```
                                    ;31689  ;HERE OFF A-FORK WHEN THERE IS A TRAP, INTERRUPT, OR CONSOLE REQUEST UP
                                    ;31690  ;AND FPA OP-CODE IN IB0.
                                    ;31691
                                    ;31692
                                    ;31693  1020:   ;-------------------------------;
U 1020, 0000,003C,0180,F800,0000,0020 ;31694            J/020
                                    ;31695
                                    ;31696  1024:   ;-------------------------------;
U 1024, 0000,003C,0180,F800,0000,0024 ;31697            J/024                       ;TRACE TRAP PENDING
                                    ;31698
                                    ;31699  1028:   ;-------------------------------;
U 1028, 0000,003C,0180,F800,0000,0028 ;31700            J/028
                                    ;31701
                                    ;31702  102C:   ;-------------------------------;
U 102C, 0000,003C,0180,F800,0000,002C ;31703            J/02C
                                    ;31704
                                    ;31705  1030:   ;-------------------------------;
U 1030, 0000,003C,0180,F800,0000,0030 ;31706            J/030                       ;INTERNAL INTERRUPT REQUEST
                                    ;31707
                                    ;31708  1034:   ;-------------------------------;;HALT PENDING
U 1034, 0000,003C,0180,F800,0000,0034 ;31709            J/034                       ;UNDO EXTRA PC ADVANCE
                                    ;31710
                                    ;31711
                                    ;31712  1038:   ;-------------------------------;
U 1038, 0000,003C,0180,F800,0000,0038 ;31713            J/038                       ;SBI INTERRUPT REQUEST
                                    ;31714
                                    ;31715  103C:   ;-------------------------------;
U 103C, 0000,003C,0180,F800,0000,003C ;31716            J/03C                       ;ARITHMETIC TRAP
                                    ;31717
                                    ;31718  1060:   ;00-----------------------------;
U 1060, 0000,003C,0180,F800,0000,0060 ;31719            J/060                       ;STOPPED WITH TB MISS.  GO FILL TB
                                    ;31720
                                    ;31721  1061:   ;01-----------------------------;
U 1061, 0000,003C,0180,F800,0000,0061 ;31722            J/061                       ;STOPPED WITH AN ERROR
                                    ;31723
                                    ;31724  1062:   ;10-----------------------------;
U 1062, 0000,003C,0180,F800,0000,0062 ;31725            J/062                       ;WAITING FOR DATA.  LOOP ON IT
```

```
                                ;31726  .TOC "  FPA Interface         : ADDF2, ADDF3, SUBF2, SUBF3 - (REG,REG), (S^#,REG)"
                                ;31727
                                ;31728  ;ASSUMPTIONS ARE: 1)  DST SPECIFIER IN IBUF<01>
                                ;31729  ;
                                ;31730  ;                 2)  IF TWO OPERAND TYPE, NO MORE DATA NEEDED BY FPA
                                ;31731  ;                     IF THREE OPERAND TYPE, OPERAND IN D-REG FOR FPA
                                ;31732
                                ;31733
                                ;31734  ;ADDF2, SUBF2(REG, REG) AFORK ENTRY
                                ;31735  1044:   ;-------------------------------;
                                ;31736          RC[TO]_LA,                      ;  ACCEPT ACCEL RESULT IF RDY
U 1044, 0000,003C,0180,F980,0000,14D2   ;31737          J/XF.A.01                       ;  TEST FOR FPA SYNC
                                ;31738
                                ;31739
                                ;31740
                                ;31741  ;ADDF2, SUBF2 (S^#, REG) A-FORK ENTRY
                                ;31742  1040:   ;-------------------------------;
                                ;31743          RC[TO]_LA,                      ;  ACCEPT ACCEL RESULT IF RDY
U 1040, 0000,003C,0180,F980,0000,14D2   ;31744          J/XF.A.01                       ;  TEST FOR FPA SYNC
                                ;31745
                                ;31746
                                ;31747
                                ;31748  ;ADDF3, SUBF3 (*, REG, REG) B-FORK ENTRY
                                ;31749  1244:   ;-------------------------------;
U 1244, 0000,007C,1580,BD80,0000,14D2   ;31750          ID_D.SYNC,RC[TO]_LA,J/XF.A.01   ;  SEND LAST OPERAND IN DREG TO ACCEL
                                ;31751
                                ;31752
                                ;31753
                                ;31754  ;ADDF3, SUBF3 (*, S^#, REG) B-FORK ENTRY
                                ;31755  1240:   ;-------------------------------;
U 1240, 0000,007C,1580,BD80,0000,14D2   ;31756          ID_D.SYNC,RC[TO]_LA,J/XF.A.01   ;  SEND LAST OPERAND IN DREG TO ACCEL
                                ;31757
                                ;31758  XF.A.01:;-------------------------------;
                                ;31759          ALU_R(SP1)+K[ZERO].RLOG,        ;  SAVE REGISTER #
U 14D2, 0018,0658,19D8,F840,0000,13D4   ;31760          Q_ACCEL&SYNC,ACCEL?             ;  ACCEPT FPA RESULT IF READY
                                ;31761
                                ;31762  =100
                                ;31763  XF.A.02:;100--------------------------;
U 13D4, 0000,067C,01D8,F800,0000,13D4   ;31764          Q_ACCEL&SYNC,ACCEL?,J/XF.A.02   ;  ACCEPT FPA RESULT IF RDY
                                ;31765
                                ;31766          ;101------------------------;  ** GOT RESULT & NO EXCEPTION **
                                ;31767          R(SP1) Q,                       ;  RESULT TO DST REG/ID_FQ1<R>
                                ;31768          LOAD.ACC.CC,                    ;  LOAD FPA CONDITION CODES INTO PSL
                                ;31769          PC_PC+2,CLR.IB0-1,              ;  PREPARE FOR NEXT INSTRUCTION
U 13D5, 4001,263C,018C,F8C5,5800,13E5   ;31770          ACCEL?,J/EXITF.SP               ;
                                ;31771
                                ;31772          ;110------------------------;  ** RESERVED OPERAND EXCEPTION **
U 13D6, 0000,003C,0180,F800,1800,14EA   ;31773          LOAD.ACC.CC,J/XD.B.09           ;  LOAD FPA CONDITION CODES IN PSL<CC>
                                ;31774
                                ;31775          ;111------------------------;  ** GOT RESULT & ARITH EXCEPT **
                                ;31776          LOAD.ACC.CC,                    ;  LOAD INTO PSL FPA CONDITION CODES
U 13D7, 4000,003C,0180,F805,5800,13E7   ;31777          PC_PC+2,CLR.IB0-1,J/OUTF.SP     ;  PREPARE FOR NEXT INSTRUCTION
```

```
                                ;31778  =101
                                ;31779  EXITF.SP:
                                ;31780          ;101----------------------------;  FINISHED
U 13E5, F80C,003B,01F1,F857,139B,6000   ;31781          IRD                         ;  NEXT INSTRUCTION
                                ;31782
                                ;31783  OUTF.SP:;111----------------------------;  EXCEPTION
U 13E7, 0840,0038,0180,F800,1C00,14D3   ;31784          D_RLOG.RIGHT                ;  REGET REGISTER #
                                ;31785
                                ;31786          ;----------------------------;
U 14D3, 0018,0038,0180,F9B8,0000,14D4   ;31787          RC[T7]_K[.8]                ;  PRELOAD T7 WITH OVERFLOW CODE
                                ;31788
                                ;31789          ;----------------------------;
                                ;31790          Q_ID[PSL],SC_D(EXP)(A),     ;  BRANCH ON V BIT
U 14D4, 0811,1A38,3DF0,2D00,0088,72FC   ;31791          D_RC[T0],PSL.V?             ;  D GETS ORIGINAL DESTINATION
                                ;31792
                                ;31793  =1100   ;1100----------------------------;  UNDERFLOW
U 12FC, 0000,003C,01A8,F800,0000,14D6   ;31794          Q_Q.LEFT,J/UNFL1            ;  MOVE PSL<FU> TO BIT POSITION 7
                                ;31795
                                ;31796  =1110   ;1110----------------------------;  OVERFLOW
U 12FE, 0001,003C,0180,F8E8,0000,12B4   ;31797          R(SC)_D,J/FLOAT.FAULT       ;  RESTORE DEST
                                ;31798
                                ;31799          ;1111----------------------------;
U 12FF, F80C,003B,01F1,F857,139B,6000   ;31800          IRD                         ;
                                ;31801
                                ;31802
                                ;31803  UNFL1:  ;----------------------------;
U 14D6, 0019,2038,F580,F9B8,1408,74D8   ;31804          STATE_Q(EXP),RC[T7]_K[.A]   ;  MOVE PSL<FU> TO STATE BIT 0
                                ;31805
                                ;31806          ;----------------------------;
U 14D8, C000,173C,0180,F800,0000,132E   ;31807          STATE0?                     ;  TEST FOR FU BIT SET
                                ;31808
                                ;31809  =1110   ;1110----------------------------;  FU NOT SET IGNORE UNDERFLOW, OUTPUT 0
U 132E, 0018,0038,1980,F8E8,0000,0062   ;31810          R(SC)_K[ZERO],J/IRD         ;  LOAD DEST WITH ZERO
                                ;31811
                                ;31812          ;1111----------------------------;  FU SET
U 132F, 0001,003C,0180,F8E8,0000,12B4   ;31813          R(SC)_D,J/FLOAT.FAULT       ;  RESTORE DEST
```

```
                                ;31814  =101
                                ;31815  EXITF.PRN:
                                ;31816          ;101--------------------;  FINISHED
U 13F5, F80C,003B,01F1,F857,139B,6000   ;31817          IRD                     ;  NEXT INSTRUCTION
                                ;31818
                                ;31819  OUTF.PRN:
                                ;31820          ;111--------------------;
U 13F7, 0018,0038,0180,F9B8,0000,14D9   ;31821          R[T7]_K[.8]             ;  PRELOAD T7 WITH OVERFLOW CODE
                                ;31822
                                ;31823          ;----------------------;  EXCEPTION
                                ;31824          D_R[T0],Q_ID[PSL],      ;  PREFETCH ORIGINAL CONTENTS OF DEST,FETCH PSL
U 14D9, 0810,1A38,3DF0,2D00,0000,133C   ;31825          PSL.V?                  ;  BRANCH ON V-BIT
                                ;31826
                                ;31827  =1100
                                ;31828          ;1100-------------------;  UNDERFLOW
U 133C, 0000,003C,01A8,F800,0000,14DA   ;31829          Q_Q.LEFT,J/UNFL2        ;  MOVE PSL<FU> TO BIT POSITION 7
                                ;31830  =1110
                                ;31831          ;1110-------------------;  OVERFLOW
                                ;31832          R(PRN)_D,               ;  RESTORE DEST
U 133E, 0001,003C,0180,F8D8,0000,12B4   ;31833          J/FLOAT.FAULT           ;
                                ;31834
                                ;31835          ;1111-------------------;
U 133F, F80C,003B,01F1,F857,139B,6000   ;31836          IRD                     ;
                                ;31837
                                ;31838  UNFL2:  ;----------------------;
U 14DA, 0001,203C,0180,F800,1408,74DB   ;31839          STATE_Q(EXP)            ;  MOVE PSL<FU> TO STATE BIT 0
                                ;31840
                                ;31841          ;----------------------;
U 14DB, 0018,1738,F580,F9B8,0000,134E   ;31842          STATE0?,R[T7]_K[.A]     ;  TEST FOR FU BIT SET
                                ;31843
                                ;31844  =1110   ;1110-------------------;  FU NOT SET IGNORE UNDERFLOW, OUTPUT 0
                                ;31845          R(PRN)_K[ZERO],         ;  LOAD DEST WITH ZERO
U 134E, 0018,0038,1980,F8D8,0000,0062   ;31846          J/IRD                   ;
                                ;31847
                                ;31848          ;1111-------------------;  FU SET
                                ;31849          R(PRN)_D,               ;  RESTORE DEST
U 134F, 0001,003C,0180,F8D8,0000,12B4   ;31850          J/FLOAT.FAULT           ;
```

```
                                ;31851  .TOC "  FPA Interface         : (ADDD,SUBD,MULD,DIVD)(2,3) - (REG,REG), (S^#,REG)"
                                ;31852
                                ;31853  ;ASSUMPTIONS ARE: 1)  DST SPECIFIER IN IBUF<01>
                                ;31854  ;                 2)  IF TWO OPERAND TYPE, NO ADDITONAL DATA NEEDED BY FPA
                                ;31855  ;                     IF THREE OPERAND TYPE, LAST OPERAND FOR FPA IN D-REG
                                ;31856
                                ;31857
                                ;31858  ;ADDD2, SUBD2 (REG, REG) A-FORK ENTRY
                                ;31859  1046:  ;-------------------------------;
U 1046, 0000,003C,0180,F980,0000,14DC   ;31860          RC[TO]_LA,J/DBL.SYNC            ; Save dest. operand in case fault
                                ;31861
                                ;31862
                                ;31863  ;DIVD2, MULD2 (REG, REG) A-FORK ENTRY
                                ;31864  1006:  ;-------------------------------;
U 1006, 0000,003C,0180,F980,0000,14DC   ;31865          RC[TO]_LA,J/DBL.SYNC            ; Save dest. operand in case fault
                                ;31866
                                ;31867
                                ;31868  ;ADDD2, SUBD2 (S^#, REG) A-FORK ENTRY
                                ;31869  1042:  ;-------------------------------;
U 1042, 0000,003C,0180,F980,0000,14DC   ;31870          RC[TO]_LA,J/DBL.SYNC            ; Save dest. operand in case fault
                                ;31871
                                ;31872
                                ;31873  ;DIVD2, MULD2 (S^#, REG) A-FORK ENTRY
                                ;31874  1002:  ;-------------------------------;
U 1002, 0000,003C,0180,F980,000C,14DC   ;31875          RC[TO]_LA,J/DBL.SYNC            ; Save dest. operand in case fault
                                ;31876
                                ;31877
                                ;31878  ;ADDD3, SUBD3 (*, REG, REG) B-FORK ENTRY
                                ;31879  1246:  ;-------------------------------; Send last operand to FPA
U 1246, 0000,007C,1580,BD80,0000,14DC   ;31880          ID_D.SYNC,RC[TO]_LA,J/DBL.SYNC  ; Save dest. operand in case fault
                                ;31881
                                ;31882
                                ;31883  ;DIVD3, MULD3 (*, REG, REG) B-FORK ENTRY
                                ;31884  1206:  ;-------------------------------; Send last operand to FPA
U 1206, 0000,007C,1580,BD80,0000,14DC   ;31885          ID_D.SYNC,RC[TO]_LA,J/DBL.SYNC  ; Save dest. operand in case fault
                                ;31886
                                ;31887
                                ;31888  ;ADDD3, SUBD3 (*, S^#, REG) B-FORK ENTRY
                                ;31889  1242:  ;-------------------------------; Send last operand to FPA
U 1242. 0000,007C,1580,BD80,0000,14DC   ;31890          ID_D.SYNC,RC[TO]_LA,J/DBL.SYNC  ; Save dest. operand in case fault
                                ;31891
                                ;31892
                                ;31893  ;DIVD3, MULD3 (*, S^#, REG) B-FORK ENTRY
                                ;31894  1202:  ;-------------------------------; Send last operand to FPA
U 1202, 0000,007C,1580,BD80,0000,14DC   ;31895          ID_D.SYNC,RC[TO]_LA,J/DBL.SYNC  ; Save dest. operand in case fault
```

```
                                    ;31896  DBL.SYNC:
                                    ;31897          ;-------------------------------;
                                    ;31898          ALU_R(SP1)+K[ZERO].RLOG,        ; SAVE REGISTER #
U 14DC, 0018,0658,19D8,F840,0000,1434 ;31899        Q_ACCEL&SYNC,ACCEL?             ; ACCEPT FIRST FPA RESULT IF RDY
                                    ;31900
                                    ;31901
                                    ;31902  =100    ;100----------------------------; FPA result not ready
U 1434, 0000,067C,01D8,F800,0000,1434 ;31903        Q_ACCEL&SYNC,ACCEL?,J/<.>       ; ACCEPT FIRST FPA RESULT IF RDY
                                    ;31904
                                    ;31905          ;101----------------------------; FPA result ready
U 1435, 0001,203C,0180,F8C0,1800,1456 ;31906        R(SP1)_Q,LOAD.ACC.CC,J/DBL.SYNC2; Dest <- FPA result, PSL <- FPA<cc>
                                    ;31907
                                    ;31908          ;110----------------------------; Reserved operand or divide by 0
U 1436, 0000,003C,0180,F800,180C,14EA ;31909        LOAD.ACC.CC,J/XD.B.C9           ; LOAD FPA CONDITION CODES IN PSL<CC>
                                    ;31910
                                    ;31911          ;111----------------------------; FPA result ready
U 1437, 0001,203C,0180,F8C0,1800,1456 ;31912        R(SP1)_Q,LOAD.ACC.CC            ; Dest <- FPA result, PSL <- FPA<cc>
                                    ;31913
                                    ;31914  =110
                                    ;31915  DBL.SYNC2:
                                    ;31916          ;110----------------------------; FPA result not ready
                                    ;31917          LAB_R(SP1+1),                   ; Save 2nd half of dest operand
U 1456, 0000,067C,01D8,F870,0000,1456 ;31918        Q_ACCEL&SYNC,ACCEL?,J/<.>       ; Accept 2nd half of result if ready
                                    ;31919
                                    ;31920          ;111----------------------------; FPA result ready
                                    ;31921          R(SP1+1)_Q,PC_PC+2,CLR.IBO-1,   ; Accept data, setup for next inst.
U 1457, 4001,263C,0180,F8F5,4000,1425 ;31922        ACCEL?,J7EXITD.SP               ; Go see if exception occured
```

```
                                    ;31923  =101
                                    ;31924  EXITD.SP:;101----------------; FINISHED
U 1425, F80C,003B,01F1,F857,139B,6000 ;31925          IRD                 ; NEXT INSTRUCTION
                                    ;31926
                                    ;31927          ;111----------------; EXCEPTION
U 1427, 0840,0038,0180,F800,1C00,14DD ;31928          D_RLOG.RIGHT        ; REGET REGISTER #,
                                    ;31929
                                    ;31930          ;-------------------;
U 14DD, 0018,0038,0180,F9B8,0000,14DE ;31931          RC[T7]_K[.8]        ; PRELOAD T7 WITH OVERFLOW CODE
                                    ;31932
                                    ;31933          ;-------------------;
                                    ;31934          SC_D(EXP)(A),LC_RC[T0], ; MOVE REG# TO SC,PSL TO Q
U 14DE, 0001,1A3C,3DF0,2D00,0088,735C ;31935          Q_ID[PSL],PSL.V?    ; BRANCH ON PSL<V> BIT
                                    ;31936
                                    ;31937  =1100
                                    ;31938          ;1100---------------; UNDERFLOW
U 135C, 0000,003C,01A8,F800,0000,14E1 ;31939          Q_Q.LEFT,J/UN.D     ; ROTATE Q
                                    ;31940
                                    ;31941  =1110   ;1110---------------; OVERFLOW
                                    ;31942          R(SC)_LC,SC_SC+K[.1], ; RESTORE DEST TO ORIGINAL VALUE
U 135E, 0010,0038,0580,F8E8,0084,94E0 ;31943          J/OVR.KJC           ;
                                    ;31944
                                    ;31945          ;1111---------------;
U 135F, 0000,003C,0180,F800,0000,0062 ;31946          J/IRD               ;
                                    ;31947
                                    ;31948  OVR.KJC:;-------------------;
U 14E0, 0000,003C,0180,F8E8,0000,12B4 ;31949          R(SC)_LA,J/FLOAT.FAULT ; RESTORE SECOND HALF OF DEST
                                    ;31950
                                    ;31951  UN.D:   ;-------------------;
U 14E1, 0001,203C,0180,F800,1408,74E2 ;31952          STATE_Q(EXP)        ; MOVE PSL<fu> TO STATE BIT ZERO
                                    ;31953
                                    ;31954          ;-------------------;
U 14E2, 0018,1738,F580,F9B8,0000,136E ;31955          STATE0?,RC[T7]_K[.A] ; BRANCH ON STATE0
                                    ;31956
                                    ;31957  =1110   ;1110---------------; PSL<fu>=0,IGNORE UNDERFLOW
                                    ;31958          R(SC)_K[ZERO],      ; ZERO OUT DESTINATION
U 136E, 0018,0038,1980,F8E8,0000,14E8 ;31959          J/FT1               ;
                                    ;31960
                                    ;31961          ;1111---------------; PSL<fu>=1, TAKE UNDERFLOW FAULT
U 136F, 0010,0038,0580,F8E8,0084,94E3 ;31962          R(SC)_LC,SC_SC+K[.1] ; RESTORE DEST
                                    ;31963
                                    ;31964          ;-------------------;
                                    ;31965          R(SC)_LA,           ; RESTORE SECOND HALF OF DEST
U 14E3, 0000,003C,0180,F8E8,0000,12B4 ;31966          J/FLOAT.FAULT       ;
                                    ;31967
                                    ;31968  FT1:    ;-------------------;
U 14E8, 0000,003C,0580,F800,0084,94E9 ;31969          SC_SC+K[.1]         ; INC SC
                                    ;31970
                                    ;31971          ;-------------------;
U 14E9, 0018,0038,1980,F8E8,0000,0062 ;31972          R(SC)_K[ZERO],J/IRD ; ZERO OUT SECOND HALF OF DEST
```

```
                                    ;31973  .TOC "  FPA Interface         : ADDF2, SUBF2, MULF2, DIVF2 - ( * ,REG)"
                                    ;31974
                                    ;31975  ;ASSUMPTIONS ARE: 1)  DST REG IN PRN-REG
                                    ;31976  ;
                                    ;31977  ;                 2)  LAST OPERAND FOR FPA IN D-REG
                                    ;31978
                                    ;31979
                                    ;31980  ;ADDF2, SUBF2 (*, REG) B-FORK ENTRY
                                    ;31981  122F:   ;-------------------------------;
U 122F, 0000,007C,1580,BD80,0000,1464 ;31982        ID_D.SYNC,RC[TO]_LA,J/XF.B.01   ; SEND LAST OPERAND IN DREG TO ACCEL
                                    ;31983
                                    ;31984
                                    ;31985
                                    ;31986  ;MULF2: (*, REG) B-FORK ENTRY
                                    ;31987  1220:   ;-------------------------------;
U 1220, 0000,007C,1580,BD80,0000,1464 ;31988        ID_D.SYNC,RC[TO]_LA,J/XF.B.01   ; SEND LAST OPERAND IN DREG TO ACCEL
                                    ;31989
                                    ;31990
                                    ;31991
                                    ;31992  ;DIVF2:  (*, REG) B-FORK ENTRY
                                    ;31993  1221:   ;-------------------------------;
U 1221, 0000,007C,1580,BD80,0000,1464 ;31994        ID_D.SYNC,RC[TO]_LA,J/XF.B.01   ; SEND LAST OPERAND TO FPA
                                    ;31995
                                    ;31996
                                    ;31997
                                    ;31998  =100
                                    ;31999  XF.B.01:;100----------------------------;
U 1464, 0000,067C,01D8,F800,0000,1464 ;32000        Q_ACCEL&SYNC,ACCEL?,J/XF.B.01   ; ACCEPT FPA RESULT IF RDY
                                    ;32001
                                    ;32002          ;101--------------------------; ** GOT RESULT & NO EXCEPTION **
                                    ;32003          R(PRN) Q,                       ; RESULT TO DST REG/PRN
                                    ;32004          LOAD.ACC.CC,                    ; LOAD FPA CONDITION CODES INTO PSL
                                    ;32005          PC_PC+1,CLR.IB.OPC,             ; SETUP FOR NEXT INSTRUCTION
U 1465, C001,263C,0180,F8DC,5800,13F5 ;32006        ACCEL?,J/EXITF.PRN              ;
                                    ;32007
                                    ;32008          ;.  0-------------------------; ** RESERVED OPERAND EXCEPTION **
U 1466, 0000,003C,0180,F800,1800,14EA ;32009        L  .0.ACC.CC,J/XD.B.09          ;
                                    ;32010
                                    ;32011          ;111--------------------------; ** GOT RESULT & ARITH EXCEPT **
                                    ;32012          D_Q,Q_ID[PSL],                  ;
                                    ;32013          LOAD.ACC.CC,                    ; LOAD INTO PSL FPA CONDITION CODES
U 1467, CC00,003C,3DF0,2C04,5800,13F7 ;32014        PC_PC+1,CLR.IB.OPC,J/OUTF.PRN   ; SETUP FOR NEXT INSTRUCTION
```

```
                                ;32015  .TOC "  FPA Interface         : ADDD2, SUBD2, MULD2, DIVD2 - ( * ,REG)"
                                ;32016
                                ;32017  ;ASSUMPTIONS ARE: 1)  DST REG/PRN & PRN+1
                                ;32018  ;                 2)  LAST OPERAND FOR FPA IN D-REG
                                ;32019
                                ;32020
                                ;32021  ;ADDD2, SUBD2 (*, REG) B-FORK ENTRY
                                ;32022  122E:   ;-------------------------------;
U 122E, 0000,007C,1580,BC58,0000,1474   ;32023          ID_D.SYNC,LAB_R(PRN),J/XD.B.01  ; SEND LAST OPERAND/D-REG TO FPA
                                ;32024
                                ;32025
                                ;32026  ;MULD2: (*, REG) B-FORK ENTRY
                                ;32027  1228:   ;-------------------------------;
U 1228, 0000,007C,1580,BC58,0000,1474   ;32028          ID_D.SYNC,LAB_R(PRN),J/XD.B.01  ; SEND LAST OPERAND/D-REG TO FPA
                                ;32029
                                ;32030
                                ;32031  ;DIVD2: (*, REG) B-FORK ENTRY
                                ;32032  1229:   ;-------------------------------;
U 1229, 0000,007C,1580,BC58,0000,1474   ;32033          ID_D.SYNC,LAB_R(PRN),J/XD.B.01  ; SEND LAST OPERAND TO FPA
                                ;32034
                                ;32035  =100
                                ;32036  XD.B.01:
                                ;32037          ;100------------------------------; ** NO FPA RESULT **
                                ;32038          Q_ACCEL&SYNC,RC[T0]_LA,         ;
U 1474, 0000,067C,01D8,F980,0000,1474   ;32039          ACCEL?,J/XD.B.01                ; ACCEPT FIRST FPA RESULT IF RDY
                                ;32040
                                ;32041          ;101------------------------------; ** GOT RESULT **
U 1475, 0001,203C,0180,F8D8,1800,14B6   ;32042          R(PRN)_Q,LOAD.ACC.CC,J/XD.B.05  ; FPA RESULT TO R/PRN, GET CC'S
                                ;32043
                                ;32044          ;110------------------------------; ** RESERVED OPERAND EXCEPTION **
U 1476, 0000,003C,0180,F800,1800,14EA   ;32045          LOAD.ACC.CC,J/XD.B.09           ;
                                ;32046
                                ;32047          ;111------------------------------; ** GOT RESULT **
U 1477, 0001,203C,0180,F8D8,1800,14B6   ;32048          R(PRN)_Q,LOAD.ACC.CC            ; FPA RESULT TO R/PRN, GET CC'S
                                ;32049
                                ;32050  =110
                                ;32051  XD.B.05:;110------------------------------; ** 2ND RESULT NOT RDY **
                                ;32052          Q_ACCEL&SYNC,LAB_R(PRN+1),      ;
U 14B6, 0000,067C,01D8,F860,0000,14B6   ;32053          ACCEL?, J/XD.B.05               ; ACCEPT 2ND RESULT OPERAND IF RDY
                                ;32054
                                ;32055          ;111------------------------------; ** GOT 2ND FPA RESULT **
                                ;32056          R(PRN+1)_Q,PC_PC+1,CLR.IB.OPC,  ; FPA RESULT TO R/PRN+1
U 14B7, C001,263C,0180,F8E4,4000,1545   ;32057          ACCEL?,J7EXITD.PRN              ;
                                ;32058
                                ;32059
                                ;32060  ;Reserved floating operand or divide by zero
                                ;32061  XD.B.09:;-------------------------------;
U 14EA, 0000,1A3C,0180,F800,0000,10BC   ;32062          PSL.V?                          ; See which it is
                                ;32063
                                ;32064  =110*   ;110*-----------------------------; ** RSVD FLT OPERAND **
U 10BC, 0000,003C,0180,F800,0000,0106   ;32065          J/RSVOPR                        ;
                                ;32066
                                ;32067          ;111*----------------------------; ** DIVIDE BY ZERO **
U 10BE, 0018,0038,D980,F9B8,0000,12B4   ;32068          RC[T7]_K[.9],J/FLOAT.FAULT      ;
```

```
                                    ;32069  =101
                                    ;32070  EXITD.PRN:;101----------------;  FINISHED
U 1545, F80C,003B,01F1,F857,139B,6000 ;32071        IRD                   ;  NEXT INSTRUCTION
                                    ;32072
                                    ;32073        ;111-----------------;  EXCEPTION
U 1547, 0018,0038,0180,F9B8,0000,14EB ;32074        RC[T7]_K[.8]          ;  PRELOAD T7 WITH OVERFLOW CODE
                                    ;32075
                                    ;32076        ;--------------------;
                                    ;32077        LC_RC[T0],Q_ID[PSL],  ;  Q GETS PSL
U 14EB, 0000,1A3C,3DF0,2D00,0000,139C ;32078        PS[.V?                ;  BRANCH ON V BIT
                                    ;32079
                                    ;32080  =1100
                                    ;32081        ;1100----------------;  UNDERFLOW
U 139C, 0000,003C,01A8,F800,0000,14ED ;32082        Q_Q.LEFT,J/UN.D.1     ;  ROTATE Q
                                    ;32083
                                    ;32084  =1110 ;1110----------------;  OVERFLOW
U 139E, 0000,003C,0180,F8E0,0000,14EC ;32085        R(PRN+1)_LA,J/OVR.DPR ;  RESTORE SECOND HALF OF DEST
                                    ;32086
                                    ;32087        ;1111----------------;
U 139F, 0000,003C,0180,F800,0000,0062 ;32088        J/IRD                 ;
                                    ;32089
                                    ;32090  OVR.DPR:;-------- ----------;
U 14EC, 0010,0038,0180,F8D8,0000,12B4 ;32091        R(PRN)_LC,J/FLOAT.FAULT ;  RESTORE FIRST HALF OF DEST
                                    ;32092
                                    ;32093  UN.D.1: ;--------------------;
U 14ED, 0001,203C,0180,F800,1408,74EE ;32094        STATE_Q(EXP)          ;  MOVE PSL<fu> TO STATE BIT ZERO
                                    ;32095
                                    ;32096        ;--------------------;
U 14EE, 0018,1738,F580,F9B8,0000,13AE ;32097        STATE0?,RC[T7]_K[.A]  ;  BRANCH ON PSL<fu> BIT
                                    ;32098
                                    ;32099  =1110 ;1110----------------;  PSL<fu>=0,IGNORE UNDERFLOW
                                    ;32100        R(PRN)_K[ZERO],       ;  ZERO OUT DEST
U 13AE, 0018,0038,1980,F8D8,0000,14F1 ;32101        J/FT2                 ;
                                    ;32102
                                    ;32103        ;1111----------------;  PSL<fu>=1,TAKE UNDERFLOW FAULT
U 13AF, 0000,003C,0180,F8E0,0000,14F0 ;32104        R(PRN+1)_LA           ;  RESTORE DEST+1
                                    ;32105
                                    ;32106        ;--------------------;
U 14F0, 0010,0038,0180,F8D8,0000,12B4 ;32107        R(PRN)_LC,J/FLOAT.FAULT ;  RESTORE DEST
                                    ;32108
                                    ;32109  FT2:  ;--------------------;
U 14F1, 0018,0038,1980,F8E0,0000,0062 ;32110        R(PRN+1)_K[ZERO],J/IRD ;  ZERO OUT SECOND HALF OF DEST
```

```
                                    ;32111  .TOC "  FPA Interface         : (ADDF,SUBF,MULF,DIVF)(2,3) - Memory destination"
                                    ;32112
                                    ;32113  ;ASSUMPTIONS ARE: 1)  IF DST/REG & TWO OPERAND TYPE, DST/PRN-REG
                                    ;32114  ;                 2)  IF DST/MEM & TWO OPERAND TYPE, DST/VA
                                    ;32115  ;                 3)  LAST OPERAND TO PASS TO FPA IN D-REG
                                    ;32116
                                    ;32117
                                    ;32118
                                    ;32119  ;ADDF2, SUBF2(*,*) & ADDF3, SUBF3(*,*,*) C-FORK ENTRY
                                    ;32120  138E:   ;-------------------------------;
                                    ;32121          ID_D.SYNC,                      ; SEND LAST OPERAND TO FPA
U 138E, 0000,1B7D,1580,BC00,0000,13BD ;32122        CALL,IR0?,J/XF.C.02             ; SET UP RETURN ADDR FOR ASPC RTN
                                    ;32123
                                    ;32124  13EE:   ;00-------------------------------; ** ASRC RETURN60, MEM DST **
U 13EE, 0A00,067C,0180,F800,0000,1554 ;32125        D_ACCEL&SYNC,ACCEL?,J/XF.C.03   ; ACCEPT FPA RESULT IF RDY
                                    ;32126
                                    ;32127  13EF:   ;01-------------------------------;  ** ASRC RETURN61, REG DST **
                                    ;32128          Q_ACCEL&SYNC,RC[TO]_LA,         ;  ACCEPT FPA RESULT IF RDY
U 13EF, 0000,067C,01D8,F980,0000,1464 ;32129        ACCEL?,J/XF.B.01                ;  MERGE INTO DST/PRN ROUTINE
                                    ;32130
                                    ;32131
                                    ;32132
                                    ;32133
                                    ;32134  ;MULF2(*,*), MULF3(*,*,*) C-FORK ENTRY
                                    ;32135  138F:   ;-------------------------------;
                                    ;32136          ID_D.SYNC,ALU_0(A),CLK.UBCC,    ;  CLEAR ALU CONDITION CODES
U 138F, 0003,007C,1580,BC00,0010,1218 ;32137        J/XF.C.01                       ;
                                    ;32138
                                    ;32139
                                    ;32140
                                    ;32141
                                    ;32142  ;DIVF2(*,*), DIVF3(*,*,*) C-FORK ENTRY
                                    ;32143  138D:   ;-------------------------------;
                                    ;32144          ID_D.SYNC,ALU_0(A),CLK.UBCC,    ;  CLEAR ALU CONDITION CODES
U 138D, 0003,007C,1580,BC00,0010,1218 ;32145        J/XF.C.01                       ;
```

```
                                ;32146  =00****0
                                ;32147  XF.C.01:
                                ;32148          ;00****0------------------------;
U 1218, 0000,1B3D,0180,F800,0000,13BD   ;32149          CALL,IRO?,J/XF.C.02             ; SET UP RETURN FROM ASPC ROUTINE
                                ;32150
                                ;32151  =11****0
                                ;32152          ;11****0------------------------;  ** ASRC RETURN60, MEM DST **
U 1278, 0A00,067C,0180,F800,0000,1554   ;32153          D_ACCEL&SYNC,ACCEL?,J/XF.C.03   ;  ACCEPT FPA RESULT IF RDY
                                ;32154
                                ;32155          ;11****1------------------------;  ** ASRC RETURN61, REG DST **
                                ;32156          Q_ACCEL&SYNC,RC[T0]_LA,         ;  ACCEPT FPA RESULT IF RDY
U 1279, 0000,067C,01D8,F980,0000,1464   ;32157          ACCEL?,J/XF.B.01                ;  MERGE INTO DST/PRN ROUTINE
                                ;32158
                                ;32159
                                ;32160  =1101
                                ;32161  XF.C.02:;1101------------------------;  ** TWO ADDRESS TYPE **
                                ;32162          D_ACCEL&SYNC,VA_RC[T7],         ;  ACCEPT FPA DATA IF RDY
U 13BD, 0A10,0678,0180,F938,0200,1554   ;32163          ACCEL?,J/XF.C.03                ;
                                ;32164
                                ;32165          ;1111------------------------;  ** THREE ADDRESS TYPE **
                                ;32166          Q_IB.DATA,                      ;  GET SPECIFIER DATA FROM ISTREAM
                                ;32167          LAB_R(SP1),                     ;  LOAD LATCHES FROM BASE SPECIFIER
                                ;32168          CLR.IB.COND,                    ;  DISCARD BASE OPERAND SPECIFIER
                                ;32169          PC_PC+N,                        ;  STEP PC OVER IT
                                ;32170          MCT/ALLOW.IB.READ,              ;  LET IB GET NEEDED DATA
U 13BF, F000,003F,01F0,F847,0000,0400   ;32171          SUB/SPEC,J/ASPC.B               ;  EVALUATE THE SPECIFIER
                                ;32172
                                ;32173
                                ;32174
                                ;32175  =100
                                ;32176  XF.C.03:;100------------------------;  ** FPA DATA NOT RDY **
U 1554, 0A00,067C,0180,F800,0000,1554   ;32177          D_ACCEL&SYNC,ACCEL?,J/XF.C.03   ;
                                ;32178
                                ;32179          ;101------------------------;  ** FPA RESULT ACCEPTED **
U 1555, 0000,063C,0180,F800,1800,1565   ;32180          LOAD.ACC.CC,ACCEL?,J/EXITF.MEM  ;  LOAD PSL<CC>/ACCEL<CC>
                                ;32181
                                ;32182          ;110------------------------;  ** RSVD OPND + DIV CHR **
U 1556, 0000,003C,01E0,F800.1800,14EA   ;32183          LOAD.ACC.CC,Q_D,J/XD.B.09       ;
                                ;32184
                                ;32185          ;111------------------------;  ** GOT RESULT & ARITH EXCEPT **
U 1557, 0000,003C,0180,F800,1800,1567   ;32186          LOAD.ACC.CC,J/OUTF.MEM          ;       ;  LOAD PSL<CC>/ACCEL<CC>
```

```
                                ;32187  =101
                                ;32188  EXITF.MEM:;101----------------;  GOOD ANSWER
                                ;32189          CACHE_D[LONG],PC_PC+1, ;  WRITE ANSWER,INC PC,NEXT INSTRUCTION
U 1565, C000,003C,0180,3004,4000,0062   ;32190          CLR.IB.OPC,J/IRD      ;
                                ;32191
                                ;32192  OUTF.MEM:
                                ;32193          ;111----------------;  EXCEPTION
U 1567, 0018,0038,0180,F988,0000,14F2   ;32194          RC[T7]_K[.8]          ;  PREFETCH OVERFLOW CODE
                                ;32195
                                ;32196          ;--------------------;
U 14F2, 0000,1A3C,3DF0,2C00,0000,13CC   ;32197          Q_ID[PSL],PSL.V?      ;  GET PSL,BRANCH ON V BIT
                                ;32198
                                ;32199  =1100   ;1100----------------;  UNDERFLOW
U 13CC, 0000,003C,01A8,F800,0000,14F3   ;32200          Q_Q LEFT,J/UNFLMEM    ;  ROTATE Q
                                ;32201
                                ;32202  =1110   ;1110----------------;  OVERFLOW
U 13CE, 0000,003C,0180,F800,0000,12B4   ;32203          J/FLOAT.FAULT         ;
                                ;32204
                                ;32205          ;1111----------------;
U 13CF, 0000,003C,0180,F800,0000,0062   ;32206          J/IRD                 ;
                                ;32207
                                ;32208  UNFLMEM:;--------------------;
U 14F3, 0001,203C,0180,F800,1408,74F8   ;32209          STATE_Q(EXP)          ;  STATE GETS PSL
                                ;32210
                                ;32211          ;--------------------;
U 14F8, 0018,1738,F580,F988,0000,13DE   ;32212          STATE0?,RC[T7]_K[.A]  ;  BRANCH ON STATE0 (PSL<fu>)
                                ;32213
                                ;32214  =1110   ;1110----------------;  PSL<fu>=0, IGNORE UNDERFLOW
U 13DE, 0818,0038,1980,F800,0000,14F9   ;32215          D_K[ZERO],J/ENDITF    ;  PREPARE TO ZERO OUT DEST
                                ;32216
                                ;32217          ;1111----------------;  PSL<fu>=1, TAKE UNDERFLOW FAULT
                                ;32218          PC_PC+1,              ;  RESTORE DEST
U 13DF, C000,003C,0180,F804,4000,12B4   ;32219          CLR.IB.OPC,J/FLOAT.FAULT;  NEXT INSTRUCTION
                                ;32220
                                ;32221  ENDITF: ;--------------------;
                                ;32222          CACHE_D[LONG],PC_PC+1, ;  ZERO OUT DEST
U 14F9, C000,003C,0180,3004,4000,0062   ;32223          CLR.IB.OPC,J/IRD      ;
```

```
                                ;32224  .TOC "  FPA Interface         : (ADDD,SUBD,MULD,DIVD)(2,3) - Memory destination"
                                ;32225
                                ;32226  ;ASSUMPTIONS ARE         1)  IF DST/REG & TWO OPERAND TYPE, DST/PRN-REG
                                ;32227  ;                        2)  IF DST/MEM & TWO OPERAND TYPE, DST/(RC[T7])
                                ;32228  ;                        3)  LAST OPERAND FOR FPA IN D-REG
                                ;32229
                                ;32230  ;ADDD2, SUBD2(*,*) & ADDD3, SUBD3(*,*,*) C-FORK ENTRY
                                ;32231
                                ;32232  138A:   ;----------------------;
                                ;32233          ID_D.SYNC,             ;  SEND LAST OPERAND TO FPA
                                ;32234          CALL,                  ;  SETUP RETURN ADDR FOR ASPC RTN
U 138A, 0000,1B7D,1580,BC00,0000,13FD   ;32235          IR0?,   J/XD.C.02      ;  TEST FOR TWO ADDR TYPE
                                ;32236
                                ;32237  13EA:   ;00--------------------;  ** ASRC RETURN60, MEM DST **
                                ;32238          D_ACCEL&SYNC,RC[T7]_D, ;  ACCEPT FPA RESULT IF RDY
U 13EA, 0A01,067C,0180,F9B8,0000,1574   ;32239          ACCEL?, J/XD.C.03      ;
                                ;32240
                                ;32241  13EB:   ;01--------------------;  ** ASRC RETURN61, REG DST **
                                ;32242          RC[T0]_LA,             ;
                                ;32243          Q_ACCEL&SYNC,          ;  ACCEPT FPA RESULT IF RDY
U 13EB, 0000,067C,01D8,F980,0000,1474   ;32244          ACCEL?, J/XD.B.01      ;  MERGE INTO DST/PRN & PRN+1 ROUTINE
                                ;32245
                                ;32246  ;MULD2(*,*) & MULD3(*,*,*) C-FORK ENTRY
                                ;32247
                                ;32248  138C:   ;----------------------;
                                ;32249          ID_D.SYNC,             ;  SEND LAST OPERAND TO FPA
                                ;32250          ALU_0(A),CLK.UBCC,     ;  CLEAR ALU CONDITION CODE
U 138C, 0003,007C,1580,BC00,0010,1288   ;32251          J/XD.C.01              ;
                                ;32252
                                ;32253  ;DIVD2(*,*) & DIVD3(*,*,*) C-FORK ENTRY
                                ;32254
                                ;32255  138B:   ;----------------------;
                                ;32256          ID_D.SYNC,             ;  SEND LAST OPERAND TO FPA
                                ;32257          ALU_0(A),  CLK.UBCC,   ;  CLEAR ALU CONDITIONS CODES
U 138B, 0003,007C,1580,BC00,0010,1288   ;32258          J/XD.C.01              ;
                                ;32259
                                ;32260  =00****0 ; CALL CONSTRAINT BLOCK
                                ;32261
                                ;32262  XD.C.01:;----------------------;
                                ;32263          CALL,                  ;  SET UP RETURN FROM ASPC ROUTINE
U 1288, 0000,1B3D,0180,F800,0000,13FD   ;32264          IR0?,   J/XD.C.02      ;
                                ;32265
                                ;32266  =11****0;00--------------------;  ** ASRC RETURN60, MEM DST **
                                ;32267          D_ACCEL&SYNC,RC[T7]_D, ;  ACCEPT FPA RESULT IF RDY
U 12E8, 0A01,067C,0180,F9B8,0000,1574   ;32268          ACCEL?, J/XD.C.03      ;
                                ;32269
                                ;32270          ;01--------------------;  ** ASRC RETURN61, REG DST **
                                ;32271          RC[T0]_LA,             ;
                                ;32272          Q_ACCEL&SYNC,          ;  ACCEPT FPA RESULT IF RDY
U 12E9, 0000,067C,01D8,F980,0000,1474   ;32273          ACCEL?, J/XD.B.01      ;  MERGE INTO DST/PRN & PRN+1 ROUTINE
                                ;32274  =;END
```

```
                                    ;32275  =1101 ;BRANCH ON NUMBER OF SPECIFIERS (BEN/ALU)
                                    ;32276
                                    ;32277  XD.C.02:;00--------------------;  ** TWO ADDRESS TYPE **
                                    ;32278          D_ACCEL&SYNC,          ;
                                    ;32279          VA_RC[T7],             ;  ACCEPT FPA DATA IF RDY
U 13FD, 0A10,0678,0180,F938,0200,1574 ;32280       ACCEL?, J/XD.C.03      ;
                                    ;32281
                                    ;32282          ;01--------------------;  ** THREE ADDRESS TYPE **
                                    ;32283          Q_IB.DATA,             ;  GET SPECIFIER DATA FROM ISTREAM
                                    ;32284          LAB_R(SP1),            ;  LOAD LATCHES FROM BASE SPECIFIER
                                    ;32285          CLR_IB.COND,           ;  DISCARD BASE OPERAND SPECIFIER
                                    ;32286          PC_PC+N,               ;  STEP PC OVER IT
                                    ;32287          MCT/ALLOW.IB.READ,     ;  LET IB GET NEEDED DATA
U 13FF, F000,003F,01F0,F847,0000,0400 ;32288       SUB/SPEC,J/ASPC.B      ;  EVALUATE THE SPECIFIER
                                    ;32289
                                    ;32290
                                    ;32291
                                    ;32292  =100    ;BRANCH ON FPA SYNC & FPA ERROR (BEN/ACCEL)
                                    ;32293
                                    ;32294  XD.C.03:;00--------------------;  ** FPA NOT FINISHED **
                                    ;32295          D_ACCEL&SYNC,          ;
U 1574, 0A00,067C,0180,F800,0000,1574 ;32296       ACCEL?, J/XD.C.03      ;  ACCEPT FPA RESULT IF RDY
                                    ;32297
                                    ;32298          ;01--------------------;  ** GOT 1ST FPA RESULT **
                                    ;32299          Q_D,VA_VA+4,           ;  STORE 1ST HALF OF RESULT
                                    ;32300          MSC/LOAD.ACC.CC,       ;  LOAD PSL<CC>/ACC<CC>
U 1575, 0000,003C,01EC,F803,1800,15B6 ;32301       J/XD.C.05              ;
                                    ;32302
                                    ;32303          ;10--------------------;  ** FPA RSVD OPND + DIV CK **
                                    ;32304          MSC/LOAD.ACC.CC,       ;
                                    ;32305          Q_D,                   ;
U 1576, 0000,003C,01E0,F800,1800,14EA ;32306       J/XD.B.09              ;
                                    ;32307
                                    ;32308          ;11--------------------;  ** GOT IT & FPA EXCEPTION **
                                    ;32309          Q_D,VA_VA+4,           ;  1ST HALF OF RESULT TO MEM
U 1577, 0000,003C,01E0,F803,1800,15B6 ;32310       MSC/LOAD.ACC.CC        ;  LOAD PSL<CC>/ACC<CC>
                                    ;32311
                                    ;32312  =110    ;BRANCH ON FPA SYNC (BEN/ACCEL)
                                    ;32313
                                    ;32314  XD.C.05:;00--------------------;  ** FPA NOT RDY **
                                    ;32315          D_ACCEL&SYNC,          ;  ACCEPT 2ND HALF IF RDY
U 15B6, 0A00,067C,0180,F800,0000,15B6 ;32316       ACCEL?, J/XD.C.05      ;
                                    ;32317
                                    ;32318          ;01--------------------;  ** GOT 2ND HALF OF RESULT **
U 15B7, 0000,063C,0180,F800,0000,15D5 ;32319       ACCEL?, J/EXITD.MEM    ;
```

```
                                ;32320  =101
                                ;32321  EXITD.MEM:;101----------------;  GOOD RESULT
                                ;32322          CACHE_D[LONG],        ;  WRITE SECOND PART OF ANSWER
U 15D5, 00C0,003C,0180,3000,0000,150E   ;32323          J/EXTRA.D             ;
                                ;32324
                                ;32325          ;111----------------;  EXCEPTION
U 15D7, 0000,1A3C,3DF0,2C00,0000,145C   ;32326          Q_ID[PSL],PSL.V?      ;  Q GETS PSL, BRANCH ON V BIT
                                ;32327
                                ;32328  =1100   ;1100----------------;  UNDERFLOW
U 145C, 0000,003C,01A8,F800,0000,14FA   ;32329          Q_Q.LEFT,J/UN.D.MEM   ;  ROTATE Q
                                ;32330
                                ;32331  =1110   ;1110----------------;  OVERFLOW
                                ;32332          RC[T7]_K[.8],         ;  LOAD OVERFLOW CODE
U 145E, 001C,0038,0180,F9B8,0000,12B4   ;32333          J/FLOAT.FAULT         ;
                                ;32334
                                ;32335          ;1111----------------;
U 145F, 0000,003C,0180,F800,0000,0062   ;32336          J/IRD                 ;
                                ;32337
                                ;32338  UN.D.MEM:;----------------------;
U 14FA, 0001,203C,0180,F800,1408,74FB   ;32339          STATE_Q(EXP)          ;  STATE GETS PSL
                                ;32340
                                ;32341          ;----------------------;
U 14FB, 0000,173C,0180,F800,0000,146E   ;32342          STATE0?               ;  BRANCH ON PSL<fu>
                                ;32343
                                ;32344  =1110   ;1110----------------;  PSL<fu>=0, IGNORE UNDERFLOW
U 146E, 0818,0038,1980,F800,0000,14FC   ;32345          D_K[ZERO],J/ENDITD    ;  PREPARE TO ZERO OUT DEST
                                ;32346
                                ;32347          ;1111----------------;  PSL<fu>=1  TAKE UNDERFLOW FAULT
                                ;32348          PC_PC+1,CLR.IB.OPC,   ;  NEXT INSTRUCTION
                                ;32349          RC[T7]_K[.A],         ;  LOAD UNDERFLOW CODE
U 146F, C018,0038,F580,F9BC,4000,12B4   ;32350          J/FLOAT.FAULT         ;
                                ;32351
                                ;32352  ENDITD: ;----------------------;
U 14FC, 0000,003C,0180,3000,0000,14FD   ;32353          CACHE_D[LONG]         ;  ZERO OUT SECOND HALF OF DEST
                                ;32354
                                ;32355          ;----------------------;
U 14FD, 0010,0038,0180,F938,0200,14FE   ;32356          VA_RC[T7]             ;  RESET ADDRESS
                                ;32357
                                ;32358          ;----------------------;
                                ;32359          CACHE_D[LONG],PC_PC+1,  ;  ZERO OUT FIRST HALF OF DEST
U 14FE, C000,003C,0180,3004,4000,0062   ;32360          CLR.IB.OPC,J/IRD      ;  PREPARE FOR NEXT INSTRUCTION
                                ;32361
                                ;32362  EXTRA.D:;----------------------;
U 150E, 0C10,0038,0180,F938,0200,1510   ;32363          VA_RC[T7],D_Q         ;
                                ;32364
                                ;32365          ;----------------------;
                                ;32366          CACHE_D[LONG],PC_PC+1,  ;
U 1510, C000,003C,0180,3004,4000,0062   ;32367          CLR.IB.OPC,J/IRD      ;
```

```
;32368  .TOC "  FPA Interface         : MULL2                        - ( * ,REG)"
;32369
;32370  ;*******************************************************
;32371
;32372  ;DESCRIPTION:
;32373  ;-----------------------------
;32374  ;THIS ROUTINE ACCEPTS THE MAGNITUDE PRODUCT OF THE MULR AND MULD(PROD)
;32375  ;FROM THE FPA.  SINCE THE FPA CREATES AN UNSIGNED 64 BIT PRODUCT USING
;32376  ;32 BIT SIGNED OPERANDS, THE CP MICRO-CODE MUST CORRECT THE RESULT BY
;32377  ;SUBTRACTING OUT EFFECTS OF NEGATIVE SIGNS ON THE MAGNITUDE RESULT AND
;32378  ;CREATE THE PROPER SIGNED RESULT, SET THE CONDITION CODES, AND TEST
;32379  ;FOR OVERFLOW.
;32380
;32381  ;THIS ROUTINE OVERLAPS SETUP FOR THE CORRECTION, EVALUATION OF THE 3RD
;32382  ;SPECIFIER, AND DETERMINES WHICH CORRECTION SEQUENCE TO USE
;32383  ;IN PARALLEL WITH THE FPA'S DOING THE MAGNITUDE MULTIPLY.
;32384  ;IT SAVES THE MULR AND MULD(PROD) IN A FORM THAT CAN BE
;32385  ;LATER USED AS SUBTRAHEND OPERANDS TO CORRECT THE PRODUCT.  IF THE
;32386  ;MULTIPLY IS A MULL3, THE 3RD SPECIFIER IS EVALUATED USING THE ASPC ROUTINE.
;32387  ;AND IT EXAMINES THE INPUT OPERAND SIGNS TO DETERMINE WHICH OF THE FOUR
;32388  ;CORRECTION SEQUENCES IS REQUIRED FOR THE MULTIPLY BEING DONE.
;32389
;32390  ;THE CORRECTIONS PERFORMED ARE:
;32391  ;                1)  SIGNS POSITIVE,  NO CORRECTION
;32392  ;                2)  MULR(+), MULD/PROD(-);  MAGNITUDE PRODUCT-MULR*(2**32)
;32393  ;                3)  MULR(-), MULD/PROD(+);  MAGNITUDE PRODUCT-MULD/PROD*(2**32)
;32394  ;                4)  MULR(-), MULD/PROD(-);  MAGNITUDE PRODUCT-MULD/PROD*(2**32)
;32395  ;                                            -MULR*(2**32)
;32396
;32397  ;MULL ENTRIES:
;32398  ;-----------------------------------
;32399
;32400  ;       THE FOLLOWING SEVEN(7) ENTRIES INTO THE MULL INTERFACE ROUTINE ARE
;32401  ;       SCATTERED THROUGH-OUT THE FOLLOWING MICRO-CODE :
;32402
;32403  ;               A-FORK: DIVF2,MULF2,MULL2(REG, REG)
;32404  ;                       DIVF2,MULF2,MULL2(S^#, REG)
;32405
;32406  ;               B-FORK: DIVF3, MULF3, MULL3(*, REG, REG)
;32407  ;                       DIVF3, MULF3, MULL3(*, S^#, REG)
;32408  ;                       MULL2(*, REG)
;32409
;32410  ;               C-FORK: MULL2(*, *)
;32411  ;                       MULL3(*, *, *)
;32412
;32413  ;               NOTES:  "*" ANY SPECIFIER
;32414  ;                       "S^#"  SHORT LITERIAL SPECIFIER
;32415  ;                       "REG"  GENERAL REGISTER SPECIFIER
```

```
                                    ;32416  ;MULL2(*,R)  B-FORK ENTRY
                                    ;32417  ;-----------------------------------
                                    ;32418
                                    ;32419  ;STATUS ON ENTRY:       1)  PROD/GENERAL REG<PRN>
                                    ;32420  ;                       2)  MULR OPERAND IN D-REG
                                    ;32421  ;                       3)  PROD OPERAND IN LAB
                                    ;32422  ;
                                    ;32423
                                    ;32424  122D:   ;-----------------------;
                                    ;32425          R[R15]_0-LB,            ;  SAVE ADJUSTED PROD OPERAND
U 122D, 000F,0040,1580,BEF8,0000,1511 ;32426        ID_D.SYNC               ;  SEND LAST OPERAND TO FPA
                                    ;32427
                                    ;32428          ;-----------------------;
                                    ;32429          RC[TO]_D-K[.1],         ;  SAVE (MULR-1) OPERAND
U 1511, 0019,0000,0580,F980,0030,1512 ;32430        N_AMX.Z_TST             ;  SAVE PROD SGN IN PSL<N>
                                    ;32431
                                    ;32432          ;-----------------------;
                                    ;32433          Q_LA,                   ;  PROD OPERAND TO Q-REG
U 1512, 0000,003C,01C0,F900,0000,12F8 ;32434        LC_RC[TO]               ;  LOAD LC WITH MULR-1
                                    ;32435
                                    ;32436
                                    ;32437  =**0    ;BRANCH ON FPA SYNC (BEN/ACCEL)
                                    ;32438
                                    ;32439  X.MUL.1:;**0--------------------;
                                    ;32440          D_ACCEL&SYNC,           ;  GET RESULT <31-00> IF RDY
U 12F8, 0A00,067C,0180,F800,0000,12F8 ;32441        ACCEL?, J/X.MUL.1       ;
                                    ;32442
                                    ;32443          ;**1--------------------;
                                    ;32444          Q_ACCEL&SYNC,           ;  GET RESULT<63-32> IF RDY
                                    ;32445          LAB_R[R15],             ;  LOAD ADJUSTED PROD OPERAND IN LA
U 12F9, 0000,0D7C,01D8,FA78,0000,15E3 ;32446        Q31?                    ;
                                    ;32447
                                    ;32448  =011    ;BRANCH ON Q31(BEN/SIGNS)
                                    ;32449
                                    ;32450  X.MUL3: ;011--------------------;  **MULD/PROD OPERAND SIGN=0 **
                                    ;32451          R(PRN)_D,               ;  RESULT TO DEST REG
                                    ;32452          N&Z_ALU.V&C_0,          ;  SET N & Z PER RESULT, CLEAR C&V
U 15E3, 0001,1A3C,0180,F8D8,0050,1497 ;32453        PSL.N?, J/X.MUL5        ;
                                    ;32454
                                    ;32455          ;111--------------------;  **MULD/PROD OPERAND SGN =1 **
                                    ;32456          R(PRN)_D,               ;  RESULT TO DEST REG
                                    ;32457          N&Z_ALU.V&C_0,          ;  SET N & Z PER RESULT, CLEAR C&V
U 15E7, 0001,1A3C,0180,F8D8,0050,14A7 ;32458        PSL.N?, J/X.MUL11       ;
```

```
                                ;32459  =0111
                                ;32460  X.MUL5:;0111--------------------;  MULR SGN=0, MULD/PROD SGN=0
                                ;32461          ALU_Q,  SET.CC(INST),   ;  TEST UPPER HALF FOR OVERFLOW
                                ;32462          Q_ID[CES],              ;  GET CP ERROR STATUS REGISTER
                                ;32463          C[R.IB.OPC, PC_PC+1,    ;  SETUP FOR NEXT INSTR
U 1497, C001,ED3C,31F0,2C04,4070,15F6   ;32464          D31?,   J/X.MUL7        ;
                                ;32465
                                ;32466          ;1111-------------------;  MULR SGN=1, MULD/PROD SGN=0
                                ;32467          ALU_Q+LB+1,     D_Q,    ;  [RESULT<63-32>+(-MULD/PROD)+1]
                                ;32468          SET.CC(INST),           ;  TEST UPPER HALF FOR OVERFLOW
                                ;32469          Q_ID[CES],              ;  GET CP ERROR STATUS REGISTER
                                ;32470          C[R.IB.OPC, PC_PC+1,    ;  SETUP FOR NEXT INSTR
U 149F, CC0D,ED10,31F0,2C04,4070,1606   ;32471          SIGNS?, J/X.MUL9        ;
                                ;32472
                                ;32473  =110
                                ;32474  X.MUL7: ;110--------------------;  ** RESULT SGN=0 **
U 15F6, F80C,003B,01F1,F857,139B,6000   ;32475          IRD                     ;  FINISHED
                                ;32476
                                ;32477          ;111--------------------;  **OVERFLOW, POS RESULT SIGN =1 **
U 15F7, 0819,2030,6580,F800,0000,1513   ;32478          D_Q.OR.K[.10]           ;  SET EXCEPTION FLAG
                                ;32479
                                ;32480  X.MUL8: ;----------------------;
                                ;32481          ID[CES]_D,              ;  RESTORE CES
                                ;32482          SET.V,                  ;
U 1513, 0000,003C,3180,3C00,0020,0062   ;32483          J/IRD                   ;  GO TAKE EXCEPTION IF ENABLED
                                ;32484
                                ;32485  =110
                                ;32486  X.MUL9:
                                ;32487          ;110--------------------;  ** OVERFLOW, NEG RESULT SIGN =0  **
                                ;32488          Q_Q.OR.K[.10], D_Q,     ;  SET EXCEPTION FLAG
U 1606, 0C19,2D30,65C0,F800,0000,1655   ;32489          D.NE.0?,  J/X.MUL10     ;
                                ;32490
                                ;32491          ;111--------------------;  **GOOD RESULT**
U 1607, F80C,003B,01F1,F857,139B,6000   ;32492          IRD                     ;  FINISHED
                                ;32493
                                ;32494  =101
                                ;32495  X.MUL10:;101--------------------;   ** ZERO RESULT**
                                ;32496          ID[CES]_D,              ;  RESTORE CES REG TO ORGINAL CONTENTS
                                ;32497          ALU_0(A),N&Z_ALU.V&C_0, ;  SET PSL<Z>
U 1655, 0003,003C,3180,3C00,0050,0062   ;32498          J/IRD                   ;
                                ;32499
                                ;32500          ;111--------------------;
U 1657, 0C00,003C,0180,F800,0000,1520   ;32501          D_Q                     ;  CES CONTENTS WITH EXCEPTION FLAG SET
                                ;32502
                                ;32503          ;----------------------;
                                ;32504          ID[CES]_D,              ;  RESTORE CES
                                ;32505          SET.V,                  ;
U 1520, 0000,003C,3180,3C00,0020,0062   ;32506          J/IRD                   ;  GO TAKE EXCEPTION IF ENABLED
```

```
                                    ;32507  =0111   ;BRANCH ON  PSL.N(BEN/PSL CC)
                                    ;32508
                                    ;32509  X.MUL11:;0111--------------------;  ** MULR SGN=0, MULD/PROD SGN=1 **
                                    ;32510          ALU_Q-LC,       D_Q,     ;  [RESULT<63-32>-(MULR-1)]
                                    ;32511          SET.CC(INST),            ;  SET OVERFLOW IF RESULT<63-32>+1 NOT ZERO
                                    ;32512          Q_ID[CES],               ;  GET CP ERROR/STATUS REG
                                    ;32513          C[R.IB.OPC,              ;
                                    ;32514          PC_PC+1,                 ;
U 14A7, CC11,ED00,31F0,2C04,4070,1606 ;32515        SIGNS?, J/X.MUL9         ;
                                    ;32516
                                    ;32517
                                    ;32518          ;1111--------------------;  **  MULR SGN=1, MULD/PROD SGN=1  **
U 14AF, 0011,2008,01C0,F800,0000,1521 ;32519        Q_Q-LC-1                 ;  [RESULT<63-32>-(MULR-1)-1]
                                    ;32520
                                    ;32521          ;----------------------;
                                    ;32522          ALU_Q+LB,                ;  [(RESULT<63-32>-MULR)+(-MULD/PROD)]
                                    ;32523          SET.CC(INST),            ;  SET OVERFLOW IF RESULT<63-32> NOT ZERO
                                    ;32524          Q_ID[CES],               ;  GET CP ERROR/STATUS REG
                                    ;32525          C[R.IB.OPC,              ;
                                    ;32526          PC_PC+1,                 ;
U 1521, C00D,ED14,31F0,2C04,4070,15F6 ;32527        D3T?,   J/X.MUL7         ;
```

```
                                        ;32528  .TOC "  FPA Interface         : MULL2, MULL3                - Memory destination"
                                        ;32529
                                        ;32530  ;
                                        ;32531  ;                        1)  IF (*,*), PROD ADDR/VA-REG
                                        ;32532  ;                        2)  MULR OPERAND/Q-REG
                                        ;32533  ;                        3)  PROD/MULD OPERAND IN D-REG
                                        ;32534
                                        ;32535  ;MULL2,3(*,*) & (*,*,*)  C-FORK ENTRY
                                        ;32536  1381:   ;-----------------------;
                                        ;32537          ID_D.SYNC,              ;  SEND LAST OPERAND TO FPA
U 1381, 001F,2040,1580,BEF8,0000,1410   ;32538          R[R15]_0-D              ;  SAVE 2'S COMPLEMENT OF MULD/PROD OPERAND
                                        ;32539
                                        ;32540  =00****0 ; CALL CONSTRAINT BLOCK
                                        ;32541
                                        ;32542          ;-----------------------;
                                        ;32543          RC[T0]_Q-K[.1],         ;  SAVE ADJUSTED MULR
                                        ;32544          N_AMX.Z_TST,            ;  SAVE MULR SGN IN 'N'
                                        ;32545          Q_D,                    ;
                                        ;32546          CALL,                   ;  SET UP RETURN FROM ASPC ROUTINE
U 1410, 0019,3B01,05E0,F980,0030,14BD   ;32547          IR0?,   J/X.MUL21       ;
                                        ;32548
                                        ;32549  =11****0
                                        ;32550
                                        ;32551  X.MUL19:;11****0----------------;  ** ASRC RETURN60, MEM DST **
                                        ;32552          D_ACCEL&SYNC,           ;  ACCEPT ANSWER<31-00>/FPA
                                        ;32553          LAB_R[R15],             ;  GET -(MULD/PROD OPERAND)
U 1470, 0A00,0D7C,0180,FA78,0000,1633   ;32554          Q31?,   J/X.MUL25       ;
                                        ;32555
                                        ;32556          ;11****1----------------;  ** ASRC RETURN61, REG DST **
                                        ;32557          D_ACCEL&SYNC,           ;ACCEPT ANSWER<31-00>/FPA
U 1471, 0A00,007C,0180,FA78,0000,1522   ;32558          LAB_R[R15]              ;  GET -(MULD/PROD OPERAND)
                                        ;32559
                                        ;32560          ;-----------------------;
                                        ;32561          LC_RC[T0],              ;  GET (MULR-1)
                                        ;32562          Q_ACCEL&SYNC,           ;  ACCEPT RESULT<63-32>/FPA
U 1522, 0000,0D7C,01D8,F900,0000,15E3   ;32563          Q31?,   J/X.MUL3        ;  MERGE IN (*,R) FLOW
                                        ;32564
                                        ;32565
                                        ;32566  =1101 ;BRANCH ON NUMBER OF SPECIFIERS (BEN/ALU)
                                        ;32567
                                        ;32568  X.MUL21:;1101--------------------;  ** TWO ADDRESS TYPE **
U 14BD, 0000,003C,0180,F800,0000,1523   ;32569          J/X.MUL23               ;  WAIT, FPA NOT READY
                                        ;32570
                                        ;32571          ;1111--------------------;  ** THREE ADDRESS TYPE **
                                        ;32572          Q_IB.DATA,              ;  GET SPECIFIER DATA FROM ISTREAM
                                        ;32573          LAB_R(SP1),             ;  LOAD LATCHES FROM BASE SPECIFIER
                                        ;32574          CLR.IB.COND,            ;  DISCARD BASE OPERAND SPECIFIER
                                        ;32575          PC_PC+N,                ;  STEP PC OVER IT
                                        ;32576          MCT/ALLOW.IB.READ,      ;  LET IB GET NEEDED DATA
U 14BF, F000,003F,01F0,F847,0000,0400   ;32577          SUB/SPEC,J/ASPC.B       ;  EVALUATE THE SPECIFIER
                                        ;32578
                                        ;32579  X.MUL23:;-----------------------;
U 1523, 0000,003C,0180,F800,0000,1470   ;32580          J/X.MUL19               ;  WAIT, FPA NOT READY WITH RESULT
```

```
                                    ;32581  =011     ;BRANCH ON Q31(BEN/SIGNS)
                                    ;32582
                                    ;32583  X.MUL25::;011--------------------;  **  MULD/PROD SGN=0  **
                                    ;32584          CACHE_D[LONG],          ;  STORE RESULT<31-00> IN MEM DST
                                    ;32585          ALU_D,  N&Z_ALU.V&C_0,  ;  SET PSL N&Z /RESULT
                                    ;32586          Q_ACCEL&SYNC,           ;  GET RESULT<63-32>
                                    ;32587          LC_RC[TO],              ;  GET (MULR-1)
U 1633, 0001,1A7C,01D8,3100,0050,1497;32588         PSL.N?, J/X.MUL5        ;
                                    ;32589
                                    ;32590          ;111--------------------;**  MULD/PROD OPERAND SGN=1  *******
                                    ;32591          CACHE_D[LONG],          ;  STORE RESULT<31-00> IN MEM DST
                                    ;32592          ALU_D,  N&Z_ALU.V&C_0,  ;  SET PSL N&Z /RESULT
                                    ;32593          Q_ACCEL&SYNC,           ;  GET RESULT<63-32>
                                    ;32594          LC_RC[TO],              ;  GET (MULR-1)
U 1637, 0001,1A7C,01D8,3100,0050,14A7;32595         PSL.N?, J/X.MUL11       ;
```

```
                                ;32596  .TOC "  FPA Interface         : (MULL, MULF, DIVF)(2,3)    - (REG,REG) and (S^#,REG)
                                ;32597
                                ;32598  ;DIVF2,MULF2,MULL2 (REG,REG) A-FORK ENTRY
                                ;32599  ;ENTRY STATUS:  1)  MULD/PROD IN GPR<SP1>
                                ;32600  ;               2)  MULR OPERAND IN D-REG
                                ;32601  ;               3)  MULD/PROD OPERAND IN LA-REG
                                ;32602
                                ;32603  1004:   ;-------------------------------;
U 1004, 001F,2600,0180,FAF8,0000,1673   ;32604          R[R15]_0-D,ACCEL?,J/X.OML       ;  2'S COMPLEMENT OF MULR OPERAND
                                ;32605
                                ;32606
                                ;32607  ;DIVF2,MULF2,MULL2 (S^#, REG) A-FORK ENTRY
                                ;32608  ;ENTRY STATUS:  1)  MULD/PROD<SP1>
                                ;32609  ;               2)  MULR OPERAND IN Q-REG
                                ;32610  ;               3)  MULD/PROD OPERAND IN LA-REG
                                ;32611
                                ;32612  1000:   ;-------------------------------;
U 1000, 0C1F,0600,0180,FAF8,0000,1673   ;32613          R[R15]_0-Q,D_Q,ACCEL?,J/X.OML   ;  2'S COMPLEMENT OF MULR OPERAND
                                ;32614
                                ;32615
                                ;32616  ;DIVF3,MULF3,MULL3 (*,REG,REG) B-FORK ENTRY
                                ;32617  ;ENTRY STATUS:  1)  MULD/PROD IN GPR<SP1>
                                ;32618  ;               2)  MULR OPERAND IN D-REG
                                ;32619  ;               3)  MULD/PROD OPERAND IN LA-REG
                                ;32620
                                ;32621  1204:   ;-------------------------------;
                                ;32622          ID_D.SYNC,R[R15]_0-D,           ; SEND MULR OPERAND, 2'S COMP OF MULR
U 1204, 001F,2640,1580,BEF8,0000,1673   ;32623          ACCEL?,J/X.OML                  ;
                                ;32624
                                ;32625  =011
                                ;32626  X.OML:  ;011----------------------------;
                                ;32627          RC[TO]_LA-K[.1],                ;  SAVE MULD/PROD OPERAND -1
U 1673, 0018,0000,0580,F980,0030,1528   ;32628          N_AMX.Z_TST,J/X.OML.1A          ;  SAVE MULD/PROD OPERAND SIGN IN PSL<N>
                                ;32629
                                ;32630          ;111----------------------------;  *** NOT MULL, GO TO FLOATING RTN ***
                                ;32631          RC[TO]_LA,                      ;  GET FPA RESULT IF READY,SAVE OPERAND
U 1677, 0000,003C,0180,F980,0000,14D2   ;32632          J/XF.A.01                       ;
                                ;32633
                                ;32634
                                ;32635  ;DIVF3,MULF3,MULL3 (*,S^#,REG) B-FORK ENTRY
                                ;32636  ;ENTRY STATUS:  1)  MULD/PROD IN GPR<SP1>
                                ;32637  ;               2)  MULR OPERAND IN D-REG
                                ;32638  ;               3)  MULD/PROD OPERAND IN Q-REG
                                ;32639
                                ;32640  1200:   ;-------------------------------;
U 1200, 001F,2640,1580,BEF8,0000,1683   ;32641          ID_D.SYNC,R[R15]_0-D,ACCEL?     ; LAST OP TO ACC,2'S COMP OF MULR
                                ;32642
                                ;32643  =011    ;011----------------------------;
                                ;32644          RC[TO]_Q-K[.1],N_AMX.Z_TST,     ; SAVE RESULT -1, PSL<N> <- SIGN
U 1683, 0019,2000,0580,F980,0030,1528   ;32645          J/X.OML.1A                      ;
                                ;32646
                                ;32647          ;111----------------------------;  *** MULF OR DIVF INSTRUCTION ***
                                ;32648          RC[TO]_LA,                      ;  GET FPA RESULT IF READY,SAVE OPERAND
U 1687, 0000,003C,0180,F980,0000,14D2   ;32649          J/XF.A.01                       ;
```

```
                                    ;32650  X.OML.1A:
                                    ;32651          ;--------------------------------;
                                    ;32652          Q_D,D_ACCEL&SYNC,                ;  GET RESULT IF READY
U 1528, 0A00,067C,01E0,F900,0000,1314 ;32653        LC_RC[TO],ACCEL?,J/X.OML.1       ;  GET MULD/PROD-1
                                    ;32654  =**0
                                    ;32655  X.OML.1:;**0-----------------------------;
U 1314, 0A00,067C,0180,F800,0000,1314 ;32656        D_ACCEL&SYNC,ACCEL?,J/X.OML.1    ;  GET RESULT <31-00> IF RDY
                                    ;32657
                                    ;32658          ;**1-----------------------------;
U 1315, 0000,0D7C,01D8,FA78,0000,1693 ;32659        Q_ACCEL&SYNC,LAB_R[R15],Q31?     ; GET RESULT<63-32>, LA_ADJUSTED MULR
                                    ;32660  =011
                                    ;32661  X.OML3: ;011-----------------------------;  ** MULR SIGN=0 **
                                    ;32662          R(SP1)_D,N&Z_ALU.V&C_0,          ;  RESULT TO DEST REG
U 1693, 0001,1A3C,0180,F8C0,0050,14C7 ;32663        PSL.N?,J/X.OML5                  ;
                                    ;32664
                                    ;32665          ;111-----------------------------;  **  MULR SGN =1 **
                                    ;32666          R(SP1)_D,N&Z_ALU.V&C_0.          ;  RESULT TO DEST REG
U 1697, 0001,1A3C,0180,F8C0,0050,14D7 ;32667        PSL.N?,J/X.OML11                 ;
                                    ;32668  =0111
                                    ;32669  X.OML5:;0111-----------------------------;  MULD/PROD SGN=0, MULR SGN=0
                                    ;32670          ALU_Q,SET.CC(INST),              ;  TEST UPPER HALF FOR OVERFLOW
                                    ;32671          Q_ID[CES],                       ;  GET CP ERROR STATUS REGISTER
U 14C7, 4001,ED3C,31F0,2C05,4070,16B6 ;32672        C[R.IB0-1,PC_PC+2,D31?,J/X.OML7  ;  SETUP FOR NEXT INSTR
                                    ;32673
                                    ;32674          ;1111----------------------------;  MULD/PROD SGN=1, MULR SGN=0
                                    ;32675          ALU_Q+LB+1,D_Q,                  ;  [RESULT<63-32>+(-MULR)+1]
                                    ;32676          SET.CC(INST),                    ;  TEST UPPER HALF FOR OVERFLOW
                                    ;32677          Q_ID[CES],CLR.IB0-1,             ;  GET CP ERROR STATUS REGISTER
U 14CF, 4C0D,ED10,31F0,2C05,4070,1606 ;32678        PC_PC+2,SIGNS?,J/X.MUL9          ;
                                    ;32679  =110
                                    ;32680  X.OML7: ;110-----------------------------;  ** RESULT SGN=0 **
U 16B6, F80C,003B,01F1,F857,139B,6000 ;32681        IRD                              ;  FINISHED
                                    ;32682
                                    ;32683          ;111-----------------------------;  **OVERFLOW, POS RESULT SIGN =1 **
U 16B7, 0819,2030,6580,F800,0000,1529 ;32684        D_Q.OR.K[.10]                    ;  SET EXCEPTION FLAG
                                    ;32685
                                    ;32686  X.OML8: ;-------------------------------;
U 1529, 0000,003C,3180,3C00,0020,0062 ;32687        ID[CES]_D,SET.V,J/IRD            ;  RESTORE CES
                                    ;32688  =0111
                                    ;32689  X.OML11:;0111----------------------------;  ** MULD/PROD SGN=0, MULR SGN=1 **
                                    ;32690          ALU_Q-LC,D_Q,                    ;  [RESULT<63-32>-(MULD/PROD-1)]
                                    ;32691          SET.CC(INST),                    ;  SET V IF RESULT<63-32>+1 NOT ZERO
                                    ;32692          Q_ID[CES],CLR.IB0-1,             ;  GET CP ERROR/STATUS REG
U 14D7, 4C11,ED00,31F0,2C05,4070,1606 ;32693        PC_PC+2,SIGNS?,J/X.MUL9          ;
                                    ;32694
                                    ;32695          ;1111----------------------------;  **  MULD/PROD SGN=1, MULR SGN=1  **
U 14DF, 0011,2008,01C0,F800,0000,152A ;32696        Q_Q-LC-1                         ;  [RESULT<63-32>-(MULD/PROD-1)-1]
                                    ;32697
                                    ;32698          ;--------------------------------;
                                    ;32699          ALU_Q+LB,                        ;  [(RESULT<63-32>-MULD/PROD)+(-MULR)]
                                    ;32700          SET.CC(INST),                    ;  SET V IF RESULT<63-32> NOT ZERO
                                    ;32701          Q_ID[CES],CLR.IB0-1,             ;  GET CP ERROR/STATUS REG
U 152A, 400D,ED14,31F0,2C05,4070,16B6 ;32702        PC_PC+2,D31?,J/X.OML7            ;
```

```
                                ;32703  .TOC "  FPA Interface         : POLYF, and POLYD"
                                ;32704
                                ;32705  ;THE FPA HANDLES DETECTION OF
                                ;32706  ;                A) UNDER & OVERFLOW
                                ;32707  ;                B) RESERVED OPERANDS ON COEFFICIENTS
                                ;32708
                                ;32709
                                ;32710
                                ;32711  ;POLYF C-FORK ENTRY
                                ;3271   ;       D-REG=DEGREE OPERAND
                                ;32713  ;       Q-REG=ARGUMENT OPERAND(ALREADY SENT TO FPA)
                                ;32714  ;
                                ;32715  13C2:   ;-------------------------------;
                                ;32716          ALU_D.ANDNOT.K[.1F],WORD,       ; TEST FOR DEGREE GREATER THAN 31
U 13C2, 0C19,4024,8DE0,F800,0010,101A ;32717    CLK.UBCC,D_Q,Q_D,J/X.POLY1      ; MOVE ARG TO D-REG, DEGREE TO Q-REG
                                ;32718
                                ;32719
                                ;32720
                                ;32721  ;POLYD C-FORK ENTRY
                                ;32722  ;       D-REG=DEGREE OPERAND
                                ;32723  ;       Q-REG=ARGUMENT OPERAND (L.O.), SENT TO FPA
                                ;32724  ;       RC[T0]=ARGUMENT OPERAND (H.O.), SENT TO FPA
                                ;32725  ;
                                ;32726  1385:   ;-------------------------------;
                                ;32727          ALU_D.ANDNOT.K[.1F],WORD,       ; TEST DEGREE FOR GREATER THAN 31
U 1385, 0C19,4024,8DE0,F800,0010,152B ;32728    CLK.UBCC,D_Q,Q_D                ; MOVE ARG(L.O.) TO D, DEGREE TO Q-REG
                                ;32729
                                ;32730          ;-------------------------------;
U 152B, 0810,0038,0180,F900,0000,101A ;32731    D_RC[T0]                        ; MOVE ARG(H0) INTO D-REG
                                ;32732
                                ;32733
                                ;32734  =00*****
                                ;32735  X.POLY1:;00*****----------------------;
                                ;32736          CALL,ALU_D,MSC/CHK.FLT.OPR,     ; TRAP IF ARG IS RESERVED OPERAND
U 101A, 0C01,013D,0180,F800,0800,132C ;32737    D_Q,Z?,J7X.POLY3                ; WAS DEGREE RESERVED OPERAND?
                                ;32738
                                ;32739  =11*****;11*****----------------------; *** RETURN 60 FROM ASPC ***
                                ;32740          RC[T2]_Q.0XT[BYTE],CLK.UBCC,    ; SAVE DEGREE IN RC<2>, CHECK FOR 0
U 107A, 0003,A03F,0180,F990,0010,1048 ;32741    SUB/SPEC                        ; ADVANCE DATA TYPE TO SINGLE OR DOUBLE
                                ;32742
                                ;32743  1048:   ;-------------------------------;
                                ;32744          D[LONG]_CACHE,RC[T3]_D+K[.4],   ; GET COEFFICIENT, ADVANCE TABLE ADDRESS
U 1048, 0019,0114,1180,4198,0000,1334 ;32745    Z?,J/X.POLY7                    ; WAS DEGREE=0?
                                ;32746
                                ;32747  =0
                                ;32748  X.POLY3:;0-----------------------------; *** DEGREE GREATER THAN 31 ***
U 132C, 0000,003C,0180,F800,0000,0106 ;32749    J/RSVOPR                        ; TAKE RESERVED OPERAND TRAP
                                ;32750
                                ;32751          ;01-----------------------------; *** DEGREE OPERAND OK ***
                                ;32752          Q_IB.DATA,LAB_R(SP1),LAB_R(SP1),; EVALUATE TABLE ADDRESS SPECIFIER
                                ;32753          CLR.IB.COND,PL_PC+N,            ;
                                ;32754          MCT/ALLOW.IB.READ,              ;
U 132D, F000,003F,01F0,F847,0000,0400 ;32755    SUB/SPEC,J/ASPC.B               ;
```

```
                                ;32756  =0100
                                ;32757  X.POLY5::;0100----------------------------; *** BASIC SYNC LOOP ***
                                ;32758          Q_RC[T3],ID_D.SYNC,               ; GET TABLE ADDRESS, COEFFICIENT TO FPA
U 14E4, 0010,0678,15C0,BD18,0000,14E4   ;32759          ACCEL?,J/X.POLY5                  ; DID FPA TAKE IT?
                                ;32760
                                ;32761          ;0101----------------------------; *** FPA TOOK IT & LAST ONE OK ***
                                ;32762          D[LONG]_CACHE,RC[T3]_Q+K[.4],     ; GET NEXT COEFF, ADVANCE TABLE ADDRESS
U 14E5, 0019,2114,1180,4198,0000,1334   ;32763          Z?,J/X.POLY7                      ; WAS DEGREE=0?
                                ;32764
                                ;32765          ;0110----------------------------; ***  EXCEPTION ON LAST COEFFICIENT ***
                                ;32766          Q_RC[T3],ID_D.SYNC,               ; GET TABLE ADDRESS, COEFFICIENT TO FPA
U 14E6, 0010,0678,15C0,BD18,0000,14E4   ;32767          ACCEL?,J/X.POLY5                  ; DID FPA TAKE IT?
                                ;32768
                                ;32769          ;0111----------------------------; ***  EXCEPTION ON LAST COEFFICIENT ***
                                ;32770          LOAD.ACC.CC,SC_K[.1B],            ; LOAD PSL<CC>S, '27' TO SC
U 14E7, 0000,003D,ED80,F800,1884,7533   ;32771          CALL,J/X.POLY40                   ; GO AND HANDLE EXCEPT
                                ;32772
                                ;32773  =1111   ;1111----------------------------; *** RETURN9, UNDERFLOW RETURN ***
                                ;32774          ACF/SYNC,D[LONG]_CACHE,           ; TELL FPA TO PROCEED, GET NEXT COEFF
U 14EF, 0019,2154,1180,4198,0000,1334   ;32775          RC[T3]_Q+K[.4],Z?,J/X.POLY7       ; ADVANCE TABLE ADDRESS, WAS DEGREE=0?
                                ;32776
                                ;32777  =0
                                ;32778  X.POLY7::;0----------------------------; *** DEGREE NEQ 0 ***
                                ;32779          VA_VA+4,Q_RC[T2],                 ; ADVANCE TAB, G[illegible] DEGREE
U 1334, 0010,0E38,01C0,F913,0000,16BE   ;32780          INTERRUPT.REQ?,J/X.POLY9          ; GO & CHECK FOR INTERRUPT REQUEST
                                ;32781
                                ;32782          ;1----------------------------; *** POLY DONE ***
U 1335, 0010,0838,01C0,F91B,C000,16D6   ;32783          Q_RC[T3],VA_VA+4,DBL?,J/X.POLY20; Q GET TAB ADDR, ADVANCE TAB ADDR IN VA
                                ;32784
                                ;32785  =110
                                ;32786  X.POLY9::;110----------------------------; *** NO INTERRUPT ***
                                ;32787          RC[T2]_Q-K[.1],CLK.UBCC,BYTE,     ; SET ALU.Z IF ZERO, DEC DEGREE BY ONE
U 16BE, 0019,A800,0580,F990,0010,16C6   ;32788          DBL?,J/X.POLY11                   ; GO CHECK FOR POLYD
                                ;32789
                                ;32790          ;111----------------------------; *** INTERRUPT PENDING ***
U 16BF, 0000,003C,0180,F800,0000,121C   ;32791          J/POLY.INT                        ;
                                ;32792
                                ;32793  =110
                                ;32794  X.POLY11:
                                ;32795          ;110----------------------------; *** SINGLE PRECISION ***
                                ;32796          Q_RC[T3],ID_D.SYNC,               ; GET TAB ADDR OP, SEND COEF TO FPA
U 16C6, 0C10,0678,15C0,BD18,0000,14E4   ;32797          ACCEL?,J/X.POLY5                  ; DID IT GET IT?
                                ;32798
                                ;32799          ;111----------------------------; *** DOUBLE PRECISION ***
U 16C7, 0010,0678,15C0,BD18,0000,16CE   ;32800          Q_RC[T3],ID_D.SYNC,ACCEL?         ; GET TABLE ADD, SEND 1st HALF TO FPA
                                ;32801
                                ;32802  =110    ;110----------------------------; WAIT FOR FPA RESPONSE
U 16CE, 0000,067C,1580,BC00,0000,16CE   ;32803          ID_D.SYNC,ACCEL?,J/<.>            ; GIVE FIRST HALF TO FPA (H.O.)
                                ;32804
                                ;32805          ;111----------------------------; *** FPA TOOK IT ***
U 16CF, 0019,2014,1180,4198,0000,152C   ;32806          D[LONG]_CACHE,RC[T3]_Q+K[.4]      ; INCREMENT TABLE ADDRESS
                                ;32807
                                ;32808          ;----------------------------;
                                ;32809          Q_RC[T3],VA_VA+4,                 ; GET TABLE ADDR OPERAND
U 152C, 0010,0678,15C0,BD18,0000,14E4   ;32810          ID_D.SYNC,ACCEL?,J/X.POLY5        ; SEND COEFFICIENT TO FPA
```

```
                                ;32811  ;POLY DONE
                                ;32812  =110
                                ;32813  X.POLY20:
                                ;32814          ;110------------------------------; *** SINGLE PRECISION ***
                                ;32815          ID_D&NO.SYNC,POLY.DONE,          ; SEND COEF TO FPA, SEND POLYDONE SIGNAL
U 16D6, 0000,06FC,1700,BC00,0000,14F4   ;32816          ACCEL?,J/X.POLY21                ; DID IT GET IT?
                                ;32817
                                ;32818          ;111------------------------------; *** DOUBLE PRECISION ***
U 16D7, 0000,06FC,1700,3C00,0000,16DE   ;32819          ID_D&NO.SYNC,POLY.DONE,ACCEL?    ; GIVE 1st HALF TO FPA, SEND DONE SIGNAL
                                ;32820
                                ;32821  =110    ;110------------------------------; WAIT FOR FPA RESPONSE
                                ;32822          ID_D&NO.SYNC,POLY.DONE,          ; GIVE 1st HALF TO FPA, SEND DONE SIGNAL
U 16DE, 0000,06FC,1700,BC00,0000,16DE   ;32823          ACCEL?,J/<.>                     ;
                                ;32824
                                ;32825          ;111------------------------------; *** FPA TOOK IT ***
U 16DF, 0019,2014,1180,4198,0000,152D   ;32826          D[LONG]_CACHE,RC[T3]_Q+K[.4]     ; INCREMENT TABLE ADDRESS
                                ;32827
                                ;32828          ;--------------------------------;
                                ;32829          VA_VA+4,                         ; UPDATE VA-REG
                                ;32830          Q_RC[T3],                        ; GET TABLE ADDR OPERAND
                                ;32831          ID_D&NO.SYNC,                    ; SEND COEFFICIENT TO FPA
U 152D, 0010,06F8,1740,BD1B,0000,14F4   ;32832          POLY.DONE,ACCEL?,J/X.POLY21      ; DID IT GET IT?
                                ;32833
                                ;32834  =0100
                                ;32835  X.POLY21:
                                ;32836          ;0100-----------------------------; *** BASIC SYNC LOOP ***
                                ;32837          Q_RC[T3],                        ; GET TABLE ADDRESS
                                ;32838          ID_D&NO.SYNC,                    ; SEND COEFFICIENT TO FPA
                                ;32839          POLY.DONE,                       ; SEND POLY DONE TO FPA
U 14F4, 0010,06F8,1740,BD18,0000,14F4   ;32840          ACCEL?,J/X.POLY21                ; DID FPA TAKE IT?
                                ;32841
                                ;32842          ;0101------------------------------; *** FPA TOOK IT & LAST ONE OK ***
                                ;32843          D_ACCEL&SYNC,                    ; GET FIRST HALF OF RESULT
                                ;32844          ALU_RC[T2],CLK.UBCC,             ; SET ALU.Z IF NO UNDERFLOW
U 14F5, 0A10,0678,0180,F910,0010,1534   ;32845          ACCEL?,J/X.POLY31                ; DID IT SET IT?
                                ;32846
                                ;32847          ;0110-----------------------------; ***  EXCEPTION ON LAST COEFFICIENT ***
                                ;32848          Q_RC[T3],                        ; GET TABLE ADDRESS
                                ;32849          ID_D&NO.SYNC,                    ; SEND COEFFICIENT TO FPA
                                ;32850          POLY.DONE,                       ; SEND POLY DONE TO FPA
U 14F6, 0010,06F8,1740,BD18,0000,14F4   ;32851          ACCEL?,J/X.POLY21                ; DID FPA TAKE IT?
                                ;32852
                                ;32853          ;0111-----------------------------; ***  EXCEPTION ON LAST COEFFICIENT ***
                                ;32854          LOAD.ACC.CC,SC_K[.1B],           ; LOAD PSL<CC>S, '27' TO SC
U 14F7, 0000,003D,ED80,F800,1884,7533   ;32855          CALL,J/X.POLY40                  ; GO AND HANDLE EXCEPT
                                ;32856
                                ;32857  =1111   ;1111-----------------------------; *** RETURN, UNDERFLOW RETURN ***
                                ;32858          ACF/SYNC,                        ; TELL FPA TO PROCEED
U 14FF, 0010,0078,0180,F910,0010,1534   ;32859          ALU_RC[T2],CLK.UBCC,J/X.POLY31   ; SET ALU.Z IF NO UNDERFLOW
```

```
                                    ;32860  ;POLY DONE CLEANUP ROUTINE
                                    ;32861  =0100
                                    ;32862  X.POLY31:
                                    ;32863          ;0100-------------------------; *** WAIT LOOP **
U 1534, 0A00,067C,0180,F800,0000,1534 ;32864        D_ACCEL&SYNC,ACCEL?,J/X.POLY31  ; ACCEPT H.O. RESULT
                                    ;32865
                                    ;32866          ;0101-------------------------; *** GOT H.O. RESUT & OK ***
U 1535, 0000,083C,0180,F800,1800,16E6 ;32867        LOAD.ACC.CC,DBL?,J/X.POLY33     ; LOAD PSL<CC>S
                                    ;32868
                                    ;32869          ;0110-------------------------; ***  EXCEPTION ON LAST COEFFICIENT ***
                                    ;32870          LOAD.ACC.CC,SC_K[.1B],          ; LOAD PSL<CC>S, '27' TO SC
U 1536, 0000,003D,ED80,F800,1884,7533 ;32871        CALL,J/X.POLY40                 ; GO AND HANDLE EXCEPT
                                    ;32872
                                    ;32873          ;0111-------------------------; ***  EXCEPTION ON LAST COEFFICIENT ***
                                    ;32874          LOAD.ACC.CC,SC_K[.1B],          ; LOAD PSL<CC>S, '27' TO SC
U 1537, 0000,003D,ED80,F800,1884,7533 ;32875        CALL,J/X.POLY40                 ; GO AND HANDLE EXCEPT
                                    ;32876
                                    ;32877  =1111   ;1111-------------------------; ***  UNDERFLOW RETURN ***
                                    ;32878          D_K[ZERO],Q_0,                  ; RESULTS GET ZERO
U 153F, 0818,0038,19F8,F800,0050,16E6 ;32879        N&Z_ALU.V&C_0,D_K[ZERO]         ; SET PSL<cc>'S
                                    ;32880
                                    ;32881  =110
                                    ;32882  X.POLY33:
                                    ;32883          ;110--------------------------; *** SINGLE PRECISION ***
U 16E6, 0000,063C,0180,F918,0000,1525 ;32884        LC_RC[T3],ACCEL?,J/X.POLY37     ; PREFETCH TABLE ADDRESS
                                    ;32885
                                    ;32886          ;111--------------------------; *** DOUBLE PRECISION ***
U 16E7, 0000,067C,01D8,F800,0000,16EE ;32887        Q_ACCEL&SYNC,ACCEL?             ; ACCEPT 2ND HALF (L.O.)
                                    ;32888
                                    ;32889  =110    ;110--------------------------; *** WAIT LOOP ***
U 16EE, 0000,067C,01D8,F800,0000,16EE ;32890        Q_ACCEL&SYNC,ACCEL?,J/<.>       ; ACCEPT 2ND HALF (L.O.)
                                    ;32891
                                    ;32892          ;111--------------------------; *** GOT 2ND HALF ***
U 16EF, 0000,063C,0180,F918,0000,1525 ;32893        LC_RC[T3],ACCEL?                ; GET TAB ADD,OVERFLOW OR UNDERFLOW?
                                    ;32894
                                    ;32895  =0101
                                    ;32896  X.POLY37:
                                    ;32897          ;0101-------------------------; *** NO EXCEPTIONS ***
U 1525, 0010,0038,0180,FA98,0000,152E ;32898        R[R3]_LC,J/X.POLY38             ; LOAD TABLE ADDRESS
                                    ;32899
                                    ;32900          ;0111-------------------------; *** EXCEPTION ***
U 1527, 0000,003D,ED80,F800,0084,7533 ;32901        SC_K[.1B],CALL,J/X.POLY40       ;
                                    ;32902
                                    ;32903  =1111   ;1111-------------------------; RET HERE IF UNDERFLOW WITH PSL<FU>=0
U 152F, 0F01,203C,01F8,FA98,0000,152E ;32904        R[R3]_Q,D_0,Q_0                 ; R3 GETS TABLE ADDRESS AND ZERO ANSWER
```

```
                                        ;32905  X.POLY38:
                                        ;32906          ;-------------------------------;
U 152E, 0001,003C,0180,FA80,0000,1530   ;32907          R[R0]_D                         ; STORE FIRST PART OF ANSWER
                                        ;32908          ;-------------------------------;
                                        ;32909
U 1530, 0018,0838,1980,FA90,0000,16F6   ;32910          R[R2]_K[ZERO],DBL?              ; ZERO REGISTER 2
                                        ;32911
                                        ;32912  =110    ;110----------------------------; SINGLE PRECISION
                                        ;32913          R[R1]_K[ZERO],                  ;
U 16F6, C018,0038,1980,FA8C,4000,0062   ;32914          CLR.IB.OPC,PC_PC+1,J/IRD        ;
                                        ;32915
                                        ;32916          ;111----------------------------; DOUBLE PRECISION
U 16F7, 0018,0038,1980,FAA0,0000,1531   ;32917          R[R4]_K[ZERO]                   ; ZERO REGISTER 4
                                        ;32918          ;-------------------------------;
                                        ;32919
U 1531, 0018,0038,1980,FAA8,0000,1532   ;32920          R[R5]_K[ZERO]                   ; ZERO REGISTER 5
                                        ;32921          ;-------------------------------;
                                        ;32922
                                        ;32923          R[R1]_Q,                        ; STORE LOW ORDER OF RESULT
U 1532, C001,203C,0180,FA8C,4000,0062   ;32924          CLR.IB.OPC,PC_PC+1,J/IRD        ;
```

```
                                    ;32925  ;POLY EXCEPTION SETUP ROUTINE
                                    ;32926  X.POLY40:
                                    ;32927          ;-------------------------------;
U 1533, 0000,003C,5DF0,2C00,0000,1538 ;32928        Q_ID[ACC.CS]                    ; GET FPA STATUS/CONTROL REG
                                    ;32929
                                    ;32930          ;-------------------------------;
U 1538, 0001,2024,0180,F800,0010,1539 ;32931        ALU_Q.ANDNOT.MASK,CLK.UBCC      ; SET ALU<Z> IF NO RESERVED OPERAND
                                    ;32932
                                    ;32933          ;-------------------------------;
U 1539, 0000,013C,3DF0,2C00,0000,1340 ;32934        Q_ID[PSL],Z?                    ; GET PSL IN CASE IT IS UNDERFLOW
                                    ;32935
                                    ;32936  =0      ;0------------------------------; RESERVED OPERAND
U 1340, 0000,003C,0180,F800,0000,0106 ;32937        J/RSVOPR                        ;
                                    ;32938
                                    ;32939          ;1------------------------------; OVERFLOW OR UNDERFLOW
U 1341, 0019,3A34,3180,F800,0082,154D ;32940        SC_Q.AND.K[.40],PSL.V?          ; ISOLATE PSL<fu> FOR UNDERFLOW
                                    ;32941
                                    ;32942  =1101   ;1101-----------------------------; UNDERFLOW
U 154D, 0000,123C,0180,F800,0000,155D ;32943        EALU?,J/X.POLY43                ; SEE IF PSL<fu> IS SET
                                    ;32944
                                    ;32945          ;1111-----------------------------; OVERFLOW
U 154F, 0018,0038,0180,F988,0000,12B4 ;32946        RC[T7]_K[.8],J/FLOAT.FAULT      ; LOAD OVERFLOW FAULT CODE
                                    ;32947
                                    ;32948  =1101
                                    ;32949  X.POLY43:
                                    ;32950          ;1101-----------------------------; PSL<fu> IS 0
U 155D, 0818,0038,4580,F800,0000,153A ;32951        Q_K[.8000],J/X.POLY45           ; VALUE TO RESET ACC.CS REG
                                    ;32952
                                    ;32953          ;1111-----------------------------; PSL<fu> IS 1
U 155F, 0018,0038,F580,F988,0000,12B4 ;32954        RC[T7]_K[.A],J/FLOAT.FAULT      ; LOAD UNDERFLOW CODE
                                    ;32955
                                    ;32956  X.POLY45:
                                    ;32957          ;-------------------------------; UNDERFLOW *******
                                    ;32958          ALU_RC[T7],CLK.UBCC,            ; SET ALU.Z IF DEGREE EQ ZERO
U 153A, 0010,0038,5D80,3D10,0010,153B ;32959        ID[ACC.CS]_D                    ; RESET FPA STATUS AND CNTL REG
                                    ;32960
                                    ;32961          ;-------------------------------;
U 153B, 0010,003A,01C0,F918,0000,0009 ;32962        Q_RC[T3],RETURN9                ; GET TABLE ADDRESS, RETURN TO CALLER
```

```
                                ;32963  .TOC "  FPA Interface          : EMODF"
                                ;32964
                                ;32965  ;-------------------------------------------------------------
                                ;32966
                                ;32967
                                ;32968  ;EMODF  (54)    MULR.RF, MULRX.RB, MULD.RF, INT.WL, FRACT.WF
                                ;32969  ;*************************************************************
                                ;32970
                                ;32971  ;THIS ROUTINE PROVIDES THE CPU INTERFACE TO THE FPA WHICH IS USED
                                ;32972  ;TO DO THE FRACTION MULTIPLY FOR EMODF.  WHILE THE FPA IS
                                ;32973  ;MULTIPLYING THE FRACTIONS, THIS ROUTINE ADDS THE
                                ;32974  ;EXPONENTS, CHECKS FOR RESERVED OPERANDS, CHECKS FOR A ZERO RESULT,
                                ;32975  ;ACCEPTS FROM THE FPA THE RESULTS OF THE FRACTION MULTIPLY, AND
                                ;32976  ;SETS-UP THE STATE NEEDED TO MERGE INTO THE BASIC EMODF FLOWS.
                                ;32977  ; IT THEN TRANSFERS CONTROL TO THE BASIC ROUTINE WHICH CONVERTS
                                ;32978  ;THE RESULT INTO ITS RESPECTIVE INTEGER AND FRACTION PARTS.
                                ;32979
                                ;32980  ;EMODF ENTRY CONDITIONS:
                                ;32981  ;-----------------------
                                ;32982  ;ENTER WITH Q = MULR.RF, D = MULRX.RB AT C.FORK.
                                ;32983
                                ;32984  13C8:
                                ;32985  X.EMODF:;-----------------------;
                                ;32986          ID_D.SYNC,              ;  SEND MULTIPLIER EXTENTION TO FPA
                                ;32987          ALU_Q,                  ;
                                ;32988          SC_Q(EXP),              ;  MUL'IER EXP
                                ;32989          SS_ALU15,               ;  MUL'IER SIGN
                                ;32990          CHK.FLT.OPR,            ;  CHECK FOR -0
                                ;32991          CALL,INTERRUPT.REQ?,    ;
U 13C8, 0001,2E7D,1581,BC00,0888,637E   ;32992          J/SPEC                  ;  GO GET M'CAND
                                ;32993
                                ;32994  13D8:   ;-----------------------;  RETURN FROM 'SPEC'
                                ;32995          ID_D.SYNC,              ;  SEND M'CAND TO FPA
                                ;32996          ALU_D,                  ;
                                ;32997          FE_D(EXP),CLK.UBCC,     ;  M'CAND EXP
                                ;32998          SS_SS.XOR.ALU15&SD_ALU15,;  SS GETS RESULT SIGN
                                ;32999          CHK.FLT.OPR,            ;  CHECK FOR -0
U 13D8, 0001,007C,1585,BC00,0918,753C   ;33000          J/X.EMDF1               ;
                                ;33001
                                ;33002  13DA:   ;-----------------------;  RETURN FROM 'SPEC'
                                ;33003          ID_D.SYNC,              ;  SEND M'CAND TO FPA
                                ;33004          ALU_D,                  ;
                                ;33005          FE_D(EXP),CLK.UBCC,     ;  M'CAND EXP
                                ;33006          SS_SS.XOR.ALU15&SD_ALU15,;  SS GETS RESULT SIGN
U 13DA, 0001,007C,1585,BC00,0918,753C   ;33007          CHK.FLT.OPR             ;  CHECK FOR -0
                                ;33008  =;END
                                ;33009
                                ;33010  X.EMDF1:;-----------------------;
                                ;33011          SC_SC+FE,               ;  SC GETS SUM OF EXP'S
U 153C, 0000,123C,0180,F800,0080,9569   ;33012          EALU?                   ;  M'IER OR M'CAND = 0?
                                ;33013
                                ;33014  =1001   ;00--------------------;  MULD/PROD = 0 (M'IER IS 0)
                                ;33015          SC_K[ZERO], Q_0, D_0,   ;
                                ;33016          TRAP.ACC[1],            ;  ABORT FPA
U 1569, 0F00,00BC,18F8,F800,0084,76FF   ;33017          J/X.EMDF5               ;
```

```
                                    ;33018          ;01--------------------;  MULD/PROD .NE. 0
                                    ;33019          SC_SC-K[.80],           ;  SAVE EXP WITH BIAS ADJUSTED
U 156B, 0000,003C,4180,F800,0084,B6FE ;33020        J/X.EMDF3               ;
                                    ;33021
                                    ;33022          ;10--------------------;  MULD/PROD = 0 (M'IER, M'CAND ARE 0)
                                    ;33023          SC_K[ZERO], Q_0, D_0,   ;
                                    ;33024          TRAP.ACC[1],            ;  ABORT FPA
U 156D, 0F00,00BC,18F8,F800,0084,76FF ;33025        J/X.EMDF5               ;
                                    ;33026
                                    ;33027          ;11--------------------;  MULD/PROD = 0 (M'CAND IS 0)
                                    ;33028          SC_K[ZERO], Q_0, D_0,   ;
                                    ;33029          TRAP.ACC[1],            ;  ABORT FPA
U 156F, 0F00,00BC,18F8,F800,0084,76FF ;33030        J/X.EMDF5               ;
                                    ;33031  =;END
                                    ;33032
                                    ;33033  =110    ;BRANCH ON FPA SYNC SIGNAL(BEN/ACCEL)
                                    ;33034
                                    ;33035  X.EMDF3:;00--------------------;
                                    ;33036          D_ACCEL&SYNC,           ;  GET FRACTION IF READY(H.O.)
U 16FE, 0A00,067C,0180,F800,0000,16FE ;33037        ACCEL?,   J/X.EMDF3     ;
                                    ;33038
                                    ;33039  X.EMDF5:;01--------------------;
                                    ;33040          SC_SC-K[.80],           ;  REMOVE BIAS FOR CONVERT RTN
                                    ;33041          RC[T1]_0,N&Z_ALU.V&C_0, ;  SET PSL<Z>
                                    ;33042          SD_SS,          Q_0,    ;
U 16FF, 0003,003C,41FC,F988,00D4,B02E ;33043        J/EMODF.6               ;  MERGE INTO NORMAL EMODF FLOWS
```

```
                                    ;33044  .TOC "  FPA Interface          : EMODD"
                                    ;33045
                                    ;33046  ;-----------------------------------------------------------------------
                                    ;33047
                                    ;33048  ;EMODD  (74)    MULR.RD, MULRX.RB, MULD.RD, INT.WL, FRACT.WD
                                    ;33049  ;*************************************************************************
                                    ;33050
                                    ;33051  ;THIS ROUTINE PROVIDES THE CPU INTERFACE TO THE FPA WHICH IS USED
                                    ;33052  ;TO DO THE FRACTION MULTIPLY FOR EMODD.  WHILE THE FPA IS
                                    ;33053  ;MULTIPLYING THE FRACTIONS, THIS ROUTINE ADDS THE
                                    ;33054  ;EXPONENTS, CHECKS FOR RESERVED OPERANDS, CHECKS FOR A ZERO RESULT,
                                    ;33055  ;ACCEPTS FROM THE FPA THE RESULTS OF THE FRACTION MULTIPLY, AND
                                    ;33056  ;SETS-UP THE STATE NEEDED TO MERGE INTO THE BASIC EMODD FLOWS.
                                    ;33057  ; IT THEN TRANSFERS CONTROL TO THE BASIC ROUTINE WHICH CONVERTS
                                    ;33058  ;THE RESULT INTO ITS RESPECTIVE INTEGER AND FRACTION PARTS.
                                    ;33059
                                    ;33060  ;EMODD ENTRY CONDITIONS:
                                    ;33061  ;-----------------------
                                    ;33062  ;ENTER WITH <RC 0, Q> = MULR.RD, D = MULRX.RB AT C.FORK.
                                    ;33063
                                    ;33064  1386:
                                    ;33065          ;-----------------------;
                                    ;33066          ID_D.SYNC,              ;  SEND MULTIPLIER EXTENTION TO FPA
                                    ;33067          SC_RC[TO](EXP),         ;  MUL'IER EXP
                                    ;33068          SS_ALU15,               ;  MUL'IER SIGN
U 1386, 0010,0078,1581,BD00,0883,10C8 ;33069        CHK.FLT.OPR             ;  CHECK FOR -0
                                    ;33070
                                    ;33071  =0**0*  ;-----------------------;
                                    ;33072          CALL,INTERRUPT.REQ?,    ;
U 10C8, 0000,0E3D,0180,F800,0000,137E ;33073        J/EMODD.SPEC            ;  GO GET M'CAND
                                    ;33074  =1**0*
                                    ;33075  X.EMDD0:;-----------------------;  RETURN WITH MULD IN <RC 2, D>
                                    ;33076          ID_D.SYNC,              ;  SEND M'CAND TO FPA
                                    ;33077          FE_SC,                  ;  MUL'ER EXP TO FE-REG
                                    ;33078          SC_RC[T2](EXP),CLK.UBCC,;  M'CAND EXP
                                    ;33079          SS_SS.XOR.ALU15&SD_ALU15,;  SS GETS RESULT SIGN
                                    ;33080          CHK.FLT.OPR,            ;  CHECK FOR -0
U 10D8, 0010,0078,1585,BD10,0993,153E ;33081        J/X.EMDD1               ;
                                    ;33082  =1**1*  ;-----------------------;
U 10DA, 0810,0038,01E0,F910,0000,153D ;33083        D_RC[T2],       Q_D     ;  REFETCH FIRST HALF OF M'CAND
                                    ;33084          ;-----------------------;
                                    ;33085          ID_D.SYNC,      D_Q,    ;  SEND FIRST HALF OF M'CAND
U 153D, 0C00,007C,1580,BC00,0000,10D8 ;33086        J/X.EMDD0               ;
                                    ;33087
                                    ;33088
                                    ;33089  X.EMDD1:;-----------------------;
                                    ;33090          SC_SC+FE,       SD_SS,  ;  SC GETS SUM OF EXP'S
U 153E, 0000,123C,0184,F800,0080,9589 ;33091        EALU?                   ;  M'IER OR M'CAND = 0?
```

```
                                ;33092  =1001   ;00--------------------;  PROD = 0 (M'IER IS 0)
                                ;33093          SC_K[ZERO], D_0, Q_0,   ;
                                ;33094          ALU_0(A),N&Z_ALU.V&C_0, ;  SET PSL<Z> AS ENTRY FLAG TO BASIC CODE
U 1589, 0F03,003C,19F8,F800,00D4,770F   ;33095          J/X.EMDD7               ;
                                ;33096
                                ;33097          ;01--------------------;  PROD .NE. 0
                                ;33098          SC_SC-K[.80],           ;  SAVE EXP WITH BIAS ADJUSTED
U 158B, 0000,003C,4180,F800,0084,B706   ;33099          J/X.EMDD3               ;
                                ;33100
                                ;33101          ;10--------------------;  PROD = 0 (M'IER, M'CAND ARE 0)
                                ;33102          ALU_0(A),N&Z_ALU.V&C_0, ;  SET PSL<Z> AS ENTRY FLAG TO BASIC CODE
                                ;33103          SC_K[ZERO], D_0, Q_0,   ;
U 158D, 0F03,003C,19F8,F800,00D4,770F   ;33104          J/X.EMDD7               ;
                                ;33105
                                ;33106          ;11--------------------;  PROD = 0 (M'CAND IS 0)
                                ;33107          SC_K[ZERO], D_0, Q_0,   ;
                                ;33108          ALU_0(A),N&Z_ALU.V&C_0, ;  SET PSL<Z> AS ENTRY FLAG TO BASIC CODE
U 158F, 0F03,003C,19F8,F800,00D4,770F   ;33109          J/X.EMDD7               ;
                                ;33110  =;END
                                ;33111
                                ;33112  =110    ;BRANCH ON FPA SYNC SIGNAL(BEN/ACCEL)
                                ;33113
                                ;33114  X.EMDD3:;00--------------------;
                                ;33115          ALU_0(A),N&Z_ALU.V&C_0, ;  SET PSL<Z> AS ENTRY FLAG TO BASIC CODE
                                ;33116          D_ACCEL&SYNC,           ;  GET FRACTION IF READY(H.O.)
U 1706, 0A03,067C,0180,F800,0050,1706   ;33117          ACCEL?,   J/X.EMDD3     ;
                                ;33118
                                ;33119          ;01--------------------;
                                ;33120          ALU_0(A),N&Z_ALU.V&C_0, ;  SET PSL<Z> AS ENTRY FLAG TO BASIC CODE
                                ;33121          Q_ACCEL&SYNC,           ;  GET FRACTION(L.O.)
U 1707, 0003,067C,01D8,F800,0050,170E   ;33122          ACCEL?,  J/X.EMDD5      ;
                                ;33123
                                ;33124  =110    ;BRANCH ON FPA SYNC SIGNAL(BEN/ACCEL)
                                ;33125
                                ;33126  X.EMDD5:;00--------------------;
                                ;33127          Q_ACCEL&SYNC,           ;  GET FRACTION IF READY
                                ;33128          ALU_0(A),N&Z_ALU.V&C_0, ;  SET PSL<Z> AS ENTRY FLAG TO BASIC CODE
U 170E, 0003,067C,01D8,F800,0050,170E   ;33129          ACCEL?,  J/X.EMDD5      ;
                                ;33130
                                ;33131  X.EMDD7:;01--------------------;
                                ;33132          STATE_K[.10],           ;  SET FLAG FOR NO OVERFLOW
                                ;33133          RC[T2] Q        Q_0,    ;  SAVE FRACTION (L.O.)
                                ;33134          TRAP.ACC[1],            ;  ABORT FPA
U 170F, 0001,20BC,64F8,F990,1404,7041   ;33135          J/EMODD.2               ;  MERGE INTO NORMAL EMODD FLOW
```

```
                                ;33136  ; Here is a copy of certain states from C.FORK, which are used by
                                ;33137  ; EMODD to obtain the third specifier when the FPA is attached.  This is
                                ;33138  ; necessary to fix a bug when the third specifier is a short literal
                                ;33139  ; because the FPA expects the base machine to explicitly pass the operand.
                                ;33140
                                ;33141  1300:   ;-------------------------------;
                                ;33142  EMODD.C.FORK:
                                ;33143
                                ;33144  1302:   ;-------------------------------;
                                ;33145          RC[T2]_Q,                       ;QUAD/DOUBLE SHORT LITERAL
U 1302, 0C01,203C,01E0,F990,0000,1540   ;33146          Q_D,D_Q                 ;GET LITERAL INTO D, SAVE D
                                ;33147
                                ;33148          ;-------------------------------;
                                ;33149          ID_D.SYNC,                      ;SEND LITERAL TO FPA
U 1540, 0F00,007E,1580,BC00,0000,0010   ;33150          D_Q,RETURN10            ;RETURN OLD D IN Q, REST IS ZERO
                                ;33151
                                ;33152  1306:   ;-------------------------------;
U 1306, 0000,003C,01E0,F990,0000,036E   ;33153          RC[T2]_LA,Q_D,J/36E     ;QUAD REGISTER, TO READ
                                ;33154
                                ;33155  1308:   ;-------------------------------;
U 1308, 0000,083C,01E0,F800,0200,02D2   ;33156          VA_LA,Q_D,DATA.TYPE?,J/C.M      ;(R)
                                ;33157
                                ;33158  1309:   ;-------------------------------;
                                ;33159          R(PRN)_LA+K[SP1.CON].RLOG,       ;(R)+ UPDATE THE STACK POINTER
U 1309, 0018,0018,1580,F8D8,0000,0308   ;33160          J/C.DR                  ;THEN LOAD UN-INCREMENTED ADDR
                                ;33161
                                ;33162  130A:   ;-------------------------------;
                                ;33163          R(PRN)_LA-K[SP1.CON].RLOG,       ;-(R) AUTO DECREMENT
U 130A, 0018,0804,15E0,F8D8,0200,02D2   ;33164          VA_ALU,Q_D,DATA.TYPE?,J/C.M     ; USE DECREMENTED ADDR
                                ;33165
                                ;33166  130B:   ;-------------------------------;
                                ;33167          VA_LA,                          ;@(R)+ AUTO INCREMENT DEFERED
U 130B, 0000,003C,01E0,F800,0200,0394   ;33168          Q_D,J/394               ;SAVE DATA WHILE GETTING INDIRECT
                                ;33169
                                ;33170  130C:   ;-------------------------------;
                                ;33171          RC[T7]_LA.CTX,                  ;INDEX MODE, CONTEXT SHIFT INDEX
U 130C, 0060,C03D,0180,F9B8,0000,047E   ;33172          CALL,J7ASPC             ; AND GO EVALUATE BASE OPERAND ADDRESS
                                ;33173
                                ;33174  136C:   ;-------------------------------;RETURN HERE WITH BASE OPERAND ADDRESS
                                ;33175          D_Q,                            ;RESTORE DATA TO BE STORED
                                ;33176          VA_D+LC,                        ;COMPUTE INDEXED ADDRESS
U 136C, 0C11,0814,0180,F800,0200,02D2   ;33177          DATA.TYPE?,J/C.M        ;GET OR STORE NORMAL OR QUAD
                                ;33178
                                ;33179  130D:   ;-------------------------------;
                                ;33180          VA_Q+LB.PC,                     ;D(R) DISPLACEMENT MODE.
                                ;33181          Q_D,                            ;SAVE OLD D
                                ;33182          C[R.IB.SPEC,                    ;DISCARD THE SPECIFIER
U 130D, D005,2814,01E0,F800,0200,02D2   ;33183          DATA.TYPE?,J/C.M        ;GO GET THE OPERAND
                                ;33184
                                ;33185  130F:   ;-------------------------------;
                                ;33186          Q_D,VA_Q+LB.PC,                 ;@D(R) DISPLACEMENT DEFERED
U 130F, D005,2014,01E0,F800,0200,03BE   ;33187          C[R.IB.SPEC,J/3BE       ;DROP THE SPECIFIER
```

```
                                ;33188  .TOC "  FPA Interface        : CFORK entry points for R=PC & special CFORK states"
                                ;33189
                                ;33190  ;HERE ARE VARIANTS OF THE C-FORK ENTRY POINTS FOR R=PC
                                ;33191  1316:   ;-------------------------------;
U 1316, 0000,003C,0180,F800,0000,0001   ;33192          J/RSVMOD                        ;PC QUAD REGISTER MODE
                                ;33193
                                ;33194  1317:   ;-------------------------------;
U 1317, 0000,003C,0180,F800,0000,0001   ;33195          J/RSVMOD                        ;ILLEGAL QUAD REGISTER MODE, R=PC
                                ;33196
                                ;33197  1318:   ;-------------------------------;
U 1318, 0000,003C,0180,F800,0000,0001   ;33198          J/RSVMOD                        ;(PC)
                                ;33199
                                ;33200  131A:   ;-------------------------------;
U 131A, 0000,003C,0180,F80C,0000,0001   ;33201          J/RSVMOD                        ;-(PC)
                                ;33202
                                ;33203  131B:   ;-------------------------------;
                                ;33204          VA_Q,Q_D,CLR.IB.SPEC,           ;@(PC)+ ABSOLUTE MODE
U 131B, D001,283C,01E0,F800,0200,02D2   ;33205          DATA.TYPE?,J/C.M                ;
                                ;33206
                                ;33207  131C:   ;-------------------------------;
U 131C, 0000,003C,0180,F800,0000,0001   ;33208          J/RSVMOD                        ;INDEX MODE, R=PC
                                ;33209
                                ;33210  131D:   ;-------------------------------;
U 131D, 0000,003C,0180,F800,0000,0001   ;33211          J/RSVMOD                        ;NESTED INDEX MODE, R=PC
                                ;33212
                                ;33213  131F:   ;-------------------------------;
U 131F, 0C01,203C,0180,F990,0000,1541   ;33214          RC[T2]_Q,D_Q                    ; SAVE THE LOW ADDRESSED HALF
                                ;33215
                                ;33216          ;-------------------------------;
U 1541, 0000,007C,1580,BC00,0000,1542   ;33217          ID_D.SYNC                       ; SEND THE LOW HALF TO FPA
                                ;33218
                                ;33219          ;-------------------------------;
                                ;33220          Q_IB.DATA,CLR.IB.COND,          ;GET SECOND PART INTO Q
U 1542, F000,0B3C,01F0,F806,0000,04D0   ;33221          PC_PC+4,IB.TEST?,J/C.IQ         ;PUSH PC PAST SECOND PART
                                ;33222
                                ;33223  ;SPECIAL CFORK STATES
                                ;33224  1387:   ;-------------------------------;HERE WE SHOULD NEVER GET
U 1387, 0000,003D,0180,F800,0000,0EE0   ;33225          CALL,J/EH.USEQ                  ;"UNUSED" LOCATION FOUND IN IB ROM
                                ;33226
                                ;33227  137C:   ;-------------------------------;IB HAD NO DATA BECAUSE OF
U 137C, 0000,003D,0180,F800,0000,0E64   ;33228          CALL,J/IB.TBM                   ;TB MISS.  REFILL IT
                                ;33229
                                ;33230  137D:   ;-------------------------------;IB HAD NO DATA BECAUSE OF
U 137D, 0000,003D,0180,F800,0000,0B80   ;33231          CALL,J/IB.ERR                   ;ANY ERROR.  FIND OUT WHAT HAPPENED
                                ;33232  137E:
                                ;33233  EMODD.SPEC:
                                ;33234          ;-------------------------------;IB IS WAITING FOR DATA
                                ;33235          LAB_R(SP1),Q_IB.DATA,PC_PC+N,   ;STALL
                                ;33236          CLR.IB.COND,ACT/ALLOW.IB.READ,  ;
U 137E, F000,003F,01F0,F847,0000,1300   ;33237          SUB/SPEC,J/EMODD.C.FORK         ;
                                ;33238
                                ;33239  137F:   ;-------------------------------;HERE IF INTERRUPT REQUEST UP
U 137F, C000,003C,0180,F800,0000,04F8   ;33240          J/INT.B                         ;GO BACKUP REGISTERS, SERVE INTERRUPT
                                ;33241  .NOLIST
```

33241; This page intentionally left blank.

```
                                        ;33242  .TOC    "GANDH.MIC"
                                        ;33243  .TOC    "Revision 0.8"
                                        ;33244  ;        Kevin J. Cassidy,P. R. Guilbault
                                        ;33245
;33246  .NOBIN
;33247  .REGION/<WCSR3L>,<WCSR3H>/<WCSR1L>,<WCSR1H>,<WCSR2L>,<WCSR2H>
;33248
;33249  .TOC    "       Revision History"
;33250
;33251  ;  08   Changes to POLYG/H,CVTRHL
;33252  ;  07   ACBH-add branch to detect cases where sign of index and sign of limit
;33253  ;       are different.
;33254  ;       CVTHF-remove SET.CC(INST) macro from SPECG which was setting vbit and
;33255  ;       causing integer overflow traps.
;33256  ;       CVTHD-replace PACK.D routine with new code in wcs that does not use
;33257  ;       SET.CC(INST) macro that caused integer overflow traps.
;33258  ;  06   ACBH-correct routine for reading and writing index when in register.
;33259  ;       initialize state during register read so ADD functions correctly.
;33260  ;       interchange ADDEND=0 and INDEX=0 return from unpack routine.
;33261  ;       and add cc macro to those paths.
;33262  ;       POLYH-change fault code in PACKH tc do POLY.FAULT when FPD is set.
;33263  ;  05   Change macro names that deal with conditions codes.
;33264  ;  04   ACBH-change ADD.SUBR to loop on bit 6 of state instead of bit 5,
;33265  ;       also change CMPH so it leaves state bit 5 alone.Remove useless branch
;33266  ;       on SS in CMPH and clear sign bits of exponents so exp comparison
;33267  ;       works correctly.
;33268  ;       EMODH- v bit cleared by ADDH.310,change cc macro
;33269  ;       ACBG-c bit lost in pack.g,save c-bit and restore after pack.
;33270  ;  03   CVTHF change constaint so fault branch works.
;33271  ;       EMODG/H CC problem fixed by clearing V and C bits at start.
;33272  ;       CVTGH problem:improper setup for call of pack routine,
;33273  ;       in ADDH.380 load LC_RC[T6] before call.
;33274  ;       CVTHD Packd subroutine includes SET.CC(INST) macro which,in this case,
;33275  ;       incorrectly sets C-bit,use SET.CC(INST) with proper conditions to
;33276  ;       zero c and v bits after return from pack.
;33277  ;       ACBG add condition code macros to ADDEND=0,INDEX=0 return paths from
;33278  ;       unpack routine to set CCs in those cases.
;33279  ;       DIVH C-bit set,modify PSL at end of instuction to clear C-bit.
;33280  ;       CVTRHL V-bit set,change SET.CC(INST) at start with N&Z_ALU.V&C_0.
;33281  ;       CVTHW/HL/HB install test for intoverflow on return to CVTHB.
;33282  ;  02   CVTGF.2-ADD SET.CC TO UNDERFLOW PATH
;33283  ;       CVTHD  ADJUST UNDERFLOW,FU=0 PATH
;33284  ;  01   Creat error detection and fault code for CVTHD and CVTHG
;33285  ;       Convert CVTGF code from trap to faults.
;33286  ;       Start of history
;33287
                                        ;33288  .BIN
```

```
                                ;33289  .TOC    ""      G & H floating point  : G-FORK Specifier Evaluation Subroutine""
                                ;33290
                                ;33291  ;FPLA trap When there is no G & H
                                ;33292  ;1180:  ;-------------------------------;
                                ;33293  ;       RC[T0]_K[.10],J/EXCPT           ; No G&H take fault
                                ;33294  ;
                                ;33295  ;FPLA trap when G & H is present
                                ;33296  ;11A3:  ;-------------------------------;
                                ;33297  ;       ALU_K[.8000],RC[T4]_ALU,        ; FOR MOVO, SHORT LITERAL DETECT
                                ;33298  ;       CLR_IB.OPC,                     ; CLEAR ESCAPE-CODE,
                                ;33299  ;       PC_PC+1,                        ; INCREMENT PC PAST OP-CODE
                                ;33300  ;       J/XFD                           ; AND STROBE INTERRUPTS
                                ;33301
                                ;33302  =0*01*
                                ;33303  XFD:    ;0*01*-------------------------; CALL SITE FOR SPECG
                                ;33304          STATE_K[ZERO],                  ; IN PARTICULAR FOR MOVH
                                ;33305          BEN/IB.0,                       ; WAIT FOR OPCODE TO SETTLE
U 182A, 0000,053D,1980,F800,1404,7902   ;33306          CALL,J/XFD.1            ; BRANCH ON IB0
                                ;33307
                                ;33308  =1*01*  ;1*01*-------------------------; RETURN FROM SPECG WITH :
                                ;33309                                          ;       G   = FORMAT 1. OPERAND
                                ;33310                                          ;       D   = OP1 LONG FRAC
                                ;33311                                          ;       RC2 = OP1 LONG EXP
                                ;33312                                          ; HERE FOR: ADDG*,SUBG*,MULG*,DIVG*,
                                ;33313                                          ; ACBG,MOVG,CMPG,MNEGG,TSTG,CVTGB,
                                ;33314                                          ; CVTGW,CVTGL,CVTRGL
U 183A, 0010,00B8,00C0,F910,0000,180A   ;33315          TRAP.ACC[1],Q_RC[T2],J/XFD.G    ; RESET ACCEL, LATCH OP1 LONG EXP IN Q
                                ;33316
                                ;33317  =1*11*  ;1*11*-------------------------; RETURN FROM SPECG WITH :
                                ;33318                                          ;       H FORMAT 1. OPERAND
                                ;33319                                          ;       Q = LONG EXP = RC[T2]
                                ;33320                                          ;       ID[T2]= OP1 LONG FRAC1
                                ;33321                                          ;       RC3 = OP1 LONG FRAC2
                                ;33322                                          ;       ID[T3]= OP1 LONG FRAC3
                                ;33323                                          ; HERE FOR THE CORRESPONDING H-SOURCES
                                ;33324          TRAP.ACC[1],RC[T0]_Q,           ; RESET ACCEL, STORE LONG EXP IN RC0
U 185E, 0001,20BC,C8F0,2D80,0000,180B   ;33325          Q_ID[T2]                ; Q GETS LONG FRAC1
                                ;33326
                                ;33327          ;-------------------------------;
                                ;33328          D_Q,Q_ID[T3],                   ; MOVE DATA AROUND
U 180B, 0C00,003C,CDF0,2D18,0000,1812   ;33329          LC_RC[T3]               ; LATCH LONG FRAC2
                                ;33330
                                ;33331          ;-------------------------------;
                                ;33332          ID[T0]_D,                       ; STORE LONG FRAC1 IN T0
                                ;33333          RC[T1]_LC,                      ; RC1 GETS LONG FRAC2
U 1812, 0C10,0038,C180,3D88,0000,1866   ;33334          D_Q                     ; D GETS LONG FRAC3
```

```
                                    ;33335  =0*11*  ;0*11*-------------------------;
                                    ;33336          ID[T1]_D,                     ; FINALLY STOR LONG FRAC3
U 1866, 0810,0039,C580,3D00,0000,147E ;33337        D_RC[T0],CALL,J/SPECG         ; D GETS LONG EXP
                                    ;33338
                                    ;33339          ;1*11*-------------------------;
                                    ;33340                                        ; RETURN WITH:
                                    ;33341                                        ;       RC0 = OP1 LONG EXP
                                    ;33342                                        ;       T0 = OP1 LONG FRAC1
                                    ;33343                                        ;       RC1 = OP1 LONG FRAC2
                                    ;33344                                        ;       T1 = OP1 LONG FRAC3
                                    ;33345                                        ;       RC2 = OP2 LONG EXP
                                    ;33346                                        ;       T2 = OP2 LONG FRAC1
                                    ;33347                                        ;       RC3 = OP2 LONG FRAC2
                                    ;33348                                        ;       T3 = OP2 LONG FRAC3
                                    ;33349                                        ;       Q = OP2 LONG EXP
U 1876, 0000,00BF,0080,F800,0000,1400 ;33350        TRAP.ACC[1],G.FORK            ; RESET THE ACCELLARATOR
                                    ;33351
                                    ;33352  =0**1*
                                    ;33353  XFD.G:  ;0**1*-------------------------;
                                    ;33354          TRAP.ACC[1],                  ; RESET THE ACCELLARATOR
                                    ;33355          RC[T0]_LC,                    ; STORE OP1 LONG EXP IN RC0
U 180A, 0010,00B9,0080,F980,0000,147E ;33356        CALL, J/SPECG                 ; EVALUATE THE SECOND OPERAND
                                    ;33357                                        ; OR EXECUTION POINT
                                    ;33358
                                    ;33359          ;1**1*-------------------------;
                                    ;33360                                        ; RETURN WITH:
                                    ;33361                                        ;       Q = OP1 LONG FRAC
                                    ;33362                                        ;       RC0 = OP1 LONG EXP
                                    ;33363                                        ;       D = OP2 LONG FRAC
                                    ;33364                                        ;       RC1 = OP2 LONG EXP
U 181A, 0000,00BF,0080,F800,0000,1600 ;33365        TRAP.ACC[1],E.FORK            ; RESET THE ACCELLARATOR
```

```
                                ;33366  =000
                                ;33367  =010
                                ;33368  XFD.1:  ;010----------------------------; BRANCH ON IB BYTE 0
                                ;33369          TRAP.ACC[1],                    ; RESET THE ACCELLARATOR
                                ;33370          RC[T5]_K[.3FF],                 ; NOTHING BETTER TO DO
U 1902, 0018,04B8,8080,F9A8,0000,1800   ;33371          BEN/IRC.ROM,J/XFD.2             ; BRANCH ON OUTPUT OF IB-ROM
                                ;33372  =100
                                ;33373  XFD.10: ;100----------------------------;
                                ;33374          TRAP.ACC[1],                    ; RESET THE ACCELLARATOR
U 1904, 0000,00BD,0080,F800,0000,0E64   ;33375          CALL,J/IB.TBM                   ; TB MISS
                                ;33376
                                ;33377          ;101----------------------------;
                                ;33378          TRAP.ACC[1],                    ; RESET THE ACCELLARATOR
U 1905, 0000,00BD,0080,F800,0000,0B80   ;33379          CALL,J/IB.ERR
                                ;33380
                                ;33381          ;110----------------------------;
                                ;33382          TRAP.ACC[1],                    ; RESET THE ACCELLARATOR
                                ;33383          MCT/ALLOW.IB.READ,              ; KEEP READING
U 1906, 0000,0BBC,0080,F800,0000,1904   ;33384          IB.TEST?,J/XFD.10               ; TEST IB
                                ;33385
                                ;33386          ;111----------------------------;
                                ;33387          TRAP.ACC[1],                    ; RESET THE ACCELLARATOR
                                ;33388          INTRPT.STROBE,                  ; STROBE INTERRUPTS
                                ;33389          RC[T5]_K[.3FF],                 ; NOTHING BETTER TO DO
U 1907, 0018,04B8,8080,F9A8,4000,1800   ;33390          BEN/IRC.ROM,J/XFD.2             ; BRANCH ON OUTPUT OF IB-ROM
```

```
                                        ;33391  =00*000
                                        ;33392  XFD.2:  ;00*000-------------------------; FLOW FOR:
                                        ;33393          TRAP.ACC[1],                     ; RESET THE ACCELLARATOR
U 1800, 0018,00B8,6480,F980,0000,08FC   ;33394          RC[T0]_K[.10],J/EXCPT            ; UNIMPLEMENTED OP-CODES
                                        ;33395
                                        ;33396          ;00*001-------------------------;
                                        ;33397          TRAP.ACC[1],                     ; RESET THE ACCELLARATOR
U 1801, F000,00BF,00F0,F847,0000,1400   ;33398          SPECG                            ; ADDG*,SUBG*,MULG*,DIVG*,
                                        ;33399                                           ; ACBG,MOVG,CMPG,MNEGG,TSTG,
                                        ;33400                                           ; CVTGB,CVTGW,CVTGL,CVTRGL,
                                        ;33401
                                        ;33402          ;00*010-------------------------;
U 1802, F000,003F,01F0,F847,0000,0300   ;33403          SPEC                             ; CVTBG,CVTWG,CVTLG,CVT*H
                                        ;33404
                                        ;33405          ;00*011-------------------------;
U 1803, 0000,003D,0180,F800,0000,147E   ;33406          CALL,J/SPECG                     ; EMOD AND POLY, EVALUATE 1st SPECIFIER
                                        ;33407
                                        ;33408          ;00*100-------------------------;
                                        ;33409          TRAP.ACC[1],                     ; RESET THE ACCELLARATOR
U 1804, 0000,00BD,0080,F800,0000,147E   ;33410          CALL,J/SPECG                     ; MOVO ONLY
                                        ;33411
                                        ;33412          ;00*101-------------------------;
                                        ;33413          TRAP.ACC[1],                     ; RESET THE ACCELLARATOR
U 1805, 0018,00B8,6480,F980,0000,08FC   ;33414          RC[T0]_K[.10],J/EXCPT            ; NO INSTRUCTION SHOULD GET HERE
                                        ;33415
                                        ;33416          ;00*110-------------------------;
                                        ;33417          TRAP.ACC[1],                     ; RESET THE ACCELLARATOR
                                        ;33418          ALU_0(A),Q_ALU,                  ; CVTFG,CVTDH,CVTFH
                                        ;33419          CLK.UBCC,                        ; Q GETS COUNT FOR ADVANC EXEC.COUNT
                                        ;33420          IBC/BDEST,                       ; ADVANCE IT ONCE
U 1806, 7003,00BD,00C0,F800,0010,1810   ;33421          CALL,J/INCR.EXEC.PTR             ; ADVANCE IT 3 MORE TIMES
                                        ;33422
                                        ;33423          ;00*111-------------------------;
                                        ;33424          TRAP.ACC[1],                     ; RESET THE ACCELLARATOR
                                        ;33425          ALU_0(A),Q_ALU,                  ; CVTGF,CVTHD,CVTHF
                                        ;33426          CLK.UBCC,                        ; Q GETS COUNT FOR ADVANC EXEC.COUNT
                                        ;33427          IBC/BDEST,                       ; ADVANCE IT ONCE
U 1807, 7003,00BD,00C0,F800,0010,1810   ;33428          CALL,J/INCR.EXEC.PTR             ; ADVANCE IT 3 MORE TIMES
```

```
                                ;33429  =01*011 ;01*011-------------------------; RETURN FROM SPECG EMODG/POLYG WITH :
                                ;33430                                          ;       D = OP1 LONG FRAC
                                ;33431                                          ;       RC2 = OP1 LONG EXP
                                ;33432          TRAP.ACC[1],                    ; RESET THE ACCELLARATOR
                                ;33433          ID[T9]_D,                       ; SAVE ARG LONG FRAC IN POLYG
                                ;33434          LC_RC[T2],                      ; LATCH OP1 LONG EXP
U 1813, 0000,1BBC,E480,3D10,0000,18AD   ;33435          IRD?,J/EMOD.POLY                ; IS THIS EMOD OR POLY ?
                                ;33436
                                ;33437  =01*100 ;01*100-------------------------; RETURN FROM SPECG WITH :
                                ;33438                                          ;       H FORMAT 1. OPERAND
                                ;33439                                          ;       Q = LONG EXP = RC[T2]
                                ;33440                                          ;       ID[T2]= OP1 LONG FRAC1
                                ;33441                                          ;       RC3 = OP1 LONG FRAC2
                                ;33442                                          ;       ID[T3]= OP1 LONG FRAC3
                                ;33443                                          ; HERE FOR THE CORRESPONDING H-SOURCES
                                ;33444          TRAP.ACC[1],RC[TO]_Q,           ; RESET ACCEL, STORE LONG EXP IN RC0
U 1814, 0001,E0BC,C8F0,2D80,0070,155B   ;33445          Q_ID[T2],SET.CC(INST),J/NEWMOVO ; Q GETS LONG FRAC1
                                ;33446
                                ;33447  =01*110 ;01*110-------------------------; THIS GETS RET16 FROM READ REG ON MOVO
                                ;33448          TRAP.ACC[1],RC[TO]_Q,           ; RESET ACCEL, STORE LONG EXP IN RC0
U 1816, 0001,E0BC,C8F0,2D80,0070,155B   ;33449          Q_ID[T2],SET.CC(INST),J/NEWMOVO ; Q GETS LONG FRAC1
                                ;33450
                                ;33451  =01*111 ;01*111-------------------------; RETURN FROM SPECG EMODH/POLYH WITH :
                                ;33452                                          ; RETURN WITH:
                                ;33453                                          ;       Q = LONG EXP = RC2
                                ;33454                                          ;       T2 = OP1 LONG FRAC1
                                ;33455                                          ;       RC3 = OP1 LONG FRAC2
                                ;33456                                          ;       T3 = OP1 LONG FRAC3
                                ;33457          TRAP.ACC[1],                    ; RESET THE ACCELLARATOR
                                ;33458          RC[TO]_Q,                       ; SAVE OP1 LONG EXP IN RC0
                                ;33459          Q_ID[T2],                       ; GET OP1 LONG FRAC1
U 1817, 0001,20BC,C8F0,2D80,0000,1680   ;33460          J/EMODH.POLYH                   ; SPECIAL FLOW
                                ;33461
                                ;33462  =10*110 ;10*110-------------------------; RETURN WITH EXE.POINTER ADVANCED BY 3
                                ;33463          IBC/BDEST,                      ; ONCE MORE
U 1826, 7000,003D,0180,F800,0000,037E   ;33464          CALL,J/SPEC                     ; CVTFG,CVTDH,CVTFH
                                ;33465
                                ;33466          ;10*111-------------------------;
                                ;33467          IBC/BDEST,                      ; ONCE MORE
                                ;33468          STATE_K[ZERO],                  ; CLEAR STATE REGISTER
U 1827, 7000,003D,1980,F800,1404,747E   ;33469          CALL,J/SPECG                    ; CVTGF,CVTHD,CVTHF
                                ;33470
                                ;33471  =11*110 ;11*110-------------------------;
                                ;33472          TRAP.ACC[1],                    ; RESET THE ACCELLARATOR
U 1836, 0000,00BF,0080,F800,0000,1400   ;33473          G.FORK                          ; CVTFG,CVTDH,CVTFH
                                ;33474
                                ;33475          ;11*111-------------------------;
                                ;33476          TRAP.ACC[1],                    ; RESET THE ACCELLARATOR
                                ;33477          ALU_0+K[.30]+1,SC_ALU,          ; SC GETS .31
U 1837, 001B,0093,7880,F800,0082,1400   ;33478          G.FORK                          ; CVTGF,CVTHD,CVTHF
```

```
                                    ;33479  =01101
                                    ;33480  EMOD.POLY:
                                    ;33481          ;01101----------------------------; BRANCH ON IR<0>
                                    ;33482          RC[T0]_LC,                         ; RC0 GETS MULTIPLIER LONG EXP
U 18AD, 0010,0038,0180,F980,0000,1883 ;33483        J/EMODG                            ; EMOD
                                    ;33484
                                    ;33485          ;01111----------------------------; POLY
                                    ;33486          RC[T0]_LC,                         ; STORE OP1 LONG EXP IN RC0
U 18AF, 0010,0039,0180,F980,0000,037E ;33487        CALL, J/SPEC                       ; EVALUATE THE DEGREE
                                    ;33488
                                    ;33489  =11111  ;11111----------------------------;
                                    ;33490                                             ; RETURN WITH:
                                    ;33491                                             ;       Q = OP1 LONG FRAC
                                    ;33492                                             ;       RC0 = OP1 LONG EXP
                                    ;33493                                             ;       D = DEGREE
U 18BF, 0000,003F,0180,F800,0000,1400 ;33494        G.FORK                             ; EXECUTION TIME
                                    ;33495
                                    ;33496
                                    ;33497
                                    ;33498  ;SUBROUTINE FOR ADVANCING EXECUTION POINT COUNTER
                                    ;33499  =0
                                    ;33500  INCR.EXEC.PTR:
                                    ;33501          ;0----------------------------; BRANCH ON ALU Z-BIT
                                    ;33502          ALU_Q-K[.1],CLK.UBCC,              ; DECREMENT COUNT
                                    ;33503          IBC/BDEST,                         ; ADVANCE POINTER
U 1810, 7019,2100,0580,F800,0010,1810 ;33504        Z?,J/INCR.EXEC.PTR                 ; FINISHED ?
                                    ;33505
                                    ;33506          ;1----------------------------;
U 1811, 7000,003E,0180,F800,0000,0022 ;33507        IBC/BDEST,RETURN[22]               ; ADVANCE POINTER A LAST TIME
```

```
                                ;33508  ;Control passes to this point by "CALL.J/SPEC" or from any 'WRITE.G.DEST" state.
                                ;33509  ;LA, LB = Register selected by bits <3:0> of IB byte 1
                                ;33510  ;D = Result to be stored, if WRITE.G.DEST; otherwise returned in Q
                                ;33511  ;Q = Instruction stream data, if any
                                ;33512  ;If the specifier evaluated is of "READ" or "MODIFY" type, control returns
                                ;33513  ; at the call site ored with 10, for literal and memory operands, or the
                                ;33514  ; call site ored with 12 for register operands.  If "WRITE" type, control
                                ;33515  ; passes to IRD to perform the next instruction.
                                ;33516
                                ;33517  1400:
                                ;33518  WRG:
                                ;33519  G.FORK: ;-------------------------------; G FORMAT SHORT LITERAL
                                ;33520          ALU_Q.ANDNOT.K[.E003],          ; ISOLATE EXP AND FRACTION BITS
U 1400, 0899,2024,99E0,F800,0000,182D   ;33521          D_ALU.RIGHT2, Q_D               ; STORE IN D, Q GETS OLD D
                                ;33522
                                ;33523          ;-------------------------------;
                                ;33524          ALU_D.OR.K[.8000],              ; ADD IN NEW BIAS
U 182D, 0F59,0032,4580,F990,0000,0010   ;33525          RC[T2]_ALU.RIGHT,D_0,RETURN10   ; RC2 GETS HIGH DATA, LONG FRAC<LS> IS 0
                                ;33526
                                ;33527  1401:   ;-------------------------------;
U 1401, 0000,003C,0180,F800,0000,0001   ;33528          J/RSVMOD                        ;RESERVED MODE
                                ;33529
                                ;33530  1402:   ;-------------------------------; H FORMAT SHORT LITERAL
                                ;33531          ALU_Q,                          ; GET I-STREAM DATA
                                ;33532          RC[T4]_ALU,                     ; SAVE IT IN RC4 FOR MOVO
                                ;33533          SC_ALU(EXP),                    ; SC GETS EXPONENT
                                ;33534          FE_K[.80],                      ; NEED TO GET RID OF OLD BIAS
                                ;33535          STATE_K[.80],                   ; SET SHORT LITERAL FLAG
                                ;33536          Q_ALU(FRAC),                    ; Q GETS FRACTION PART
U 1402, 0F01,203C,41C8,F9A0,1587,7834   ;33537          D_0                             ; GET READY TO CLEAR ID[T3]
                                ;33538
                                ;33539          ;-------------------------------;
                                ;33540          ID[T3]_D,                       ; CLEAR LONG FRAC3
                                ;33541          Q_Q.LEFT2,                      ; Q GETS REAL FRACTION PART
                                ;33542          RC[T3]_0,                       ; CLEAR LONG FRAC2
U 1834, 0003,0C3C,CD88,3D98,0080,8841   ;33543          SC_SC-FE,SC.NE.0?               ; CLEAR BIAS
                                ;33544
                                ;33545  =0**    ;0**----------------------------; MOVO
                                ;33546          ID[T3]_D,Q_RC[T4],              ;
U 1841, 0010,0038,CDC0,3D20,0000,1844   ;33547          J/MOVO                          ;
                                ;33548
                                ;33549          ;1**----------------------------;
U 1845, 0019,2030,1DC0,F918,0000,1838   ;33550          LC_RC[T3],Q_Q.OR.K[SC]          ; LATCH LONG FRAC2, GENERATE LONG EXP
                                ;33551
                                ;33552          ;-------------------------------;
U 1838, 0019,2030,B1C0,F800,0000,1844   ;33553          Q_Q.OR.K[.4000]                 ; Q GETS BIAS
                                ;33554  MOVO:   ;-------------------------------;
U 1844, 0000,003C,C980,3C00,0000,184D   ;33555          ID[T2]_D                        ;
                                ;33556          ;-------------------------------;
                                ;33557          K[.30],                         ; HAVE TO CLEAR ID[T2] AS WELL
U 184D, 0001,203E,7980,F990,0000,0014   ;33558          RC[T2]_Q,RETURN[14]             ; RETURN 14 FOR H FORMAT WITH :
                                ;33559                                          ;       Q = RC[T2] = LONG EXP
                                ;33560                                          ;       D = ID[T2] = LONG FRAC1
                                ;33561                                          ;       RC[T3] = LONG FRAC2
                                ;33562                                          ;       ID[T3] = LONG FRAC3
```

```
                                ;33563  1403:   ;-------------------------------;
U 1403, 0000,003C,0180,F800,0000,0001   ;33564          J/RSVMOD                        ; RESERVED MODE
                                ;33565
                                ;33566  1404:   ;-------------------------------; G FORMAT REGISTER
U 1404, 0000,003C,01E0,F990,0000,1850   ;33567          Q_D,RC[T2]_LA                   ; SAVE D IN Q, RC2 GETS FIRST PART
                                ;33568
                                ;33569          ;-------------------------------;
U 1850, 0800,003E,0180,F860,0000,0012   ;33570          D_R(PRN+1),RETURN12             ; D GETS LONG FRAC
                                ;33571  1424:
                                ;33572  WRDG.R: ;-------------------------------; WRITE G FORMAT REGISTER
U 1424, 0001,00BC,0080,F8D8,0000,1855   ;33573          TRAP.ACC[1],R(PRN)_D            ; RESET ACCEL, WRITE LOW REGISTER FIRST
                                ;33574
                                ;33575          ;-------------------------------;
U 1855, 0810,0038,0180,F908,0000,185A   ;33576          D_RC[T1]                        ; GET DATA FOR HIGH ADDRESS
                                ;33577
                                ;33578          ;-------------------------------;
                                ;33579          R(PRN+1)_D,                     ; WRITE HIGH REGISTER
U 185A, C001,003C,0180,F8E4,4000,0062   ;33580          CLR.IB.OPC,PC_PC+1,J/IRD        ; CLEAR IB
                                ;33581
                                ;33582  1405:   ;-------------------------------;
U 1405, 0000,003C,0180,F800,0000,0001   ;33583          J/RSVMOD                        ;
                                ;33584
                                ;33585  1406:   ;-------------------------------; *****HUGE REGISTER*****
                                ;33586          RC[T2]_LA,                      ; HUGE REGISTER, TO READ
U 1406, 0F00,003C,4180,F990,0000,1862   ;33587          K[.80]_D_0                      ; SET UP SLOW CONSTANT
                                ;33588
                                ;33589  HUGE.REG.READ.1:
                                ;33590          ;-------------------------------;
                                ;33591          LAB_R(PRN+1),                   ; LATCH LONG FRAC1
                                ;33592          ALU_D+K[.80].RLOG,              ; PUT PRN+1 IN RLOG
U 1862, 0839,0018,4180,F860,0000,1882   ;33593          D_ALU.LEFT                      ; D GETS 100
                                ;33594
                                ;33595          ;-------------------------------;
U 1882, 0041,0014,01C0,F800,1C00,1884   ;33596          ALU_D+RLOG,Q_ALU.RIGHT          ; Q GETS PRN+2 (IN WRONG PLACE)
                                ;33597
                                ;33598          ;-------------------------------;
                                ;33599          D_LB,                           ; D GETS LONG FRAC1
U 1884, 080D,2038,0180,F800,0088,7891   ;33600          SC_Q(EXP)                       ; SC GETS PRN+2
                                ;33601
                                ;33602          ;-------------------------------;
                                ;33603          ID[T2]_D,                       ; STORE LONG FRAC1 IN T2
                                ;33604          Q_R(SC),                        ; Q GETS LONG FRAC2
U 1891, 0000,003C,C9C0,3C68,0080,D8B4   ;33605          SC_SC+1                         ; POINT TO NEXT REGISTER
                                ;33606
                                ;33607          ;-------------------------------;
U 18B4, 0800,003C,0180,F868,0000,18C2   ;33608          D_R(SC)                         ; D GETS LONG FRAC3
                                ;33609
                                ;33610          ;-------------------------------;
                                ;33611          RC[T3]_Q,                       ; RC3 GETS LONG FRAC3
U 18C2, 0001,203C,CD80,3D98,0000,18D1   ;33612          ID[T3]_D                        ; T3 GETS LONG FRAC3
                                ;33613
                                ;33614          ;-------------------------------;
                                ;33615          Q_RC[T2],                       ; Q GETS LONG EXP
U 18D1, 0010,003A,01C0,F910,0000,0016   ;33616          RETURN[16]                      ; RETURN16 FOR H FORMAT REGISTER
```

```
                                ;33617  1426:   ;--------------------------------; *****HUGE REGISTER *****
                                ;33618                                           ;ENTER WITH:
                                ;33619                                           ;       RC[T0] = DST LONG EXP
                                ;33620                                           ;       T0 = DST LONG FRAC1
                                ;33621                                           ;       RC1 = DST LONG FRAC2
                                ;33622                                           ;       T1 = DST LONG FRAC3
                                ;33623          TRAP.ACC[1],                     ; RESET THE FPA
U 1426, 0810,00B8,0080,F900,0000,18D4   ;33624          D_RC[T0]                         ; GET DST LONG FRAC
                                ;33625
                                ;33626  HUGE.REG.WRITE.1:
                                ;33627          ;--------------------------------;
                                ;33628          R(PRN)_0+D.RLOG,D_0,             ; QUAD REGISTER. STUFF LOW ADDRESS PART
U 18D4, 0F1F,2018,C1F0,2CD8,0000,18DA   ;33629          Q_ID[T0]                         ; Q GETS LONG FRA1
                                ;33630
                                ;33631          ;--------------------------------;
                                ;33632          ALU_D+RLOG,Q_ALU.RIGHT,          ; Q GETS REGISTER NUMBER
                                ;33633          SC_R[.2],D_Q,                    ;
U 18DA, 0C41,0014,09C0,F908,1C84,78F5   ;33634          LC_RC[T1]                        ; LATCH UP LONG FRAC2
                                ;33635
                                ;33636          ;--------------------------------;
                                ;33637          SC_SC+EXP(Q)(A),                 ; SC GETS PRN+2
                                ;33638          ALU_D(B),R(PRN+1)_ALU,           ; STORE LONG FRAC1
U 18F5, 001D,2038,C5F0,2CE0,0088,98FD   ;33639          Q_ID[T1]                         ; Q GETS LONG FRAC3
                                ;33640
                                ;33641  WRITE.HUGE.REGISTER:
                                ;33642          ;--------------------------------;
                                ;33643          ALU_LC,R(SC)_ALU,                ;  STORE LONG FRAC2
U 18FD, 0010,0038,0180,F8E8,0080,D900   ;33644          SC_SC+1                          ;
                                ;33645
                                ;33646          ;--------------------------------;
                                ;33647          ALU_Q,R(SC)_ALU,                 ;  STORE LONG FRAC3 IN R(SC)
U 1900, C001,203C,0180,F8EC,4000,0062   ;33648          CLR.IB.OPC,PC_PC+1,J/IRD         ;
                                ;33649
                                ;33650  WRITE.DEST.0:
                                ;33651          ;--------------------------------;
                                ;33652          SET.CC(INST),                    ;
U 190E, 0001,C03C,C180,3D80,0070,191C   ;33653          RC[T0]_D,ID[T0]_D                ;  CLEAR OPERAND
                                ;33654
                                ;33655          ;--------------------------------;
                                ;33656          RC[T1]_D,ID[T1]_D,N_AMX.Z_TST,   ;
U 191C, 0001,003C,C580,3D88,0030,1B80   ;33657          J/WRITE.G                        ;
```

```
                                ;33658  1407:   ;-------------------------------;
U 1407, 0000,003C,0180,F800,0000,0001   ;33659          J/RSVMOD
                                ;33660  1408:
                                ;33661  G.DR:   ;-------------------------------;
                                ;33662          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;33663          VA_LA,                          ;(R)  G OR H FORMAT, READ OR WRITE
                                ;33664          Q_ALU,                          ;  SAVE ADDRESS IN Q FOR 2-OPERAND
U 1408, 0000,08BC,00C0,F800,0200,1910   ;33665          DATA.TYPE?,J/G.M                ;  TEST FOR READ/WRITE, G/H-FORMAT
                                ;33666
                                ;33667  1409:   ;-------------------------------;  G OR H FORMAT, (R)+
                                ;33668          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;33669          R(PRN)_LA+K[SP1.CON].RLOG,      ;(R)+ UPDATE THE STACK POINTER
U 1409, 0018,0098,14C0,F8D8,0000,192C   ;33670          Q_ALU
                                ;33671
                                ;33672          ;-------------------------------;
                                ;33673          R(PRN)_Q+K[SP1.CON].RLOG,       ;  UPDATE POINTER TWICE
U 192C, 0019,2018,1580,F8D8,0000,1408   ;33674          J/G.DR                          ;THEN LOAD UN-INCREMENTED ADDR
                                ;33675
                                ;33676  140A:   ;-------------------------------;  G OR H FORMAT, -(R)
                                ;33677          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;33678          R(PRN)_LA-K[SP1.CON].RLOG,      ;-(R) AUTO DECREMENT
U 140A, 0018,0084,14C0,F8D8,0000,193B   ;33679          Q_ALU
                                ;33680
                                ;33681          ;-------------------------------;
                                ;33682          R(PRN)_Q-K[SP1.CON].RLOG,       ;  AUTO DECREMENT TWICE
                                ;33683          VA_ALU,                         ;  USE DECREMENTED ADDR
                                ;33684          Q_ALU,                          ;  SAVE ADDRESS IN Q FOR 2-OPERAND
U 193B, 0019,2804,15C0,F8D8,0200,1910   ;33685          DATA.TYPE?,J/G.M                ;  TEST FOR READ/WRITE, G/H-FORMAT
                                ;33686
                                ;33687  140B:   ;-------------------------------;  G OR H FORMAT
                                ;33688          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;33689          VA_LA,                          ;@(R)+ AUTO INCREMENT DEFERED
U 140B, 0000,00BC,00E0,F800,0200,1944   ;33690          Q_D                             ;SAVE DATA WHILE GETTING INDIRECT
                                ;33691
                                ;33692          ;-------------------------------;
                                ;33693          D[LONG]_CACHE,                  ;GET INDIRECT WORD
                                ;33694          R(PRN)_[A+K[.4].RLOG,           ; WHILE UPDATING REGISTER
U 1944, 0018,0018,1180,40D8,0000,195D   ;33695          J/G.DF                          ;THEN JOIN COMMON CODE
```

```
                                ;33696  140C:   ;--------------------------------; INDEX MODE, G OR H FORMAT
                                ;33697          RC[T7]_LA+LB.CTX,                ; INDEX MODE, CONTEXT SHIFT INDEX
                                ;33698          INTERRUPT.REQ?,                  ; TEST FOR INTERRUPTS
U 140C, 006C,CE15,0180,F9B8,0000,157E   ;33699          CALL,J/ASPCG                     ; AND GO EVALUATE BASE OPERAND ADDRESS
                                ;33700                                           ; RETURN WITH:
                                ;33701                                           ;       D = BASE ADDRESS
                                ;33702                                           ;       LC = INDEX
                                ;33703                                           ;       Q = OLD D-VALUE
                                ;33704
                                ;33705  146C:   ;--------------------------------; RETURN HERE WITH BASE OPERAND ADDRESS
                                ;33706          TRAP.ACC[1],                     ; RESET THE FPA
                                ;33707          D_Q,                             ; RESTORE DATA TO BE STORED
                                ;33708          VA_D+LC,                         ; COMPUTE INDEXED ADDRESS
                                ;33709          Q_ALU,                           ; SAVE ADDRESS IN Q FOR 2-OPERAND
U 146C, 0C11,0894,00C0,F800,0200,1910   ;33710          DATA.TYPE?,J/G.M                 ; GET OR STORE G OR H-FORMAT
                                ;33711
                                ;33712  140D:   ;--------------------------------; G OR H FORMAT
                                ;33713          TRAP.ACC[1],                     ; RESET THE FPA
                                ;33714          VA_Q+LB.PC,                      ; D(R) DISPLACEMENT MODE,
                                ;33715          Q_ALU,                           ; SAVE ADDRESS IN Q FOR 2-OPERAND
                                ;33716          C[R.IB.SPEC,                     ; DISCARD THE SPECIFIER
U 140D, D005,2894,00C0,F800,0200,1910   ;33717          DATA.TYPE?,J/G.M                 ; GO GET THE OPERAND
                                ;33718
                                ;33719  140F:   ;--------------------------------; G OR H FORMAT
                                ;33720          TRAP.ACC[1],                     ; RESET THE FPA
                                ;33721          Q_D,VA_Q+LB.PC,                  ; @D(R) DISPLACEMENT DEFERED
U 140F, D005,2094,00E0,F800,0200,194E   ;33722          C[R.IB.SPEC                      ; DROP THE SPECIFIER
                                ;33723
                                ;33724          ;--------------------------------;
U 194E, 0000,003C,0180,4000,0000,195D   ;33725          D[LONG]_CACHE                    ;GET INDIRECT, GO USE IT AS ADDR
                                ;33726
                                ;33727  G.DF:   ;--------------------------------;
                                ;33728          VA_D,                            ;USE POINTER AS ADDRESS
                                ;33729          TRAP.ACC[1],                     ; RESET THE FPA
                                ;33730          D_Q,                             ;GET DATA TO STORE BACK TO D
                                ;33731          Q_ALU,                           ; SAVE ADDRESS IN Q FOR 2-OPERAND
U 195D, 0C01,08BC,00C0,F800,0200,1910   ;33732          DATA.TYPE?,J/G.M                 ; READ/WRITE, G OR H ?
```

```
                                ;33733  ;HERE ARE VARIANTS OF THE G-FORK ENTRY POINTS FOR R=PC
                                ;33734
                                ;33735  1414:   ;-------------------------------;
U 1414, 0000,003C,0180,F800,0000,0001   ;33736          J/RSVMOD                        ;PC REGISTER MODE
                                ;33737
                                ;33738  1415:   ;-------------------------------;
U 1415, 0000,003C,0180,F800,0000,0001   ;33739          J/RSVMOD                        ;ILLEGAL REGISTER MODE, R=PC
                                ;33740
                                ;33741  1416:   ;-------------------------------;
U 1416, 0000,003C,0180,F800,0000,0001   ;33742          J/RSVMOD                        ;PC QUAD REGISTER MODE
                                ;33743
                                ;33744  1417:   ;-------------------------------;
U 1417, 0000,003C,0180,F800,0000,0001   ;33745          J/RSVMOD                        ;ILLEGAL QUAD REGISTER MODE, R=PC
                                ;33746
                                ;33747  1418:   ;-------------------------------;
U 1418, 0000,003C,0180,F800,0000,0001   ;33748          J/RSVMOD                        ;(PC)
                                ;33749
                                ;33750  1419:   ;-------------------------------;
                                ;33751          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;33752          RC[T2]_Q,                       ;  (PC)+, G IMMEDIATE
                                ;33753          Q_IB.DATA,                      ;  GET SECOND PART INTO Q
                                ;33754          C[R.IB.COND,                    ;  CLEAR IT
                                ;33755          PC_PC+4,                        ;  PUSH PC PAST SECOND PART
U 1419, F001,28BC,00F0,F996,0000,04D0   ;33756          IB.TEST?,J/C.IQ                 ;  USE OLD C-FORK
                                ;33757
                                ;33758  141A:   ;-------------------------------;
                                ;33759          TRAP.ACC[1],                    ;  RESET THE FPA
U 141A, 0000,00BC,0080,F800,0000,0001   ;33760          J/RSVMOD                        ;-(PC)
                                ;33761
                                ;33762  141B:   ;-------------------------------;
                                ;33763          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;33764          VA_Q,                           ;@(PC)+ ABSOLUTE MODE
                                ;33765          CLR.IB.SPEC,
U 141B, D001,28BC,0080,F800,0200,1910   ;33766          DATA.TYPE?,J/G.M                ;  TEST FOR READ/WRITE, G/H-FORMAT
                                ;33767
                                ;33768  141C:   ;-------------------------------;
U 141C, 0000,003C,0180,F800,0000,0001   ;33769          J/RSVMOD                        ;INDEX MODE, R=PC
                                ;33770
                                ;33771  141D:   ;-------------------------------;
U 141D, 0000,003C,0180,F800,0000,0001   ;33772          J/RSVMOD                        ;NESTED INDEX MODE, R=PC
                                ;33773
```

```
                                ;33774  141F:   ;-------------------------------;****HUGE FORMAT IMMEDIATE****
                                ;33775          RC[T2]_Q,                       ;QUAD IMMEDIATE
U 141F, 0001,203C,0180,F996,0000,1885   ;33776          PC_PC+4                         ;PUSH PC PAST SECOND PART
                                ;33777
                                ;33778  =0****  ;0*******************************;
                                ;33779          Q_IB.DATA,                      ;GET SECOND PART INTO Q
U 1885, F000,0B3D,01F0,F800,0000,1870   ;33780          C[R.IB.COND,IB.TEST?,CALL,J/G.IQ;
                                ;33781
                                ;33782          ;1*******-------- ---------------;
                                ;33783          ID[T2]_D,                       ;  STORE LONG FRAC1
U 1895, 0000,003C,C980,3C06,0C00,188C   ;33784          PC_PC+4                         ;PUSH PC PAST SECOND PART
                                ;33785
                                ;33786  =0****  ;0*******************************;
                                ;33787          Q_IB.DATA,                      ;GET SECOND PART INTO Q
U 188C, F000,0B3D,01F0,F800,0000,1870   ;33788          C[R.IB.COND,IB.TEST?,CALL,J/G.IQ;
                                ;33789
                                ;33790          ;1*******-----------------------;
                                ;33791          RC[T3]_D,                       ;  STORE LONG FRAC2 IN RC3
U 189C, 0001,003C,0180,F99E,0000,188D   ;33792          PC_PC+4                         ;  PUSH PC PAST SECOND PART
                                ;33793
                                ;33794  =0****  ;0*******************************;
                                ;33795          Q_IB.DATA,                      ;GET SECOND PART INTO Q
U 188D, F000,0B3D,01F0,F800,0000,04D0   ;33796          C[R.IB.COND,IB.TEST?,CALL,J/C.IQ;
                                ;33797
                                ;33798          ;1*******-----------------------;
                                ;33799          ID[T3]_D,                       ;  STORE LONG FRAC3 IN T3
                                ;33800          Q_RC[T2],                       ;  Q GETS LONG EXP
U 189D, 0010,003C,CDC0,3D10,0000,0014   ;33801          RETURN[14]                      ;  RETURN14 FOR H FORMAT IMMEDIATE
                                ;33802  =00
                                ;33803  G.IQ:   ;00----------------------------;
U 1870, 0000,003D,0180,F800,0000,0E64   ;33804          CALL,J/IB.TBM                   ;  TB MISS IN IB
                                ;33805
                                ;33806          ;01---------------------------;
U 1871, 0000,003D,0180,F800,0000,0B80   ;33807          CALL,J/IB.ERR                   ;  IB ERROR
                                ;33808
                                ;33809          ;10---------------------------;
                                ;33810          Q_IB.DATA,                      ;  KEEP ON TRYING
                                ;33811          C[R.IB.COND,
U 1872, F000,0B3C,01F0,F800,0000,1870   ;33812          IB.TEST?,J/G.IQ
                                ;33813
                                ;33814          ;11---------------------------;
U 1873, 0C00,003E,01E0,F800,0000,0010   ;33815          D_Q,Q_D,RETURN10                ;  GOT THE DATA, LEAVE IT IN D
```

```
                                ;33816  ;SPECIAL G-FORK STATES
                                ;33817
                                ;33818  1487:   ;-------------------------------; HERE WE SHOULD NEVER GET
J 1487, 0000,003D,0180,F800,0000,0EE0   ;33819          CALL,J/EH.USEQ                  ; 'UNUSED' LOCATION FOUND IN IB ROM
                                ;33820
                                ;33821  147C:   ;-------------------------------; IB HAD NO DATA BECAUSE OF
J 147C, 0000,003D,0180,F800,0000,0E64   ;33822          CALL,J/IB.TBM                   ; TB MISS.  REFILL IT
                                ;33823
                                ;33824  147D:   ;-------------------------------; IB HAD NO DATA BECAUSE OF
J 147D, 0000,003D,0180,F800,0000,0B80   ;33825          CALL,J/IB.ERR                   ; ANY ERROR.  FIND OUT WHAT HAPPENED
                                ;33826
                                ;33827  147E:
                                ;33828  SPECG:  ;-------------------------------; IB IS WAITING FOR DATA
                                ;33829          LAB_R(SP1),                     ; STALL
                                ;33830          Q_IB.DATA,CLR.IB.COND,PC_PC+N,  ;
                                ;33831          MCT/ALLOW.IB.READ,              ;
J 147E, F000,003F,01F0,F847,0000,1400   ;33832          SUB/SPEC,J/G.FORK               ;
                                ;33833
                                ;33834  147F:   ;-------------------------------; HERE IF INTERRUPT REQUEST UP
U 147F, 0000,003C,0180,F800,0000,04F8   ;33835          J/INT.B                         ; GO BACKUP REGISTERS, SERVE INTERRUPT
                                ;33836
                                ;33837
                                ;33838  ;HERE TO WRITE BACK THE RESULT OF AN INSTRUCTION WITH A MODIFY DESTINATION.
                                ;33839  ; ASSIGNED AN ADDRESS ON CFORK BECAUSE MANY 2-OPERAND INSTRUCTIONS ARE
                                ;33840  ; EXECUTED BY THE SAME CODE AS THE 3-OPERAND COUNTERPART, AND CONCLUDE WITH
                                ;33841  ; THE WRITE.G.DEST OPERATION, WHICH EVALUATES THE THIRD SPECIFIER IN THE
                                ;33842  ; 3-OPERAND FORM, AND COMES HERE FOR THE 2-OPERAND FORM.
                                ;33843  1441:
                                ;33844  STORE.G:;-------------------------------; STORE MODIFIED G FORMAT
                                ;33845          TRAP.ACC[1],                    ; RESET THE FPA
                                ;33846          VA_RC[T7],                      ; RESTORE OPERAND ADDRESS
U 1441, 0010,17B8,0080,F938,0200,1858   ;33847          STATE0?                         ; IS THIS MOVH ?
                                ;33848
                                ;33849  =0      ;0-------------------------------;
U 1858, 0000,003C,0180,F800,0000,1558   ;33850          J/MOVH
                                ;33851
                                ;33852          ;1-------------------------------;
                                ;33853          CACHE_D[LONG],                  ; WRITE LONG EXP
U 1859, 0000,003C,0180,3000,0000,03F8   ;33854          J/C.WQ1                         ; USE OLD DOUBLE WRITE ROUTINE
                                ;33855
                                ;33856  ;HERE IS THE SAME FUNCTION FOR OCTO/HUGE  OPERATIONS
                                ;33857  1444:
                                ;33858  STOR.H: ;-------------------------------; STORE MODIFIED H FORMAT
                                ;33859          TRAP.ACC[1],                    ; RESET THE FPA
                                ;33860          VA_RC[T7],Q_ALU,                ; RELOAD OPERAND ADDRESS, WHICH GOT
U 1444, 0010,00B8,00C0,F938,0200,1911   ;33861          J/G.WH                          ; INCREMENTED IN FETCHING OPERAND (???
```

```
                                ;33862  ;HERE FOR THE SECOND AND SUBSEQUENT STATES OFF C-FORK SUBROUTINE
                                ;33863  =000
                                ;33864  G.M:    ;000-------------------------------; GET HERE BY DATA.TYPE, WRITE G -FORMAT
                                ;33865          TRAP.ACC[1],                      ; RESET THE FPA
                                ;33866          RC[T0]_Q,                         ; DO NOT WRITE YET
                                ;33867          Q_D,                              ; SAVE DATA
                                ;33868          D[LONG]_CACHE.WCHK,               ; READ FIRST LONGWORD WITH WRITECHECK
U 1910, 0001,20BC,00E0,5180,0000,1966   ;33869          J/WRITE.GRAND
                                ;33870
                                ;33871  G.WH:   ;001-------------------------------; WRITE HUGE FORMAT
                                ;33872          TRAP.ACC[1],                      ; RESET THE FPA
                                ;33873          ALU_Q+K[.8],VA_ALU,               ; VA GETS ADDRESS OF LONG FRAC2
U 1911, 0019,2094,0080,F800,0200,1982   ;33874          J/WRITE.HUGE
                                ;33875
                                ;33876  =100    ;100-------------------------------;
                                ;33877          TRAP.ACC[1],                      ; RESET THE FPA
                                ;33878          D[LONG]_CACHE.IBCHK,              ; READ LONG EXP WITH IB-CHECK
                                ;33879          RC[T7]_Q,                         ; SAVE ADDRESS IN RC7
                                ;33880          Q_D,                              ; SAVE PREVIOUS OPERAND
U 1914, 0001,20BC,00E0,59B8,0000,1A41   ;33881          J7G.RQ                            ; LABEL IN C-FORK (=C.M+4)
                                ;33882
                                ;33883          ;101-------------------------------;READ HUGE FORMAT
                                ;33884          TRAP.ACC[1],                      ; RESET THE FPA
                                ;33885          D[LONG]_CACHE.IBCHK,              ; READ LONG EXP WITH IB-CHECK
                                ;33886          RC[T7]_Q,                         ; SAVE ADDRESS IN RC7
                                ;33887          Q_D,                              ; SAVE PREVIOUS OPERAND
U 1915, 0001,20BC,00E0,59B8,0000,19E9   ;33888          J7READ.HUGE
                                ;33889
                                ;33890          ;110-------------------------------;
U 1916, 0000,003E,0180,F800,0000,0010   ;33891          RETURN10                          ; Q HAS ADDRESS
                                ;33892
                                ;33893          ;111-------------------------------; H-FORMAT ADDRESS
                                ;33894          ALU_Q,SET.CC(INST),               ; MOVAH OR PUSHAH
                                ;33895          D_Q,LAB_R[SP],                    ; D GETS ADDRESS,LATCH SP
U 1917, 0C01,FB3C,0180,FA70,0070,191D   ;33896          IR0?                              ; WHICH INSTRUCTION IS IT ?
                                ;33897
                                ;33898  =1101   ;1101-------------------------------;
U 191D, F000,003F,01F0,F847,0000,0200   ;33899          B.FORK                            ;  MOVAH (7EFD)
                                ;33900
                                ;33901          ;1111-------------------------------;  PUSHAH (7FFD)
                                ;33902          R[SP]&VA_LA-K[.4].RLOG,           ;  LOAD DECREMENTED ADDR INTO SP
U 191F, 0018,0004,1180,FAF0,0200,0341   ;33903          J/STORE                           ;  STORE ADDRESS ON STACK
                                ;33904
                                ;33905  WRITE.GRAND:
                                ;33906          ;--------------------------------;
                                ;33907          TRAP.ACC[1],                      ;  RESET THE FPA
                                ;33908          VA_VA+4,                          ;  POINT TO NEXT LONGWORD
U 1966, 0810,00B8,0080,F90B,0000,196A   ;33909          D_RC[T1]                          ;  GET LONG FRAC
                                ;33910
                                ;33911          ;--------------------------------;
U 196A, 0000,003C,0180,3000,0000,197E   ;33912          CACHE_D[LONG]                     ;  WRITE LONG FRAC
                                ;33913
                                ;33914          ;--------------------------------;
                                ;33915          VA_RC[T0],                        ;  LOAD ADDRESS OF LONG EXP
U 197E, 0C10,0038,0180,F900,0200,03FD   ;33916          D_Q,J/STOR.L                      ;  WRITE LONG EXP
```

```
                                ;33917  WRITE.HUGE:
                                ;33918          ;-------------------------------;
                                ;33919          RC[T2]_Q,                       ;  RC2 GETS ADDRESS OF LONG EXP
U 1982, 0001,E03C,0180,5190,0000,1991   ;33920          D[INST.DEP]_CACHE.WCHK          ;  TEST LAST TWO LONGWORDS
                                ;33921
                                ;33922          ;-------------------------------;
                                ;33923          VA_RC[T2],                      ;  VA GETS LOW ADDRESS
U 1991, 0010,0038,C1F0,2D10,0200,19A3   ;33924          Q_ID[T0]                        ;  Q GETS LONG FRAC1
                                ;33925
                                ;33926          ;-------------------------------;
U 19A3, 0810,0038,0180,F900,0000,19A9   ;33927          D_RC[T0]                        ;  D GETS LONG EXP
                                ;33928
                                ;33929  WRITE.HUGE.1:
                                ;33930          ;-------------------------------; ENTER HERE FROM EMODH
U 19A9, 0000,003C,0180,3000,0000,19B1   ;33931          CACHE_D[LONG]                   ;  WRITE LONG EXP
                                ;33932
                                ;33933          ;-------------------------------;
                                ;33934          D_Q,                            ;  D GETS LONG FRAC1
                                ;33935          VA_VA+4,                        ;  POINT TO ITS ADDRESS
U 19B1, 0C10,0038,01C0,F90B,0000,19B6   ;33936          Q_RC[T1]                        ;  Q GETS LONG FRAC2
                                ;33937
                                ;33938          ;-------------------------------;
U 19B6, 0000,003C,0180,3000,0000,19C1   ;33939          CACHE_D[LONG]                   ;  WRITE LONG FRAC1
                                ;33940
                                ;33941          ;-------------------------------;
                                ;33942          VA_VA+4,                        ;  KEEP ON INCREMENTING
                                ;33943          D_Q,                            ;  D GETS LONG FRAC2
U 19C1, 0C00,003C,C5F0,2C03,0000,19D4   ;33944          Q_ID[T1]                        ;  Q GETS LONG FRAC3
                                ;33945
                                ;33946          ;-------------------------------;
                                ;33947          CACHE_D[LONG],                  ;  WRITE LONG FRAC2
                                ;33948          RC[T1]_Q,                       ;  TO USE OLD CODE
U 19D4, 0001,203C,0180,3188,0000,03F8   ;33949          J/C.WQT                         ;
```

```
                                ;33950  READ.HUGE:
                                ;33951          ;-------------------------------;
                                ;33952          RC[T2]_D,                       ;  STORE IT IN RC2
U 19E9, 0001,003C,0180,F993,0000,19FB   ;33953          VA_VA+4                         ;  POINT TO LONG FRAC1
                                ;33954
                                ;33955          ;-------------------------------;
U 19FB, 0000,003C,0180,4000,0000,1A09   ;33956          D[LONG]_CACHE                   ;  READ LONG FRAC1
                                ;33957
                                ;33958          ;-------------------------------;
                                ;33959          Q_D,ID[T2]_D,                   ;  STORE IT IN T2
U 1A09, 0000,003C,C9E0,3C03,0000,1A1D   ;33960          VA_VA+4                         ;  INCREMENT ADDRESS TO LONG FRAC2
                                ;33961
                                ;33962          ;-------------------------------;
U 1A1D, 0000,003C,0180,4000,0000,1A25   ;33963          D[LONG]_CACHE                   ;  READ LONG FRAC2
                                ;33964
                                ;33965          ;-------------------------------;
                                ;33966          RC[T3]_D,                       ;  STORE LONG FRAC2 IN RC3
U 1A25, 0001,003C,0180,F99B,0000,1A29   ;33967          VA_VA+4                         ;  POINT TO LONG FRAC3
                                ;33968
                                ;33969          ;-------------------------------;
U 1A29, 0000,003C,0180,4000,0000,1A3A   ;33970          D[LONG]_CACHE                   ;  D GETS LONG FRAC3
                                ;33971
                                ;33972          ;-------------------------------;
                                ;33973          ID[T3]_D,                       ;  STORE LONG FRAC3 IN ID[T3]
                                ;33974          Q_RC[T2],                       ;  Q GETS LONG EXP
                                ;33975          D_Q,                            ;  D GETS LONG FRAC1
U 1A3A, 0C10,003A,CDC0,3D10,0000,0014   ;33976          RETURN[14]                      ;  RETURN14 FOR H FORMAT
                                ;33977
                                ;33978  G.RQ:   ;-------------------------------; HERE TO READ 2nd PART OF A G OPERAND
                                ;33979          RC[T2]_D,                       ;  SAVE FIRST PART IN RC2
U 1A41, 0001,003C,0180,F993,0000,1A49   ;33980          VA_VA+4
                                ;33981
                                ;33982          ;-------------------------------;
                                ;33983          D[LONG]_CACHE.IBCHK,            ;  READ SECOND LONGWORD
U 1A49, 0000,003E,0180,5800,0000,0010   ;33984          RETURN10
```

```
                                ;33985  .TOC    ""      G & H floating point  : Address Specifier Evaluation''
                                ;33986  ;Control passes to this point by CALL,J/ASPCG
                                ;33987  ; and returns to the call site ''or'' 60, if the specifier is valid,
                                ;33988  ; or the call site ''or'' 61 if the specifier is register mode, which will
                                ;33989  ; be invalid unless the specifier is a field source.  At the return, Q
                                ;33990  ; contains the longword that was in D at the entry, D and VA contain
                                ;33991  ; the address implied by the specifier.
                                ;33992  157E:
                                ;33993  ASPCG:  ;-------------------------------;HANG HERE IF IB MUST STALL
                                ;33994          Q_IB.DATA,                      ;GET SPECIFIER DATA FROM ISTREAM
                                ;33995          LAB_R(SP1),                     ;LOAD LATCHES FROM BASE SPECIFIER
                                ;33996          CLR.IB.COND,                    ;DISCARD BASE OPERAND SPECIFIER
                                ;33997          PC_PC+N,                        ;STEP PC OVER IT
                                ;33998          MCT/ALLOW.IB.READ,              ;LET IB GET NEEDED DATA
U 157E, F000,003F,01FC,F847,0000,1500   ;33999          SUB/SPEC,J/ASPCG.B              ;EVALUATE THE SPECIFIER
                                ;34000
                                ;34001  1500:
                                ;34002  ASPCG.B:;-------------------------------;
U 1500, 0000,003C,0180,F800,0000,0001   ;34003          J/RSVMOD                        ;S^# SHORT LITERAL NOT LEGAL AS ASRC
                                ;34004
                                ;34005  1501:   ;-------------------------------;
U 1501, 0000,003C,0180,F800,0000,0001   ;34006          J/RSVMOD                        ;RESERVED MODE
                                ;34007
                                ;34008  1502:   ;-------------------------------;
U 1502, 0000,003C,0180,F800,0000,0001   ;34009          J/RSVMOD                        ;QUAD/DOUBLE SHORT LITERAL
                                ;34010
                                ;34011  1503:   ;-------------------------------;
U 1503, 0000,003C,0180,F800,0000,0001   ;34012          J/RSVMOD                        ;RESERVED MODE
                                ;34013
                                ;34014  1504:   ;-------------------------------;
                                ;34015          Q_D,                            ;REGISTER
                                ;34016          R(PRN)_LA+K[ZERO].RLOG,          ;RECORD REGISTER NUMBER IN RLOG
                                ;34017          D_ALU,                          ;GET REGISTER CONTENTS TO D
U 1504, 0818,001A,19E0,F8D8,0000,0061   ;34018          RETURN61                        ;RETURN IT IN CASE FIELD SOURCE
                                ;34019
                                ;34020  1524:   ;-------------------------------;
                                ;34021          Q_D,                            ;WRITE REGISTER
                                ;34022          R(PRN)_LA+K[ZERO].RLOG,          ;RECORD REGISTER NUMBER IN RLOG
                                ;34023          D_ALU,                          ;GET REGISTER CONTENTS TO D
U 1524, 0818,001A,19E0,F8D8,0000,0061   ;34024          RETURN61                        ;RETURN IT IN CASE FIELD SOURCE
                                ;34025
                                ;34026  1505:   ;-------------------------------;
U 1505, 0000,003C,0180,F800,0000,0001   ;34027          J/RSVMOD
                                ;34028
                                ;34029  1506:   ;-------------------------------;
                                ;34030          Q_D,                            ;QUAD REGISTER
                                ;34031          R(PRN)_LA+K[ZERO].RLOG,          ;RECORD REGISTER NUMBER IN RLOG
                                ;34032          D_ALU,                          ;RETURN CONTENTS TO CALLER
U 1506, 0818,001A,19E0,F8D8,0000,0061   ;34033          RETURN61
                                ;34034
                                ;34035  1526:   ;-------------------------------;
                                ;34036          Q_D,                            ;WRITE QUAD REGISTER
                                ;34037          R(PRN)_LA+K[ZERO].RLOG,          ;RECORD REGISTER NUMBER IN RLOG
                                ;34038          D_ALU,                          ;RETURN CONTENTS TO CALLER
U 1526, 0818,001A,19E0,F8D8,0000,0061   ;34039          RETURN61                        ;
```

```
                                ;34040  1507:   ;--------------------------------;
U 1507, 0000,003C,0180,F800,0000,0001   ;34041          J/RSVMOD                        ;ILLEGAL QUAD REG
                                ;34042  1508:
                                ;34043  ASPCG.DR:
                                ;34044          ;--------------------------------;(R)
                                ;34045          Q_D,D&VA_LA,LC_RC[T7],           ;RECOVER INDEX(PROPERLY SHIFTED),IF ANY
U 1508, 0800,003E,01E0,F938,0200,0060   ;34046          RETURN60                        ;RETURN IT
                                ;34047
                                ;34048  1509:   ;--------------------------------;
                                ;34049          R(PRN)_LA+K[SP1.CON].RLOG,       ;UPDATE THE STACK POINTER
U 1509, 0018,0018,15C0,F8D8,0000,1A5D   ;34050          Q_ALU                           ;
                                ;34051
                                ;34052          ;--------------------------------;
                                ;34053          R(PRN)_Q+K[SP1.CON].RLOG,        ;UPDATE THE STACK POINTER
U 1A5D, 0019,2018,1580,F8D8,0000,1508   ;34054          J/ASPCG.DR                      ;THEN LOAD UN-INCREMENTED ADDR
                                ;34055
                                ;34056  150A:   ;--------------------------------;
U 150A, 0018,0004,15C0,F8D8,0000,1A95   ;34057          R(PRN)_LA-K[SP1.CON].RLOG,Q_ALU ;  -(R) AUTO DECREMENT
                                ;34058
                                ;34059          ;--------------------------------;
                                ;34060          Q_D,R(PRN)_Q-K[SP1.CON].RLOG,    ;-(R) AUTO DECREMENT
                                ;34061          D_ALU,                          ; USE DECREMENTED ADDR
U 1A95, 0819,2004,15E0,F8D8,0000,1AAD   ;34062          J7ASPCG.DF                      ;GO LOAD LC BEFORE RETURNING
                                ;34063
                                ;34064  150B:   ;--------------------------------;
U 150B, 0000,003C,01E0,F800,0200,1A9E   ;34065          Q_D,VA_LA                       ;@(R)+ AUTO INCREMENT DEFERED
                                ;34066
                                ;34067          ;--------------------------------;
                                ;34068          D[LONG]_CACHE,                   ;GET INDIRECT WORD
                                ;34069          R(PRN)_LA+K[.4].RLOG,            ; WHILE UPDATING REGISTER
U 1A9E, 0018,0018,1180,40D8,0000,1AAD   ;34070          J/ASPCG.DF                      ;THEN JOIN COMMON CODE
                                ;34071
                                ;34072  150C:   ;--------------------------------;
                                ;34073          RC[T7]_LA+LB.CTX,                ;INDEX MODE, CONTEXT SHIFT INDEX
U 150C, 006C,C015,0180,F9B8,0000,157E   ;34074          CALL,J7ASPCG                    ;GO AROUND AGAIN
                                ;34075
                                ;34076  156C:   ;--------------------------------;
U 156C, 0811,0016,0180,F800,0200,0060   ;34077          D&VA_D+LC,RETURN60              ;RETURN THE INDEXED VALUE
                                ;34078
                                ;34079  150D:   ;--------------------------------;
                                ;34080          Q_D,D&VA_Q+LB.PC,                ;D(R) DISPLACEMENT MODE.
                                ;34081          CIR.IB.SPEC,                     ;DISCARD THE SPECIFIER
U 150D, D805,2016,01E0,F938,0200,0060   ;34082          LC_RC[T7],RETURN60              ;LOAD UP INDEX, IF ANY
                                ;34083
                                ;34084  150F:   ;--------------------------------;
                                ;34085          Q_D,VA_Q+LB.PC,                  ;@D(R) DISPLACEMENT DEFERED
U 150F, D005,2014,01E0,F800,0200,1AA1   ;34086          CIR.IB.SPEC                     ;DROP THE SPECIFIER
                                ;34087
                                ;34088          ;--------------------------------;
U 1AA1, 0000,003C,0180,4000,0000,1AAD   ;34089          D[LONG]_CACHE                   ;GET INDIRECT, GO USE IT AS ADDR
                                ;34090
                                ;34091  ASPCG.DF:
                                ;34092          ;--------------------------------;****COULD USE OLD CODE HERE****
                                ;34093          VA_D,                            ;USE POINTER AS ADDRESS
U 1AAD, 0001,003E,0180,F938,0200,0060   ;34094          LC_RC[T7],RETURN60              ;RECOVER INDEX, IF ANY
```

```
                                ;34095  ;HERE ARE VARIANTS OF THE ASPC ENTRY POINTS FOR R=PC
                                ;34096
                                ;34097  1514:  ;-------------------------------;
J 1514, 0000,003C,0180,F800,0000,0001   ;34098          J/RSVMOD                        ;PC REGISTER MODE
                                ;34099
                                ;34100  1515:  ;-------------------------------;
J 1515, 0000,003C,0180,F800,0000,0001   ;34101          J/RSVMOD                        ;ILLEGAL REGISTER MODE, R=PC
                                ;34102
                                ;34103  1516:  ;-------------------------------;
J 1516, 0000,003C,0180,F800,0000,0001   ;34104          J/RSVMOD                        ;PC QUAD REGISTER MODE
                                ;34105
                                ;34106  1517:  ;-------------------------------;
J 1517, 0000,003C,0180,F800,0000,0001   ;34107          J/RSVMOD                        ;ILLEGAL QUAD REGISTER MODE, R=PC
                                ;34108
                                ;34109  1518:  ;-------------------------------;
J 1518, 0000,003C,0180,F800,0000,0001   ;34110          J/RSVMOD                        ;(PC)
                                ;34111
                                ;34112  1519:  ;-------------------------------;
                                ;34113          Q_D,                            ;(PC)+ IMMEDIATE MODE
                                ;34114          D_PC,                           ;GET PC WHICH THE IB INCREMENTED
U 1519, D814,0038,01E0,F800,0000,1AB2   ;34115          C[R.IB.SPEC
                                ;34116
                                ;34117          ;-------------------------------;
                                ;34118          D&VA_D-K[SP1.CON],              ;COMPUTE THE UNINCREMENTED ADDRESS
                                ;34119          LC_RC[T7],                      ;RECOVER INDEX, IF ANY
U 1AB2, 0819,0002,1580,F938,0200,0060   ;34120          RETURN60
                                ;34121
                                ;34122  151A:  ;-------------------------------;
U 151A, 0000,003C,0180,F800,0000,0001   ;34123          J/RSVMOD                        ;-(PC)
                                ;34124
                                ;34125  151B:  ;-------------------------------;
                                ;34126          Q_D,D&VA_Q,                     ;@(PC)+ ABSOLUTE MODE
                                ;34127          LC_RC[T7],                      ;GET BACK INDEX
                                ;34128          CLR.IB.SPEC,
U 151B, DC01,203E,01E0,F938,0200,0060   ;34129          RETURN60
                                ;34130
                                ;34131  151C:  ;-------------------------------;
U 151C, 0000,003C,0180,F800,0000,0001   ;34132          J/RSVMOD                        ;INDEX MODE, R=PC
                                ;34133
                                ;34134  151D:  ;-------------------------------;
U 151D, 0000,003C,0180,F800,0000,0001   ;34135          J/RSVMOD                        ;NESTED INDEX MODE, R=PC
```

```
                                    ;34136  ;HERE WHEN IB ROMS BELIEVE WE SHOULDN'T GET HERE
                                    ;34137
                                    ;34138  1587:   ;-------------------------------;
U 1587, 0000,003D,0180,F800,0000,0EE0   ;34139          CALL,J/EH.USEQ                  ;"UNUSED" LOCATION IN IB ROM
                                    ;34140
                                    ;34141  ;HERE OFF ASPCG, WHEN INSTRUCTION BUFFER DOES NOT HAVE ENOUGH DATA
                                    ;34142
                                    ;34143  157C:   ;-------------------------------;
U 157C, 0000,003D,0180,F800,0000,0E64   ;34144          CALL,J/IB.TBM                   ;TB MISS.  REFILL IT
                                    ;34145
                                    ;34146  157D:   ;-------------------------------;
U 157D, 0000,003D,0180,F800,0000,0B80   ;34147          CALL,J/IB.ERR                   ;ANY ERROR.  FIND OUT WHAT HAPPENED
                                    ;34148
                                    ;34149  ;157E:  ;-------------------------------;
                                    ;34150  ;       ASPCG:                          ;STALL, WAITING FOR THE DATA
                                    ;34151
                                    ;34152  157F:   ;-------------------------------;INTERRUPT REQUEST DETECTED
U 157F, 0000,003C,0180,F800,0000,04F8   ;34153          J/INT.B                         ;BACKUP REGISTERS AND TAKE INTERRUPT
```

```
                                    ;34154  .TOC    ""      G & H floating point  : E-FORK"
                                    ;34155
                                    ;34156  ;BRANCH POINT FOR EMOD-EXTENDER AND OTHER ODD SPECIFIER EVALUATIONS.
                                    ;34157
                                    ;34158  1600:
                                    ;34159  E.FORK: ;-------------------------------;
U 1600, 0C00,003E,01E0,F800,0000,0010 ;34160        D_Q,Q_D, RETURN10               ;  SHORT LITERAL
                                    ;34161
                                    ;34162  1604:   ;-------------------------------;
                                    ;34163          Q_D,D_LA,                       ;  D GETS REGISTER VALUE
U 1604, 0800,003E,01E0,F800,0000,0010 ;34164        RETURN10
                                    ;34165  1608:
                                    ;34166  E.DR:   ;-------------------------------;
                                    ;34167          VA_LA,                          ;  MODE IS (R)
                                    ;34168          Q_D,                            ;  SAVE OLD D-VALUE IN Q
U 1608, 0000,003C,01E0,F800,0200,1AC9 ;34169        J7E.M                           ;  JOIN EMOD-MEMORY-REFERENCE
                                    ;34170
                                    ;34171  1609:   ;-------------------------------;
                                    ;34172          R(PRN)_LA+K[.2].RLOG,           ;  (R)+
U 1609, 0018,0018,0980,F8D8,0000,1608 ;34173        J/E.DR
                                    ;34174
                                    ;34175  160A:   ;-------------------------------;
                                    ;34176          R(PRN)_LA-K[.2].RLOG,           ;  (R)-
                                    ;34177          VA_ALU, Q_D,                    ;  LOAD VA WITH OPERAND ADDRESS
U 160A, 0018,0004,09E0,F8D8,0200,1AC9 ;34178        J/E.M
                                    ;34179
                                    ;34180  160B:   ;-------------------------------;
                                    ;34181          VA_LA,                          ;  @(R)+
U 160B, 0000,003C,01E0,F800,0200,1AB4 ;34182        Q_D
                                    ;34183
                                    ;34184          ;-------------------------------;
                                    ;34185          D[LONG]_CACHE,                  ;  D GETS OPERAND ADDRESS
                                    ;34186          R(PRN)_LA+K[.4].RLOG,           ;  AUTO INCREMENT REGISTER
U 1AB4, 0018,0018,1180,40D8,0000,1AB7 ;34187        J/E.DF
                                    ;34188
                                    ;34189  E.DF:   ;-------------------------------;
U 1AB7, 0001,003C,0180,F800,0200,1AC9 ;34190        VA_D, J/E.M                     ;  VA GETS OPERAND ADDRESS
                                    ;34191
                                    ;34192  E.M:    ;-------------------------------;
                                    ;34193          D[WORD]_CACHE,                  ;  READ OPERAND
U 1AC9, 0000,403E,0180,4000,0000,0010 ;34194        RETURN10
                                    ;34195
                                    ;34196  160C:   ;-------------------------------;
                                    ;34197          RC[T7]_LA+LB.CTX,               ;  INDIRECT
U 160C, 006C,C015,0180,F9B8,0000,157E ;34198        CALL,J7ASPCG
                                    ;34199
                                    ;34200  166C:   ;-------------------------------;
                                    ;34201          D_Q,VA_D+LC,                    ;  VA GETS OPERAND ADDRESS
U 166C, 0C11,0014,0180,F800,0200,1AC9 ;34202        J7E.M                           ;  MAKE MEMORY REFERENCE
                                    ;34203
                                    ;34204  160D:   ;-------------------------------;
                                    ;34205          VA_Q+LB.PC,                     ;  INDIRECT
U 160D, D005,2014,01E0,F800,0200,1AC9 ;34206        Q_D,CLR.IB.SPEC,J/E.M           ;
```

```
                                ;34207  160F:   ;--------------------------------;
                                ;34208          Q_D,VA_Q+LB.PC,
U 160F, D005,2014,01E0,F800,0200,1ACC   ;34209          CLR.IB.SPEC
                                ;34210
                                ;34211          ;--------------------------------;
U 1ACC, 0000,003C,0180,4000,0000,1AB7   ;34212          D[LONG]_CACHE, J/E.DF
                                ;34213
                                ;34214  1619:   ;--------------------------------;
                                ;34215          D_Q,                            ;  IMMMEDIATE
                                ;34216          SC_K[.FFF8],                    ;  SHIFT COUNT OF -8.
U 1619, 0C00,003C,7180,F804,0084,78C5   ;34217          PC_PC+1
                                ;34218
                                ;34219  =0****  ;0*******************************;
                                ;34220          Q_IB.DATA,                      ;  GET SECOND PART OF WORD
                                ;34221          CLR.IB.COND,                    ;  CLEAR SPECIFIER
                                ;34222          IB.TEST?,                       ;  NEED ONE MORE BYTE
U 18C5, F000,0B3D,01F0,F800,0000,04D0   ;34223          CALL,J/C.IQ
                                ;34224
                                ;34225          ;1*******************************;
                                ;34226          D_DAL.SC,                       ;  GET NEW BYTE INTO D<31:24>
                                ;34227          Q_D,                            ;  Q GETS LOW BYTE
U 18D5, 0D00,003C,69E0,F800,0084,7ACE   ;34228          SC_K[.FFE8]                     ;  -24
                                ;34229
                                ;34230          ;--------------------------------;
                                ;34231          Q_ID[T9],                       ;  RESTORE MULTIPLIER FRAC LONG
U 1ACE, 0D00,003E,E5F0,2C00,0000,0010   ;34232          D_DAL.SC, RETURN10              ;  GET THE DATA IN RIGHT ORDER
                                ;34233
                                ;34234  161B:   ;--------------------------------;
                                ;34235          VA_Q, Q_D,                      ;  IMMEDIATE DEFFERRED
                                ;34236          CLR.IB.SPEC,
U 161B, D001,203C,01E0,F800,0200,1AC9   ;34237          J/E.M
                                ;34238
                                ;34239  167C:   ;--------------------------------;
U 167C, 0000,003D,0180,F800,0000,0E64   ;34240          CALL,J/IB.TBM
                                ;34241
                                ;34242  167D:   ;--------------------------------;
U 167D, 0000,003D,0180,F800,0000,0B80   ;34243          CALL,J/IB.ERR
                                ;34244
                                ;34245  167E:
                                ;34246  SPECE:  ;--------------------------------;
                                ;34247          LAB_R(SP1),                     ;  LA AND LB GETS REGISTER VALUE
                                ;34248          Q_IB.DATA,                      ;  Q GETS INSTRUCTION STREAM DATA
                                ;34249          CLR.IB.COND,
                                ;34250          PC_PC+N,                        ;  INCREMENT PC
                                ;34251          MCT/ALLOW.IB.READ,              ;  TURN ON IB-READ
U 167E, F000,003F,01F0,F847,0000,1600   ;34252          SUB/SPEC,J/E.FORK
                                ;34253
                                ;34254  167F:   ;--------------------------------;
U 167F, 0000,003C,0180,F800,0000,04F8   ;34255          J/INT.B
```

```
;34256  .TOC    ''      G & H floating point  : GRAND FLOATING NOTATION''
;34257
;34258  ; THE FOLLOWING NOTATION IS USED IN COMMENTS BELONGING TO THE
;34259  ; MICROCODE FOR THE GRAND FLOATING INSTRUCTION SET.
;34260  ;
;34261  ;       OP1 LONG EXP = LONGWORD OF 1. OPERAND CONTAINNING EXPONENT
;34262  ;                      AND HIGH FRACTION BITS.
;34263  ;       OP1 LONG FRAC = LONGWORD OF 1. OPERAND CONTAINING FRACTION BITS ONLY
;34264  ;       OP1 EXP = EXPONENT OF OPERAND 1
;34265  ;       OP1 FRAC <LS> = LONGWORD CONTAINING LOW PART OF ALIGNED FRACTION
;34266  ;       OP1 FRAC <MS> = LONGWORD CONTAINING MOST SIGNIFICANT BITS OF ALIGNED
;34267  ;                       FRACTION.
;34268  ;       SRC = OP1
;34269  ;       DST = OPX WHERE X IS NUMBER OF OPERANDS.
;34270  ;
;34271  ;       OP1 FRAC <MS> LOOKS LIKE THIS:
;34272  ;       31                                  00
;34273  ;       ;-----------------------------------;
;34274  ;       ;  ;  ;                            ;
;34275  ;       ;  ;  ;                            ;
;34276  ;       ;0 ;1 ;   FRACTION <52:23>          ;
;34277  ;       ;  ;  ;                            ;
;34278  ;       ;-----------------------------------;
;34279  ;
;34280  ;       OP1 FRAC <LS> LOOKS LIKE THIS:
;34281  ;       31                          09      00
;34282  ;       ;-----------------------------------;
;34283  ;       ;                           ;       ;
;34284  ;       ;                           ;       ;
;34285  ;       ;   FRAC <22:0>             ;  00   ;
;34286  ;       ;                           ;       ;
;34287  ;       ;-----------------------------------;
```

```
                                ;34288  .TOC    ''      G & H floating point  : MODIFIED DOUBLE FLOATING -- CMPG''
                                ;34289
                                ;34290  ;DOUBLE PRECISION FLOATING POINT CMP.
                                ;34291
                                ;34292  ;INPUTS:
                                ;34293  ;               Q = OP1 LONG FRAC
                                ;34294  ;               RC0 = OP1 LONG EXP
                                ;34295  ;               D = OP2 LONG FRAC
                                ;34296  ;               RC2 = OP2 LONG EXP
                                ;34297
                                ;34298  ;ALGORITHM:
                                ;34299  ;       THE FOLLOWING TESTS ARE BEING PERFORMED IN ORDER:
                                ;34300  ;       1.      EXPONENTS
                                ;34301  ;       2.      FRACTION <0:20>
                                ;34302  ;       3.      FRACTION <21:36>
                                ;34303  ;       4.      FRACTION <37:52>
                                ;34304
                                ;34305  ;OUTPUTS:
                                ;34306  ;               SETS CONDITION CODES ON SRC-DST
                                ;34307
                                ;34308  ;TEMPORARIES:
                                ;34309  ;
                                ;34310
                                ;34311  16C1:   ;-------------------------------;  C1 + 1400
                                ;34312  CMPG:   TRAP.ACC[1],                    ;  RESET THE FPA
                                ;34313          ID[T1]_D,                       ;  Save OP2 LONG FRAC IN T1
                                ;34314          D_Q,                            ;  D GETS OP1 LONG FRAC
                                ;34315          Q_RC[T0],                       ;  Get OP1 LONG EXP FROM RC0
J 16C1, 0C10,00B8,C4C1,3D00,0000,1AD6   ;34316          SS_ALU15                        ;  STORE IT IN Q-REG.  SS GETS OP1 SIGN
                                ;34317
                                ;34318          ;-------------------------------;  CMPG
                                ;34319          TRAP.ACC[1],                    ;  RESET THE FPA
J 1AD6, 0019,20B4,3880,F800,0010,1ADE   ;34320          ALU_Q.AND.K[.7FF0], CLK.UBCC    ;  CLOCK OP1 EXPONENT FOR 0
                                ;34321
                                ;34322          ;-------------------------------;
                                ;34323          ID[T2]_D,                       ;  SAVE OP1 LONG FRAC IN T2
                                ;34324          D_RC[T2],                       ;  D GETS OP2 LONG EXP FROM RC2
                                ;34325          SS_SS.XOR.ALU15&SD_ALU15,       ;  SS gets XOR of two signs,
                                ;34326                                          ;  SD GETS OP2 SIGN
J 1ADE, 0810,0138,C985,3D10,0000,186C   ;34327          Z?                              ;  Test OP1 for ZERO
                                ;34328          ;-------------------------------;
                                ;34329  =0       ;BRANCH ON ALU Z-BIT
                                ;34330          ;0------------------------------;  OP1 NE 0
                                ;34331  CMPG0:  ALU_D.AND.K[.7FF0],             ;  ISOLATE EXPONENT BITS OF OP2
J 186C, 0019,0034,3980,F800,0010,1AE2   ;34332          CLK.UBCC, J/CMPG01
                                ;34333
                                ;34334          ;1------------------------------;  OP1 = 0
                                ;34335          ALU_Q.AND.K[.8000],             ;  ISOLATE SIGN BIT
J 186D, 0019,2034,4580,F800,0800,186C   ;34336          CHK.FLT.OPR,                    ;  CHECK FOR -0
                                ;34337          J/CMPG0
                                ;34338          ;-------------------------------;
```

```
U 1AE2, 0000,013C,0180,F800,0000,1874   ;34339  CMPG01: Z?                              ;  TEST FOR OP2 = 0
                                        ;34340          ;------------------------------;
                                        ;34341  =0      ;Branch on ALU Z-bit
                                        ;34342          ;0-----------------------------;  OP2 .NE. 0
                                        ;34343          ALU_Q-D, WORD, CLK.UBCC,       ;  Compare exponents, OP1 - OP2
U 1874, 001D,7200,0180,F800,0010,196E   ;34344          EALU?,J/CMPG1                  ;  Test XOR of signs
                                        ;34345
                                        ;34346          ;1-----------------------------;  OP2 .EQ. 0
                                        ;34347          ALU_D.AND.K[.8000],            ;  ISOLATE SIGN-BIT OF OP2
U 1875, 0019,0034,4580,F800,0800,1AE6   ;34348          CHK.FLT.OPR                    ;  CHECK FOR -0
                                        ;34349
                                        ;34350          ;------------------------------;
                                        ;34351          ALU_Q.AND.K[.FFF0],            ;  Get exponent part of OP1
                                        ;34352          N&Z_ALU.V&C_0, WORD,           ;  Set CC FROM OP1
U 1AE6, C019,6034,6D80,F804,4050,0062   ;34353          CLR.IB.OPC,PC_PC+1,J/IRD       ;  Complete
                                        ;34354
                                        ;34355          ;------------------------------;
                                        ;34356  =1110   ;Branch on SS-bit
                                        ;34357          ;1110--------------------------;  SAME SIGN
                                        ;34358  CMPG1:  ALU_Q-D, LONG, CLK.UBCC,       ;  CLOCK MS FRACTION BITS, OP1 - OP2
U 196E, 001D,3B00,0180,F800,0010,1923   ;34359          ALU?, J/CMPG2                  ;  Same sign, BRANCH ON exponents
                                        ;34360
                                        ;34361          ;1111--------------------------;  OPPOSITE SIGNS
                                        ;34362          ALU_D.XOR.K[.8000],            ;  Set condition codes on -dst
                                        ;34363                                         ;  SINCE SRC MAY BE 0
                                        ;34364          N&Z_ALU.V&C_0,WORD,            ;  Set N and Z from -dst
U 196F, C019,4020,4580,F804,4050,0062   ;34365          CLR.IB.OPC,PC_PC+1,J/IRD       ;  Complete
                                        ;34366
                                        ;34367          ;------------- -------------;
                                        ;34368  =0011   ;Branch on ALU z and n-bits
                                        ;34369          ;0011--------------------------;
                                        ;34370  CMPG2:  ALU_Q.AND.K[.FFF0],            ;  Dst exp is < src exp
                                        ;34371          N&Z_ALU.V&C_0, WORD,           ;  Set N and Z from src
U 1923, C019,6034,6D80,F804,4050,0062   ;34372          CLR.IB.OPC, PC_PC+1, J/IRD     ;  COMPLETE
                                        ;34373
                                        ;34374          ;0111--------------------------;
                                        ;34375          R[R15]_D,                      ;  Save dst exponent in R15
                                        ;34376          Q_ID[T2],                      ;  Same exponents, get OP1 LONG FRAC
U 1927, 0001,1B3C,C9F0,2EF8,0000,197A   ;34377          ALU?, J/CMPG3                  ;  Test high fraction part
                                        ;34378
                                        ;34379          ;1011--------------------------;
                                        ;34380          ALU_D.XOR.K[.8000],            ;
                                        ;34381          N&Z_ALU.V&C_0,WORD,            ;  Set cc from -dst
U 192B, C019,4020,4580,F804,4050,0062   ;34382          CLR.IB.OPC, PC_PC+1, J/IRD     ;  COMPLETE
                                        ;34383  =;END   ;------------------------------;
```

```
                                        ;34384  ;       Q = OP1 LONG FRAC
                                        ;34385  ;       R15 = OP2 LONG EXP
                                        ;34386  ;       RC0 = OP1 LONG EXP
                                        ;34387  ;       T1 = OP2 LONG FRAC
                                        ;34388  ;       T2 = OP1 LONG FRAC
                                        ;34389
                                        ;34390  =1010   ;Branch on ALU Z and C31-bits
                                        ;34391          ;1010-------------------------;  BORROW
                                        ;34392  CMPG3:  ALU_R[R15].XOR.K[.8000],      ;  Set CC on -dst
                                        ;34393          N&Z_ALU.V&C_0, WORD,
U 197A, C018,4020,4580,FA7C,4050,0062   ;34394          CLR.IB.OPC, PC_PC+1, J/IRD
                                        ;34395
                                        ;34396          ;1011-------------------------;  NO BORROW, NON-ZERO
                                        ;34397          ALU_RC[T0], N&Z_ALU.V&C_0, WORD,;  Set CC on src
U 197B, C010,4038,0180,F904,4050,0062   ;34398          CLR.IB.OPC, PC_PC+1, J/IRD
                                        ;34399
                                        ;34400          ;1111-------------------------;  NO BORROW, ZERO
                                        ;34401  =1111   ALU_Q.0XT[WORD],              ;  ISOLATE BITS <47:32> OF DST
                                        ;34402          D_ALU,                        ;  SAVE IT IN D
U 197F, 0803,603C,C5F0,2C00,0000,1AEA   ;34403          Q_ID[T1]                      ;  GET OP2 LONG FRAC
                                        ;34404  =;END
                                        ;34405          ;-----------------------------;
                                        ;34406          ALU_Q.0XT[WORD]-D,            ;  CLOCK DIFFERENCE IN BITS <47:32>
                                        ;34407          CLK.UBCC, WORD,               ;  DST-SRC
U 1AEA, 0C1F,6000,C9F0,2C00,0010,1AEE   ;34408          D_Q,Q_ID[T2]                  ;  D GETS DST, Q GETS SRC
                                        ;34409
                                        ;34410          ;-----------------------------;
U 1AEE, 001D,1B00,0180,F800,0010,198A   ;34411  CMPG4:  ALU_D-Q,CLK.UBCC,ALU?         ;  Clock difference of high fract bits
                                        ;34412                                        ;  CLOCK DST - SRC
                                        ;34413          ;-----------------------------;
                                        ;34414  =1010   ;Branch on ALU Z and C31 bits
                                        ;34415          ;1010-------------------------;  NO BORROW, SRC > DST
                                        ;34416          ALU_RC[T0], WORD,             ;  Set CC on src
                                        ;34417          N&Z_ALU.V&C_0, WORD,          ;  Set N and Z from src
U 198A, C010,4038,0180,F904,4050,0062   ;34418          CLR.IB.OPC, PC_PC+1, J/IRD
                                        ;34419
                                        ;34420          ;1011-------------------------;  BORROW, DST > SRC
                                        ;34421          ALU_R[R15].XOR.K[.8000],      ;  set CC on -dst
                                        ;34422          N&Z_ALU.V&C_0, WORD,
U 198B, C018,4020,4580,FA7C,4050,0062   ;34423          CLR.IB.OPC, PC_PC+1, J/IRD
                                        ;34424
                                        ;34425          ;1111-------------------------;
U 198F, 0000,1B3C,0180,F800,0000,199A   ;34426  =1111   ALU?, J/CMPG5                 ;  TEST DST - SRC <31:0>
                                        ;34427  =;END
                                        ;34428          ;-----------------------------;
                                        ;34429  =1010   ;Branch on ALU Z and C31 bits
                                        ;34430          ;1010-------------------------;  BORROW, SRC > DST
                                        ;34431  CMPG5:  ALU_RC[T0], WORD,
                                        ;34432          N&Z_ALU.V&C_0,
U 199A, C010,4038,0180,F904,4050,0062   ;34433          CLR.IB.OPC, PC_PC+1, J/IRD
                                        ;34434
                                        ;34435          ;1011-------------------------;  NO BORROW, DST > SRC
                                        ;34436          ALU_R[R15].XOR.K[.8000], WORD,;  Set CC on -dst
                                        ;34437          N&Z_ALU.V&C_0,
U 199B, C018,4020,4580,FA7C,4050,0062   ;34438          CLR.IB.OPC, PC_PC+1, J/IRD
```

```
                                        ;34439
                                        ;34440          ;1111-------------------------;
                                        ;34441  =1111   ALU_D-Q, N&Z_ALU.V&C_0,         ;  EQUALITY
U 199F, C01D,0000,0180,F804,4050,0062   ;34442          CLR.IB.OPC, PC_PC+1, J/IRD
                                        ;34443  =;END   ;-------------------------------;
```

```
;34444  .TOC     ""      G & H floating point  : G FORMAT UNPACK ROUTINE"
;34445
;34446  ;ROUTINE TO UNPACK MODIFIED DOUBL FLOATING OPERANDS.
;34447
;34448  ;INPUTS:
;34449  ;               Q = OP1 LONG FRAC
;34450  ;               RC0 = OP1 LONG EXP
;34451  ;               D = OP1 LONG FRAC (WORD SWAPPED)
;34452  ;               RC2 = OP2 LONG EXP
;34453  ;               RC[T6] = OP2 LONG FRAC
;34454  ;               RC3 = OP1 LONG FRAC
;34455  ;               SC = -7
;34456  ;               IR<2:1> = TYPE OF OPERATION BEING PERFORMED:
;34457  ;               00 = ADD, 01 = SUB, 10 = MUL, 11 = DIV
;34458  ;               IN CASE OF EMODG, HIGH WORD OF RC4 EQUALS EXTENDER
;34459  ;               SS=0 EXCEPT FOR ACBG, WHEN IT IS 1
;34460
;34461
;34462  ;OUTPUTS:
;34463  ;               ID[T0] = OP1 FRAC <MS>
;34464  ;               Q = ID[T2] = OP1 FRAC <LS>
;34465  ;               ID[T1] = OP2 FRAC <MS>
;34466  ;               D = OP2 FRAC <LS>
;34467  ;               RC1 = EXP2-EXP1 IF DIVG
;34468  ;               RC[T5] = OP1 EXPONENT
;34469  ;               RC[T4] = OP2 EXPONENT
;34470  ;               SS : (SIGN OF OPERAND 1) .XOR. (SIGN OF OPERAND 2) .XOR. IR<1>
;34471  ;                       EXCEPT FOR ACBG WHEN SS IS COMPLEMENT OF ABOVE
;34472  ;               R15 = (EXPONENT OF OPERAND 2 ) - (EXP OF OPER 1)
;34473  ;               FE = NABS (LOW BITS OF EXPONENT DIFFERENCE)
;34474  ;                       IF IR<2> = 0 (ADD OR SUB)
;34475
;34476  ;TEMPORARIES:
;34477  ;               ID[T3] = OP1 LONG FRAC (SWAPPED)
;34478
;34479  ;RETURNS:
;34480  ;               RETURN @ 1 IF OPERAND 1 = 0
;34481  ;               RETURN @ 2 IF OPERAND 1 <> 0, OPERAND 2 = 0
;34482  ;               RETURN @ 3 IF BOTH OPERANDS ARE NON-ZERO
;34483
;34484
;34485
```

```
                                ;34486  UNPACKG:
                                ;34487          ;-------------------------------;
                                ;34488          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;34489          ID[T3]_D,Q_D,                   ;  SAVE OP1 LONG FRAC (SWAPPED) IN T3
                                ;34490          D_RC[T0],                       ;  D GETS OP1 LONG EXP
U 1AF2, 0810,00B8,CCE5,3D00,0000,1AF3   ;34491          SGN/SS.XOR.ALU                  ;  SS GETS SIGN (EXCEPT FOR ACBG)
                                ;34492
                                ;34493          ;-------------------------------;
                                ;34494          ALU_D.AND.K[.7FF0],             ;  ISOLATE EXPONENT
                                ;34495          R[R15]_ALU.RIGHT2,              ;  STORE IT IN R15
U 1AF3, 0099,0034,3980,FAF8,0010,1AF4   ;34496          CLK.UBCC                        ;  CLOCK ALU Z-BIT
                                ;34497
                                ;34498          ;-------------------------------;
                                ;34499          D_D.OR.K[.10],                  ;  SET HIDDEN BIT
U 1AF4, 0819,0030,6580,FA78,0000,1AF5   ;34500          LAB_R[R15]                      ;  LATCH EXPONENT
                                ;34501
                                ;34502          ;-------------------------------;
                                ;34503          ALU_D.ANDNOT.K[.20],            ;  CLEAR BIT 5
U 1AF5, 0919,0024,7580,F800,0000,1AF6   ;34504          D_ALU(FRAC)                     ;  UNPACK OP1
                                ;34505
                                ;34506          ;-------------------------------;
                                ;34507          D_DAL.SC,                       ;  SHIFT RIGHT 7 BITS (SC=-7)
                                ;34508          SD_SS,                          ;  SAVE SS IN SD
                                ;34509          SC_K[.A],                       ;  SC GETS 10.
                                ;34510          ALU_LA,R[R15]_ALU.RIGHT2,       ;  R15 GETS EXPONENT
U 1AF6, 0D80,173C,F584,FAF8,0084,792D   ;34511          STATE1?                         ;  IS THIS EMODG ?
                                ;34512
                                ;34513          ;-------------------------------;
                                ;34514  =1101   ;BRANCH ON STATE<1>
                                ;34515          ;1101-----------------------------;
                                ;34516          LAB_R[R[ ],                     ;  LATCH EXPONENT
                                ;34517          D_DAL.SC,                       ;  GENERATE SRC FRACTION <MS>
U 192D, 0D00,003C,0180,FA78,0000,1AFA   ;34518          J/UNPKG.1
                                ;34519
                                ;34520          ;1111-----------------------------;  EMODG, SHIFT IN EXTENSION
                                ;34521          LAB_R[R15],                     ;  LATCH EXPONENT
U 192F, 00U0,003C,0580,FA78,0084,9AF8   ;34522          SC_SC+K[.1]
                                ;34523
                                ;34524          ;-------------------------------;
U 1AF8, 0D00,003C,0180,F800,0000,1AF9   ;34525          D_DAL.SC
                                ;34526
                                ;34527          ;-------------------------------;
                                ;34528          ID[T0]_D,                       ;  SAVE M'PLIER FRAC <LS> IN T0
                                ;34529          D_Q,                            ;  D GETS SRC FRACTION <LS>
                                ;34530          Q_RC[T4],                       ;  Q GETS EXTENDER
U 1AF9, 0C10,0138,C1C0,3D20,0000,18BC   ;34531          Z?,J/UNPKG.2
                                ;34532
                                ;34533          ;-------------------------------;
                                ;34534  UNPKG.1:
                                ;34535          ID[T0]_D,                       ;  SAVE SRC FRACTION <MS> IN T0
                                ;34536          D_Q,                            ;  GET LOW FRACTION READY FOR SHIFT
                                ;34537          Q_0,                            ;  GET READY FOR SHIFT
U 1AFA, 0C00,013C,C1F8,3C00,0000,18BC   ;34538          Z?                              ;  TEST OP1 EXPONENT
```

```
                                ;34539          ;-------------------------------;
                                ;34540  =0      ;Branch on alu z-bit
                                ;34541          ;0-------------------------------;  SRC NE 0
                                ;34542  UNPKG.2:
                                ;34543          D_DAL.SC,                       ;  D GETS SRC FRACTION <LS>
                                ;34544          Q_RC[T2],                       ;  GET OPER 2 LONG EXP
                                ;34545          SGN/ADD.SUB,
                                ;34546          FE_K[.10],                      ;  GET READY TO SWAP OP2 LONG FRAC
U 18BC, 0D10,0038,65C6,F910,0104,7B00   ;34547  J/SRCL0
                                ;34548
                                ;34549          ;1-------------------------------;  SRC EQ 0
                                ;34550          ALU_RC[T0],                     ;  PASS OP1 LONG EXP THRU ALU
U 18BD, 0010,0038,0180,F900,0800,1AFB   ;34551  CHK.FLT.OPR                     ;  TEST FOR RESERVED OPERAND
                                ;34552
                                ;34553          ;-------------------------------;
U 1AFB, 0810,0038,0180,F910,0000,1AFC   ;34554  D_RC[T2]                         ;  D gets DST LONG EXP
                                ;34555
                                ;34556          ;-------------------------------;
                                ;34557          ALU_D.AND.K[.7FF0],
                                ;34558          Q_ALU.RIGHT2,
                                ;34559          CLK.UBCC,
U 1AFC, 0099,0034,39C0,F930,0010,1AFD   ;34560  LC_RC[T6]                        ;  GET DST <L>
                                ;34561
                                ;34562          ;-------------------------------;
                                ;34563          RC[T4]_Q.RIGHT2,                ;  RC4 GETS OP2 EXPONENT
U 1AFD, 0081,213C,0180,F9A0,0000,18D8   ;34564  Z?                              ;  TEST FOR OPER 2 = 0
                                ;34565
                                ;34566          ;-------------------------------;
                                ;34567  =0      ;Branch on alu z-bit
                                ;34568          ;0-------------------------------;  DST.NE.0, SRC = 0
                                ;34569          RC[T1]_LC,                      ;  RC1 GETS DST0 <L>
                                ;34570          Q_0,                            ;  CLEAR Q
U 18D8, 0010,003A,01F8,F988,0000,0001   ;34571  RETURN1                          ;  RETURN1 FOR SRC=0, DST.NE. 0
                                ;34572
                                ;34573          ;1-------------------------------;  SRC = 0, DST = 0
                                ;34574          ALU_D.AND.K[.8000],             ;  CHECK FOR -0
                                ;34575          CHK.FLT.OPR,                    ;  WHICH IS RESERVED OPERAND
U 18D9, 0F19,0034,45F8,F800,0800,1AFE   ;34576  D_0,Q_0                          ;  EVERYTHING IS 0
                                ;34577
                                ;34578          ;-------------------------------;
                                ;34579          RC[T1]_K[ZERO],                 ;  MAKE RESULT 0
U 1AFE, 0018,003A,1980,F988,0000,0001   ;34580  RETURNT
                                ;34581          ;-------------------------------;
                                ;34582
                                ;34583
```

```
                                ;34584  SRCL0:  Q_Q.AND.K[.7FF0].RIGHT2,        ;  ISOLATE OP2 EXPONENT BITS
                                ;34585          C[K.UBCC,                       ;  CLOCK OPER 2 EXP
U 1B00, 0099,2034,39C0,F928,0010,1B01   ;34586          LC_RC[T5]                       ;  LATCH OPER 1 EXPONENT
                                ;34587
                                ;34588          ;-------------------------------;
                                ;34589          ID[T2]_D,                       ;  SAVE OPER 1 FRACTION <LS>
                                ;34590          RC[T4]_Q.RIGHT2,Q_SHF,          ;  STORE OPER 2 EXP IN RC4 AND Q
U 1B01, 0081,213C,C9C0,3DA0,0000,18E4   ;34591          Z?                              ;  TEST OPER 2 EXPONENT
                                ;34592
                                ;34593          ;-------------------------------;
                                ;34594  =0       ;BRANCH ON ALU Z-BIT
                                ;34595          ;0------------------------------;  DST NE 0
                                ;34596          FE_K[ZERO],                     ;  FE GETS 16. FOR SWAPPING
                                ;34597          D_RC[T6],                       ;  D GETS OP2 LONG FRAC
U 18E4, 0810,0938,1980,F930,0104,7898   ;34598          IR2-1?,J/UNPG2                  ;  TEST OPCODE
                                ;34599
                                ;34600          ;1------------------------------;  OPER 2 = 0, OPER 1 NE 0
                                ;34601          ALU_RC[T2],                     ;  NEED TO CHECK OP2 FOR -0
U 18E5, 0010,0038,0180,F910,0800,1B02   ;34602          CHK.FLT.OPR
                                ;34603
                                ;34604          ;-------------------------------;
U 1B02, 0810,0038,0180,F900,0000,1B03   ;34605          D_RC[T0]                        ;  D GETS OP1 LONG EXP
                                ;34606
                                ;34607          ;-------------------------------;
                                ;34608          LC_RC[T3],                      ;  LC GETS OP1 LONG FRAC
U 1B03, 0000,003E,0180,F918,0000,0002   ;34609          RETURN2
                                ;34610
                                ;34611          ;-------------------------------;
                                ;34612  =00      ;Branch on low two bits of opcode
                                ;34613  UNPG2:  ;00------------------------------;  ADDG
                                ;34614          ALU_Q-LB,R[R15]_ALU,            ;  R15 GETS EXPONENT DIFFERENCE
                                ;34615          SC_ALU,CLK.UBCC,                ;  CLOCK DIFFERENCE
U 1898, 000D,2000,0180,FAF8,0092,1B04   ;34616          J/GH.DSTTST
                                ;34617
                                ;34618          ;01------------------------------;  SUBG
                                ;34619          ALU_Q-LB,R[R15]_ALU,            ;  R15 GETS EXPONENT DIFFERENCE
                                ;34620          SC_ALU,CLK.UBCC,                ;  CLOCK DIFFERENCE
U 1899, 000D,2000,0180,FAF8,0092,1B04   ;34621          J/GH.DSTTST
                                ;34622
                                ;34623          ;10------------------------------;  MULG
                                ;34624          ALU_Q+LB,RC[T4]_ALU,            ;  RC5 GETS EXPONENT SUM
                                ;34625          SC_ALU,CLK.UBCC,                ;  CLOCK SUM
U 189A, 000D,2014,0180,F9A0,0092,1B04   ;34626          J/GH.DSTTST
                                ;34627
                                ;34628          ;11------------------------------;  DIVG & ACBG
                                ;34629          ALU_Q-LB,R[R15]_ALU,            ;  R15 GETS EXPONENT DIFFERENCE
                                ;34630          SC_ALU,CLK.UBCC,                ;  CLOCK DIFFERENCE
U 189B, 000D,2000,0180,FAF8,0092,1B04   ;34631          J/GH.DSTTST
                                ;34632          ;-------------------------------;
```

```
                                ;34633  GH.DSTTST:
                                ;34634          FE_NABS(SC-FE),                 ;  GET NEGATIVE EXPONENT DIFFERENCE
                                ;34635          ALU_K[.10],SC_ALU,              ;  SC GETS 16.
U 1B04, 0018,0038,65E0,F800,0182,FB05   ;34636          Q_D                             ;  GET OP2 LONG FRAC
                                ;34637
                                ;34638          ;-------------------------------;
                                ;34639          D_DAL.SC,                       ;  SWAP WORD DST LONG FRAC
                                ;34640          Q_RC[T2],                       ;  GET DST LONG EXP
U 1B05, 0D10,0038,BDC0,F910,0084,7B06   ;34641          SC_K[.FFF9]                     ;  SC GETS -7
                                ;34642
                                ;34643          ;-------------------------------;
U 1B06, 0019,2030,65C0,F800,0000,1B07   ;34644          Q_Q.OR.K[.10]                   ;  SET HIDDEN BIT
                                ;34645
                                ;34646          ;-------------------------------;
                                ;34647          ALU_Q.ANDNOT.K[.20],            ;  CLEAR NEXT BIT
                                ;34648          Q_D,                            ;  Q GETS LONG FRAC (SWAPPED)
U 1B07, 0919,2024,75E0,F800,0000,1B08   ;34649          D_ALU(FRAC)                     ;  UNPACK DST LONG EXP
                                ;34650
                                ;34651          ;-------------------------------;
                                ;34652          ALU_LA,RC[T5]_ALU,              ;  RC[T5] GETS OP2 EXPONENT
                                ;34653          D_DAL.SC,                       ;  SHIFT DST FRAC RIGHT 7 PLACES
U 1B08, 0D00,003C,F580,F9A8,0084,7B09   ;34654          SC_K[.A]                        ;  SHIFT COUNT OF 10.
                                ;34655
                                ;34656          ;-------------------------------;
U 1B09, 0D00,003C,0180,F800,0000,1B0A   ;34657          D_DAL.SC                        ;  GET DST FRAC <MS>
                                ;34658
                                ;34659          ;-------------------------------;
                                ;34660          LAB_R[R15],                     ;  LATCH EXPONENT DIFFERENCE
                                ;34661          ID[T1]_D,                       ;  SAVE DST FRAC <MS> IN T1
U 1B0A, 0C00,003C,C5F8,3E78,0000,1B0B   ;34662          D_Q, Q_0                        ;  SET UP FRAC <LS>
                                ;34663
                                ;34664          ;-------------------------------;
                                ;34665          D_DAL.SC,                       ;  D GETS DST FRAC <LS>
                                ;34666          Q_ID[T2],                       ;  Q GETS SRC FRAC <LS>
                                ;34667          ALU_RC[T4],SC_ALU,              ;  SC GET DST EXPONENT
U 1B0B, 0D10,003A,C9F0,2D20,0082,0003   ;34668          RETURN3                         ;  RETURN ADDRESS 3 FOR DST AND SRC NE 0
                                ;34669          ;-------------------------------;
```

```
                                ;34670  .TOC    ""      G & H floating point  : GRAND FLOATING PACK ROUTINE''
                                ;34671
                                ;34672  ;ROUTINE TO PACK GRAND FLOATING POINT RESULT.
                                ;34673
                                ;34674  ;ENTRY: AT 'PACKG''
                                ;34675
                                ;34676  ;INPUTS:
                                ;34677  ;               D=NORMALIZED FRACTION <MS>, UNROUNDED
                                ;34678  ;               Q=NORMALIZED FRACTION <LS>, ROUNDED
                                ;34679  ;               C31=REFLECTS RESULT OF ROUNDING
                                ;34680  ;               R15=BIASED EXPONENT
                                ;34681  ;               SD=SIGN
                                ;34682  ;               STATE<0>=1 IF CALLED FROM POLYG
                                ;34683  ;               STATE<0>=0 IF NOT CALLED FROM POLYG
                                ;34684  ;               FE=
                                ;34685  ;               SC=.FFF5
                                ;34686
                                ;34687  ;OUTPUTS:
                                ;34688  ;               D=RESULT LONG EXP
                                ;34689  ;               Q=R[T1]=RESULT LONG FRAC
                                ;34690  ;               CONDITION CODES SET ON RESULT
                                ;34691  ;               TRAP CODES SET IN ID[CES] IF UNDERFLOW OR OVERFLOW OCCURS
                                ;34692
                                ;34693  ;TEMPORARIES:
                                ;34694  ;               T9 = RESULT LONG FRAC
                                ;34695
                                ;34696          ;-------------------------------;
                                ;34697  =0*      ;BRANCH ON ALU C31
                                ;34698          ;0*-----------------------------;
                                ;34699  PACKG:  TRAP.ACC[1],                    ;  RESET THE FPA
                                ;34700          SC_SC.ANDNOT.K[.FFE0],          ;  GET SHIFT COUNT OF 21. (=.15)
                                ;34701          LAB_R[R15],                     ;  LOAD LATCHES WITH EXPONENT
U 1889, 0000,03BC,A080,FA78,0084,58B0   ;34702          C31?,J/PACKG.0                  ;  TEST CARRY FROM ROUNDING
                                ;34703
                                ;34704          ;1*-----------------------------;
U 188B, 0018,0014,0580,FAF8,0000,1889   ;34705          R[R15]_LA+K[.1], J/PACKG        ;  INCREMENT EXPONENT
                                ;34706
                                ;34707          ;-------------------------------;
                                ;34708  =0*      ;BRANCH ON ALU C31
                                ;34709          ;0*-----------------------------;
                                ;34710  PACKG.0:                                ;  NO CARRY
                                ;34711          Q_D,                            ;  SAVE FRAC <MS>
                                ;34712          D_DAL.SC,                       ;  D GETS FRAC <LS>
                                ;34713          A[U_LA+LB,R[R15]_ALU.LEFT3,     ;  SHIFT EXPONENT LEFT 4 PLACES
                                ;34714          SC_R[.FFFC], CLK.UBCC,          ;  SC GETS -4
U 18B0, 0DAC,0014,F1E0,FAF8,0094,7B0D   ;34715          J/PACKG.2
                                ;34716
                                ;34717          ;1*-----------------------------;
U 18B2, 0819,0014,0580,F800,0010,1B0C   ;34718          D_D+K[.1], CLK.UBCC             ;  INCREMENT FRAC <MS>
                                ;34719
                                ;34720          ;-------------------------------;
                                ;34721          ALU_0+K[.1], CLK.UBCC,          ;  CLEAR CARRY
U 1B0C, 001B,0314,0580,F800,0010,1889   ;34722          C31?, J/PACKG                   ;  DID ROUNDING RESULT IN CARRY?
                                ;34723          ;-------------------------------;
```

```
                                    ;34724  PACKG.2:
                                    ;34725          SC_K[.10],                      ;  16. FOR SWAPPING FRAC <LS>
                                    ;34726          RC[T1]_Q.RIGHT2,                ;  START SHIFTING FRAC <MS> RIGHT
U 1B0D, 0081,373C,65E0,F988,0084,79AB ;34727        Q_D,STATE2?                     ;  DUPLICATE FRAC <LS>
                                    ;34728
                                    ;34729  =1011   ;1011----------------------------;  NOT ACBG
U 19AB, 0000,003C,0180,F800,0050,19AF ;34730        N&Z_ALU.V&C_0                   ;  CLEAR V&C
                                    ;34731
                                    ;34732          ;1111----------------------------;  ACBG,LEAVE V&C ALONE
                                    ;34733          D_DAL.SC,                       ;  GET LONG FRAC IN ORDER
                                    ;34734          ALU_R[R15], CLK.USCC, WORD,     ;  CLOCK EXPONENT
                                    ;34735          Q_ALU,
U 19AF, 0D00,573C,01C0,FA78,0010,19BE ;34736        STATE0?                         ;  IS THIS POLYG?
                                    ;34737
                                    ;34738          ;----------------------------------;
                                    ;34739  =1110   ;BRANCH ON STATE<0>
                                    ;34740          ;1110----------------------------;
                                    ;34741          ID[T9]_D,                       ;  SAVE LONG FRAC IN T9
                                    ;34742          ALU_RC[T1],D_ALU.RIGHT,         ;  GET FRAC <MS>
U 19BE, 0850,1B38,E580,3D08,0000,19C3 ;34743        ALU?,J/PACKG.3                  ;  TEST FOR OVER/UNDERFLOW
                                    ;34744
                                    ;34745          ;1111----------------------------;
                                    ;34746          ID[T9]_D,                       ;  SAVE LONG FRAC IN T9
U 19BF, 0850,0038,E580,3D08,0000,19C3 ;34747        ALU_RC[T1],D_ALU.RIGHT          ;  GET FRAC <MS>
                                    ;34748
                                    ;34749          ;----------------------------------;
                                    ;34750  =0011   ;BRANCH ON ALU Z AND N-BITS
                                    ;34751          ;0011----------------------------;
                                    ;34752  PACKG.3:
                                    ;34753          D_PACK.FP,                      ;  PACK D IN FLOATING FORMAT
U 19C3, 0808,0038,0190,F800,0000,1B11 ;34754        QR/RIGHT2,J/PACKG.4
                                    ;34755
                                    ;34756          ;0111----------------------------;
                                    ;34757          D_0,Q_0,EALU_K[ZERO],           ;  FLOATING UNDERFLOW
                                    ;34758          RC[T1]_K[ZERO],                 ;  SET RESULT TO 0
                                    ;34759          N&Z_ALU,                        ;  SET Z, CLEAR N
U 19C7, 0F18,0038,19F8,F988,0064,7867 ;34760        J/PACKG.9
                                    ;34761
                                    ;34762          ;1011----------------------------;
                                    ;34763          D_0,QK/LEFT,                    ;  FLOATING OVERFLOW
                                    ;34764          RC[T1]_K[ZERO],                 ;  SET RESULT TO 0
U 19CB, 0F18,0038,19A8,F988,0060,1B10 ;34765        N&Z_ALU                 ;  SET Z, CLEAR N
                                    ;34766  =;END
                                    ;34767          ;----------------------------------;
U 1B10, 0000,0D3C,01F8,F800,0000,1863 ;34768        Q_0,Q31?,J/PACKG.7              ;  TEST FOR NEGATIVE EXPONENT
                                    ;34769          ;----------------------------------;
```

```
                                ;34770
                                ;34771  PACKG.4:
                                ;34772          D_D.ANDNOT.K[.7FF0],                    ;  CLEAR OUT EXPONENT BITS
U 1B11, 0819,1724,3990,F800,0000,19CE   ;34773          QR/RIGHT2,STATE0?                       ;  KEEP SHIFTING EXPONENT RIGHT
                                ;34774          ;---------------------------------;
                                ;34775  =1110   ;BRANCH ON STATE0
                                ;34776          ;1110-----------------------------;
                                ;34777          D_D.OR.R[R15],                          ;  ASSEMBLE DATA
                                ;34778          N&Z_ALU, WORD,                          ;  SET CC
                                ;34779          Q_ID[T9],                               ;  GET BACK LONG FRAC
U 19CE, 080D,4030,E5F0,2E78,0060,1B14   ;34780          J/PACKG.40
                                ;34781
                                ;34782          ;1111-----------------------------;
                                ;34783          D_D.OR.R[R15],
U 19CF, 080D,4030,0180,FA78,0060,1B12   ;34784          N&Z_ALU, WORD
                                ;34785          ;---------------------------------;
U 1B12, 0001,203C,0180,F988,0000,1B13   ;34786          RC[T1]_Q
                                ;34787
                                ;34788          ;---------------------------------;
                                ;34789          Q_R[R15],ALU?,                          ;  TEST EXPONENT,WORD
U 1B13, 0000,1B3E,01C0,FA78,0000,0010   ;34790          RETURN[10]
                                ;34791          ;---------------------------------;
                                ;34792  PACKG.40:
U 1B14, 0001,203E,0180,F988,0000,0010   ;34793          RC[T1]_Q, RETURN[10]                    ;  FINISHED
                                ;34794
                                ;34795          ;---------------------------------;
                                ;34796  =011    ;BRANCH ON Q31
                                ;34797          ;011------------------------------;
                                ;34798  PACKG.7:
                                ;34799          RC[T7]_K[.8],                           ;  OVERFLOW CODE
                                ;34800          N&Z_ALU,WORD,
U 1863, 0018,4038,0180,F9B8,0060,12B4   ;34801          J/FLOAT.FAULT                           ;
                                ;34802
                                ;34803          ;111------------------------------;     UNDERFLOW
U 1867, 0000,003C,3DF0,2C00,0000,19D5   ;34804  PACKG.9:Q_ID[PSL]                        ;
                                ;34805
                                ;34806          ;0101-----------------------------;
U 19D5, 0000,003D,0180,F800,0000,12AE   ;34807  =0101   CALL[PSL.FU.A]                          ;  TEST FU
                                ;34808
                                ;34809  =1101   ;1101-----------------------------;     FU NOT SET IGNORE UNDERFLOW
U 19DD, 0018,003A,19F8,F988,0000,0010   ;34810          RC[T1]_K[ZERO],Q_0,RETURN[10]           ;
                                ;34811
                                ;34812          ;1111-----------------------------;     FU SET, TAKE UNDERFLOW
U 19DF, 0018,0038,F580,F9B8,0000,12B4   ;34813          RC[T7]_K[.A],J/FLOAT.FAULT              ;
```

```
                                ;34814
                                ;34815  ;G-FORMAT FLOATING ADD AND SUBTRACT, *-R.
                                ;34816  ;ENTER WITH SRC OPND IN RC0,D, AND DST OPND IN LA.
                                ;34817  142F:
                                ;34818          ;-------------------------------;
                                ;34819          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;34820          STATE_K[ZERO],                  ;  CLEAR POLY FLAG
U 142F, 0000,00BC,1880,F990,1404,7B15   ;34821  RC[T2]_LA                       ;  RC2 GETS OP2 LONG EXP
                                ;34822          ;-------------------------------;
                                ;34823          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;34824          Q_D,                            ;  Q GETS SRC LONG FRAC
U 1B15, 0800,00BC,00E0,F860,0000,18E1   ;34825  D_R(PRN+1)                      ;  D GETS DST LONG FRAC
                                ;34826          ;0****--------------------------;
                                ;34827  =0****
                                ;34828          RC[T6]_D,                       ;  SAVE DST LONG FRAC
                                ;34829          D_Q,                            ;  D GETS OP1 LONG FRAC
                                ;34830          SC_K[.10],                      ;  SC GETS 16. FOR WORD SWAP
U 18E1, 0C01,003D,6580,F9B0,0084,78A8   ;34831  CA[L,J/ADDG.S                    ;  CALL ADDG SUBROUTINE
                                ;34832          ;1****--------------------------;
                                ;34833  ;RETURN WITH RESULT IN <D,RC1>
                                ;34834          TRAP.ACC[1],                    ;  RESET THE FPA
U 18F1, 0001,00BC,0080,F8D8,0000,0796   ;34835  R(PRN)_D, J/ADDD.P               ;  USE DOUBLE ROUTINE
                                ;34836  =;END   ;-------------------------------;
                                ;34837
                                ;34838  ;G-FORMAT FLOATING ADD AND SUBTRACT *-*-*, AND *-MEMORY.
                                ;34839  ;ENTER WITH SRC OPND IN <RC0, Q>, DST OPND INN <RC2, D>
                                ;34840  ;YIELDS NORMALIZED AND ROUNDED RESULTS.
                                ;34841
                                ;34842  168E:                                   ;  1600+8E
                                ;34843  ADDG:   ;-------------------------------;
                                ;34844          TRAP.ACC[1],                    ;  RESET THE FPA
U 168E, 0000,00BC,1880,F800,1404,78EC   ;34845  STATE_K[ZERO]                    ;  CLEAR POLY/EMOD FLAG
                                ;34846          ;0****--------------------------;
                                ;34847  =0****
                                ;34848          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;34849          RC[T6]_D,                       ;  SAVE OP2 LONG FRAC IN RC6
                                ;34850          D_Q,                            ;  D GETS OP1 LONG FRAC
                                ;34851          SC_K[.10],                      ;  SC GETS 16. FOR SWAP WORD OF FRAC <L>
U 18EC, 0C01,00BD,6480,F9B0,0084.78A8   ;34852  CA[L, J/ADDG.S                   ;  CALL ADDG/SUBG SUBROUTINE
                                ;34853
                                ;34854          ;1****--------------------------;
                                ;34855          STATE_K[.1],
U 18FC, F000,003F,05F0,F847,1404,7400   ;34856  WRITE.G.DEST, J/WRG              ;  WRITE RESULT
                                ;34857  =;END   ;-------------------------------;
```

```
;34858  ;       COMMON GRAND FLOATING ADD/SUB ROUTINE
;34859
;34860  ;INPUTS:
;34861  ;               RC[T0] = OP1 LONG EXP
;34862  ;               D = Q = OP1 LONG FRAC
;34863  ;               RC[T2] = OP2 LONG EXP
;34864  ;               RC[T6] = OP2 LONG FRAC
;34865  ;               SC = .10 (HEX)
;34866  ;               STATE = 0
;34867  ;               IR<1> = 0 IF ADD, 1 IF SUB
;34868
;34869  ;OUTPUTS:
;34870  ;               D= PACKED ROUNDED SUM LONG EXP
;34871  ;               RC[T1] = PACKED ROUNDED SUM LONG FRAC
;34872  ;               CONDITION CODES AND ID[CES] SET ON SUM
;34873
;34874  ;TEMPORARIES:
;34875  ;               RC3 = OP1 LONG FRAC
;34876  ;               T0 = OP1 FRAC <MS>
;34877  ;               T1  = OP2 FRAC <MS>
;34878  ;               T2 = OP1 FRAC <LS>
;34879  ;               R15 = EXPONENT DIFFERENCE
;34880  ;
;34881
;34882  ;RETURNS:
;34883  ;               RETURNS AT 10
;34884
;34885
```

```
                                ;34886  =00
                                ;34887  ADDG.S:
                                ;34888          ;00------------------------------;
                                ;34889          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;34890          SGN/CLR.SD+SS,                  ;  CLEAR OUT SS
                                ;34891          RC[T3]_D,                       ;  SAVE OP1 LONG FRAC IN RC3
                                ;34892          D_DAL.SC,                       ;  SWAP WORDS OF OP1 LONG FRAC
                                ;34893          SC_K[.FFF9],                    ;  SET UP SHIFT AMOUNT OF -7
U 18A8, 0D01,00BD,BC87,F998,0084,7AF2   ;34894          CALL, J/UNPACKG                 ;  UNPACK MODIFIED DOUBLE
                                ;34895
                                ;34896          ;01------------------------------;  RETURN1, SRC=0, DST MAY BE 0
                                ;34897          ALU_D.AND.K[.FFF0],             ;  CLOCK CONDITON CODES ON EXP &SIGN
                                ;34898          N&Z_ALU.V&C_0,WORD,             ;  NO OVERFLOW POSSIBLE
U 18A9, 0019,4036,6D80,F800,0050,0010   ;34899          RETURN10
                                ;34900
                                ;34901          ;10------------------------------;  RETURN2, DST = 0, SRC NE 0
                                ;34902          RC[T1]_LC,                      ;  RC1 GETS SRC0 <L>
U 18AA, 0010,0938,0180,F988,0000,1822   ;34903          IR1?, J/ADDG.8                  ;  TEST FOR ADD OR SUB
                                ;34904
                                ;34905          ;11------------------------------;  RETURN3,  SRC NE0, DST NE 0
                                ;34906  ;       T0 = OP1 FRAC <MS>
                                ;34907  ;       T2 = Q = OP1 FRAC <LS>
                                ;34908  ;       T1 = OP2 FRAC <MS>
                                ;34909  ;       D = OP2 FRAC <LS>
                                ;34910  ;       RC4 = OP2 EXP
                                ;34911  ;       RC5 = OP1 EXP
                                ;34912  ;       R15 = OP2 EXP - OP1 EXP
                                ;34913  ;       FE=NABS(LOW 8 BITS OF EXPONENT DIFFERENCE)
                                ;34914  ;       ALU CC REFLECT R15
                                ;34915  ;       RC0 AND RC1 ARE UNCHANGED
                                ;34916  ;       LAB = R[R15] = EXPONENT DIFFERENCE
                                ;34917
                                ;34918  ADDG.6:
                                ;34919          ;RETURN HERE WITH BOTH OPERANDS NON-ZERO.
                                ;34920          ;ALU CC REFLECT EXP2-EXP1
                                ;34921
                                ;34922          LC_RC[T4],                      ;  LATCH UP OP1 EXP
U 18AB, 0000,1B3C,0180,F920,0000,19E3   ;34923          ALU?, J/ADDG.10                 ;  TEST ALU N,Z (FROM EXP2-EXP1)
                                ;34924
                                ;34925          ;------------------------------;
                                ;34926  =10      ;BRANCH ON IR1
                                ;34927          ;10------------------------------;
                                ;34928  ADDG.8:                                 ;  CALLED BY ADDH AS WELL
                                ;34929          ALU_D,RC[T0]_ALU,N&Z_ALU.V&C_0, ;  RC[T0] GETS LONG EXP
U 1822, 0001,403E,0180,F980,0050,0010   ;34930          WORD, RETURNT0                  ;  SET CC ON SRC
                                ;34931
                                ;34932          ;11------------------------------;
                                ;34933          D_D.XOR.K[.8000],RC[T0]_ALU,    ;  RESULT IS -SRC
                                ;34934          N&Z_ALU.V&C_0,WORD,             ;  SET CC
U 1823, 0819,4022,4580,F980,0050,0010   ;34935          RETURN10
                                ;34936          ;------------------------------;
```

```
                                ;34937  ;       MAIN BRANCHING POINT OF ADDG - ALSO USED BY POLYG AND ACBG.
                                ;34938  ;       AT THIS POINT, MACHINE STATE IS AS FOLLOWS:
                                ;34939  ;INPUTS:
                                ;34940  ;               D = OP2 FRAC <LS>
                                ;34941  ;               Q = ID[T2] = OP1 FRAC <LS>
                                ;34942  ;               ID[T0] = OP1 FRAC <MS>
                                ;34943  ;               ID[T1] = OP2 FRAC <MS>
                                ;34944  ;               SD = OP2 SIGN
                                ;34945  ;               LC = RC4 = OP2 EXP
                                ;34946  ;               RC6 = OP2 LONG FRAC
                                ;34947  ;               RC0 = OP1 LONG EXP
                                ;34948  ;               SS = OP2 SGN.XOR.OP1 SGN.XOR.SUBG
                                ;34949  ;               R15 = EXP2-EXP1
                                ;34950  ;               RC4 = EXP2
                                ;34951  ;               RC5 = EXP1
                                ;34952  ;               FE = NABS(ALIGNMENT SHIFT)
                                ;34953
                                ;34954  ;TEMPORARIES:
                                ;34955  ;               ID[T4] = OP2 FRAC <LS>
                                ;34956
                                ;34957  =0011
                                ;34958  ADDG.10:
                                ;34959          ;0011-------------------------;  DST EXP > SRC EXP
                                ;34960          RC[T5]_LC,                      ;  RC5 GETS BIGGEST EXPONENT
                                ;34961          ID[T4]_D,                       ;  SAVE DST FRAC <LS> IN T4
                                ;34962          D_Q,                            ;  D GETS OP1 FRAC <LS>
                                ;34963                                          ;  SETUP FOR ALIGN SRC FRACTION
                                ;34964          SC_FE,                          ;  GET SHIFT AMOUNT
                                ;34965          EALU?,                          ;  ADD OR SUBTRACT ?
U 19E3, 0C10,1238,D180,3DA8,0081,184E   ;34966          J/ADDG.100
                                ;34967
                                ;34968          ;0111-------------------------;  DST EXP = SRC EXP
                                ;34969          SC_K[ZERO],                     ;  CLEAR EXP-DIFFERENCE
                                ;34970          FE_K[ZERO],                     ;  DO IT THOROUGHLY
                                ;34971          LC_RC[T5],                      ;  LATCH UP EXPONENT
U 19E7, 0000,123C,1980,F928,0184,7A0E   ;34972          EALU?, J/ADDG.14                ;  ADD OR SUBTRACT ?
                                ;34973
                                ;34974          ;1011-------------------------;  DST EXP < SRC EXP
                                ;34975          SC_FE,                          ;  GET EXPONENT DIFFERENCE
                                ;34976          Q_ID[T1],                       ;  Q GETS DST FRAC <MS>
                                ;34977          R[R15]_0-LB,                    ;  NEGATE EXP2-EXP1
U 19EB, 000F,1200,C5F0,2EF8,0081,180E   ;34978          EALU?                           ;  ADD OR SUBTRACT ?
                                ;34979  =;END   ;-------------------------------;
```

```
                                ;34980  ;       ENTER HERE TO DO ADD OR SUBTRACT WHEN DST EXP < SRC EXP
                                ;34981  ; EXPECTS:
                                ;34982  ;       <D,Q> = DST FRACTION, WHICH NEEDS TO BE ALIGNED
                                ;34983  ;       RC[T5] = SRC EXP
                                ;34984  ;       SD = DST SIGN
                                ;34985  ;       SS = SRC SGN.XOR. DST SGN.XOR.IR<1>
                                ;34986
                                ;34987  =001110 ;BRANCH ON SS
                                ;34988          ;001110----------------------------;  DO AN ADDITION
                                ;34989          LC_RC[T5],                          ;  LATCH UP EXPONENT
U 180E, 0000,003D,0180,F928,0000,182B   ;34990          CALL,J/ALNPOSG                      ;  ALIGN DST FRACTION
                                ;34991
                                ;34992          ;001111----------------------------;  DO A SUBTRACTION
                                ;34993          SD_NOT.SD,                          ;  COMPLEMENT DST-SIGN
                                ;34994          LC_RC[T5],                          ;  LATCH UP EXPONENT
U 180F, 0000,003D,0183,F928,0000,1B38   ;34995          CALL, J/ALNNEGG
                                ;34996
                                ;34997  =011110 ;011110----------------------------;  EXP DIFF < 32, ADD
                                ;34998          RC[T2]_D,                           ;  SAVE ALIGNED DST FRAC <LS>
                                ;34999          D_Q,Q_0,                            ;  GET READY TO ALIGN DST <MS>
U 181E, 0C01,003C,01F8,F990,0000,1B22   ;35000          J7ADDG.24
                                ;35001
                                ;35002          ;011111----------------------------;  EXP DIFF < 32, SUB
                                ;35003          RC[T2]_D,                           ;  SAVE ALIGNED DST FRAC <LS>
U 181F, 0C01,003C,0180,F990,0000,1B21   ;35004          D_Q,J/ADDG.23
                                ;35005
                                ;35006  =101111 ;101111----------------------------;  31<EXP DIFF<64, ADD/SUB
                                ;35007          RC[T2]_D,                           ;  STORE ALIGNED DST FRAC <LS>
                                ;35008          D_Q,                                ;  D GETS 0 OR 1'S
                                ;35009          Q_ID[T2],                           ;  GET SRC FRAC <LS>
U 182F, 0C01,003C,C9F0,2D90,0000,1B24   ;35010          J7ADDG.25
                                ;35011
                                ;35012  =111111 ;111111----------------------------;  63< EXP DIFF, ADD/SUB
                                ;35013          Q_ID[T0],                           ;  Q GETS OP1 FRAC <MS>
                                ;35014          SC_FE,
U 183F, 0000,003C,C1F0,2D28,0081,1B16   ;35015          LC_RC[T5]
                                ;35016  =;END
                                ;35017          ;--------------------------------;
                                ;35018          D_Q,Q_ID[T2],                       ;  <D,Q> GETS FRACTION OF OP1
                                ;35019          SC_SC.ANDNOT.FE,                    ;  CLEAR SC
U 1B16, 0C00,173C,C9F0,2C00,0080,59EE   ;35020          STATE0?                             ;  IS THIS POLYG?
                                ;35021
                                ;35022  =1110   ;--------------------------------;  NO
U 19EE, 0000,003C,0180,F800,0000,1A1B   ;35023          J/ADDG.31                           ;
                                ;35024
                                ;35025          ;1111------------------------------;  YES
U 19EF, 0000,123C,0180,F800,0000,19FE   ;35026          SS?                                 ;  SIGNS DIFFERENT?
                                ;35027
                                ;35028  =1110   ;--------------------------------;  NO
U 19FE, 0000,003C,0180,F800,0000,1A1B   ;35029          J/ADDG.31                           ;
                                ;35030
                                ;35031          ;1111------------------------------;  YES
U 19FF, 0E00,003C,0180,F800,008C,9B17   ;35032          SC_SC+SHF.VAL,D_DAL.NORM            ;  D GETS NORMALIZED FRAC<MS>
```

```
                                ;35033
                                ;35034          ;-------------------------------;
U 1B17, 0CC1,003C,01F8,FAF8,0000,1B18   ;35035          R[R15]_D,D_Q,Q_0                ;  SAVE FRAC<MS>,SET UP NORM FRAC<LS>
                                ;35036
                                ;35037          ;-------------------------------;
U 1B18, 0D1B,0010,81C0,F800,0000,1B19   ;35038          D_DAL.SC.Q_K[.3FF]+1            ;  NORM FRAC<LS>,CREATE CONSTANT
                                ;35039
                                ;35040          ;-------------------------------;
U 1B19, 001D,2034,0180,F800,0010,1B1A   ;35041          ALU_Q.AND.0,CLK.UBCC            ;  SEE IF ROUND BIT IS SET
                                ;35042
                                ;35043          ;-------------------------------;
U 1B1A, 0000,013C,01C0,FA78,0000,18F8   ;35044          Q_R[R15],Z?                     ;
                                ;35045
                                ;35046  =0      ;0------------------------------;  ROUND BIT IS ONE
U 18F8, 0819,0000,0580,F800,0000,18F9   ;35047          D_D-K[.1]                       ;  DECREMENT GUARD BITS
                                ;35048
                                ;35049          ;1------------------------------;
U 18F9, 0C01,003C,01F8,FAF8,0000,1B31   ;35050          R[R15]_D,D_Q,Q_0,J/ADDG.37      ;
                                ;35051
                                ;35052          ;-------------------------------;
                                ;35053  =001110 ;BRANCH ON SS
                                ;35054          ;001110---------------------------;
                                ;35055  ADDG.100:                               ;  ADD
                                ;35056          Q_ID[T0],                       ;  GET SRC FRAC <MS>
U 184E, 0000,003D,C1F0,2C00,0000,182B   ;35057          CALL[ALNPOSG]
                                ;35058
                                ;35059          ;001111---------------------------;  SUBTRACT
                                ;35060          Q_ID[T0],                       ;  GET SRC FRAC <
U 184F, 0000,003D,C1F0,2C00,0000,1B38   ;35061          CALL[ALNNEGG]                   ;  NEGATE BEFOR' ALIGNING
                                ;35062
                                ;35063          ;011110---------------------------;
                                ;35064  ADDG.100.R10:
                                ;35065  ;       D = ALIGNED SRC FRAC <LS>
                                ;35066  ;       Q = SRC FRAC <MS>
                                ;35067
                                ;35068          ID[T2]_D,                       ;  ALIGNED SRC FRAC <LS>
                                ;35069          D_Q, Q_0,                       ;  .T READY TO ALIGN SRC FRAC <MS>
U 185E, 0C00,003C,C9F8,3C00,0000,1B1C   ;35070          J7ADDG.21
                                ;35071
                                ;35072          ;011111-------------------------;  SUBTRACT, DST EXP >SRC EXP, <32.
                                ;35073          ID[T2]_D,                       ;  SAVE ALIGNED SRC FRAC <LS> IN T2
                                ;35074          D_Q,                            ;  D GETS SRC FRAC <MS>
                                ;35075          ALU_-1, Q_ALU,                  ;  NEED TO SHIFT IN 1'S
U 185F, 0C03,0028,C9C0,3C00,0000,1B1C   ;35076          J/ADDG.21                       ;  GO TO ALIGN <MS>
                                ;35077
                                ;35078  =101111 ;101111---------------------------;  ADD/SUB, 31<DIFF EXP < 64
                                ;35079          ID[T2]_D,                       ;  SAVE ALIGNED FRAC <LS>
                                ;35080          D_Q,                            ;  D GETS SIGN EXTENSION
                                ;35081          Q_D,                            ;  Q GETS ALIGNED FRAC <LS>
U 186F, 0C00,003C,C9E0,3C00,0000,1B1E   ;35082          J7ADDG.22
                                ;35083
                                ;35084  =111111 ;111111---------------------------;  ADD/SUB, 63 < EXP DIFF
                                ;35085          Q_ID[T1],                       ;  Q GETS OP2 FRAC <MS>
U 187F, 0000,003C,C5F0,2C00,0081,1B1B   ;35086          SC_FE
                                ;35087  =;END
```

```
                                ;35088
                                ;35089          ;--------------------------------;
                                ;35090          D_Q,Q_ID[T4],                   ;  <D,Q> GETS FRACTION (=OP2)
                                ;35091          SC_SC.ANDNOT.FE,                ;   CLEAR SC
U 1B1B, 0C00,003C,D1F0,2C00,0080,5A1B   ;35092          J/ADDG.31                       ;
                                ;35093
                                ;35094  =1110   ;BRANCH ON SS-BIT
                                ;35095  ADDG.14:
                                ;35096          ;1110-----------------------------;  DST EXP=SRC EXP,ADD
                                ;35097          R[R15]_D+Q,                     ;  ADD FRAC <LS>'S
                                ;35098          CLK.UBCC,                       ;  CLOCK CARRY IF ANY
                                ;35099          Q_ID[T1],                       ;  Q GETS DST FRAC <MS>
U 1A0E, 001D,0014,C5F0,2EF8,0010,1B25   ;35100          J7ADDG.26
                                ;35101
                                ;35102  ADDG.16:;1111-----------------------------;  DST EXP= SRC EXP, SUBTRACT
                                ;35103          R[R15]_D-Q,                     ;  SUBTRACT FRACTIONS <LS>
                                ;35104          CLK.UBCC,                       ;  CLOCK BORROW IF ANY
                                ;35105          Q_ID[T1],                       ;  GET DST FRAC <MS>
U 1A0F, 001D,0000,C5F0,2EF8,0010,1B28   ;35106          J7ADDG.30
                                ;35107
                                ;35108          ;--------------------------------;
                                ;35109  ADDG.21:D_DAL.SC,                       ;  D GETS ALIGNED SRC FRAC <MS>
U 1B1C, 0D00,003C,C9F0,2C00,0000,1B1E   ;35110          Q_ID[T2]                        ;  Q GETS SRC FRAC <LS>
                                ;35111
                                ;35112  ADDG.22:;-------------------------------;  COME HERE IF SRC FRACTION IS ALIGNED
                                ;35113          ID[T0]_D,                       ;  SAVE ALIGNED SRC FRAC <MS> IN T0
U 1B1E, 0C00,003C,C180,3C00,0000,1B20   ;35114          D_Q                             ;  D GETS SRC FRAC <LS>
                                ;35115
                                ;35116          ;--------------------------------;
U 1B20, 0000,003C,D1F0,2C00,0000,1A0E   ;35117          Q_ID[T4],J/ADDG.14              ;  Q GETS DST FRAC <LS>
                                ;35118
                                ;35119  ADDG.23:;-------------------------------;
U 1B21, 0003,0028,01C0,F800,0000,1B22   ;35120          ALU_-1, Q_ALU
                                ;35121
                                ;35122  ADDG.24:;-------------------------------;
                                ;35123          D_DAL.SC,                       ;  ALIGN DST FRAC <MS>
U 1B22, 0D00,003C,C9F0,2C00,0000,1B24   ;35124          Q_ID[T2]                        ;  Q GETS SRC FRAC <LS>
                                ;35125
                                ;35126  ADDG.25:;-------------------------------;
                                ;35127          ID[T1]_D,                       ;  STORE AWAY DST FRAC <MS>
                                ;35128          D_RC[T2],                       ;  RETRIEVE DST FRAC <LS>
U 1B24, 0810,0038,C580,3D10,0000,1A0E   ;35129          J7ADDG.14                       ;  GO AND DO THE ADD
```

```
                                ;35130
                                ;35131  ADDG.26:;--------------------------------;
                                ;35132          D_Q,                            ;  D GETS ALIGNED DST FRAC <MS>
                                ;35133          Q_ID[TO],                       ;  Q GETS ALIGNED SRC FRAC <MS>
U 1B25, 0C00,033C,C1F0,2C00,0000,1901   ;35134          C31?                            ;  CARRY FROM LS ADD ?
                                ;35135
                                ;35136          ;--------------------------------;
                                ;35137  =0*     ;BRANCH ON ALU CARRY
                                ;35138          ;0*--------------------------------;
                                ;35139          D_D+Q,                          ;  ADD ALIGNED FRAC <MS>'S
                                ;35140          LAB_R[R15], CLK.UBCC,           ;  LATCH RESULT FRAC <LS>,
U 1901, 081D,0014,0180,FA78,0010,1B26   ;35141          J/ADDG.27                       ;  SET Z ON FRAC <MS>
                                ;35142
                                ;35143          ;1*--------------------------------;
                                ;35144          D_D+Q+1,                        ;  ADD ALIGNED FRAC <MS>'S +1
U 1903, 081D,0010,0180,FA78,0010,1B26   ;35145          LAB_R[R15], CLK.UBCC            ;  LATCH RESULT FRAC <LS>
                                ;35146  =;END
                                ;35147          ;--------------------------------;
                                ;35148  ADDG.27:
                                ;35149          SC_K[ZERO],                     ;  INITIALIZE SHIFT VALUE
                                ;35150          Q_R[R15],                       ;  Q GETS RESULT FRAC <LS>
U 1B26, 0000,1B3C,19C0,FA78,0084,7A1B   ;35151          ALU?, J/ADDG.31                 ;  GO TO NORMALIZE
                                ;35152
                                ;35153          ;--------------------------------;
                                ;35154  ADDG.30:
                                ;35155          D_Q,                            ;  D GETS DST FRAC <MS>
                                ;35156          Q_ID[TO],                       ;  Q GETS SRC FRAC <MS>
U 1B28, 0C00,033C,C1F0,2C00,0000,1928   ;35157          C31?                            ;  WAS THERE A BORROW?
                                ;35158
                                ;35159          ;--------------------------------;
                                ;35160  =0*     ;BRANCH ON ALU C31
                                ;35161          ;0*--------------------------------;
                                ;35162          D_D-Q-1,                        ;  SUBTRAC FRAC'S <MS> WITH BORROW
                                ;35163          CLK.UBCC,                       ;  CLOCK ALU N-BIT
                                ;35164          LAB_R[R15],                     ;  LATCH RESULT FRAC <LS>
U 1928, 081D,0008,0180,FA78,0010,1B29   ;35165          J/ADDG.32
                                ;35166
                                ;35167          ;1*--------------------------------;
                                ;35168          D_D-Q,                          ;  NO BORROW, SUBTRACT FRAC <MS>
                                ;35169          CLK.UBCC,                       ;  CLOCK ALU N-BIT
                                ;35170          LAB_R[R15],                     ;  LATCH RESULT FRAC <LS>
U 192A, 081D,0000,0180,FA78,0010,1B29   ;35171          J/ADDG.32
                                ;35172  =;END
```

```
                                ;35173
                                ;35174          ;-------------------------------;
                                ;35175  ADDG.32:
                                ;35176          Q_R[R15],                       ;  Q GETS FRAC <LS>
U 1B29, 0000,1B3C,01C0,FA78,0000,1A1A   ;35177          ALU?                            ;  IS FRAC NEGATIVE ?
                                ;35178
                                ;35179          ;-------------------------------;
                                ;35180  =1010   ;BRANCH ON ALU Z-BIT AND C31
                                ;35181          ;1010---------------------------;  BORROW
                                ;35182          SD_NOT.SD,                      ;  CHANGE SIGN
                                ;35183          Q_0-Q,                          ;  COMPLEMENT FRAC <LS>
                                ;35184          CLK.UBCC,                       ;  CLOCK BORROW IF ANY
U 1A1A, 001F,0000,01C3,F800,0010,1B2A   ;35185          J/ADDG.33
                                ;35186
                                ;35187          ;1011---------------------------;
                                ;35188  ;       ENTER HERE FROM EMODG-ROUTINE.
                                ;35189  ;       EXPECTS:
                                ;35190  ;               RC[T5] = BIASSED EXONENT -1
                                ;35191  ;               <D,Q> = FRACTION
                                ;35192  =1011
                                ;35193  ADDG.31:
                                ;35194          SC_SC+SHF.VAL,                  ;  POS. DIFF,NORMALIZE RESULT
                                ;35195          D_BAL.NORM,                     ;  D GETS NORMALIZED FRAC<MS>
U 1A1B, 0E00,003C,0180,F800,008C,9B2C   ;35196          J7ADDG.36
                                ;35197
                                ;35198          ;1111---------------------------;
                                ;35199  =1111
                                ;35200          SC_K[.20],                      ;  SHIFT VALUE OF 32
                                ;35201          D_0, Q_0,                       ;  SET FRAC <MS>, CLEAR FRAC <LS>
U 1A1F, 0C00,003C,75F8,F800,0084,7A1B   ;35202          J7ADDG.31
                                ;35203
                                ;35204          ;-------------------------------;
                                ;35205  ADDG.33:
                                ;35206          D_NOT.D, CLK.UBCC,              ;  1'S COMPLEMENT OF FRAC <MS>
U 1B2A, 0801,0128,0180,F800,0010,1924   ;35207          Z?                              ;  IS FRAC <LS> = 0 ?
                                ;35208
                                ;35209          ;-------------------------------;
                                ;35210  =0       ;BRANCH ON ALU Z-BIT
                                ;35211          ;0---------------------------;
U 1924, 0000,1B3C,0180,F800,0000,1A1B   ;35212          ALU?, J/ADDG.31                 ;  NO, IS FRAC<MS> = 0 ?
                                ;35213
                                ;35214          ;1---------------------------;
                                ;35215          D_D+K[.1],                      ;  YES, MAKE FRAC <MS> 2'S COMPLEMENT
U 1925, 0819,0014,0580,F800,0000,1A1B   ;35216          J7ADDG.31
                                ;35217  =;END    ;-------------------------------;
                                ;35218
```

```
                                ;35219  ;       D = FRAC <MS>
                                ;35220  ;       Q = FRAC <LS>
                                ;35221  ;       LC = EXP
                                ;35222  ;       SC = NUMBER OF LEADING 0'S
                                ;35223
                                ;35224  ADDG.36:
                                ;35225          ;-------------------------------;  ENTER WITH FRAC<MS> NORMALIZED
                                ;35226          R[R15]_D,                       ;  SAVE FRACTION <MS> IN R15
U 1B2C, 0C01,003C,01F8,FAF8,0000,1B2E   ;35227          D_Q, Q_0                        ;  SETUP FOR NORMALIZE FRAC <LS>
                                ;35228
                                ;35229          ;-------------------------------;
                                ;35230          D_DAL.SC,                       ;  NORMALIZE FRAC <LS>
U 1B2E, 0D00,003C,01C0,FA78,0000,1B30   ;35231          Q_R[R15]                        ;  Q GETS BACK FRAC <MS>
                                ;35232
                                ;35233          ;-------------------------------;
                                ;35234          R[R15]_D,                       ;  SAVE NORMALIZED FRAC <LS>
U 1B30, 0C01,003C,01F8,FAF8,0000,1B31   ;35235          D_Q,Q_0                         ;  D GETS FRAC <MS>
                                ;35236
                                ;35237  ADDG.37:;-------------------------------;
                                ;35238          LAB_R[R15],                     ;  LATCH FRAC <LS>
                                ;35239          ALU_LA[OR]D,CLK.UBCC,           ;  CHECK IF RESULT FRAC IS 0
U 1B31, 001C,2030,7D80,FA78,0114,7B32   ;35240          FE_R[.18]                       ;  GET READY TO PACK RESULT
                                ;35241
                                ;35242          ;-------------------------------;
                                ;35243          LC_RC[T5],                      ;  LATCH EXPONENT
U 1B32, 0000,013C,8180,F928,0000,1934   ;35244          K[.3FF], Z?                     ;  SLOW CONSTANT FOR ROUNDING
                                ;35245
                                ;35246  =0      ;0-------------------------------;
                                ;35247          R[R15]_LA+K[.3FF]+1,            ;  ROUND FRACTION
                                ;35248          CLK.UBCC,                       ;  CLOCK CARRY IF ANY
U 1934, 0018,0010,8180,FAF8,0010,1B34   ;35249          J/ADDG.38                       ;  PACK UP RESULT
                                ;35250
                                ;35251          ;1-------------------------------;
                                ;35252          RC[T1]_K[ZERO], D_0,
                                ;35253          SET.CC(INST),
U 1935, 0F18,C038,19F8,F988,0070,1B33   ;35254          Q_0
                                ;35255  =;END
                                ;35256          ;-------------------------------;
U 1B33, 0000,003E,E580,3C00,0000,0010   ;35257          ID[T9]_D,RETURN10               ;  CLEAR LOW FRAC AND RETURN
                                ;35258
                                ;35259  ADDG.38:
                                ;35260          ;-------------------------------;
                                ;35261          Q_0+LC+1,                       ;  INCREMENT EXPONENT TO COMPENSATE
                                ;35262                                          ;   FOR SHIFT VALUE
U 1B34, 0013,0010,01C0,FA78,0000,1B36   ;35263          LAB_R[R15]                      ;  LATCH FRAC <LS> ROUNDED
                                ;35264
                                ;35265          ;-------------------------------;
U 1B36, 0019,2000,1D80,FAF8,0000,1B37   ;35266          R[R15]_Q-K[SC]                  ;  SUBTRACT # OF LEADING 0'S FROM EXP
                                ;35267
                                ;35268          ;-------------------------------;
                                ;35269          SC_K[.FFF5],                    ;  GET READY FOR PACKG-ROUTINE
                                ;35270          Q_LA,                           ;  Q GETS FRAC <LS>
U 1B37, 0000,003C,E1C0,F800,0084,7889   ;35271          J7PACKG                         ;  PACK UP THE RESULT
```

```
                                        ;35272  :ADDG/SUBG ALIGNMENT SUBROUITNE, ENTER FROM ADDG.100 BRANCH TARGETS WITH:
                                        ;35273  ;INPUTS:
                                        ;35274  ;       <Q,D>=FRACTION TO BE ALIGNED
                                        ;35275  ;       TO = SRC FRAC <MS>
                                        ;35276  ;       FE=SC=AMOUNT TO ALIGN IT
                                        ;35277  ;       LC=LARGER OF SRC EXP AND DST EXP
                                        ;35278  ;       R15 = OP2 EXP - OP1 EXP
                                        ;35279  ;       ALU CC REFLECT R15
                                        ;35280  ;       ENTRY IS AT ALNNEGG IF <Q,D> SHOULD BE NEGATED FIRST. ELSE AT ALNPOSG.
                                        ;35281  ;
                                        ;35282  ;       THERE ARE SEVERAL RETURN POINTS:
                                        ;35283  ;       RETURN10 IF     EXP DIFF < 32.
                                        ;35284  ;       RETURN21 IF     32.<=EXP DIFF
                                        ;35285  ;       RETURN31 IF     EXP DIFF > 63
                                        ;35286
                                        ;35287  ALNNEGG:
                                        ;35288          ;-------------------------------;
U 1B38, 081F,2000,0180,F800,0010,1B39   ;35289          D_0-D, CLK.UBCC                 ;  NEGATE LOW PART FIRST
                                        ;35290
                                        ;35291          ;-------------------------------;
U 1B39, 001F,0300,01C0,F800,0000,1829   ;35292          Q_0-Q, C31?                     ;  NEGATE HIGH PART, CHECK BORROW
                                        ;35293
                                        ;35294  =01     ;01-----------------------------;BRANCH ON BORROW
U 1829, 0019,2000,05C0,F800,0000,182B   ;35295          Q_Q-K[.1]                       ;  SUBTRACT BORROW FROM HIGH PART
                                        ;35296
                                        ;35297  ALNPOSG:
                                        ;35298          ;11-----------------------------;
                                        ;35299          LAB_R[R15],                     ;  LATCH UP EXPONENT DIFFERENCE
U 182B, 0018,0024,8D80,FA78,0010,1B3A   ;35300          ALU_LA.ANDNOT.K[.1F], CLK.UBCC  ;  CLEAR ALU Z IF EXP DIFF >= 32.
                                        ;35301
                                        ;35302          ;-------------------------------;
                                        ;35303          ALU_LA.ANDNOT.K[.3F], CLK.UBCC, ;  CLEAR ALU Z IF EXP DIFF > 63
U 1B3A, 0018,0124,5580,F800,0010,193C   ;35304          Z?                              ;  TEST FOR >= 32.
                                        ;35305
                                        ;35306  =0      ;0-----------------------------;BRANCH ON ALU Z-BIT
                                        ;35307          SC_SC+K[.20],D_Q,ALU_-1,Q_ALU,  ;  EXP DIFF >= 32
U 193C, 0C03,1228,75C0,F800,0084,9A2E   ;35308          EALU?,J/ALN.00                  ;  TEST FOR NEGATIVE FRACTION
                                        ;35309
                                        ;35310          ;1------------------------------;
U 193D, 0D00,003E,0180,F800,0000,0010   ;35311          D_DAL.SC, RETURN[10]            ;  EXP DIFF < 32
                                        ;35312
                                        ;35313  =1110
                                        ;35314  ALN.00: ;1110---------------------------;BRANCH ON SS
U 1A2E, 0000,013C,01F8,F800,0000,1940   ;35315          Q_0,Z?,J/GH.ALN.01
                                        ;35316
                                        ;35317          ;1111---------------------------;
U 1A2F, 0000,013C,0180,F800,0000,1940   ;35318          Z?                              ;  TEST FOR EXP DIFF >= 64
                                        ;35319  =0
                                        ;35320  GH.ALN.01:
                                        ;35321          ;0-----------------------------;BRANCH ON ALU Z-BIT
                                        ;35322          D_RC[T2],                       ;  EXP DIFF > 63, READ DST LONG EXP
U 1940, 0810,003A,C1F0,2D10,0000,0031   ;35323          Q_ID[T0], RETURN[31]            ;  HOPELESS, GET SRC FRAC <MS>
                                        ;35324
                                        ;35325          ;1------------------------------;
U 1941, 0D00,003E,0180,F800,0000,0021   ;35326          D_DAL.SC, RETURN[21]            ;  31< EXP DIFF < 64
```

```
                                        ;35327  .TOC    ''      G & H floating point  : GRAND FLOATING POINT MULTIPLY''
                                        ;35328
                                        ;35329  ;       GRAND FLOATING MULTIPLY ROUTINE.
                                        ;35330
                                        ;35331  ;       FIRST MULG2, *-R
                                        ;35332
                                        ;35333  ;INPUTS:
                                        ;35334  ;               D = OP1 LONG FRAC
                                        ;35335  ;               RC0 = OP1 LONG EXP
                                        ;35336  ;               LA = LB = OP2 LONG EXP
                                        ;35337
                                        ;35338          ;-------------------------------;
                                        ;35339  1420:   TRAP.ACC[1],                    ;  RESET THE FPA
                                        ;35340          STATE_K[ZERO],                  ;  CLEAR POLYG FLAG
                                        ;35341          RC[T2]_LA,                      ;  RC2 GETS OP2 LONG EXP
U 1420, 0000,00BC,18E0,F990,1404,7921   ;35342          Q_D                             ;  Q GETS OP1 LONG FRAC
                                        ;35343
                                        ;35344  =0****  ;0****-------------------------;  CALL SITE FOR ADD/SUB
                                        ;35345          D_R(PRN+1),                     ;  D GETS OP2 LONG FRAC
U 1921, 0800,003D,0180,F860,0000,184C   ;35346          CALL, J/MULG.00                 ;  CALL MULTIPLY ROUTINE
                                        ;35347
                                        ;35348          ;1****----------------------;
                                        ;35349          R(PRN)_D,                       ;  STORE RESULT LONG EXP
U 1931, 0001,003C,0180,F8D8,0000,0796   ;35350          J/ADDD.P                        ;  STORE RESULT LONG FRAC
                                        ;35351  =;END   ;-------------------------------;
                                        ;35352
                                        ;35353  ;MULG, *-MEMORY, AND *-*-*
                                        ;35354
                                        ;35355  ;INPUTS:
                                        ;35356  ;               <RC0,Q> = PACKED MULTIPLIER
                                        ;35357  ;               <RC2,D> = PACKED MULTIPLICAND
                                        ;35358
                                        ;35359  168F:                                   ;  1600+8F
                                        ;35360  MULG:   ;-------------------------------;
                                        ;35361          TRAP.ACC[1],                    ;  RESET THE FPA
                                        ;35362          STATE_K[ZERO],                  ;  Clear POLY-flag
U 168F, 0000,00BD,1880,F800,1404,7B4C   ;35363          CALL,J/MULG.00                  ;  MULTIPLY SUBROUTINE
                                        ;35364
                                        ;35365  169F:   ;-------------------------------;
                                        ;35366          STATE_K[.1],                    ;  FLAG TO SIGNAL WRITE-DEST AT 1441
U 169F, F000,003F,05F0,F847,1404,7400   ;35367          WRITE.G.DEST, J/WRG             ;  WRITE RESULT
                                        ;35368          ;-------------------------------;
```

```
                                ;35369  .TOC    ''      G & H floating point  : DIVIDE GRAND FLOATING ROUTINE''
                                ;35370
                                ;35371  ;DIVG2, *-R
                                ;35372  ;INPUTS:
                                ;35373  ;               D = OP1 LONG FRAC
                                ;35374  ;               RC0 = OP1 LONG EXP
                                ;35375  ;               LA = LB = OP2 LONG EXP
                                ;35376
                                ;35377  1421:   ;-------------------------------;
                                ;35378          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;35379          RC[T2]_LA,                      ;  RC2 GETS OP2 LONG EXP
U 1421, 0000,00BC,00E0,F990,0000,1B3B   ;35380          Q_D
                                ;35381
                                ;35382          ;-------------------------------;
U 1B3B, 0800,003C,0180,F860,0000,196C   ;35383          D_R(PRN+1)                      ;  D GETS DST LONG FRAC
                                ;35384
                                ;35385  =0****  ;0****----------------------------;  CALLSITE FOR DIVIDE SUBR
                                ;35386          RC[T6]_D,                       ;  SAVE DST LONG FRAC
                                ;35387          D_Q,                            ;  D GETS OP1 LONG FRAC
                                ;35388          SC_K[.10],                      ;  16. FOR WORD SWAP
U 196C, 0C01,003D,6580,F9B0,0084,7888   ;35389          CALL, J/DIVG.S
                                ;35390
                                ;35391          ;1****----------------------------;
                                ;35392          ALU_D,N&Z_ALU.V&C_0, WORD,      ;  SET CONDITION CODES
U 197C, 0001,5A3C,0180,F800,0050,00FC   ;35393          PSL.V?, J7ADDD.M                ;  TEST FOR OVERFLOW
                                ;35394  =;END   ;-------------------------------;
```

```
                                        ;35395  ;DIVG, *-MEMORY, AND *-*-*
                                        ;35396
                                        ;35397  ;INPUTS:
                                        ;35398  ;               D = OP2 LONG FRAC
                                        ;35399  ;               RC2 = OP2 LONG EXP
                                        ;35400  ;               Q = OP1 LONG FRAC
                                        ;35401  ;               RC0 = OP1 LONG EXP
                                        ;35402
                                        ;35403  168D:   ;-------------------------------;  1600 + 8D
                                        ;35404  DIVG:   TRAP.ACC[1],                    ;  RESET THE FPA
                                        ;35405          RC[T6]_D,                       ;  SAVE OP2 LONG FRAC IN RC6
                                        ;35406          D_Q,                            ;  D GETS OP1 LONG FRAC
                                        ;35407          SC_K[.10],                      ;  16. FOR WORD SWAP
U 168D, 0C01,008D,6480,F9B0,0084,78B8   ;35408          CALL, J/DIVG.S                  ;  CALL DIVIDE GRAND SUBROUTINE
                                        ;35409
                                        ;35410  169D:   ;-------------------------------;
                                        ;35411          ALU_D,N&Z_ALU.V&C_0, WORD,      ;  SET CC ON DST
U 169D, 0001,5A3C,0180,F800,0050,182C   ;35412          PSL.V?                          ;  NEED TO RESET V?
                                        ;35413  =;END
                                        ;35414          ;-------------------------------;
                                        ;35415  =110*   ;BRANCH ON PSL V-BIT
                                        ;35416          ;110*-------------------------;
                                        ;35417          STATE_K[.1],                    ;  SIGNAL WRITE-DEST AT 1441
U 182C, F000,003F,05F0,F847,1404,7400   ;35418          WRITE.G.DEST, J/WRG             ;  NO, JUST WRITE DESTINATION
                                        ;35419
                                        ;35420          ;111*-------------------------;
                                        ;35421          STATE_K[.1],                    ;  SIGNAL WRITE-DEST AT 1441
                                        ;35422          SET.V,                          ;  SET V
U 182E, F000,003F,05F0,F847,1424,7400   ;35423          WRITE.G.DEST, J/WRG
                                        ;35424          ;---------------------------;
```

```
                                ;35425  ;GRAND FLOATING DIVIDE SUBROUTINE
                                ;35426
                                ;35427  ;ALGORITHM:
                                ;35428  ;               NON-RESTORING DIVIDE IN TWO LOOPS, ONE FOR
                                ;35429  ;               FIRST 32 BITS OF QUOTIENT, AND ONE FOR NEXT 18 BITS.
                                ;35430
                                ;35431  ;INPUTS:       RC0 = OP1 LONG EXP
                                ;35432  ;               D = Q = OP1 LONG FRAC
                                ;35433  ;               RC1 = OP2 LONG EXP
                                ;35434  ;               RC6 = OP2 LONG FRAC
                                ;35435  ;               SC = 16.
                                ;35436  ;OUTPUTS:
                                ;35437  ;               D = PACKED ROUNDED QUOTIENT (LONG EXP)
                                ;35438  ;               RC1 = ROUNDED QUOTIENT LONG FRAC
                                ;35439  ;               CC AND ID[CES] SET FROM QUOTIENT
                                ;35440  ;TEMPORARIES:
                                ;35441  ;               RC3 = OP1 LONG FRAC
                                ;35442  ;               RC4 = DIVISOR FRAC <MS>
                                ;35443  ;               R15 =
                                ;35444  ;               ID[T3] = R1 DURING MAIN LOOP
                                ;35445  ;               ID[T4] = OP2 FRAC <LS>
                                ;35446
                                ;35447  ;RETURNS:
                                ;35448  ;               RETURN @ 10
                                ;35449
                                ;35450  =00     ;-------------------------------;
                                ;35451  DIVG.S: TRAP.ACC[1],                    ;  RESET THE FPA
                                ;35452          SGN/CLR.SD+SS,                  ;  CLEAR OUT SS
                                ;35453          RC[T3]_D,                       ;  SAVE OP1 LONG FRAC IN RC3
                                ;35454          D_DAL.SC,                       ;  SWAP THE TWO WORDS OF OP1 LONG FRAC
                                ;35455          SC_K[.FFF9],                    ;  SET UP SHIFT AMOUNT OF -7
U 18B8, 0D01,00BD,BC87,F998,0084,7AF2   ;35456          CALL, J/UNPACKG                 ;  UNPACK GRAND SUBROUTINE
                                ;35457
                                ;35458          ;01-----------------------------;  DIVISOR IS 0
                                ;35459          D_K[.8000],                     ;  RESERVED OPERAND
                                ;35460          N&Z_ALU.V&C_0,                  ;  CLOCK CONDITION CODES
U 18B9, 0818,0038,4580,F800,0050,0470   ;35461          J/DIVD.20                       ;  USE DIVIDE DOUBLE ROUTINE
                                ;35462
                                ;35463          ;10-----------------------------;  DIVIDEND IS 0
                                ;35464          RC[T1]_K[ZERO], D_0,            ;  CLEAR OUT QUOTIENT
                                ;35465          N&Z_ALU.V&C_0,WORD,             ;  SET CONDITION CODES
U 18BA, 0F18,403A,1980,F988,0050,0010   ;35466          RETURN10
                                ;35467
                                ;35468          ;11-----------------------------;  SRC NE 0, DST NE 0
                                ;35469
                                ;35470  ;       ID[T0] = OP1 FRAC <MS> = DIVISOR FRAC <MS>
                                ;35471  ;       Q = ID[T2] = OP1 FRAC <LS> = DIVISOR FRAC <LS>
                                ;35472  ;       ID[T1] = OP2 FRAC <MS> = DIVIDEND FRAC <MS>
                                ;35473  ;       D = OP2 FRAC <LS> = DIVIDEND FRAC <LS>
                                ;35474  ;       RC5 = OP1 EXP
                                ;35475  ;       RC4 = OP2 EXP
                                ;35476  ;       R15 = LAB = EXP2-EXP1
                                ;35477
U 18BB, 0000,0C3C,0180,F988,0000,1B3C   ;35478          ALU_LA,RC[T1]_ALU               ;  STORE EXP2-EXP1 IN RC1
```

```
                                ;35479          ;--------------------------------;
                                ;35480          R[R15]_Q,                       ;  STORE DIVISOR FRAC <LS> IN R15
                                ;35481          Q_D,                            ;  Q GETS DIVIDEND FRAC <LS>
                                ;35482          SD_SS,                          ;  SD GETS MINUS RESULT SIGN
J 1B3C, 0001,203C,51E4,FAF9,0184,7B3E           ;35483          FE_K[.1E],SC_K[.1E]             ;  LOOP COUNT
                                ;35484
                                ;35485          ;--------------------------------;
                                ;35486          SD_NOT.SD,                      ;  FIX UP RESULT SIGN
U 1B3E, 0800,003C,0183,FA08,0000,1B40           ;35487          D_R[R1]                         ;  NEED TO SAVE R1
                                ;35488
                                ;35489          ;--------------------------------;
                                ;35490          ID[T3]_D,                       ;  SAVE R1 IN T3
                                ;35491          D_Q,                            ;  D GETS DIVIDEND FRAC <LS>
                                ;35492          A[U_-1,N&Z_ALU.V&C_0,           ;  SET PSL N-BIT
U 1B40, 0C03,0028,CD80,3E78,0050,1B41           ;35493          LAB_R[R15]                      ;  LATCH D'SOR FRAC <LS>
                                ;35494
                                ;35495          ;--------------------------------;
                                ;35496          ALU_D-LB.RLOG,                  ;  SUBTRACT <LS> FRACTIONS
                                ;35497          R[RT]_ALU.LEFT,SI/ZERO,         ;  STORE RESULT IN R1
                                ;35498          CLK.UBCC,                       ;  CLOCK BORROW
U 1B41, 002D,0004,C1F0,2E88,0010,1B42           ;35499          Q_ID[T0]                        ;  GET D'SOR FRAC <MS>
                                ;35500
                                ;35501          ;--------------------------------;
                                ;35502          RC[T4]_Q,                       ;  RC4 GETS D'SOR FRAC <MS>
                                ;35503          Q_ID[TT],                       ;  Q GETS DIVIDEND FRAC <MS>
U 1B42, 0001,203C,C5F0,2DA0,1C00,1854           ;35504          MSC/READ.RLOG                   ;  POP REGISTER-LOG
                                ;35505
                                ;35506          ;10*-------------------------------;
                                ;35507  =10*
                                ;35508          LC_RC[T4],                      ;  LATCH D'SOR FRAC <MS>
                                ;35509          D_Q,Q_0,                        ;  D GETS D'END FRAC <MS>
                                ;35510          SC_SC-K[.1],                    ;  ADJUST LOOP COUNT
                                ;35511          ALU?,                           ;  TEST FOR BORROW
U 1854, 0C00,1B3D,05F8,F920,0084,BA36           ;35512          CALL,J/DIVG.0
                                ;35513
                                ;35514          ;11*-------------------------------;
                                ;35515          D_Q,                            ;  D GETS QUOTIEN FRAC <LS>
                                ;35516          Q_ID[T3],                       ;  Q GETS OLD VALUE OF R1
                                ;35517          LAB_R[R15],                     ;  LATCH QUOTIEN FRAC <MS>
U 1856, 0C00,003C,CDF0,2E78,0000,1B46           ;35518          J/DIVG.16
                                ;35519  =;END    ;--------------------------------;
```

```
                                ;35520  =0110
                                ;35521  DIVG.0: ;0110-------------------------;  BORROW
                                ;35522          ALU_D-LC-1,                  ;  SUBTRACT <MS> FRACS WITH BORROW
                                ;35523          D_ALU.LEFT,                  ;  SHIFT TO GET READY FOR NEXT OPERATION
                                ;35524          Q_Q.LEFT,SI/DIV,             ;  SHIFT IN QUOTIENT BIT (=ALU31=0)
                                ;35525          C[K.UBCC,                    ;  FLAG RESULT FOR NEXT OPERATION, +/-
U 1A36, 0831,0008,02A8,F888,0010,1B44   ;35526          LA_RA[R1],J/DIVG.00          ;  GET LA READY
                                ;35527
                                ;35528          ;0111-------------------------;  NO BORROW
                                ;35529          ALU_D-LC,                    ;  SUBTRACT <MS> FRACS
                                ;35530          D_ALU.LEFT,                  ;  SHIFT TO GET READY FOR NEXT OPERATION
                                ;35531          Q_Q.LEFT,SI/DIV,             ;  SHIFT IN QUOTIENT BIT (=ALU31)
                                ;35532          C[K.UBCC,                    ;  FLAG RESULT FOR NEXT OPERATION, +/-
U 1A37, 0831,0000,02A8,F888,0010,1B44   ;35533          LA_RA[R1],J/DIVG.00          ;  GET LA READY
                                ;35534
                                ;35535          ;1110-------------------------;  BORROW
                                ;35536          ALU_D-LC-1,                  ;  SUBTRACT <MS> FRACS WITH BORROW
                                ;35537          D_ALU.LEFT,                  ;  SHIFT TO GET READY FOR NEXT OPERATION
                                ;35538          Q_Q.LEFT,SI/DIVD,            ;  SHIFT IN QUOTIENT BIT (=ALU31=1)
                                ;35539          C[K.UBCC,                    ;  FLAG RESULT FOR NEXT OPERATION, +/-
U 1A3E, 0831,0008,0028,F888,0010,1B44   ;35540          LA_RA[R1],J/DIVG.00          ;  GET LA READY
                                ;35541
                                ;35542          ;1111-------------------------;  NO BORROW
                                ;35543          ALU_D-LC,                    ;  SUBTRACT <MS> FRACS
                                ;35544          D_ALU.LEFT,                  ;  SHIFT TO GET READY FOR NEXT OPERATION
                                ;35545          Q_Q.LEFT,SI/DIVD,            ;  SHIFT IN QUOTIENT BIT (=ALU31=1)
                                ;35546          C[K.UBCC,                    ;  FLAG RESULT FOR NEXT OPERATION, +/-
U 1A3F, 0831,0000,0028,F888,0010,1B44   ;35547          LA_RA[R1]                    ;  GET LA READY
                                ;35548  =;END
                                ;35549  DIVG.00:;-------------------------;
U 1B44, 0000,033C,0180,F800,0000,18D0   ;35550          C31?                         ;  ADD OR SUBTRACT NEXT ?
                                ;35551
                                ;35552  =00*    ;00*--------------------------;  BORROW
                                ;35553          LC_RC[T4]&R1_(LA+LB+PSL.C).LEFT,;  ADD DIVISOR<LS> TO D'END <LS>
                                ;35554          SI7ZERO,                     ;  SHIFT IN ZEROS
                                ;35555          CLK.UBCC,                    ;  CLOCK ALU CARRY
                                ;35556          SC_SC+1,                     ;  INCREMENT LOOP COUNT BY 1
U 18D0, 002C,002D,0180,FBA0,0090,D9CA   ;35557          CALL,J/DIVG.081              ;  SINCE FIRST QUOTIENT BIT IS 0
                                ;35558
                                ;35559          ;01*--------------------------;  NO BORROW
                                ;35560          STATE_STATE.OR.K[.80],       ;  SET FLAG TO INCREMENT EXP BY 1
                                ;35561          LC_RC[T4]&R1_(LA-LB.RLOG).LEFT,;  SUBTRACT DIVISOR<LS> FROM DIV<MS>
                                ;35562          SI7ZERO,                     ;  SHIFT IN ZEROS
                                ;35563          CLK.UBCC,                    ;  CLOCK ALU BORROW
U 18D2, 002C,0005,4180,FBA0,1414,398E   ;35564          CALL,J/DIVG.061              ;
                                ;35565
                                ;35566          ;11*--------------------------;
                                ;35567  =11*    SC_SC-K[.1],                 ;  ADJUST LOOP COUNT
                                ;35568          R[R15]_Q,                    ;  FRACTION <MS>
                                ;35569          Q_0,                         ;  INITAILIZE QUOTIENT
U 18D6, 0001,233C,05F8,FAF8,0084,B99C   ;35570          C31?,J/DIVG.063
                                ;35571  =;END
```

```
                                    ;35572  ; ENTER HERE TO DO THE SECOND HALF OF A SUBTRACT-OPERATION
                                    ;35573  ; BRANCH ON SC GT 0.
                                    ;35574
                                    ;35575  =*0*    ;BRANCH ON SC GT 0
                                    ;35576          ;*0*-----------------------------;
                                    ;35577  DIVG.06:SC_K[.18],                      ;  LOOP COUNT FOR NEXT PASS
                                    ;35578          MSC/READ.RLOG,                  ;  POP REGISTER-LOG
U 198C, 0000,1B3C,7D80,F800,1C84,7A66 ;35579        ALU?,J/DIVG.064                 ;  END OF FIRST LOOP
                                    ;35580
                                    ;35581          ;*1*-----------------------------;
U 198E, 0000,003C,0180,F800,1C00,1B45 ;35582  DIVG.061:MSC/READ.RLOG                 ;  POP REGISTER-LOG
                                    ;35583  =;END
                                    ;35584          ;-------------------------------;
                                    ;35585          ALU_D-LC,CLK.UBCC,              ;  CLOCK DIFFERENCE
U 1B45, 0011,1B00,0180,F800,0010,1A46 ;35586        ALU?                            ;  TEST RESULT OF <LS> SUBTRACT
                                    ;35587
                                    ;35588          ;-------------------------------;
                                    ;35589  =0110   ;BRANCH ON ALU N&Z-BITS
                                    ;35590          ;0110-----------------------------;  BORROW
                                    ;35591          ALU_D-LC-1,                     ;  SUBTRACT WITH BORROW
                                    ;35592          D_ALU.LEFT,SI/DIV,              ;  SHIFT IN QUOTIENT BIT
                                    ;35593          Q_Q.LEFT,
                                    ;35594          LA_RA[R1],
                                    ;35595          CLR.UBCC,
U 1A46, 0831,1B08,02A8,F888,0010,1A5A ;35596        ALU?,J/DIVG.062
                                    ;35597
                                    ;35598          ;0111-----------------------------;  NO BORROW
                                    ;35599          ALU_D-LC,                       ;  SUBTRACT
                                    ;35600          D_ALU.LEFT,SI/DIV,              ;  SHIFT IN QUOTIENT BIT
                                    ;35601          Q_Q.LEFT,
                                    ;35602          LA_RA[R1],
                                    ;35603          CLR.UBCC,
U 1A47, 0831,0300,02A8,F888,0010,199C ;35604        C31?,J/DIVG.063
                                    ;35605
                                    ;35606          ;1110-----------------------------;  BORROW
                                    ;35607          ALU_D-LC-1,                     ;  SUBTRACT WITH BORROW
                                    ;35608          D_ALU.LEFT,SI/DIVD,             ;  SHIFT IN QUOTIENT BIT
                                    ;35609          Q_Q.LEFT,
                                    ;35610          LA_RA[R1],
                                    ;35611          CLR.UBCC,
U 1A4E, 0831,1B08,0028,F888,0010,1A5A ;35612        ALU?,J/DIVG.062
                                    ;35613
                                    ;35614          ;1111-----------------------------;  NO BORROW
                                    ;35615          ALU_D-LC,                       ;  SUBTRACT
                                    ;35616          D_ALU.LEFT,SI/DIVD,             ;  SHIFT IN QUOTIENT BIT
                                    ;35617          Q_Q.LEFT,
                                    ;35618          LA_RA[R1],
                                    ;35619          CLR.UBCC,
U 1A4F, 0831,0300,0028,F888,0010,199C ;35620        C31?,J/DIVG.063
```

```
                                ;35621          ;-------------------------------;
                                ;35622  =1010   ;BRANCH ON ALU Z AND C-BITS
                                ;35623          ;1010---------------------------;
                                ;35624  DIVG.062:LC_RC[T4]&R1_(LA+LB+PSL.C).LEFT,;  R1 GETS D'END FRAC <LS>
                                ;35625          SI/ZERO,                        ;  SHIFT IN ZEROES
                                ;35626          CLK.UBCC,                       ;  CLOCK THE CARRY
                                ;35627          SC_SC-K[.1],
U 1A5A, 002C,142C,0580,FBA0,0094,B9C8   ;35628  SC.GT.0?,J/DIVG.08
                                ;35629
                                ;35630          ;1011---------------------------;
                                ;35631          LC_RC[T4]&R1_(LA-LB.RLOG).LEFT, ;  R1 GETS D'END FRAC <LS>
                                ;35632          SI7ZERO,                        ;  SHIFT IN ZEROES
                                ;35633          CLK.UBCC,                       ;  CLOCK THE CARRY
                                ;35634          SC_SC-K[.1],
U 1A5B, 002C,1404,0580,FBA0,0094,B98C   ;35635  SC.GT.0?,J/DIVG.06
                                ;35636
                                ;35637  =1111   ;1111---------------------------;  ACTUALLY A BORROW RESULTED
                                ;35638          LC_RC[T4]&R1_(LA+LB+PSL.C).LEFT, ;  R1 GETS D'END FRAC <LS>
                                ;35639          SI7ZERO,                        ;  SHIFT IN ZEROES
                                ;35640          CLK.UBCC,                       ;  CLOCK THE CARRY
                                ;35641          SC_SC-K[.1],
U 1A5F, 002C,142C,0580,FBA0,0094,B9C8   ;35642  SC.GT.0?,J/DIVG.08
                                ;35643  =;END
                                ;35644          ;-------------------------------;
                                ;35645  =0*     ;BRANCH ON ALU C31
                                ;35646          ;0*-----------------------------;
                                ;35647  DIVG.063:LC_RC[T4]&R1_(LA+LB+PSL.C).LEFT,;  R1 GETS D'END FRAC <LS>
                                ;35648          SI/ZERO,                        ;  SHIFT IN ZEROES
                                ;35649          CLK.UBCC,                       ;  CLOCK THE CARRY
                                ;35650          SC_SC-K[.1],
U 199C, 002C,142C,0580,FBA0,0094,B9C8   ;35651  SC.GT.0?,J/DIVG.08
                                ;35652
                                ;35653          ;1*-----------------------------;
                                ;35654          LC_RC[T4]&R1_(LA-LB.RLOG).LEFT, ;  R1 GETS D'END FRAC <LS>
                                ;35655          SI7ZERO,                        ;  SHIFT IN ZEROES
                                ;35656          CLK.UBCC,                       ;  CLOCK THE CARRY
                                ;35657          SC_SC-K[.1],
U 199E, 002C,1404,0580,FBA0,0094,B98C   ;35658  SC.GT.0?,J/DIVG.06
                                ;35659  =;END
```

```
                                ;35660          ;-------------------------------;
                                ;35661  =0110    ;BRANCH ON ALU N&C-BITS
                                ;35662          ;0110---------------------------;  BORROW
                                ;35663  DIVG.064:ALU_D-LC-1,                    ;  SUBTRACT WITH BORROW
                                ;35664          D_ALU.LEFT,SI/DIV,              ;  SHIFT IN QUOTIENT BIT
                                ;35665          Q_Q.LEFT,
                                ;35666          !A_RA[R1],
                                ;35667          CLR.UBCC,
U 1A66, 0831,000A,02A8,F888,0010,0006   ;35668          RETURN[6]
                                ;35669
                                ;35670          ;0111---------------------------;  NO BORROW
                                ;35671          ALU_D-LC,                       ;  SUBTRACT
                                ;35672          D_ALU.LEFT,SI/DIV,              ;  SHIFT IN QUOTIENT BIT
                                ;35673          Q_Q.LEFT,
                                ;35674          !A_RA[R1],
                                ;35675          CLR.UBCC,
U 1A67, 0831,0002,02A8,F888,0010,0006   ;35676          RETURN[6]
                                ;35677
                                ;35678          ;1110---------------------------;
                                ;35679          ALU_D-LC-1,                     ;  SUBTRACT WITH BORROW
                                ;35680          D_ALU.LEFT,SI/DIVD,             ;  SHIFT IN QUOTIENT BIT
                                ;35681          Q_Q.LEFT,
                                ;35682          !A_RA[R1],
                                ;35683          CLR.UBCC,
U 1A6E, 0831,000A,0028,F888,0010,0006   ;35684          RETURN[6]
                                ;35685
                                ;35686          ;1111---------------------------;  NO BORROW
                                ;35687          ALU_D-LC,                       ;  SUBTRACT
                                ;35688          D_ALU.LEFT,SI/DIVD,             ;  SHIFT IN QUOTIENT BIT
                                ;35689          Q_Q.LEFT,
                                ;35690          !A_RA[R1],
                                ;35691          CLR.UBCC,
U 1A6F, 0831,0002,0028,F888,0010,0006   ;35692          RETURN[6]
                                ;35693
```

```
                                ;35694          ;-------------------------------;
                                ;35695  =0*     ;BRANCH ON SC GT 0
                                ;35696          ;0*-----------------------------;
                                ;35697  DIVG.08:SC_K[.18],ALU?,
U 19C8, 0000,1B3C,7D80,F800,0084,7A86   ;35698          J/DIVG.084
                                ;35699
                                ;35700          ;1*-----------------------------;
                                ;35701  DIVG.081:ALU_D+LC+1,
                                ;35702          CLK.UBCC,
                                ;35703          LA_RA[R1],
U 19CA, 0011,1B10,0180,F888,0010,1A76   ;35704          ALU?                            ;  CHECK FOR CARRY AND N-BIT
                                ;35705
                                ;35706          ;-------------------------------;
                                ;35707  =0110    ;BRANCH ON ALU N&C-BITS
                                ;35708          ;0110---------------------------;  NO CARRY
                                ;35709          ALU_D+LC,
                                ;35710          D_ALU.LEFT,SI/DIV,              ;  ALU N-BIT IS 0
                                ;35711          Q_Q.LEFT,
                                ;35712          LA_RA[R1],
                                ;35713          CLK.UBCC,
U 1A76, 0831,1B14,02A8,F888,0010,1A5A   ;35714          ALU?,J/DIVG.062                 ;  CHECK FOR CARRY OR ZERO
                                ;35715
                                ;35716          ;0111---------------------------;
                                ;35717          ALU_D+LC+1,                     ;  ADD WITH CARRY
                                ;35718          D_ALU.LEFT,SI/DIV,
                                ;35719          Q_Q.LEFT,
                                ;35720          LA_RA[R1],
                                ;35721          CLK.UBCC,
U 1A77, 0831,0310,02A8,F888,0010,199C   ;35722          C31?,J/DIVG.063
                                ;35723
                                ;35724          ;1110---------------------------;
                                ;35725          ALU_D+LC,
                                ;35726          D_ALU.LEFT,SI/DIVD,             ;  ALU N-BIT IS 1
                                ;35727          Q_Q.LEFT,
                                ;35728          LA_RA[R1],
                                ;35729          CLK.UBCC,
U 1A7E, 0831,1B14,0028,F888,0010,1A5A   ;35730          ALU?,J/DIVG.062                 ;  CHECK FOR CARRY OR ZERO
                                ;35731
                                ;35732          ;1111---------------------------;
                                ;35733          ALU_D+LC+1,
                                ;35734          D_ALU.LEFT,SI/DIVD,
                                ;35735          Q_Q.LEFT,
                                ;35736          LA_RA[R1],
                                ;35737          CLK.UBCC,
U 1A7F, 0831,0310,0028,F888,0010,199C   ;35738          C31?,J/DIVG.063
                                ;35739
```

```
                                    ;35740          ;--------------------------------;
                                    ;35741  =0110   ;BRANCH ON ALU N&C BITS
                                    ;35742          ;0110----------------------------;  NO CARRY
                                    ;35743  DIVG.084:ALU_D+LC,
                                    ;35744          D_ALU.LEFT,SI/DIV,
                                    ;35745          Q_Q.LEFT,
                                    ;35746          LA_RA[R1],
                                    ;35747          CLR.UBCC,
U 1A86, 0831,0016,02A8,F888,0010,0006 ;35748        RETURN[6]
                                    ;35749
                                    ;35750          ;0111----------------------------;
                                    ;35751          ALU_D+LC+1,
                                    ;35752          D_ALU.LEFT,SI/DIV,
                                    ;35753          Q_Q.LEFT,
                                    ;35754          LA_RA[R1],
                                    ;35755          CLR.UBCC,
U 1A87, 0831,0012,02A8,F888,0010,0006 ;35756        RETURN[6]
                                    ;35757
                                    ;35758          ;1110----------------------------;
                                    ;35759          ALU_D+LC,
                                    ;35760          D_ALU.LEFT,SI/DIVD,
                                    ;35761          Q_Q.LEFT,
                                    ;35762          LA_RA[R1],
                                    ;35763          CLR.UBCC,
U 1A8E, 0831,0016,0028,F888,0010,0006 ;35764        RETURN[6]
                                    ;35765
                                    ;35766          ;1111----------------------------;
                                    ;35767          ALU_D+LC+1,
                                    ;35768          D_ALU.LEFT,SI/DIVD,
                                    ;35769          Q_Q.LEFT,
                                    ;35770          LA_RA[R1],
                                    ;35771          CLR.UBCC,
U 1A8F, 0831,0012,0028,F888,0010,0006 ;35772        RETURN[6]
                                    ;35773  =;END
                                    ;35774          ;--------------------------------;
                                    ;35775  DIVG.16:LC_RC[T1]&R1_Q,                  ;  RESTORE R1,LATCH EXPONENT
                                    ;35776          SC_K[.8],                        ;  SC GET 32.-#OF BITS IN <LS>
U 1B46, 0001,203C,01F8,FB88,0084,7B48 ;35777        Q_0
                                    ;35778
                                    ;35779          ;--------------------------------;
                                    ;35780          D_DAL.SC,                        ;  RIGHT ADJUST <LS> FRACTION
                                    ;35781          ALU_K[.80],Q_ALU.LEFT3,          ;  Q GETS BIAS OF .400
U 1B48, 0DB8,0038,41C0,FA78,0000,1B49 ;35782        LAB_R[R15]                       ;  LATCH FRACTION <MS>
                                    ;35783
                                    ;35784          ;--------------------------------;
                                    ;35785          ALU_D+Q,Q_ALU,D_Q,CLK.UBCC,      ;  ROUND FRACTION <LS>
                                    ;35786          SC_K[.FFF5],
U 1B49, 0C1D,1614,E1C0,F800,0094,7A97 ;35787        STATE7-4?                        ;  NEED TO ADJUST EXPONENT ?
                                    ;35788
```

```
                                        ;35789          ;--------------------------------;
                                        ;35790  =0111   ;BRANCH ON STATE 7
                                        ;35791          ;0111----------------------------;
                                        ;35792  DIVG.170:
                                        ;35793          ALU_D+LC,R[R15]_ALU,            ;  EXPONENT IS RIGHT
U 1A97, 0011,0014,0180,FAF8,0000,1B4A   ;35794          J/DIVG.17
                                        ;35795
                                        ;35796          ;1111----------------------------;
                                        ;35797          ALU_D+LC+1,R[R15]_ALU,           ;  INCREMENT EXPONENT
U 1A9F, 0011,0010,0180,FAF8,0000,1B4A   ;35798          J/DIVG.17
                                        ;35799
                                        ;35800          ;--------------------------------;
U 1B4A, 0800,003C,0180,F800,0000,1889   ;35801  DIVG.17:D_LA, J/PACKG                    ;  D GETS FRACTION <MS>
                                        ;35802          ;--------------------------------;
```

```
                                ;35803  .TOC    ''      G & H floating point  : MULTIPLY GRAND FLOATING''
                                ;35804
                                ;35805  ;       USED BY POLYG AND EMODG AS WELL AS MULG.
                                ;35806  ;INPUTS:
                                ;35807  ;               RC0 = OP1 LONG EXP
                                ;35808  ;               Q = OP1 LONG FRAC
                                ;35809  ;               RC2 = OP2 LONG EXP
                                ;35810  ;               D = OP2 LONG FRAC
                                ;35811
                                ;35812  ;TEMPORARIES:
                                ;35813  ;               T0 =
                                ;35814  ;               T9 = SCRATCH REGISTER
                                ;35815  ;               R15 = SCRATCH REGISTER
                                ;35816
                                ;35817          ;-------------------------------;
                                ;35818  MULG.00:TRAP.ACC[1],                    ;  RESET THE FPA
                                ;35819          RC[T6]_D,                       ;  SAVE OP2 LONG FRAC IN RC6
                                ;35820          D_Q,                            ;  DUPLICATE OP1 LONG FRAC
U 1B4C, 0C01,00BC,6480,F9B0,0084,78C8   ;35821          SC_K[.10]                       ;  16. FOR SWAPPING
                                ;35822
                                ;35823  =00     ;-------------------------------;  ENTER HERE FROM POLYG
                                ;35824  MULG.02:SGN/CLR.SD+SS,                  ;  CLEAR OUT SS
                                ;35825          RC[T3]_D,                       ;  SAVE OP1 LONG FRAC IN RC3
                                ;35826          D_DAL.SC,                       ;  WORD SWAP OP1 LONG FRAC
                                ;35827          SC_K[.FFF9],                    ;  SETUP SHIFT AMOUNT OF -7
U 18C8, 0D01,003D,BD87,F998,0084,7AF2   ;35828          CALL,J/UNPACKG
                                ;35829
                                ;35830          ;01-----------------------------;  RETURN1, SRC = 0
                                ;35831          D_0,Q_0,                        ;  PRODUCT IS 0
                                ;35832          RC[T1]_0,                       ;  CLEAR PROD LONG EXP
                                ;35833          SC_ALU, FE_K[.10],
                                ;35834          N&Z_ALU.V&C_0,                  ;  CLOCK CONDITION CODES
U 18C9, 0F03,003E,65F8,F988,01D6,6010   ;35835          RETURN10
                                ;35836
                                ;35837          ;10-----------------------------;  RETURN2, DST = 0
                                ;35838          D_0,Q_0,                        ;  PRODUCT IS 0
                                ;35839          RC[T1]_0,
                                ;35840          SC_ALU, FE_K[.10],
                                ;35841          N&Z_ALU.V&C_0,                  ;  CLOCK CONDITION CODES
U 18CA, 0F03,003E,65F8,F988,01D6,6010   ;35842          RETURN10
                                ;35843
                                ;35844          ;11-----------------------------;  RETURN3, SRC NE 0, DST NE 0
                                ;35845
                                ;35846  ;       T0 = M'PLIER FRAC <MS>
                                ;35847  ;       T2 = Q = M'PLIER FRAC <LS>
                                ;35848  ;       M'CAND<MS> = T1
                                ;35849  ;       D = M'CAND FRAC <LS>
                                ;35850  ;       RC4 = EXP2+EXP1
                                ;35851
                                ;35852          RC[T2]_D,                       ;  SAVE M'CAND FRAC <LS> IN RC2
                                ;35853          D_D.RIGHT, SI/ZERO,             ;  DIVIDE IT BY 2
                                ;35854          SC_FE,                          ;  SC GETS NABS(EXP2+EXP1)
                                ;35855                                          ;  LOW BITS ONLY
U 18CB, 0601,003C,0184,F990,0081,1B4D   ;35856          SD_SS                           ;  SD GETS XOR OF SIGNS
                                ;35857  =;END   ;-------------------------------;  NOT EMODG
```

```
                                ;35858          ;--------------------------------;
                                ;35859          R[R15]_D, CLK.UBCC,              ;  STORE M'CAND/2 IN R15
                                ;35860          D_Q, Q_0,                        ;  D GETS M'PLIER FRAC <LS>
U 1B4D, 0C01,003C,01F8,FAF8,0010,19DC   ;35861  J7MULG.03
                                ;35862
                                ;35863  =0*     ;0*--------------------------------;  CALL SITE FOR MULTIPLY ROUTINE
                                ;35864  MULG.03:LC_RC[T2],                       ;  LATCH M'CAND FRAC <LS>
                                ;35865          SC_K[.F],                        ;  LOAD LOOP COUNT FOR 16 PASSES
                                ;35866          CAL, Z?,                         ;  CALL WHILE BRANCHING ON M'CAND
U 19DC, 0000,013D,6180,F910,0084,70E8   ;35867  J/MULDMPY                        ;  USE MULTIPLY DOUBLE ROUTINE
                                ;35868
                                ;35869  ;RETURN WITH:
                                ;35870  ;       <Q,D> = NEWLY FORMED PRODUCT
                                ;35871  ;       NOTICE THAT WE NEED ONLY HIGH BIT OF D (=LOW BIT OF MIDDLE PRODUCT)
                                ;35872
                                ;35873          ;1*--------------------------------;  RETURN2
                                ;35874          ALU_D, N_AMX.Z_TST,              ;  STORE LOW BIT IN PSL<N>
                                ;35875          D_Q,                             ;  D GETS PROD <LS>
                                ;35876          Q_ID[T0],                        ;  Q GETS M'PLIER <MS>
U 19DE, 0C01,003C,C1F0,2D10,0030,19E8   ;35877  LC_RC[T2]                         ;  LATCH M'CAND <LS>
                                ;35878  =;END
                                ;35879  =0*     ;0*--------------------------------;  CALL SITE FOR MULTIPLY ROUTINE
                                ;35880          D_Q,                             ;  D GETS M'PLIER <MS>
                                ;35881          Q_D,                             ;  Q GETS <MS> OF LOW PRODUCT
                                ;35882          SC_K[.F],                        ;  SET LOOP COUNT TO 16 PASSES
U 19E8, 0C00,013D,61E0,F800,0084,70E8   ;35883  CAL, Z?, J/MULDMPY               ;  CALL MULTIPLY LOOP
                                ;35884                                           ;  STILL BRANCHING ON M'CAND<LS>
                                ;35885  ;RETURN WITH:
                                ;35886  ;       <Q,D> = NEWLY FORMED PRODUCT (A MIDDLE PRODUCT)
                                ;35887
                                ;35888          ;1*--------------------------------;  RETURN2
U 19EA, 0C00,003C,01E0,F800,0000,1B50   ;35889  D_Q, Q_D                          ;  PREPARE FOR LEFT SHIFT
                                ;35890  =;END
                                ;35891          ;--------------------------------;
                                ;35892          ALU_Q, RC[T6]_ALU.LEFT,          ;  SHIFT DATA TO MAKE
                                ;35893          D_D.LEFT, SI/DIVD,               ;  <D,RC6> = 64 <LS> BITS OF
U 1B50, 0521,203C,C470,2DB0,0000,1B51   ;35894  Q_ID[T1]
                                ;35895                                           ;  M'CAND<LS>xM'PLIER
                                ;35896          ;--------------------------------;
                                ;35897          R[R15]_Q,                        ;  STORE M'CAND <MS> IN R15
U 1B51, 0001,203C,CD80,3EF8,0000,1B52   ;35898  ID[T3]_D                          ;  SAVE PROD <LS> FOR LATER ADD IN T3
                                ;35899
                                ;35900          ;--------------------------------;
                                ;35901          RC[T0]_Q.LEFT,                   ;  RC0 GETS M'CAND <MS> *2
                                ;35902          SI/ZERO,
U 1B52, 0021,203C,C9F0,2D80,0000,1A1C   ;35903  Q_ID[T2]                          ;  GET M'PLIER <LS>
                                ;35904          ;--------------------------------;
```

```
                                ;35905  =0*      ;0*-------------------------------;  CALL SITE FOR MULTIPLY ROUTINE
                                ;35906          D_Q, Q_0,                         ;  D GETS M'PLIER <LS>
                                ;35907          LC_RC[T0],                        ;  LATCH 2*M'CAND IN LC
                                ;35908          SC_K[.F],                         ;  LOOP COUNT GETS 16.
U 1A1C, 0C00,003D,61F8,F900,0084,70E8 ;35909    CALL, J/MULDMPY                   ;  CALL MULTIPLY LOOP
                                ;35910
                                ;35911  ;RETURN WITH:
                                ;35912  ;       <Q,D> = NEWLY FORMED PRODUCT (ANOTHER MIDDLE PRODUCT)
                                ;35913
                                ;35914          ;1*-------------------------------;  RETURN2
                                ;35915          D_D+LC,                           ;  ADD [DST<LS>XSRC<MS>]<LS> TO
U 1A1E, 0811,0014,0180,F800,0010,1B53 ;35916    C[K.UBCC                          ;  [DST<MS>XSRC<LS>]<LS>
                                ;35917                                            ;  AND CLOCK CARRY
                                ;35918  =;END
                                ;35919          ;-------------------------------;
                                ;35920          ALU_D, N_AMX.Z_TST,               ;  SAVE CURRENT LOW BIT IN PSL<N>
                                ;35921          D_Q,                              ;  D GETS M'PLIER <MS>
                                ;35922          Q_ID[T3],                         ;  Q GETS PROD<LS>
U 1B53, 0C01,033C,CDF0,2C00,0030,1A5C ;35923    C31?                              ;  TEST CARRY FROM PREVIOUS ADD
                                ;35924
                                ;35925          ;-------------------------------;
                                ;35926  =C*     ;BRANCH ON ALU C31
                                ;35927          ;0*-------------------------------;  NO CARRY
                                ;35928          ALU_D+Q, RC[T6]_ALU,              ;  ADD AND SAVE RESULT IN RC6
                                ;35929          CLK.UBCC,LONG,                    ;  WE MIGHT HAVE A CARRY HERE
                                ;35930          Q_ID[T0],                         ;  Q GETS M'PLIER <MS>
U 1A5C, 001D,0014,C1F0,2DB0,0010,1A68 ;35931    J/MULG.04
                                ;35932
                                ;35933          ;1*-------------------------------;  CARRY
                                ;35934          ALU_D+Q+1, RC[T6]_ALU,            ;  ADD AND SAVE RESULT IN RC6
                                ;35935          CLK.UBCC,LONG,                    ;  WE MIGHT HAVE A CARRY HERE
                                ;35936          Q_ID[T0],                         ;  Q GETS M'PLIER <MS>
U 1A5E, 001D,0010,C1F0,2DB0,0010,1A68 ;35937    J/MULG.04
                                ;35938  =;END
                                ;35939  =0*     ;0*-------------------------------;  CALL SITE FOR MULTIPLY ROUTINE
                                ;35940  MULG.04:LC_RC[T0],                        ;  LATCH UP 2*M'CAND<MS>
                                ;35941          D_Q,                              ;  D GETS M'PLIER <MS>
                                ;35942          Q_0,
                                ;35943          SC_K[.F],                         ;  LOOP COUNT IS SET TO 16 PASSES
U 1A68, 0C00,003D,61F8,F900,0084,70E8 ;35944    CALL, J/MULDMPY
                                ;35945
                                ;35946          ;1*-------------------------------;  RETURN2
                                ;35947          SC_K[.FFF5],                      ;  GET SC READY FOR PACKG-ROUTINE
                                ;35948          Q_D+LC, CLK.UBCC,                 ;  ADD RESULTS TOGETHER
                                ;35949                                            ;  CLOCK CARRY
U 1A6A, 0C11,0314,E1C0,F800,0094,7A69 ;35950    D_Q,C31?                          ;  WAS THERE AN OLD CARRY ?
                                ;35951  =;END
```

```
                                ;35952          ;--------------------------------;
                                ;35953  =0*     ;BRANCH ON ALU C31
                                ;35954          ;0*--------------------------------;  NO CARRY
                                ;35955          LC_RC[T4],                        ;  LATCH PRODUCT EXPONENT
U 1A69, 0000,033C,0180,F920,0000,1A70   ;35956  C3T?,J/MULG.05                    ;  TEST CARRY
                                ;35957
                                ;35958          ;--------------------------------;
                                ;35959          LC_RC[T4],                        ;  LATCH PRODUCT EXPONENT
                                ;35960          D_D+K[.1],                        ;  ADD IN PREVIOUS CARRY
U 1A6B, 0819,0314,0580,F920,0000,1A70   ;35961  C31?,J/MULG.05                    ;  TEST CARRY
                                ;35962  =;END
                                ;35963          ;--------------------------------;
                                ;35964  =0*     ;BRANCH ON ALU C31
                                ;35965          ;0*--------------------------------;  NO CARRY
                                ;35966  MULG.05:ALU_Q, Q_ALU.LEFT,                ;  SHIFT <D,Q,PSL<N>> LEFT ONCE
                                ;35967          D_D.LEFT, SJ/DIVD,
                                ;35968          FE_SHF.VAL,                       ;  NEED IT TO CLR STATE0 IF FRACT<.5
U 1A70, 0521,203C,0040,F800,010C,7B54   ;35969  J/MULG.51
                                ;35970
                                ;35971          ;1*--------------------------------;  CARRY
U 1A72, 0819,0014,0580,F800,0000,1A70   ;35972  D_D+K[.1], J/MULG.05               ;  ADD IN CARRY TO PROD<MS>
                                ;35973
                                ;35974  MULG.51:;--------------------------------;
                                ;35975          ID[T9]_D,                         ;  NEED D TEMPORARILY
U 1B54, 0010,0038,E580,3EF8,0000,1B56   ;35976  R[R15]_LC                         ;  R15 GETS EXPONENT
                                ;35977
                                ;35978          ;--------------------------------;
                                ;35979          LAB_R[R15],                       ;  LATCH EXPONENT
U 1B56, 0818,0038,8180,FA78,0000,1B57   ;35980  D_K[.3FF]                         ;  D GETS .400-1 (=BIAS-1)
                                ;35981
                                ;35982          ;--------------------------------;
                                ;35983          R[R15]_LA-D,                      ;  GET BIASED EXPONENT
                                ;35984          D_Q, Q_ID[T9],                    ;  RESTORE PREVIOUS D AND Q
                                ;35985          STATE_STATE.ANDNOT.FE,            ;  MAYBE CLEAR STATE0
U 1B57, 0C1C,3700,E5F0,2EF8,1400,586A   ;35986  STATE0?                           ;  SEPARATE MULG FROM EMODG/POLYG
                                ;35987
                                ;35988          ;--------------------------------;
                                ;35989  =**10   ;BRANCH ON LOW BIT OF STATE
                                ;35990          ;**10------------------------------;  MULG
                                ;35991          D_Q,Q_D,                          ;  GET FRACTION IN ORDER
U 186A, 0C00,003C,7DE0,F800,0104,7B58   ;35992  FE_K[.18],J/MULG.07
                                ;35993
                                ;35994          ;**11------------------------------;  POLYG/EMODG
                                ;35995          LAB_R[R15],                       ;  LATCH UP EXPONENT
U 186B, 0C00,173E,01E0,FA78,0000,0012   ;35996  D_Q,Q_D, STATE0?, RETURN12
                                ;35997
                                ;35998  MULG.07:;--------------------------------;
                                ;35999          LAB_R[R15],                       ;  LATCH EXPONENT
U 1B58, 0000,0D3C,8180,FA78,0000,1886   ;36000  K[.3FF], D31?                     ;  GET READY TO ROUND, NORMALIZED?
                                ;36001
```

```
                                    ;36002          ;--------------------------------;
                                    ;36003  =110    ;BRANCH ON D31 (NORMALIZED?)
                                    ;36004          ;110-----------------------------; NOT NORMALIZED
                                    ;36005          D_D.LEFT,Q_Q.LEFT,                ; NORMALIZE
                                    ;36006          SI/DIVD,
                                    ;36007          R[R15]_LA-K[.1],                  ; DECREMENT EXPONENT
U 1886, 0518,0000,0428,FAF8,0000,1858 ;36008        J/MULG.07
                                    ;36009
                                    ;36010          ;------- ----------------------;
                                    ;36011          Q_Q+K[.3FF]+1,                    ; ROUND FRAC <LS>
                                    ;36012          C[K.UBCC,                         ; CLOCK CARRY
U 1887, 0019,2010,81C0,F800,0010,1889 ;36013        J/PACKG                           ; PACK UP RESULT
                                    ;36014          ;--------------------------------;
```

```
                                ;36015  .TOC    ""      G & H floating point  : GRAND FLOATING POLY''
                                ;36016
                                ;36017  ;       GRAND FLOATING POINT POLYNOMIAL EVALUATION.
                                ;36018  ;INPUTS:
                                ;36019  ;               RC0 = ARG LONG EXP
                                ;36020  ;               Q = ARG LONG FRAC
                                ;36021  ;               D = DEGREE
                                ;36022
                                ;36023  ;OUTPUTS:
                                ;36024  ;               R0 = RESULT LONG EXP
                                ;36025  ;               R1 = RESULT LONG FRAC
                                ;36026  ;               R2 = 0
                                ;36027  ;               R3 = TABLE ADDRESS + DEG*8 + 8
                                ;36028  ;               R4 = 0
                                ;36029  ;               R5 = 0
                                ;36030
                                ;36031  ;TEMPORARIES:
                                ;36032  ;               ID[T0] = DEGREE
                                ;36033  ;               ID[T1] =
                                ;36034  ;               ID[T2] =
                                ;36035  ;               ID[T3] =
                                ;36036  ;               ID[T4] =
                                ;36037  14C2:   ;-------------------------------;
U 14C2, 0000,11BC,0080,F800,0000,1AA7   ;36038          TRAP.ACC[1],FPD?                ;  RESET THE FPA
                                ;36039
                                ;36040  =0111
                                ;36041  POLYG.FPD:
                                ;36042          ;0111---------------------------;  FPD IS SET
                                ;36043          D_R[R2],Q_0,                    ;  SETUP TO GET PC DELTA
                                ;36044          SC_K[.FFF8],                    ;  GET READY TO SHIFT IT
U 1AA7, 0800,003C,71F8,FA10,0084,7B59   ;36045          J/POLYG.FPD.1
                                ;36046
                                ;36047  POLYG:  ;1111---------------------------;  FPD NOT SET
                                ;36048          D_D.OXT[WORD],                  ;  GET DEGREE
                                ;36049          STATE_K[.1],                    ;  SET POLYG FLAG IN STATE
U 1AAF, 0803,403C,0580,F800,1404,7B5B   ;36050          J/POLYG.0
                                ;36051
                                ;36052  POLYG.FPD.1:
                                ;36053          ;-------------------------------;
U 1B59, 0D00,003C,0180,F800,0000,1B5A   ;36054          D_DAL.SC                        ;  PC DELTA NOW IN D<07:00>
                                ;36055
                                ;36056          ;-------------------------------;
U 1B5A, 0819,0000,0580,F800,0000,1908   ;36057          D_D-K[.1]                       ;  ADJUST FOR THE ESCD
                                ;36058
                                ;36059  =00     ;00---------------------------;
                                ;36060          PC&VA_D.OXT[BYTE]+PC,           ;  BYPASS SPECIFIERS
U 1908, 0017,8015,0180,F801,0200,0E16   ;36061          CALL, J/SETFPD
                                ;36062
                                ;36063  =10     ;10---------------------------;
U 190A, 0014,0038,01C0,F800,0000,0EB8   ;36064          Q_PC, J/BAKUP.PC                ;  MEM MGMT CODE HERE ON READ ERRORS
                                ;36065
                                ;36066  =11     ;11---------------------------;
U 190B, 0000,003C,0180,FA18,0200,1896   ;36067          VA_R[R3], J/POLYG.9             ;  VA GETS TABLE ADDRESS
```

```
                                ;36068          ;--------------------------------;
                                ;36069  POLYG.0:
                                ;36070          LC_RC[TO],                      ;  LATCH ARG LONG EXP
                                ;36071          AL[]_D.ANDNOT.K[.1F],           ;  CHECK IF DEGREE > 31
U 1B5B, 0019,0024,8D80,F900,0010,1B5C   ;36072          CLK.UBCC                        ;  CLOCK ALU Z-BIT
                                ;36073
                                ;36074          ;--------------------------------;
                                ;36075          ID[TO]_D,                       ;  SAVE DEGREE IN TO
                                ;36076          D_LC,                           ;  D GETS ARG LONG EXP
                                ;36077          INTRPT.STROBE,                  ;  STROBE FOR INTERRUPTS
U 1B5C, 0810,0138,C180,3C00,4000,1978   ;36078          Z?                              ;  TEST FOR RESERVED DEGREE
                                ;36079
                                ;36080          ;--------------------------------;
                                ;36081  =0       ;BRANCH ON ALU Z-BIT
                                ;36082          ;0-------------------------------;
U 1978, 0000,003C,0180,F800,0000,0106   ;36083          J/RSVOPR                        ;  RESERVED OPERAND
                                ;36084
                                ;36085          ;1-------------------------------;
U 1979, 0019,0034,3980,F800,0010,1B5E   ;36086          ALU_D.AND.K[.7FF0],CLK.UBCC     ;  SEE IF ARG EXP IS ZERO
                                ;36087
                                ;36088          ;--------------------------------;
U 1B5E, 0000,013C,0180,F800,0000,1984   ;36089          Z?                              ;
                                ;36090
                                ;36091  =0       ;0-------------------------------;
U 1984, 0000,003C,0180,F800,0000,1824   ;36092          J/POLYG.A1                      ;  EXP NOT ZERO,CARRY  ON AS USUAL
                                ;36093
                                ;36094          ;1-------------------------------;
U 1985, 0001,003C,0180,F800,0800,1824   ;36095          ALU_D,CHK.FLT.OPR               ;  EXP=0,CHECK FOR RES OP
                                ;36096
                                ;36097  =01*****;--------------------------------;
                                ;36098  POLYG.A1:ID[T2]_D,                      ;  SAVE ARG LONG EXP IN T2
                                ;36099          RC[T2]_Q,                       ;  SAVE ARG LONG FRAC IN RC2
                                ;36100          INTERRUPT.REQ?,                 ;  TEST FOR INTERRUPTS
U 1824, 0001,2E3D,C980,3D90,0000,047E   ;36101          CALL,J/ASPC                     ;  EVALUATE TABLE ADDRESS
                                ;36102
                                ;36103          ;--------------------------------;
                                ;36104  =11*****                                ;  RETURN WITH TABLE ADDR IN D
                                ;36105          R[R15]_D,                       ;  SAVE TABLE ADDRESS IN R15
U 1864, 0001,003C,0180,42F8,0000,1B60   ;36106          D[LONG]_CACHE                   ;  GET FIRST COEFFICIENT LONG EXP
                                ;36107  =;END
                                ;36108          ;--------------------------------;
                                ;36109          ALU_D.AND.K[.7FF0],CLK.UBCC,    ;  CLOCK EXPONENT
                                ;36110          LAB_R[R15],                     ;  LATCH TABLE ADDRESS IN LAB
U 1B60, 0019,0034,3980,FA7B,C010,1B61   ;36111          VA_VA+4                         ;  VA POINTS TO C0 LONG FRAC
                                ;36112
                                ;36113          ;--------------------------------;
                                ;36114          ID[T1]_D,                       ;  STORE C0 LONG EXP IN T1
U 1B61, 0000,013C,C580,3C00,0000,1918   ;36115          Z?                              ;  TEST FOR 0
```

```
                                ;36116          ;00------------------------------;
                                ;36117  =00
                                ;36118  POLYG.1:
                                ;36119          D[LONG] CACHE,                  ;  READ CO LONG FRAC
                                ;36120          LC_RC[T0],                      ;  LATCH ARG LONG EXP
U 1918, 0000,003D,0180,4100,0000,0E16   ;36121          CALL,J/SETFPD
                                ;36122
                                ;36123          ;01------------------------------;
                                ;36124          ALU_D,CHK.FLT.OPR,              ;  CHECK FOR -0
U 1919, 0001,003C,0180,F8C0,0800,1918   ;36125          J/POLYG.1
                                ;36126
                                ;36127          ;10------------------------------;
U 191A, 0014,0038,01C0,F800,0000,0EB8   ;36128  =10      Q_PC, J/BAKUP.PC                ;  READ ERROR, BACK UP PC
                                ;36129
                                ;36130          ;11------------------------------;
                                ;36131  =11      ID[T4]_D,                       ;  SAVE CO LONG FRAC IN T4
U 191B, 0010,0038,D180,3EA0,0000,1B62   ;36132          R[R4]_LC                        ;  STORE ARG LONG EXP IN R4
                                ;36133  =;END
                                ;36134          ;------------------------------;
U 1B62, 0000,003C,0180,FA98,0000,1988   ;36135          R[R3]_LA                        ;  R3 GETS THE TABLE ADDRESS
                                ;36136
                                ;36137  =0       ;0------------------------------;
                                ;36138          ALU_0(A),                       ;  CLEAR RUNNING PRODUCT <LS>
                                ;36139          LC_RC[T2]&R1_ALU,               ;  STORE IT IN R1
                                ;36140                                          ;  LATCH ARG LONG FRAC IN LC
U 1988, 0003,003D,0180,FB90,0000,13E8   ;36141          CALL,J/POLY.PC                  ;  GO GET PC-DELTA
                                ;36142
                                ;36143          ;1-------------------------------;
                                ;36144          R[R2]_D.OR.Q,                   ;  R2 STORES PC DELTA, DEGREE
U 1989, 001D,0030,C5F0,2E90,0000,1B64   ;36145          Q_ID[T1]                        ;  Q GETS CO LONG EXP
                                ;36146  =;END
                                ;36147          ;------------------------------;
                                ;36148          R[R5]_LC,                       ;  STORE ARG LONG FRAC IN R5
                                ;36149          D_Q,                            ;  Q GETS CO LONG EXP
U 1B64, 0C10,0038,C1F0,2EA8,0000,1B65   ;36150          Q_ID[T0]                        ;  Q GETS DEGREE
                                ;36151
                                ;36152          ;------------------------------;
                                ;36153          ALU_D.AND.K[.7FF0], CLK.UBCC,   ;  CLOCK CO EXPONENT
U 1B65, 0019,4034,3980,F800,0010,1B66   ;36154          WORD
                                ;36155
                                ;36156          ;------------------------------;
                                ;36157          R[R0]_D,                        ;  STORE CO LONG EXP IN R0
                                ;36158          D_Q,                            ;  D GETS DEGREE
U 1B66, 0C01,003C,D1F0,2E80,0000,1B68   ;36159          Q_ID[T4]                        ;  Q GETS CO LONG FRAC
                                ;36160
                                ;36161          ;------------------------------;
                                ;36162          R[R1]_Q,                        ;  STORE CO LONG FRAC AS SUM LONG FRAC
U 1B68, 0101,203C,0180,FA88,0000,1B69   ;36163          D_D.LEFT2, SI/ZERO              ;  D GETS DEGREE * 4
                                ;36164
```

```
                                ;36165          ;--------------------------------;
                                ;36166          R[R3]&VA_LA+K[.8],              ;  SET UP TABLE ADDRESS IN R3 AND VA
                                ;36167          D_D.LEFT, SI/ZERO,              ;  D GETS DEGREE * 8
                                ;36168          SGN/CLR.SD+SS,                  ;  CLEAR SS FOR BRANCH
U 1B69, 0518,0D14,0187,FA98,0200,18F4   ;36169          D.NE.0?                         ;  DEGREE NE 0 ?
                                ;36170
                                ;36171          ;--------------------------------;
                                ;36172  =10*    ;BRANCH ON D.NE.0
                                ;36173          ;10*-------------------------------;  DEGREE EQ 0
U 18F4, 0000,013C,0180,F800,0000,1998   ;36174          Z?, J/POLYG.4                   ;  TEST C0 = 0
                                ;36175
                                ;36176          ;11*-------------------------------;
                                ;36177          ALU_R[R4].AND.K[.7FF0],
U 18F6, 0018,0034,3980,FA20,0010,1B6A   ;36178          CLK.UBCC                        ;  CLOCK EXPONENT OF ARGUMENT
                                ;36179
                                ;36180          ;--------------------------------;
   1B6A, 0000,013C,0180,FA78,0000,1994  ;36181          LAB_R[R15],Z?                   ;  TEST ARGUMENT FOR 0
                                ;36182
                                ;36183          ;--------------------------------;
                                ;36184  =0       ;BRANCH ON ALU Z-BIT
                                ;36185          ;0-------------------------------;  ARG NE 0
U 19 4, 0800,003C,0180,FA20,0000,1B71   ;36186          D_R[R4], J/POLYG.10             ;  D GETS ARG LONG EXP
                                ;36187
                                ;36188          ;1-------------------------------;  ARG = 0
U 199 , 001C,2014,0180,FA98,0200,1B6C   ;36189          R[R3]_LA+D,  VA_ALU             ;  ADVANCE TABLE TO LAST COEFF
                                ;36190  =;END
                                ;36191          ;--------------------------------;
                                ;36192          ALU_R[R4],CHK.FLT.OPR,          ;  CHECK FOR RESERVED OPERAND
U 1B6C, 0000,003C,0180,FA20,0800,1B70   ;36193          J/POLYG.8                       ;  BY ADDING 8*DEGREE
                                ;36194
                                ;36195          ;--------------------------------;
                                ;36196  =0       ;BRANCH ON ALU Z-BIT
                                ;36197          ;0-------------------------------;
                                ;36198  POLYG.4:                                 ;  DEGREE EQ 0, C0 NE 0
                                ;36199          R[R4]_0,                        ;  CLEAR R4
U 1998, 0003,003C,0180,FAA0,0000,19AD   ;36200          J/POLYG.35                      ;  AND EXIT
                                ;36201
                                ;36202          ;1-------------------------------;
U 1999, 0003,003C,0180,FA80,0000,1B6D   ;36203          R[R0]_0                         ;  C0 EQ 0, DEGREE EQ 0
                                ;36204
                                ;36205          ;--------------------------------;
U 1B6D, 0003,003C,0180,FA88,0000,1998   ;36206          R[R1]_0, J/POLYG.4              ;  CLEAR RESULT
                                ;36207
                                ;36208          ;--------------------------------;
                                ;36209  POLYG.8:
U 1B70, 0018,8038,0580,FA90,0000,1896   ;36210          R[R2]_K[.1], BYTE               ;  ARG =0, SET DEGREE TO ONE FOR
                                ;36211                                          ;  SINGLE PASS THRU LOOP
                                ;36212          ;--------------------------------;
```

```
                                ;36213
                                ;36214          ;POLYG LOOP BEGINS HERE
                                ;36215  =110    ;BRANCH ON PENDING INTERRUPTS
                                ;36216          ;-------------------------------;
                                ;36217  POLYG.9:
U 1896, 0800,003C,0180,FA20,0000,1B71   ;36218          D_R[R4], J/POLYG.10             ;  GET ARG LONG EXP
                                ;36219
                                ;36220          ;-------------------------------;
U 1897, 0014,0038,01C0,F800,0000,1A71   ;36221          Q_PC                            ;  BAKUP.PC WANTS PC IN Q
                                ;36222  =;END
                                ;36223          ;0*-------------------------------;
U 1A71, 0000,003D,0180,F800,0000,0EB8   ;36224  =0*     CALL,J/BAKUP.PC                 ;  GO TO BACK UP PC
                                ;36225
                                ;36226          ;1*-------------------------------;
U 1A73, 0000,003C,0180,F800,0000,04FA   ;36227          J/INT.I                         ;  TAKE PENDING INTERRUPT
                                ;36228
                                ;36229          ;-------------------------------;
                                ;36230  POLYG.10:
U 1B71, 0001,003C,0180,F980,0000,1B72   ;36231          RC[T0]_D                        ;  RC0 GETS ARG LONG EXP
                                ;36232
                                ;36233          ;-------------------------------;
U 1B72, 0000,003C,01C0,FA28,0000,1B73   ;36234          Q_R[R5]                         ;  Q GETS ARG LONG FRAC
                                ;36235
                                ;36236          ;-------------------------------;
U 1B73, 0800,003C,0180,FA00,0000,1B74   ;36237          D_R[R0]                         ;  D GETS PARTIAL POLY. LONG EXP
                                ;36238
                                ;36239          ;-------------------------------;
                                ;36240          ALU_D, STATE_K[.1],             ;  SET POLYG FLAG FOR RETURN
U 1B74, 0001,003C,0580,FB10,1404,7820   ;36241          LAB_R1&RC[T2]_ALU               ;  RC1 GETS PARTIAL POLY. LONG EXP
                                ;36242
                                ;36243          ;0**00----------------------;
                                ;36244  =0**00
                                ;36245          D_Q, SC_K[.10],                 ;  D GETS ARG LONG FRAC
                                ;36246          RC[T6]_LA,                      ;  SAVE ARG LONG FRAC IN T6
U 1820, 0C00,003D,6580,F9B0,0084,78C8   ;36247          CALL, J/MULG.02                 ;  CALL MULTIPLY ROUTINE
                                ;36248
                                ;36249          ;1**00----------------------;  POLYNOMIAL EQ 0
                                ;36250  =1**00  ;  RETURN WITH VA POINTING TO COEFF.
                                ;36251          D[LONG]_CACHE,                  ;  GET NEXT COEF LONG EXP
                                ;36252          LAB_R[R3],                      ;  LATCH TABLE ADDRESS
U 1830, 0000,003C,0180,4218,0000,1B77   ;36253          J/POLYG.12
                                ;36254
                                ;36255          ;1**10----------------------;
                                ;36256  =1**10                                  ;  .25 <= PARTIAL POLY FRACTION < .5
                                ;36257          D_D.LEFT, SI/DIVD,              ;  NORMALIZE
                                ;36258          R[R15]_LA-K[.1],                ;  DECREMENT EXPONENT
U 1832, 0518,0000,0400,FAF8,0000,1B76   ;36259          J/POLYG.11
                                ;36260
                                ;36261          ;1**11----------------------;
                                ;36262  =1**11                                  ;  POLY FRACTION IS NORMALIZED
                                ;36263          ID[T0]_D,                       ;  SAVE POLY FRAC <MS> IN T0
                                ;36264          LAB_R[R15],
                                ;36265          D_Q, Q_D,                       ;  SWAP HALVES TO SAVE <LS> PART
U 1833, 0C00,003C,C1E0,3E78,0000,1B78   ;36266          J/POLYG.13
                                ;36267  =;END   ;-------------------------------;
```

```
                                ;36268  POLYG.11:
                                ;36269          ;-------------------------------;
                                ;36270          ID[TO]_D,                       ;  SAVE POLY FRAC <MS> IN TO
                                ;36271          LAB_R[R15],                     ;  LATCH EXPONENT
                                ;36272          ALU_Q,D_ALU.LEFT,Q_D,           ;  SWAP HALVES TO SAVE <LS> PART
U 1B76, 0821,203C,C1E0,3E78,0000,1B7B   ;36273          J/POLYG.13
                                ;36274
                                ;36275          ;-------------------------------;
                                ;36276  POLYG.12:                               ;  PARTIAL POLY IS 0
                                ;36277          ALU_D.AND.K[.7FF0],             ;  CLOCK EXP OF NEXT COEFF
                                ;36278          CLK.UBCC,
                                ;36279          Q_D,                            ;  Q GETS COEFF LONG EXP
U 1B77, 0019,0034,39E0,F803,0010,1B78   ;36280          VA_VA+4                         ;  VA NOW POINTS TO COEFF LONG FRAC
                                ;36281
                                ;36282          ;-------------------------------;
                                ;36283          D[LONG]_CACHE,                  ;  READ NEXT COEFF LONG FRAC
U 1B78, 0000,013C,0180,4000,0000,19A4   ;36284          Z?                              ;  IS COEFF = 0 ?
                                ;36285
                                ;36286          ;-------------------------------;
                                ;36287  =0       ;BRANCH ON ALU Z-BIT
                                ;36288          ;0------------------------------;  NOT 0
                                ;36289          R[R1]_D,                        ;  STORE COEFF LONG FRAC
                                ;36290          D_Q,Q_D,                        ;  MAIN FLOW EXPECTS THEM THIS WAY
U 19A4, 0C01,003C,01E0,FA88,0000,1B7A   ;36291          J7POLYG.12A
                                ;36292
                                ;36293          ;1------------------------------;
                                ;36294          ALU_Q.AND.K[.8000],CHK.FLT.OPR, ;  CHECK FOR RESERVED OP
U 19A5, 0F19,2034,45F8,F800,0800,1B79   ;36295          Q_0,D_0                         ;  COEFF IS TRUE 0
                                ;36296
                                ;36297          ;-------------------------------;
                                ;36298          R[R0]_K[ZERO],                  ;  ZERO PARTIAL PRODUCT
U 1B79, 0018,0038,1980,FA80,0000,1B7A   ;36299          J/POLYG.12A
                                ;36300  =;END
                                ;36301          ;-------------------------------;
                                ;36302  POLYG.12A:
U 1B7A, 0018,0014,0180,FA98,0000,1B89   ;36303          R[R3]_LA+K[.8],J/POLYG.22       ;  INC TABLE ADDRESS
```

```
                                ;36304  POLYG.13:
                                ;36305          ;-------------------------------;
                                ;36306          ID[T2]_D,                       ;  SAVE POLY FRAC <LS> IN T2
                                ;36307          ALU_Q_D_ALU.LEFT, SI/ZERO,      ;  D GETS NORMALIZED POLY FRAC <MS>
                                ;36308          LAB_R[R15],                     ;  LATCH POLYNOMIAL EXPONENT
U 1B7B, 0821,203C,C980,3E78,0000,1B7C   ;36309          J/POLYG.14
                                ;36310
                                ;36311          ;-------------------------------;
                                ;36312  POLYG.14:
                                ;36313  ;       RC[TO] GETS PACKED EXPONENT
                                ;36314          ;-------------------------------;
                                ;36315          ALU_LA-K[.1], RC[T5]_ALU,       ;  SAVE POLY EXP IN RC5
U 1B7C, 0018,0000,0580,41A8,0010,1B7E   ;36316          CLK.UBCC,D[LONG]_CACHE          ;  READ NEXT COEFF LONG EXP
                                ;36317
                                ;36318          ;-------------------------------;
                                ;36319          RC[T1]_D,                       ;  SAVE IT IN RC1
U 1B7E, 0001,003C,01E0,F98B,0000,1B80   ;36320          Q_D, VA_VA+4                    ;  AND IN Q
                                ;36321
                                ;36322          ;-------------------------------;
                                ;36323          LAB_R[R3],                      ;  LATCH COEFF TABLE ADDRESS
U 1B80, 0000,003C,0180,4218,0000,1938   ;36324          D[LONG]_CACHE                   ;  READ COEFF LONG FRAC
                                ;36325
                                ;36326          ;-------------------------------;
                                ;36327  =00                                     ;  CALL CONSTRAINT BLOCK
                                ;36328  POLYG.14A:STATE_STATE.OR.K[.1],         ;  SET POLYG FLAG AGAIN FOR ADDG
                                ;36329          RC[T6]_D,                       ;  SAVE COEFF LONG FRAC IN RC6
U 1938, 0001,003D,0580,F9B0,1404,3B93   ;36330          CALL, J/UNPKG                   ;  UNPACK COEF
                                ;36331
                                ;36332          ;-------------------------------;
                                ;36333          SD_SS, Q_ID[T2],                ;  GET POLY FRAC <LS>
U 1939, 0000,003C,C9F4,2C00,0000,1B84   ;36334          J/POLYG.T6
                                ;36335
                                ;36336          ;-------------------------------;  COEFF NE 0
                                ;36337  ;       D = COEFF FRAC <LS>
                                ;36338  ;       RC4 = COEFF EXP
                                ;36339  ;       RC5 = POLY EXP
                                ;36340  ;       RC6 = COEFF LONG FRAC
                                ;36341  ;       R[R15] = COEFF EXP-POLY EXP
                                ;36342  ;       ID[T1] = COEFF FRAC <MS>
                                ;36343  ;       ALU CC REFLECT R15
                                ;36344  ;       SC = LOW BITS OF EXP DIFF
                                ;36345  ;       FE = 0
                                ;36346
                                ;36347          LAB_R[R3],                      ;  LATCH TABLE ADDRESS
U 193A, 0000,003C,0180,FA18,0000,1B81   ;36348          J/POLYG.140
                                ;36349  =;END
                                ;36350          ;-------------------------------;
                                ;36351  POLYG.140:
U 1B81, 0018,0014,0180,FA98,0000,1B82   ;36352          R[R3]_LA+K[.8]                  ;  INC TABLE ADDRESS
                                ;36353
                                ;36354          ;-------------------------------;
U 1B82, 0000,003C,0180,FA78,0000,18C3   ;36355          LAB_R[R15]                      ;  LATCH EXPONENT DIFFERENCE
```

```
                                ;36356  =00011  ;00011-------------------------;  CALL SITE FOR ADDG-ROUTINE
                                ;36357          Q_ID[T2],                       ;  Q GETS RUNNING POLY FRAC <LS>
                                ;36358          LC_RC[T4],                      ;  LATCH POLY EXPONENT
                                ;36359          FE_NABS(SC-FE),                 ;  FE GETS NEGATIVE EXP DIFF
                                ;36360          ALU?,                           ;  TEST EXPONENT DIFFERENCE
U 18C3, 0000,1B3D,C9F0,2D20,0100,F9E3   ;36361          CALL,J/ADDG.10
                                ;36362
                                ;36363  =10011  ;10011-------------------------;  RETURN FROM ADDG THRU PACKG
U 18D3, 0001,003C,E5F0,2E80,0000,1B8A   ;36364          Q_ID[T9],R[RO]_D,J/POLYG.22A    ;  NORMAL FLOW,STORE NEW ANSWER AND CON
                                ;36365
                                ;36366          ;10111-------------------------;  EXPONENT WAS ZERO,UNDERFLOW
U 18D7, 0000,003C,0180,F800,0000,1AB5   ;36367          J/POLYG.17                      ;  GO TEST PSL<FU> SET?
                                ;36368
                                ;36369          ;11011-------------------------;  BIT 15 WAS SET
U 18DB, 0000,0D3C,E5F0,2C00,0000,1933   ;36370          Q_ID[T9],Q31?                   ;  OVERFLOW OR UNDERFLOW?
                                ;36371  =;END
                                ;36372          ;---------------------------------;
                                ;36373  =011    ;       BRANCH ON Q 31            ;
                                ;36374  POLYG.15:;---------------------------------;  OVERFLOW
U 1933, 0818,0038,0180,FA18,0000,13F1   ;36375          D_K[.8],LAB_R[R3],J/POLYD.FAULT ;  TAKE FAULT
                                ;36376
                                ;36377          ;---------------------------------;  UNDERFLOW
U 1937, 0000,003C,0180,F800,0000,1AB5   ;36378          J/POLYG.17                      ;
                                ;36379
                                ;36380          ;---------------------------------;
                                ;36381  POLYG.16:
                                ;36382          D_Q,LAB_R[R3],                  ;  D GETS RESULT FRAC <MS>
U 1B84, 0C00,003C,C1F0,2E18,0000,1B85   ;36383          Q_ID[T0]                        ;  Q GETS RESULT FRAC <LS>
                                ;36384
                                ;36385          ;---------------------------------;
U 1B85, 0018,0014,0180,FA98,0000,1B86   ;36386          R[R3]_LA+K[.8]                  ;  INC TABLE ADDRESS
                                ;36387
                                ;36388          ;---------------------------------;
                                ;36389          D_D.LEFT, Q_Q.LEFT,             ;  DOUBLE SHIFT FRACTION PART
U 1B86, 0500,003C,8128,F800,0000,1B88   ;36390          SI/ASHL, K[.3FF]                ;  GET READY TO ROUND
                                ;36391
                                ;36392          ;---------------------------------;
                                ;36393          D_Q, Q_D+K[.3FF]+1,             ;  ROUND LOW FRACTION PART
U 1B88, 0C19,0010,81C0,F800,0010,18E3   ;36394          C[K.UBCC                        ;  CLOCK CARRY IF ANY
                                ;36395
                                ;36396          ;00011-------------------------;
                                ;36397  =00011
                                ;36398          SC_K[.FFF5],                    ;  CONSTANT NEEDED FO PACKG-ROUTINE
U 18E3, 0000,003D,E180,F800,0084,7889   ;36399          CALL, J/PACKG                   ;  PACK IT UP
                                ;36400
                                ;36401  =10011  ;10011-------------------------;  OK
U 18F3, 0001,003C,E5F0,2E80,0000,1B8A   ;36402          Q_ID[T9],R[RO]_D,J/POLYG.22A    ;  RESTORE LOW FRAC
                                ;36403
                                ;36404          ;10111-------------------------;  UNDERFLOW
U 18F7, 0000,003C,0180,F800,0000,1AB5   ;36405          J/POLYG.17                      ;
                                ;36406
                                ;36407          ;11011-------------------------;
U 18FB, 0000,0D3C,E5F0,2C00,0000,1933   ;36408          Q_ID[T9],Q31?,J/POLYG.15        ;
                                ;36409  =;END
```

```
                                ;36410  =0101   ;--------------------------------;
U 1AB5, 0000,003D,3DF0,2C00,0000,12AE   ;36411  POLYG.17:Q_ID[PSL],CALL[PSLFU.A]          ;  TEST FU BIT
                                ;36412
                                ;36413  =1101   ;--------------------------------;
U 1ABD, 0F00,003C,01F8,F800,0000,1B89   ;36414          Q_0,D_0,J/POLYG.22              ;  PSL<FU>=0,IGNORE UNDERFLOW
                                ;36415
                                ;36416          ;--------------------------------;
U 1ABF, 0818,0038,F580,FA18,0000,13F1   ;36417          LAB_R[R3],D_K[.A],J/POLYD.FAULT ;  TAKE UNDERFLOW FAULT
                                ;36418
                                ;36419          ;--------------------------------;
                                ;36420  POLYG.22:
U 1B89, 0001,003C,0180,FA80,0000,1B8A   ;36421          R[R0]_D                         ;  R0 GETS NEW PARTIAL PROD<MS>
                                ;36422
                                ;36423          ;--------------------------------;
                                ;36424  POLYG.22A:
U 1B8A, 0000,003C,0180,FA10,0000,1B8C   ;36425          LAB_R[R2]                       ;  LATCH UP DEGREE
                                ;36426
                                ;36427  POLYG.26:;--------------------------------;
                                ;36428          VA_VA+4,                        ;  INCREMENT COEFF ADDR
                                ;36429          R[R2]_LA-K[ 1], BYTE,           ;  DECREMENT DEGREE,
U 1B8C, 0018,8000,0580,FA93,0010,1B8E   ;36430          CLK.UBCC                        ;  WITHOUT CHANGING PC-DELTA
                                ;36431
                                ;36432          ;--------------------------------;
                                ;36433          R[R1]_Q,                        ;  STORE RESULT <LS>
                                ;36434          SC_K[.1F],                      ;  GENERATE UNDERFLOW MASK BIT
                                ;36435          INTRPT.STROBE,                  ;  STROBE INTERRUPTS
U 1B8E, 0001,213C,8D80,FA88,4084,79AC   ;36436          Z?                              ;  ARE WE DONE ?
                                ;36437
                                ;36438          ;--------------------------------;
                                ;36439  =0       ; BRANCH ON ALU Z-BIT
                                ;36440          ;0-------------------------------;
U 19AC, 0000,0E3C,0180,F800,0000,1896   ;36441          INTERRUPT.REQ?, J/POLYG.9       ;  KEEP LOOPING
                                ;36442
                                ;36443          ;1-------------------------------;  FINISH
                                ;36444  POLYG.35:
                                ;36445          CLR.FPD,                        ;  CLEAR FPD-FLAG IN PSL
U 19AD, 0003,003C,0180,FAA8,2000,1B90   ;36446          R[R5]_0                         ;  CLEAR R5
                                ;36447
                                ;36448          ;--------------------------------;
U 1B90, 0003,003C,0180,FAA0,0000,1B91   ;36449          R[R4]_0                         ;  CLEAR R4
                                ;36450
                                ;36451          ;--------------------------------;
U 1B91, 0003,003C,0180,FA90,0000,1B92   ;36452          R[R2]_0                         ;  CLEAR R2
                                ;36453
                                ;36454          ;--------------------------------;
                                ;36455  POLYG.C:
                                ;36456          ALU_R[R0].AND.K[.FFF0],
                                ;36457          N&Z_ALU.V&C_0,WORD,             ;  CLOCK PSL CONDITION CODES
U 1B92, 0018,4034,6D80,FA00,0050,13C3   ;36458          J/POLY.0                        ;  JOIN COMMON CODE
                                ;36459          ;--------------------------------;
```

```
                                ;36460  .TOC    ''      G & H floating point  : UNPACK SINGLE G FLOATING''
                                ;36461
                                ;36462  ;ROUTINE TO UNPACK A SINGLE G FLOATING POINT NUMBER.
                                ;36463
                                ;36464  ;INPUTS:
                                ;36465  ;               D = OP LONG FRAC
                                ;36466  ;               Q = OP LONG EXP
                                ;36467  ;               RC[T5] = OP1 EXPONENT
                                ;36468
                                ;36469  ;OUTPUTS:
                                ;36470  ;               SS = SD = SIGN
                                ;36471  ;               ID[T1] = OP FRAC <MS>
                                ;36472  ;               D = OP FRAC <LS>
                                ;36473  ;               RC4 = EXPONENT
                                ;36474  ;               R15 = RC5 - RC4
                                ;36475  ;               ALU CC REFLECT R15
                                ;36476  ;               SC = LOW 10 BITS OF R15
                                ;36477  ;               FE = 0
                                ;36478
                                ;36479  ;TEMPORARIES:
                                ;36480  ;               SC = VARIOUS SHIFT COUNTS
                                ;36481
                                ;36482  ;RETURNS:
                                ;36483  ;               RETURN1 IF RESULT IS 0
                                ;36484  ;               RETURN2 IF RESULT IS NOT 0
                                ;36485
                                ;36486  UNPKG:
                                ;36487          ;-------------------------------;
                                ;36488          R[R15]_Q,                       ;  STORE LONG EXP IN R15
                                ;36489          Q_D,                            ;  DUPLICATE LONG FRAC FOR SWAP
U 1B93, 0001,203C,65E0,FAF8,0084,7B94   ;36490          SC_K[.10]                       ;  SHIFTER GETS 16.
                                ;36491
                                ;36492          ;-------------------------------;
                                ;36493  UNPKG.0:
                                ;36494          D_DAL.SC,                       ;  SWAP WORDS OF LONG FRAC
                                ;36495          Q_R[R15].OR.K[.10],             ;  SET HIDDEN BIT
U 1B94, 0D18,0030,65C5,FA78,0000,1B95   ;36496          SS_SS.XOR.ALU15&SD_ALU15        ;  SD GETS SIGN BIT
                                ;36497
                                ;36498          ;-------------------------------;
U 1B95, 0019,2024,75C0,F800,0000,1B96   ;36499          Q_Q.ANDNOT.K[.20]               ;  CLEAR NEXT BIT
                                ;36500
                                ;36501          ;-------------------------------;
                                ;36502          Q_Q(FRAC),                      ;  UNPACK IT
U 1B96, 0001,203C,B9C8,F800,0084,7B97   ;36503          SC_K[.19]                       ;  SHIFT COUNT OF 25
                                ;36504          ;-------------------------------;
```

```
                                ;36505          D_DAL.SC,                       ;  RIGHT JUSTIFIED FRAC <MS>
                                ;36506          Q_D,                            ;  Q GETS LONG FRAC (SWAPPED)
U 1B97, 0D00,003C,F5E0,F800,0084,7B98   ;36507          SC_K[.A]                        ;  SHIFT COUNT OF 10
                                ;36508
                                ;36509          ;-------------------------------;
                                ;36510          LC_RC[T5],                      ;  LATCH UP OP1 EXPONENT
U 1B98, 0D00,003C,0180,F928,0000,1B99   ;36511          D_DAL.SC                        ;  GENERATE FRAC <MS>
                                ;36512
                                ;36513          ;-------------------------------;
                                ;36514          ID[T1]_D,                       ;  SAVE FRAC <MS>
U 1B99, 0C00,003C,C5F8,3C00,0000,1B9A   ;36515          D_Q, Q_0
                                ;36516
                                ;36517          ;-------------------------------;
                                ;36518          D_DAL.SC,                       ;  GENERATE FRAC <LS>
                                ;36519          ALU_R[R15].AND.K[.7FF0],
                                ;36520          Q_ALU.RIGHT2,                   ;  Q GETS EXPONENT/4
U 1B9A, 0D98,0034,39C0,FA78,0010,1B9B   ;36521          CLK_UBCC                        ;  CLOCK IT ON ITS WAY THRU
                                ;36522
                                ;36523          ;-------------------------------;
                                ;36524          RC[T4]_Q.RIGHT2,                ;  RC4 GETS EXPONENT
U 1B9B, 0081,213C,01C0,F9A0,0006,19BC   ;36525          Q_ALU.RIGHT2,Z?                 ;  IS IT 0?
                                ;36526
                                ;36527          ;-------------------------------;
                                ;36528  =0      ;BRANCH ON ALU Z-BIT
                                ;36529          ;0------------------------------;  NO
                                ;36530          ALU_Q-LC,                       ;  SUBTRACT EXP1 FROM EXP2
                                ;36531          R[RT5]_ALU,                     ;  STORE RESULT IN R15
                                ;36532          SC_ALU,                         ;  SC GETS LOW BITS OF DIFFERENCE
                                ;36533          FE_K[ZERO],
                                ;36534          CLR.UBCC,                       ;  CLOCK IT FOR USE IN ADDG
U 19BC, 0011,2002,1980,FAF8,0196,6002   ;36535          RETURN2                         ;
                                ;36536
                                ;36537          ;1------------------------------;  YES
                                ;36538          ALU_LA.AND.K[.8000],            ;  PASS JUST SIGN BIT THROUGH ALU
                                ;36539          CHK.FLT.OPR,                    ;  IS IT -0 ?
U 19BD, 0F18,0034,4580,F800,0800,1B9C   ;36540          D_0
                                ;36541  =;END
                                ;36542          ;-------------------------------;
                                ;36543          ALU_LC,R[R15]_ALU,              ;  STORE ORIGINAL EXP IN R15
U 1B9C, 0010,003A,0180,FAF8,0000,0001   ;36544          RETURN1
                                ;36545          ;-------------------------------;
```

```
                                        ;36546  .TOC    ''       G & H floating point  : FLOATING POINT ARITHMETIC -- EMODG''
                                        ;36547
                                        ;36548
                                        ;36549  ;  EMODG        MULR.RG, MULRX.RB, MULD.RG, INT.WL, FRACT.WD
                                        ;36550
                                        ;36551  ;INPUTS:
                                        ;36552  ;               RC0 = MULTIPLIER LONG EXP
                                        ;36553  ;               Q = MULTIPLIER LONG FRAC
                                        ;36554  ;               D = MULTIPLIER EXTENSION
                                        ;36555
                                        ;36556  ;TEMPORARIES:
                                        ;36557  ;               RC1 = M'CAND LONG EXP
                                        ;36558  ;               RC2 = M'CAND LONG EXP
                                        ;36559  ;               RC4 = M'PLIER EXTENSION IN HIGH 11 BITS
                                        ;36560                                  C
                                        ;36561  =0**1*  ;0**1*-------------------------;
                                        ;36562  EMODG:
U 1883, 0000,003D,0180,F800,0000,167E   ;36563          CALL,J/SPECE                    ;  EVALUATE EXTENSION
                                        ;36564          ;1**1*-------------------------;
                                        ;36565          SC_K[.10],
U 1893, 0000,003F,6580,F800,0084,7400   ;36566          SUB/SPEC,J/G.FORK
                                        ;36567  =;END
                                        ;36568          ;-------------------------------;
                                        ;36569  14C8:                                   ;  EMODG
                                        ;36570          TRAP.ACC[1],                    ;  RESET THE FPA
                                        ;36571          INTRPT.STROBE,                  ;  STROBE INTERRUPTS
U 14C8, 0D00,00BC,0080,F800,4000,18CE   ;36572          D_DAL.SC                        ;  PUT EXTENDER IN HIGH WORD
                                        ;36573
                                        ;36574          ;0**1*-------------------------;
                                        ;36575  =0**1*  RC[T4]_D,                       ;  GET 16 BIT EXTENDED MULTIPLIER
                                        ;36576          D_Q,                            ;  D GETS M'PLIER LONG FRAC
                                        ;36577          CALL, INTERRUPT.REQ?,           ;  EVALUATE MUL'CAND
U 18CE, 0C01,0E3D,0180,F9A0,0000,147E   ;36578          J/SPECG
                                        ;36579
                                        ;36580          ;-------------------------------;
                                        ;36581          ;RETURN WITH :
                                        ;36582          ;       D = M'CAND LONG FRAC
                                        ;36583          ;       RC2 = M'CAND LONG EXP
                                        ;36584          ;       Q = M'PLIER LONG FRAC
                                        ;36585          ;1**1*-------------------------;
                                        ;36586          TRAP.ACC[1],                    ;  RESET THE FPA
                                        ;36587          LC_RC[T2],                      ;  LATCH UP M'CAND LONG EXP
U 18DE, 0000,00BC,0080,F910,0000,1809   ;36588          J/EMODG.1
                                        ;36589  =;END   ;-------------------------------;
```

```
                                ;36590  =0**01  ;0**01-------------------------;
                                ;36591  EMODG.1:
                                ;36592          RC[T1]_LC,                      ;  STORE M'CAND LONG EXP IN RC1
                                ;36593          STATE_R[.3],                    ;  SET STATE-CODE FOR EMODG
U 1809, 0010,0039,0D80,F988,1404,7B4C   ;36594          CALL,J/MULG.00                  ;  CALL MULTIPLY ROUTINE
                                ;36595
                                ;36596  ; WHEN WE ENTER MULTIPLY-ROUTINE:
                                ;36597  ;       D = M'CAND LONG FRAC
                                ;36598  ;       RC1 = RC2 = LC = M'CAND LONG EXP
                                ;36599  ;       RC4 = EXTENDER IN HIGH 11 BITS
                                ;36600  ;       RC0 = M'PLIER LONG EXP
                                ;36601  ;       Q = M'PLIER LONG FRAC
                                ;36602  ;       STATE = 3
                                ;36603
                                ;36604          ;1**01-------------------------;  PRODUCT = 0
                                ;36605  ;       FE = 10
                                ;36606  ;       D = Q = RC1 = 0
                                ;36607  =1**01
                                ;36608  EMODG.ZERO:
                                ;36609          ID[T0]_D,                       ;  SAVE ? IN T0
                                ;36610          STATE_FE,                       ;  SHARE CODE WITH EMODF
                                ;36611          RC[T1]_0,                       ;  RESULT IS 0
                                ;36612          N&Z_ALU.V&C_0,                  ;  CLOCK PSL CONDITION CODES
U 1819, 0003,003C,C180,3D88,1450,7BA3   ;36613          J/EMODG.0000                    ;  SHARE CODE WITH EMODD AND EMODF
                                ;36614
                                ;36615          ;1**11-------------------------;  PRODUCT NE 0
                                ;36616  ;       < D,Q> = FRACTION
                                ;36617  ;        AB = R15 = BIASSED EXPONENT + 1 OR 2
                                ;36618
                                ;36619          RC[T2]_Q,                       ;  SAVE PROD FRAC <LS> IN RC2
                                ;36620          STATE_R[.10],                   ;  SET DOUBLE FLAG
U 181B, 0001,203C,65F8,F990,1404,7B9D   ;36621          Q_0                             ;  BRANCHING CONVENIENCE
                                ;36622
                                ;36623          ;-------------------------------;
                                ;36624          ALU_K[.80],Q_ALU.LEFT3,         ;  Q GETS BIAS OF 400
U 1B9D, 00B8,0038,41C0,F910,0000,1BA0   ;36625          LC_RC[T2]                       ;  LATCH PROD FRAC <LS>
                                ;36626
                                ;36627          ;-------------------------------;
                                ;36628          ALU_LA-Q-1,SC_ALU,              ;  UNBIAS EXPONENT
U 1BA0, 001C,0008,0180,FAF8,0092,196D   ;36629          R[RT5]_ALU,CLR.UBCC
                                ;36630          ;-------------------------------;
```

```
                                ;36631  =0****  ;0****----------------------------;
                                ;36632  EMODG.2:
                                ;36633          Q_LC,                              ;  RESTORE Q TO PROD FRAC <LS>
                                ;36634          SC_K[.1],                          ;  IN CASE WE NEED TO SHIFT
                                ;36635          SET.CC(INST),                      ;  CLEAR VBIT
                                ;36636          LAB_R[R15],D31?,                   ;  TEST LEADING BIT OF D
U 196D, 0010,CD39,05C0,FA78,00F4,787A   ;36637          CAL[[EMODG.5]
                                ;36638
                                ;36639          ;1****----------------------------;
                                ;36640  ;return with:
                                ;36641  ;       RC1 = FRACTION <MS>
                                ;36642  ;       RC2 = FRACTION <LS>
                                ;36643  ;       SC = EXPONENT MODULO 32.
                                ;36644  ;       R15 = EXPONENT
                                ;36645  ;       D = INTEGER
                                ;36646  ;       PSL<V> SIGNALS OVERFLOW
                                ;36647
                                ;36648          ALU_R[R15].ANDNOT.K[.3FF],         ;  ISOLATE UPPER BITS
U 197D, 0018,0024,8180,FA78,0010,1BA1   ;36649          CLK.UBCC                           ;  SEE IF THEY ARE ZERO
                                ;36650
                                ;36651          ;-------------------------------;
U 1BA1, 0000,1B3C,0180,F800,0000,1AC3   ;36652          ALU?                           ;
                                ;36653  =0011   ;0011----------------------------;  UPPER BITS NOT ZERO
U 1AC3, 0000,003C,0180,F800,0020,1AC7   ;36654          SET.V                          ;  SET V-BIT
                                ;36655
                                ;36656          ;0111----------------------------;
                                ;36657          ALU_R[R15],CLK.UBCC,               ;  CLOCK EXPONENT
U 1AC7, 0000,003C,A180,FA78,0094,5BA2   ;36658          SC_SC.ANDNOT.K[.FFE0],J/EMODG.2A;  CLEAR OUT IRRELEVANT BITS
                                ;36659
                                ;36660          ;1011----------------------------;
                                ;36661          ALU_R[R15],CLK.UBCC,               ;  CLOCK EXPONENT
U 1ACB, 0000,003C,A180,FA78,0094,5BA2   ;36662          SC_SC.ANDNOT.K[.FFE0]          ;  CLEAR OUT IRRELEVANT BITS
                                ;36663  =;END
                                ;36664  EMODG.2A:;-------------------------------;
                                ;36665          ID[T0]_D,                          ;  STORE INTEGER IN T0
                                ;36666          D_RC[T2],Q_0,                      ;  D GETS MANTISSA <L>
                                ;36667          FE_STATE,                          ;  TRICK TO CLEAR STATE
U 1BA2, 0810,1A38,C1F8,3D10,1500,1ACD   ;36668          PSL.V?                             ;  TEST FOR INTEGER OVERFLOW
                                ;36669
                                ;36670  =1101   ;1101----------------------------;  NO OVERFLOW
                                ;36671          Q_RC[T1],                          ;  GET MANTISSA <MS> IN Q
                                ;36672          CLK.UBCC,                          ;  SET CONDITION CODES ON IT
                                ;36673          ID[T2]_D,                          ;  SAVE FRACTION <LS> IN T2
                                ;36674          D_DAL.SC,                          ;  FIRST PART OF 64 BIT FRACTION SHIFT
U 1ACD, 0D10,1B38,C9C0,3D08,0010,1AD7   ;36675          ALU?, J/EMODG.30                   ;  CHECK FOR POSITIVE EXPONENT
                                ;36676
                                ;36677          ;-------------------------------;  OVERFLOW
                                ;36678          STATE_STATE.ANDNOT.FE,             ;  CLEAR STATE FOR OVERFLOW
                                ;36679          Q_RC[T1],                          ;  GET MANTISSA <MS> IN Q
                                ;36680          CLK.UBCC,                          ;  SET CONDITION CODES ON IT
                                ;36681          ID[T2]_D,                          ;  SAVE FRACTION <LS> IN T2
                                ;36682          D_DAL.SC,                          ;  FIRST PART OF 64 BIT FRACTION SHIFT
U 1ACF, 0D10,1B38,C9C0,3D08,1410,5AD7   ;36683          ALU?, J/EMODG.30                   ;  CHECK FOR POSITIVE EXPONENT
                                ;36684          ;-------------------------------;
```

```
                                ;36685
                                ;36686          ;-------------------------------;
                                ;36687  EMODG.0000:
                                ;36688          SGN/CLR.SD+SS,                  ;  RESULT IS 0
U 1BA3, 0001,003C,0187,F988,0000,181C   ;36689          RC[T1]_D,J/EMODFG.11
                                ;36690
                                ;36691          ;-------------------------------;
                                ;36692  =0111    ;BRANCH ON ALU N-BIT
                                ;36693          ;0111---------------------------;
                                ;36694  EMODG.30:                               ;  EXPONENT IS >= 0
                                ;36695          RC[T2]_D,                       ;  SAVE NEW FRACTION <LS> IN RC2
                                ;36696          D_Q, Q_ID[T2],                  ;  GET READY TO SHIFT FRAC <MS>
U 1AD7, 0C01,003C,C9F0,2D90,0000,1BA4   ;36697          J7EMODG.4
                                ;36698
                                ;36699          ;1111---------------------------;  EXP < 0, DON'T SHIFT
                                ;36700  EMODG.3:
                                ;36701          D_Q,                            ;  RESTORE ORIGINAL FRACTION
                                ;36702          Q_RC[T2],                       ;  Q GETS FRACTION <MS>
                                ;36703          SC_FE,                          ;  NEEDED FOR CALL
U 1ADF, 0C10,0038,81C0,F910,0081,198D   ;36704          K[.3FF]
                                ;36705  =;END
                                ;36706          ;0****-------------------------;
                                ;36707  =0****
                                ;36708          ALU_LA+K[.3FF],RC[T5]_ALU,      ;  ADD BIAS-1 TO EXPONENT
                                ;36709          SC_SC-FE,                       ;  CLEAR SC
                                ;36710          ALU?,                           ;  BRANCH ON FRAC <MS>
U 198D, 0018,1B15,8180,F9A8,0080,BA1B   ;36711          CALL,J/ADDG.31
                                ;36712
                                ;36713          ;1****-------------------------;
                                ;36714  ;RETURN FROM ADDG-ROUTINE.
                                ;36715          STATE_K[.8], FE_K[.8],
                                ;36716          Q_K[.8].RIGHT,                  ;  USED FOR FLAG TO PROBE-ROUTINE
                                ;36717          INTRPT.STROBE,                  ;
U 199D, 0058,1638,01C0,F800,5504,79C4   ;36718          STATE4?                         ;  WAS THERE OVERFLOW?
                                ;36719  =;END
                                ;36720          ;-------------------------------;
                                ;36721  =***0   ;BRANCH ON STATE<4>
                                ;36722          ;***0---------------------------;  YES
                                ;36723          SET.V,                          ;  SET INTEGER OVERFLOW FLAG
                                ;36724          STATE_STATE+FE,                 ;  STATE GETS 10
                                ;36725          SC_Q,                           ;  SC GETS 4
                                ;36726          Q_ID[CES],                      ;  GET CES-REGISTER FOP OVERFLOW
U 19C4, 0001,203C,3180,2C00,14A2,9BA6   ;36727          J7EMODFG.V
                                ;36728
                                ;36729          ;***1---------------------------;  NO
                                ;36730          STATE_STATE+FE,                 ;  STATE GETS 10
                                ;36731          INTRPT.STROBE,                  ;  STROBE FOR INTERRUPTS
                                ;36732          SC_Q,                           ;  SC GETS 4
U 19C5, 0001,203C,0180,F800,5482,981C   ;36733          J/EMODFG.11
```

```
                                ;36734          ;-------------------------------;
                                ;36735  EMODG.4:
                                ;36736          R[R15]_0,                       ;  CLEAR EXPONENT
U 1BA4, 0D03,003C,0180,FAF8,0000,1BA5   ;36737          D_DAL.SC                        ;  KEEP SHIFTING FRACTION
                                ;36738
                                ;36739          ;-------------------------------;
                                ;36740          SC_K[ZERO],
                                ;36741          LAB_R[R15],                     ;  LATCH 0 IN LAB
                                ;36742          Q_D, ALU_D, CLK.UBCC,           ;  CLOCK ALU CC ON FRAC <MS>
U 1BA5, 0001,003C,19E0,FA78,0094,7ADF   ;36743          J/EMODG.3
                                ;36744
                                ;36745          ;00****0--------------------------;
                                ;36746  =00****0
                                ;36747  EMODFG.11:
                                ;36748          ID[T1]_D,                       ;  SAVE FRACTION <MS>
                                ;36749          D_RC[TT],                       ;  GET FRACTION <LS>
                                ;36750          CALL,INTERRUPT.REQ?,
U 181C, 0810,0E39,C580,3D08,0000,047E   ;36751          J/ASPC                          ;  EVALUATE INTEGER ADDRESS
                                ;36752
                                ;36753          ;01****0--------------------------;  MEMORY ADDRESS
                                ;36754  =01****0
                                ;36755  EMODFG.110:
                                ;36756          ID[T1]_D,                       ;  SAVE FRACTION <MS>
                                ;36757          D_RC[TT],                       ;  GET FRACTION <LS>
                                ;36758          CALL,
U 183C, 0810,0039,C580,3D08,0000,047E   ;36759          J/ASPC                          ;  EVALUATE INTEGER ADDRESS
                                ;36760
                                ;36761          ;11****0--------------------------;  MEMORY ADDRESS
                                ;36762  =11****0
                                ;36763          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;36764          RC[T2]_D,                       ;  SAVE INT.WL ADDRESS IN RC2
                                ;36765          D_Q,                            ;  D GETS FRACTION <LS>
                                ;36766          STATE_STATE+1,                  ;  SET FLAG FOR MEMORY MODE
                                ;36767          INTRPT.STROBE,                  ;  STROBE FOR INTERRUPTS
U 187C, 0C01,00BC,0080,F990,5400,D888   ;36768          J/EMODGF.12
                                ;36769
                                ;36770          ;11****1--------------------------;
                                ;36771  =11****1                                ;  REGISTER OPERAND
                                ;36772          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;36773          D_Q,                            ;  D GETS FRACT<LS>
                                ;36774          RC[T2]_RLOG.RIGHT,              ;  SAVE REGISTER NUMBER
                                ;36775          INTRPT.STROBE,                  ;  STROBE FOR INTERRUPTS
U 187D, 0C40,00B8,0080,F990,5C00,1888   ;36776          J/EMODGF.12
                                ;36777  =;END
                                ;36778          ;-------------------------------;
                                ;36779  EMODFG.V:
                                ;36780          Q_D,D_Q.OR.K[.10],              ;  SET INTEGER OVERFLOW CODE IN CES
U 1BA6, 0819,2030,65E0,F800,4000,1BA7   ;36781          INTRPT.STROBE                   ;  STROBE FOR INTERRUPTS
                                ;36782
                                ;36783          ;-------------------------------;
                                ;36784          ID[CES]_D,D_Q,                  ;  WRITE TRAP CODE IN CES
U 1BA7, 0C00,003C,3180,3C00,0000,183C   ;36785          J/EMODFG.110                    ;  DON'T TEST FOR INTERRUPTS, TOO LATE
                                ;36786          ;-------------------------------;
```

```
                                ;36787          ;00****0----------------------;
                                ;36788  =00****0
                                ;36789  EMODGF.12:
                                ;36790          ID[T2]_D,                       ;  T2 SAVES FRACTION <LS>
                                ;36791          CALL,
U 1888, 0000,003D,C980,3C00,0000,157E   ;36792          J/ASPCG                         ;  NEED TO DOUBLE THE CONTEXT
                                ;36793
                                ;36794          ;11****0----------------------;
                                ;36795  =11****0
                                ;36796          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;36797          RC[T4]_D,                       ;  SAVE FRACT ADDRESS
                                ;36798          STATE_STATE+K[.2],              ;  REMEMBER MEMORY MODE
U 18E8, 0001,00BC,0880,F9A0,1404,9114   ;36799          J/EMODF.14
                                ;36800
                                ;36801          ;11****0----------------------;
                                ;36802          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;36803          Q_ID[T0],                       ;  REGISTER MODE,
U 18E9, 0000,17BC,C0F0,2C00,0000,12EA   ;36804          STATE0?,J/EMODF.16
                                ;36805  =;END
                                ;36806
                                ;36807          ;--------------------------------;
                                ;36808  =*10    ;BRANCH ON D31 (WHETHER PRODUCT IS NORMALIZED)
                                ;36809          ;*10-------------------------------;
                                ;36810  EMODG.5:                                ;  NOT NORMALIZED
                                ;36811          Q_Q.LEFT,                       ;  SHIFT FRACTION 1 BIT
                                ;36812          D_DAL.SC, SI/ZERO,              ;  <MS> FRACTION AS WELL
                                ;36813          ALU_LA-K[.1], R[R15]_ALU,       ;  ADJUST EXPONENT
                                ;36814          SC_ALU,CLK.UBCC,
U 187A, 0D18,0000,05A8,FAF8,0092,1BA8   ;36815          J/EMODG.50
                                ;36816
                                ;36817          ;*11-------------------------------;  NORMALIZED
                                ;36818          ALU_R[R15],CLK.UBCC,LONG,
U 187B, 0000,003C,0180,FA78,0092,1BA9   ;36819          SC_ALU,J/EMODG.6
                                ;36820  =;END
                                ;36821          ;--------------------------------;
                                ;36822  EMODG.50:
U 1BA8, 0001,203C,0180,F990,0000,1BA9   ;36823          RC[T2]_Q                        ;  SAVE FRACTION <LS> IN RC2
                                ;36824
                                ;36825          ;--------------------------------;
                                ;36826  EMODG.6:
                                ;36827          FE_SC,                          ;  BACK UP EXPONENT IN FE
                                ;36828          Q_0,
                                ;36829          RC[T1]_0,                       ;
U 1BA9, 0003,1B3C,01F8,F988,0100,1AE3   ;36830          ALU?                            ;  INITIALIZE AND CALL CONVERT ROUTINE
                                ;36831
                                ;36832          ;0011-------------------------;  EXP>0
                                ;36833  =0011   ALU_R[R15].ANDNOT.K[.1F],       ;  MASK OUT LOWER FIVE BITS
U 1AE3, 0018,0024,8D80,FA78,0010,1AEF   ;36834          CLK.UBCC,J/EMODG.N1             ;
                                ;36835
                                ;36836          ;0111-------------------------;  EXP=0
U 1AE7, 0F01,003E,01E0,F988,0000,0010   ;36837          Q_D,RC[T1]_D,D_0,RETURN10       ;  Q,RC1 GETS FRAC<MS>,INT=0
                                ;36838
                                ;36839          ;1011-------------------------;  EXP<0
U 1AEB, 0F01,003E,01F8,F988,0000,0010   ;36840          RC[T1]_D,D_0,Q_0,RETURN10       ;  RC1 GETS FRAC<MS>,INT=0,Q=0
```

```
                                ;36841  EMODG.N1:;-------------------------------;
                                ;36842          SC_SC-K[.20],FE_SC-K[.20],Q_0,  ;  SUBTRACT 32 FROM EXP
U 1AEF, 0000,013C,75F8,F800,0184,B9CC   ;36843          Z?                              ;  EXP>=32?
                                ;36844
                                ;36845  =0      ;0------------------------------;  EXP>=32
                                ;36846          ALU_R[R15].ANDNOT.K[.3F],       ;
U 19CC, 0018,0024,5580,FA78,0010,1BAA   ;36847          CLK.UBCC,J/EMODG.N2             ;  MASK LOWER SIX BITS
                                ;36848
                                ;36849          ;1------------------------------;  1<=NUMBER<2**31
                                ;36850          RC[T1]_D,Q_D,D_DAL.SC,SS?,      ;  D GETS INTEGER,SAVE FRAC<MS> IN RC2,Q
U 19CD, 0D01,123C,01E0,F988,0000,1B6E   ;36851          J/CVTGI.4                       ;
                                ;36852
                                ;36853  EMODG.N2:;-------------------------------;
U 1BAA, 0010,0138,01C0,F910,0000,19E4   ;36854          Q_RC[T2],Z?                     ;  EXP>=64?
                                ;36855
                                ;36856  =0      ;0------------------------------;  EXP>=64
                                ;36857          ALU_R[R15]-K[.60],CLK.UBCC,     ;  SUBTRACT 96
U 19E4, 0018,0000,A580,FA78,0010,1BAB   ;36858          J/EMODG.N3                      ;
                                ;36859
                                ;36860          ;1------------------------------;  2**31<=NUMBER<2**63
                                ;36861          ALU_FE,Q_D.OXT[BYTE].OR.PACK.FP,;  GENERATE FUNNY PACKED NUMBER
U 19E5, 0D0B,8030,01C0,F800,0000,7BE6   ;36862          D_DAL.SC,J/CVTGI.3              ;  GET INTEGER
                                ;36863
                                ;36864  EMODG.N3:;-------------------------------;
                                ;36865          SC_SC-K[.20],RC[T2]_0,D_Q,      ;
U 1BAB, 0C03,1B3C,75F8,F990,0084,BAF7   ;36866          Q_0,ALU?                        ;
                                ;36867
                                ;36868  =0111   ;0111-----------------------------;  EXP>=96
U 1AF7, 0F03,003E,01F8,F988,0020,0010   ;36869          D_0,Q_0,RC[T1]_0,SET.V,RETURN10 ;  SET OVERFLOW BIT
                                ;36870
                                ;36871          ;1111-----------------------------;  2**63<NUMBER<2**95
                                ;36872          D_DAL.SC,RC[T1]_0,Q_0,          ;  D GETS LOW BITS OF INTEGER
U 1AFF, 0D03,123C,01F8,F988,0020,1B0E   ;36873          SET.V,SS?                       ;  SAVE MANTISSA,SET OVERFLOW BIT
                                ;36874
                                ;36875  =1110   ;1110-----------------------------;  SS CLEAR
U 1B0E, 0000,003E,0180,F800,0000,0010   ;36876          RETURN10                        ;  WE ARE DONE
                                ;36877
                                ;36878          ;1111-----------------------------;  SS SET
U 1B0F, 081F,2002,0180,F800,0000,0010   ;36879          D_0-D,RETURN10                  ;  COMPLIMENT INTEGER
```

```
                                ;36880  .TOC    ""      G & H floating point  : G FORMAT MOVE AND TEST INSTRUCTIONS''
                                ;36881
                                ;36882  ;INPUT:
                                ;36883  ;               D = SRC LONG FRAC
                                ;36884  ;               RC0 = SRC LONG EXP
                                ;36885
                                ;36886          ;-------------------------------;
                                ;36887  1485:
                                ;36888  MOVG:   TRAP.ACC[1],                    ;  RESET THE FPA
                                ;36889          ALU_RC[T0], Q_ALU,              ;  GET SRC LONG EXP
U 1485, 0010,00B8,00C0,F900,0070,1BAC ;36890    SET.CC(LONG)                    ;
                                ;36891
                                ;36892          ;-------------------------------;  MOVG/MNEG
                                ;36893          ALU_Q.AND.K[.7FF0],             ;  CHECK EXPONENT FOR 0
U 1BAC, 0019,2034,3980,F800,0010,1BAD ;36894    CLK.UBCC,J/MOVG.1
                                ;36895
                                ;36896          ;-------------------------------;
                                ;36897  MOVG.1:
                                ;36898          ALU_D,RC[T1]_ALU,               ;  STORE LONG FRAC WHERE G-FORK WANTS IT
U 1BAD, 0C01,013C,0180,F938,0000,19EC ;36899    D_Q,Z?                          ;  IS SRC 0 ?
                                ;36900
                                ;36901          ;-------------------------------;
                                ;36902  =0       ;BRANCH ON ALU Z-BIT
                                ;36903          ;0------------------------------;
                                ;36904          D_D[INST.DEP]K[.8000],          ;  CHANGE SIGN IF MNEGG
                                ;36905          N&Z_ALU,WORD,                   ;  CLOCK Z & N-BITS,  LEAVE C AND V
U 19EC, F819,400F,45F0,F847,0060,1400 ;36906    WRITE.G.DEST                    ;  WRITE DESTINATION
                                ;36907
                                ;36908          ;1------------------------------;
                                ;36909          RC[T1]_D.AND.K[.8000],
                                ;36910          CHK.FLT.OPR,D_0,                ;  CHECK FOR RESERVED OPERAND
                                ;36911          N&Z_ALU,                        ;  CLOCK Z & N-BITS,  LEAVE C AND V
U 19ED, 0F19,4034,4580,F988,0860,1BB0 ;36912    WORD,J/WRITE.G
                                ;36913  =;END
                                ;36914          ;-------------------------------;
U 1BB0, F000,003F,01F0,F847,0000,1400 ;36915  WRITE.G:       WRITE.G.DEST
                                ;36916          ;-------------------------------;
```

```
                                ;36917  ;       TESTG INSTRUCTION
                                ;36918  ;INPUTS:        RC0 = LONG EXP
                                ;36919
                                ;36920  14A9:   STATE_K[SP1.CON],               ;  STATE GETS 4 OR 8
U 14A9, 0000,003C,15C0,F800,1404,7BB1   ;36921          Q_LA                    ;  Q GETS SRC EXP
                                ;36922
                                ;36923          ;-----------------------------;
                                ;36924          STATE3-0?,                      ;  TEST FOR G OR H-FORMAT
U 1BB1, 0000,173C,0180,F800,0000,1815   ;36925          J/TSTG.0
                                ;36926
                                ;36927  148D:   ;-----------------------------;
U 148D, 0010,0038,01C0,F900,0000,1815   ;36928          Q_RC[T0]                ;  Q GETS SRC EXP
                                ;36929
                                ;36930          ;-----------------------------;
                                ;36931  =01**   ;BRANCH ON STATE<3>
                                ;36932          ;01**-------------------------;
                                ;36933  TSTG.0:
                                ;36934          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;36935          ALU_Q.AND.K[.7FF0],
                                ;36936          CLK.UBCC,                       ;  CLOCK Z-BIT ON EXPONENT
U 1815, 0019,20B4,3880,F800,0010,1BB4   ;36937          J/TSTG.01
                                ;36938
                                ;36939          ;11**-------------------------;
                                ;36940  TSTH.00:
                                ;36941          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;36942          ALU_Q.ANDNOT.K[.8000],          ;  CLOCK EXPONENT
U 181D, 0019,60A4,4480,F800,0010,1BB2   ;36943          CLK.UBCC,WORD
                                ;36944
                                ;36945          ;-----------------------------;
                                ;36946          ALU_Q,N&Z_ALU.V&C_0,WORD,
U 1BB2, 0001,613C,0180,F800,0050,19F4   ;36947          Z?,J/TSTG.1
                                ;36948
                                ;36949          ;-----------------------------;
                                ;36950  TSTG.01:
                                ;36951          ALU_Q.AND.K[.FFF0],             ;
U 1BB4, 0019,6134,6D80,F800,0050,19F4   ;36952          N&Z_ALU.V&C_0,WORD,Z?
                                ;36953
                                ;36954          ;-----------------------------;
                                ;36955  =0      ;BRANCH ON ALU Z-BIT
                                ;36956          ;0--------------------------;
                                ;36957  TSTG.1: PC_PC+1,                        ;  INCREMENT PC
                                ;36958          CLR.IB.OPC,                     ;  CLEAR OPCODE FROM IB
U 19F4, C000,003C,0180,F804,4000,0062   ;36959          J/IRD
                                ;36960
                                ;36961          ;1--------------------------;
                                ;36962          ALU_Q,
                                ;36963          CHK.FLT.OPR,                    ;  TEST FOR -0
                                ;36964          PC_PC+1,                        ;  INCREMENT PC
                                ;36965          CLR.IB.OPC,                     ;  CLEAR OPCODE FROM IB
U 19F5, C001,203C,0180,F804,480C,0062   ;36966          J/IRD
                                ;36967  =;END   ;-----------------------------;
```

```
                                ;36968  .TOC    ''      G & H floating point  : FORMAT CONVERSION -- CVTFG,CVTDH,CVTFH''
                                ;36969  ;        FOR CVTFG AND CVTFH :D = SRC OPERAND, A F FLOATING POINT NUMBER
                                ;36970  ;       FOR CVTDH: RC[T2]= LONG EXP, D = LONG FRAC
                                ;36971  ;       OPCODES ARE: 99FD FOR CVTFG, 98FD FOR CVTFH, 32FD FOR CVTDH
                                ;36972
                                ;36973  144C:   ;-------------------------------;
                                ;36974  CVTFG:  TRAP.ACC[1],                    ;  RESET THE FPA
                                ;36975          ALU_D,                          ;  D HAS SRC OPERAND
                                ;36976          SC_D(EXP),                      ;  SC GETS BIASSED EXP
                                ;36977          SS_ALU15,                       ;  SS GETS SIGN
                                ;36978          N&Z_ALU.V&C_0,WORD,             ;  CLOCK CONDTION CODES
                                ;36979          RC[T3]_ALU,                     ;  SAVE IT IN RC3
                                ;36980          D_ALU(FRAC),                    ;  D GETS UNPACKED FRACTION
                                ;36981          Q_D,                            ;  Q GETS SRC
U 144C, 0901,5B8C,00E1,F998,00D3,1B1D   ;36982          IR0?                            ;  IS THIS CVTFG OR CVTFH ?
                                ;36983
                                ;36984          ;-------------------------------;
                                ;36985  =1101   ;BRANCH ON IR<0>
                                ;36986          ;1101-----------------------------;  CVTFH OR CVTDH
                                ;36987          Q_D,D_0,
U 1B1D, 0F00,093C,01E0,F800,0000,19B2   ;36988          IR2-1?,J/CVTFH
                                ;36989
                                ;36990          ;1111-----------------------------;  CVTFG
                                ;36991          R[R15]_K[.3FF],                 ;  NEW BIAS -1
                                ;36992          DK/RIGHT2,                      ;  SHIFT FRACTION RIGHT TWICE
                                ;36993          FE_SC,CLK.UBCC,                 ;  DUPLICATE EXPONENT IN FE
                                ;36994          SD_SS,                          ;  SD GETS SIGN
U 1B1F, 0218,0C38,8184,FAF8,0110,1A01   ;36995          SC.NE.0?                        ;  IS EXPONENT = 0 ?
                                ;36996
                                ;36997          ;-------------------------------;
                                ;36998  =0**    ;BRANCH ON SC NE 0
                                ;36999          ;0**------------------------------;
                                ;37000          ALU_Q.AND.K[.8000],RC[T1]_ALU,  ;  CLEAR RC1
                                ;37001          CHK.FLT.OPR,                    ;  CHECK FOR -0
                                ;37002          D_0,Q_0,                        ;  MAKE IT A CLEAN 0
U 1A01, 0F19,2034,45F8,F988,0850,1B8C   ;37003          N&Z_ALU.V&C_0,J/CVTFG.2         ;  SET PSL Z-BIT
                                ;37004
                                ;37005          ;1**------------------------------;
                                ;37006          QK/RIGHT2,                      ;  SHIFT SRC RIGHT TWICE
U 1A05, 0808,0038,0190,F800,0000,1B85   ;37007          D_PACK.FP                       ;  PACK NEW FRACTION <MS>
                                ;37008  ;END
                                ;37009          ;-------------------------------;
                                ;37010          ALU_Q.ANDNOT.R[R15],            ;  GET RID OF BITS <9:0>
                                ;37011          Q_ALU.RIGHT,                    ;  SHIFT 3 LOW ORDER FRACTION BITS
U 1B85, 004D,2024,41C0,FA78,0084,B8B6   ;37012          SC_SC-K[.80]                    ;  UNBIAS EXPONENT
                                ;37013
                                ;37014          ;-------------------------------;
                                ;37015          ALU_Q.AND.K[.E003],             ;  ISOLATE UPPER 3 BITS OF FRAC <LS>
U 1B86, 0019,2034,9980,F988,0000,1B88   ;37016          RC[T1]_ALU
                                ;37017
                                ;37018          ;-------------------------------;
U 1B88, 0018,0038,1DC0,F800,0000,1B89   ;37019          ALU_K[SC],Q_ALU                 ;  Q GETS LOW 8 BITS OF EXPONENT
```

```
                                        ;37020
                                        ;37021          ;-------------------------------;
U 1BB9, 00AE,A010,01C0,F800,0000,1BBA   ;37022          ALU_Q.SXT[BYTE]+LB+1,Q_ALU.LEFT3 ;  GET NEW EXPONENT*4
                                        ;37023
                                        ;37024          ;-------------------------------;
                                        ;37025          ALU_D.ANDNOT.K[.7FF0],D_ALU,     ;  CLEAR EXPONENT BITS
U 1BBA, 0819,0024,39A8,F800,0000,1BBC   ;37026          Q_Q.LEFT                         ;  Q GETS EXPONENT * 16.
                                        ;37027
                                        ;37028          ;-------------------------------;
                                        ;37029  CVTFG.2:
U 1BBC, F81D,0033,01F0,F847,0000,1400   ;37030          D_D.OR.Q, WRITE.G.DEST           ;  NOT ZERO, WRITE DESTINATION
                                        ;37031          ;-------------------------------;
```

```
                                ;37032  .TOC    "       G & H floating point  : CONVERT G/H-FORMAT TO F/D-FORMAT"
                                ;37033
                                ;37034  ; ENTRY POINT FOR CVTHF (OPC = F6), CVTHD ( F7 ), CVTGF ( 33 ).
                                ;37035  ;INPUTS:
                                ;37036  ;       FOR CVTGF:
                                ;37037  ;               RC2 = SRC LONG EXP
                                ;37038  ;               D = SRC LONG FRAC
                                ;37039  ;       FOR CVTHF AND CVTHD:
                                ;37040  ;               RC2-ID3 = SRC OPERAND
                                ;37041  ;               STATE = 0
                                ;37042  ;               SC = .31
                                ;37043
                                ;37044  144B:   ;-------------------------------;
                                ;37045  CVTGF:  TRAP.ACC[1],                    ;  RESET THE FPA
                                ;37046          SC_SC+1,FE_SC+1,                ;  SC AND FE GET .32
                                ;37047          LC_RC[T2],                      ;  LATCH SRC LONG FRAC
                                ;37048          ALU_K[.80], Q_ALU.LEFT3,        ;  Q GETS .400 (=OLD BIAS)
U 144B, 00B8,09B8,40C0,F910,0180,D839   ;37049          IR2-1?                          ;  IS SRC = H OR G ?
                                ;37050
                                ;37051          ;-------------------------------;
                                ;37052  =01     ; BRANCH ON IR<2>
                                ;37053          ;01-----------------------------;
                                ;37054          ALU_Q-K[.80], RC[T5]_ALU,       ;  RC5 GETS DIFFERENCE BETWEEN BIASSES
U 1839, 0019,2000,4180,F9A8,0000,1948   ;37055          J/CVTGF.0
                                ;37056
                                ;37057          ;11-----------------------------;
                                ;37058          STATE_K[.80],                   ;  SET NORMALIZE BIT
U 183B, 0000,003C,4180,F800,1404,7980   ;37059          J/CVTH
                                ;37060  =;END
                                ;37061  =00     ;00-----------------------------;
                                ;37062  CVTGF.0:                                ;  CALL SITE FOR UNPACK ROUTINE
                                ;37063          R[R15]_LC,                      ;  R15 GETS SRC LONG FRAC
                                ;37064          Q_D,                            ;  Q GETS SRC LONG FRAC
                                ;37065          SC_K[.10],
U 1948, 0010,0039,65E0,FAF8,0084,7B94   ;37066          CALL, J/UNPKG.0
                                ;37067
                                ;37068          ;01-----------------------------;  SRC = 0
                                ;37069          D_0, ALU_K[ZERO],               ;  SET DEST = 0
                                ;37070          NEZ_ALU.V&C_0,                  ;  SET Z, CLEAR OTHER CC
U 1949, FF18,003B,19F0,F847,0050,0300   ;37071          WRITE.DEST                      ;  GO AND STORE IT (F-FORMAT)
                                ;37072
                                ;37073          ;10-----------------------------;  SRC NE 0
                                ;37074  ;RETURN WITH:
                                ;37075  ;       SS = SD = SIGN BIT
                                ;37076  ;       ID[T1] = SRC FRAC <MS>
                                ;37077  ;       D = SRC FRAC <LS>
                                ;37078  ;       RC4 = SRC BIASSED EXPONENT
                                ;37079  ;       R15 = DST BIASSED EXPONENT
                                ;37080  ;       SC = LOW 10 BITS OF R15
                                ;37081  ;       ALU CC REFLECT R15
                                ;37082
U 194A, 0818,0038,3180,F800,0000,1BBD   ;37083          D_K[.40]                        ;  ROUNDING CONSTANT
                                ;37084  =;END
```

```
                                ;37085          ;--------------------------------;
U 1BBD, 0000,003C,C5F0,2C00,0000,1BC0   ;37086          Q_ID[T1]
                                ;37087
                                ;37088          ;--------------------------------;
                                ;37089          ALU_D+Q,                        ;  ADD 80 TO ROUND
                                ;37090          D_ALU,                          ;  STORE ROUNDED FRAC <MS> IN D
U 1BC0, 081D,0014,0180,F800,0050,1BC1   ;37091          N&Z_ALU.V&C_0                   ;  SET PSL CONDITION CODES
                                ;37092
                                ;37093          ;--------------------------------;
                                ;37094  CVTGF.01:       ; JOIN HERE FROM CVTHF-FLOWS
                                ;37095          ALU_R[R15],CLK.UBCC,            ;  CLOCK WHOLE EXPONENT
U 1BC1, 0000,1A3C,0180,FA78,0010,1835   ;37096          PSL.N?                          ;  DID ROUNDING PROPAGATE ?
                                ;37097
                                ;37098          ;--------------------------------;
                                ;37099  =01**    ;BRANCH ON PSL N-BIT
                                ;37100          ;01**----------------------------;
                                ;37101  CVTGF.1:
                                ;37102          D_D.LEFT, SI/ZERO,              ;  LEFT ADJUST FRACTION <MS>
                                ;37103          ALU_R[R15].AND.K[.FF00],        ;  ISOLATE UPPER BITS
                                ;37104          CLK.UBCC,                       ;  CLOCK THEM FOR OVERFLOW TEST
U 1835, 0518,1B34,4D80,FA78,0010,1B23   ;37105          ALU?, J/CVTGF.2                 ;  TEST FOR UNDERFLOW
                                ;37106
                                ;37107          ;11**----------------------------;
                                ;37108          R[R15]_LA+K[.1],CLK.UBCC,       ;  INCREMENT EXPONENT
                                ;37109          SC_SC+T,                        ;  INCREMENT LOW 10 BITS AS WELL
                                ;37110          D_D.RIGHT,                      ;  TO COMPENSATE FOR NEXT SHIFT
U 183D, 0618,0014,0580,FAF8,0090,D835   ;37111          J7CVTGF.1
                                ;37112  =;END
                                ;37113          ;--------------------------------;
                                ;37114  =0011   ;BRANCH ON ALU Z AND N-BITS
                                ;37115          ;0011----------------------------;
                                ;37116  CVTGF.2:
                                ;37117          EALU_SC,                        ;  GET EXPONENT
                                ;37118          D_PACK.FP,                      ;  PACK RESULT
                                ;37119          N&Z_ALU.V&C_0,WORD,             ;  CLOCK  CC ON EXPONENT AND SIGN
U 1B23, 0808,4138,0180,F800,0050,19FC   ;37120          Z?,J/CVTGF.3                    ;  TEST FOR OVERFLOW
                                ;37121
                                ;37122          ;0111----------------------------;
                                ;37123          D_0,ALU_K[ZERO],N&Z_ALU.V&C_0,  ;  ZERO EXPONENT on underflow
U 1B27, 0F18,0038,1980,F800,0050,1B35   ;37124          J7CVTGF.4
                                ;37125
                                ;37126          ;--------------------------------;
                                ;37127          D_0,ALU_K[ZERO],N&Z_ALU.V&C_0,  ;  NEGATIVE EXPONENT
U 1B2B, 0F18,0038,1980,F800,0050,1B35   ;37128          J7CVTGF.4                       ;  UNDERFLOW
                                ;37129  =;END
                                ;37130          ;--------------------------------;
                                ;37131  =0      ;BRANCH ON ALU Z-BIT
                                ;37132          ;0--------------------------------;  HIGH BITS NOT 0, OVERFLOW
U 19FC, 0018,0038,0180,F988,0000,12B4   ;37133  CVTGF.3:RC[T7]_K[.8],J/FLOAT.FAULT        ;  OVERFLOW FAULT
                                ;37134
                                ;37135          ;1--------------------------------;  HIGH BITS ZERO.OK
U 19FD, F000,003F,01F0,F847,0000,0300   ;37136          WRITE.DEST                      ;
```

```
                                        ;37137  =0101
                                        ;37138  CVTGF.4::0101-------------------------;  UNDERFLOW
U 1B35, 0000,003D,3DF0,2C00,0000,12AE   ;37139          Q_ID[PSL],CALL[PSLFU.A]       ;  TEST FU
                                        ;37140
                                        ;37141  =1101   ;1101-------------------------;  FU NOT SET, IGNORE UNDERFLOW
U 1B3D, F000,003F,01F0,F847,0000,0300   ;37142          WRITE.DEST                    ;
                                        ;37143
                                        ;37144          ;1111-------------------------;  FU SET, TAKE UNDERFLOW
U 1B3F, 0018,0038,F580,F9B8,0000,12B4   ;37145          RC[T7]_K[.A],J/FLOAT.FAULT    ;
```

```
                                ;37146  1447:   ;  ENTER HERE FOR CVTGH, OPCODE 56FD
                                ;37147          ;  EXPECTS: RC2 = SRC LONG EXP
                                ;37148          ; D = SRC LONG FRAC
                                ;37149
                                ;37150  CVTGH:  TRAP.ACC[1],                    ;  CALM DOWN THE ACC.
                                ;37151          CLR.SD&SS,
                                ;37152          ALU_K[.80],Q_ALU.LEFT3,         ;  Q GETS 400
U 1447, 00B8,00B8,40C7,F9A8,0000,1958   ;37153          RC[T5]_ALU.LEFT3                ;  RC5 GETS 400
                                ;37154
                                ;37155          ;00-----------------------------;
                                ;37156  =00     STATE_K[.7],                    ;  STATE-REGISTER SET TO 7
                                ;37157          SET.CC(INST),                   ;  CLEAR V
                                ;37158          Q_RC[T2],                       ;  Q GETS LONG EXP
U 1958, 0010,C039,5DC0,F910,1474,7B93   ;37159          CALL,J/UNPKG                    ;  UNPACK THE G-SRC
                                ;37160
                                ;37161          ;01-----------------------------;
U 1959, 0000,003C,0180,F800,0000,144D   ;37162          J/CLRH                          ;  0 SOURCE, CLEAR EVERYTHING
                                ;37163
                                ;37164          ;10-----------------------------;
                                ;37165  ; RETURN WITH:
                                ;37166  ;       SS = SD = SIGN
                                ;37167  ;       ID[T1] = SRC FRAC <MS>
                                ;37168  ;       D = SRC FRAC <LS>
                                ;37169  ;       RC4 = BIASSED EXP
                                ;37170  ;       R15 = 400-BIASSED EXP
                                ;37171  ;       SC = LOW 10 BITS OF R15
                                ;37172
                                ;37173          Q_K[.4000],                     ;  GET NEW BIAS
U 195A, 0018,0038,B1C0,FA78,0000,1BC2   ;37174          LAB_R[R15]                      ;  LATCH 400-EXP
                                ;37175  =;END
                                ;37176          ;-----------------------------;
                                ;37177          ALU_Q+LB,RC[T6]_ALU,            ;  RC6 GETS DST EXP BIASSED
U 1BC2, 000D,2014,C5F0,2DB0,0000,1BC3   ;37178          Q_ID[T1]
                                ;37179
                                ;37180          ;-----------------------------;
                                ;37181          ; NOW <Q,D> = FRACTION PARTS
                                ;37182          ;  RC6 = BIASSED EXP
                                ;37183          ;  SD = SIGN
                                ;37184          RC[T0]_Q,                       ;  RC0 GETS FRAC <MS>
                                ;37185          ID[T0]_D,                       ;  T0 GETS FRAC <LS>
U 1BC3, 0F01,203C,C1F8,3D80,0000,19C9   ;37186          D_0,Q_0
                                ;37187
                                ;37188          ;0****--------------------------;
                                ;37189  =0****
                                ;37190  CVTGH.W:        ;  ENTER HERE FROM CVTDH
                                ;37191          RC[T1]_D,ID[T1]_D,
                                ;37192          SC_ALU,                         ;  ALL GETS 0
U 19C9, 0001,003D,C580,3D88,0082,15A5   ;37193          CALL,J/ADDH.310
                                ;37194
                                ;37195          ;1****--------------------------;
U 19D9, F000,003F,01F0,F847,0000,1400   ;37196          WRITE.G.DEST,J/WRG              ;  WRITE THE H-FORMAT DEST
                                ;37197  =;END   ;-----------------------------;
```

```
                                    ;37198  14CD: ;  ENTER HERE FOR CVTHG, WITH OPCODE 76
                                    ;37199  CVTHG:
                                    ;37200          CLR.SD&SS,                      ;  INITIALIZE SIGN-BITS
                                    ;37201          TRAP.ACC[1],                    ;  CALM DOWN THE ACC.
U 14CD, 0000,00BC,4087,F800,1404,7BC4 ;37202        STATE_K[.80]                    ;  SET NORMALIZE-BIT
                                    ;37203
                                    ;37204          ;-------------------------------;
U 1BC4, 0000,003C,7980,F800,0084,7968 ;37205        SC_K[.30]                       ;  SET UP POINTER
                                    ;37206
                                    ;37207          ;00-----------------------------;
                                    ;37208  =00
                                    ;37209          CLR.SD&SS,                      ;  INITIALIZE SIGN-BITS
                                    ;37210          SC_SC.OR.K[.2],
                                    ;37211          FE_SC.OR.K[.2],                 ;  FE GETS .32 AS WELL
U 1968, 0000,003D,0987,F800,0184,363A ;37212        CA[L,J/UNPACK1H                 ;  UNPACK OPERAND IN RC2-ID3
                                    ;37213
                                    ;37214          ;01-----------------------------;
                                    ;37215  =01     D_0,ALU_0(A),RC[T1]_ALU,        ;  STORE 0 IN D AND RC1
                                    ;37216          N&Z_ALU.V&C_0,                  ;  SET PSL Z-BIT
U 1969, 0F03,003C,0180,F988,0050,1BB0 ;37217        J/WRITE.G                       ;  WRITE F OR D -FORMAT
                                    ;37218
                                    ;37219          ;11-----------------------------;
                                    ;37220  =11     ;RETURN WITH:
                                    ;37221          ;       RC2-ID3 = UNPACKED FRACTION
                                    ;37222          ;       RC6 = BIASSED EXPONENT
                                    ;37223          ;       R15 = UNBIASSED EXPONENT
                                    ;37224          ;       ALU CC REFLECT R15
                                    ;37225          ;       SC = LOW 10 BITS OF R15
                                    ;37226          STATE_FE,                       ;  NEED TO CLEAR STATE
U 196B, 0000,003C,C9F0,2C00,1400,7BC5 ;37227        Q_ID[T2]                        ;  Q GETS FRAC<AMS>
                                    ;37228  =;END
                                    ;37229          ;-------------------------------;
                                    ;37230          STATE_STATE.ANDNOT.FE,          ;  CLEAR STATE
                                    ;37231          K[.3FF],                        ;  NEED TO ROUND
U 1BC5, 0000,003C,8180,FA78,1400,5BC6 ;37232        LAB_R[R15]                      ;  LATCH UNBIASSED EXP
                                    ;37233
                                    ;37234          ;-------------------------------;
                                    ;37235          ALU_LA+K[.3FF]+1,               ;  BIAS EXPONENT
U 1BC6, 0018,0010,8180,FAF8,0010,1BC7 ;37236        R[R15]_ALU,CLK.UBCC             ;  WITH G-BIAS
                                    ;37237
                                    ;37238          ;-------------------------------;
U 1BC7, 0000,003C,0180,FA78,0000,1BC8 ;37239        LAB_R[R15]                      ;
```

```
                                        ;37240
                                        ;37241          ;-------------------------------;
                                        ;37242          RC[T7]_K[.1F00].LEFT3,SI/ZERO,  ;  CREATE MASK FOR HIGH BITS OF EXPONENT
U 1BC8, 00B8,1B38,9180,F9B8,0000,1B43   ;37243          ALU?                            ;  TEST N,Z BITS FOR UNDERFLOW
                                        ;37244
                                        ;37245  =0011
                                        ;37246          ;0011-------------------------;  NOT UNDERFLOW
U 1B43, 0000,003C,0180,F938,0000,1BC9   ;37247          LC_RC[T7],J/CVTHG.0             ;  PREFETCH MASK FOR NEXT TEST
                                        ;37248
                                        ;37249          ;0111-------------------------;  EXP=0,UNDERFLOW
U 1B47, 0000,003C,3DF0,2C00,0000,1B55   ;37250          Q_ID[PSL],J/UNFLG               ;  PREPARE TO TAKE FAULT
                                        ;37251
                                        ;37252          ;1011-------------------------;  EXP<0,UNDERFLOW
U 1B4B, 0000,003C,3DF0,2C00,0000,1B55   ;37253          Q_ID[PSL],J/UNFLG               ;  PREPARE TO TAKE FAULT
                                        ;37254  =;END
                                        ;37255  =0101
                                        ;37256  UNFLG:  ;0101-------------------------;  CALL SITE FOR TEST OF PSL<FU>
U 1B55, 0000,003D,0180,F800,0000,12AE   ;37257          CALL[PSLFU.A]                   ;
                                        ;37258  =1101
                                        ;37259          ;1101-------------------------;  RETURN HERE IF FU=0,IGNORE UNDERFL
U 1B5D, 0018,0038,1980,FAF8,0000,1A0D   ;37260          R[R15]_K[ZERO],J/CVTHG.1        ;  ZERO OUT EXP,REJOIN CODE
                                        ;37261
                                        ;37262          ;1111-------------------------;  RETURN HERE IF FU=1
U 1B5F, 0018,0038,F580,F9B8,0000,12B4   ;37263          RC[T7]_K[.A],J/FLOAT.FAULT      ;  TAKE FAULT
                                        ;37264
                                        ;37265  CVTHG.0:;-------------------------------;
U 1BC9, 0010,0034,0180,FA78,0010,1BCC   ;37266          ALU_R[R15].AND.LC,CLK.UBCC      ;  TEST HIGH BITS OF EXP
                                        ;37267
                                        ;37268          ;-------------------------------;
U 1BCC, 0000,013C,8180,F800,0000,1A0C   ;37269          K[.3FF],Z?                      ;
                                        ;37270  =0
                                        ;37271          ;0----------------------------- -;  HIGH BITS NOT 0,OVERFLOW
U 1A0C, 0018,0038,0180,F9B8,0000,12B4   ;37272          RC[T7]_K[.8],J/FLOAT.FAULT      ;
                                        ;37273
                                        ;37274  CVTHG.1:;1-------------------------------;  HIGH BITS ZERO,OK
                                        ;37275          ALU_Q+K[.3FF]+1,Q_ALU.          ;  ROUND LOW FRACTION
U 1A0D, 0019,2010,81C0,F800,0010,19E1   ;37276          CLK.UBCC                        ;  CLOCK CARRY
                                        ;37277
                                        ;37278  =0****  ;0****-------------------------;
                                        ;37279          SC_K[.FFF5],                    ;
                                        ;37280          D_RC[T2],                       ;  GET FRAC<MS>
U 19E1, 0810,0039,E180,F910,0084,7889   ;37281          CALL,J/PACKG                    ;  CALL PACKG-ROUTINE
                                        ;37282
                                        ;37283          ;1****-------------------------;
                                        ;37284  =1****  ;  RETURN WITH <D,RC1> = RESULT
U 19F1, F000,003F,01F0,F847,0000,1400   ;37285          WRITE.G.DEST
                                        ;37286          ;-------------------------------;
```

```
                                ;37287  =00
                                ;37288  CVTH:
                                ;37289          TRAP.ACC[1],                    ;  CALM DOWN THE ACC.
                                ;37290          SC_SC.OR.K[.2],
                                ;37291          FE_SC.OR.K[.2],                 ;  FE GETS .32 AS WELL
U 1980, 0000,00BD,0880,F800,0184,363A   ;37292          CALL,J/UNPACK1H                 ;  UNPACK OPERAND IN RC2-ID3
                                ;37293
                                ;37294          ;-------------------------------;
                                ;37295  =01     D_0,ALU_0(A),RC[T1]_ALU,        ;  STORE 0 IN D AND RC1
                                ;37296          N&Z_ALU.V&C_0,Q_0,              ;  SET PSL Z-BIT
U 1981, 0F03,003C,01F8,F988,0050,1A2C   ;37297          J/CVTHD.2C                      ;  WRITE F OR D -FORMAT
                                ;37298
                                ;37299          ;-------------------------------;
                                ;37300  =11     ;RETURN WITH:
                                ;37301          ;       RC2-ID3 = UNPACKED FRACTION
                                ;37302          ;       RC6 = BIASSED EXPONENT
                                ;37303          ;       R15 = UNBIASSED EXPONENT
                                ;37304          ;       ALU CC REFLECT R15
                                ;37305          ;       SC = LOW 10 BITS OF R15
                                ;37306          ; RC[T2],                       ;  D GETS FRAC0
                                ;37307          Q_ID[T2],                       ;  Q GETS FRAC1
U 1983, 0810,1B38,C9F0,2D10,0000,1B2D   ;37308          IR0?                            ;  TEST FOR CVTHF OR CVTHD
                                ;37309  =;END
                                ;37310          ;-------------------------------;
                                ;37311  =1101   ;BRANCH ON IR<0>
                                ;37312          ;1101-------------------------;
                                ;37313          DK/RIGHT,SI/ZERO,               ;  UNNORMALIZE FRAC<MS>
                                ;37314          Q_K[.40],LAB_R[R15],            ;  LATCH UNBIASSED EXPONENT
U 1B2D, 0618,0038,31C0,FA78,0000,1BCD   ;37315          J/CVTHF.2
                                ;37316
                                ;37317          ;1111-------------------------;
                                ;37318          FE_K[.18],LAB_R[R15],           ;  TOO BAD
U 1B2F, 0000,003C,7D80,FA78,0104,7BD0   ;37319          J/CVTHD.2
                                ;37320  =;END
                                ;37321          ;-------------------------------;
                                ;37322  CVTHF.2:
                                ;37323          K[.80],                         ;  GET SLOW CONSTANT
                                ;37324          ALU_D+Q,D_ALU,                  ;  ROUND FRAC0
U 1BCD, 081D,0014,4180,F800,0050,1BCE   ;37325          N&Z_ALU.V&C_0                   ;  CLOCK PSL N-BIT
                                ;37326
                                ;37327          ;-------------------------------;
                                ;37328          ALU_LA+K[.80],R[R15]_ALU,
                                ;37329          SC_ALU,                         ;  SC AND R15 GET BIASSED EXP
                                ;37330          CLR.UBCC,                       ;  CLOCK CC ON IT
U 1BCE, 0018,0014,4180,FAF8,0092,1BC1   ;37331          J/CVTGF.01                      ;  JOIN CVTGF-FLOWS
                                ;37332          ;-------------------------------;
                                ;37333
```

```
                                ;37334  CVTHD.2:
                                ;37335          SC_ALU,R[R15]_LA+K[.80],        ;  ADD IN BIAS
U 1BD0, 0018,0014,4180,FAF8,00D2,1BD1   ;37336          N&Z_ALU.V&C_0                   ;  CLEAR V&C
                                ;37337
                                ;37338          ;-------------------------------;
                                ;37339          Q_ALU,ALU_Q+K[.80],CLK.UBCC,    ;  ADD ROUND BIT TO LOW FRAC
U 1BD1, 0019,2014,41C0,F800,0010,1A79   ;37340          J/KJC.0                         ;
                                ;37341
                                ;37342  =0*     ;0*-----------------------------;
                                ;37343  KJC.1:  D_DAL.SC,SC_FE,                 ;  D GETS FRAC<L>,SC GETS EXP
U 1A78, 0D00,003C,0580,F800,0095,1BD3   ;37344          EALU_K[.1],CLK.UBCC,J/KJC.2     ;  CLEAR EALUCC'S
                                ;37345
                                ;37346          ;1*-----------------------------;
U 1A7A, 0819,0014,0580,F800,0191,1BD2   ;37347          D_D+K[.1],SC_FE,FE_SC,CLK.UBCC  ;  INC FRAC<H>,SC GETS EXP,FE GETS 24
                                ;37348  =;END
                                ;37349          ;-------------------------------;
                                ;37350          ALU_0+K[.1],CLK.UBCC,           ;  CLR C31,MUST WE INC EXP?
U 1BD2, 001B,0314,0580,FA78,0010,1A79   ;37351          LAB_R[R15],C31?                 ;
                                ;37352  =0*
                                ;37353  KJC.0:  ;0*-----------------------------;  ;  NO PACK RESULT <H>
                                ;37354          SC_FE,FE_SC,RC[T1]_PACK.FP,     ;  SC GETS 24,FE GETS EXP,PACK RESULT
U 1A79, 0008,0338,0180,F988,0181,1A78   ;37355          C31?,J/KJC.1                    ;  NEED TO INC FRAC <H> ?
                                ;37356
                                ;37357          ;1*-----------------------------;  ;  YES INC EXP BY 1
U 1A7B, 0018,0014,0580,FAF8,0080,DA79   ;37358          SC_SC+1,R[R15]_LA+K[.1],J/KJC.0 ;
                                ;37359  =;END
                                ;37360  KJC.2:  ;-------------------------------;
U 1BD3, 0018,0038,65E0,F800,0182,1BD4   ;37361          SC_K[.10].ALU,FE_SC,Q_D         ;  SC GETS 16 FE GETS EXP
                                ;37362
                                ;37363          ;-------------------------------;
U 1BD4, 0D10,0038,01C0,F908,0000,1BD5   ;37364          D_DAL.SC,Q_RC[T1]               ;  SWAP WORD,Q GETS RESULT<H>
                                ;37365
                                ;37366          ;-------------------------------;
                                ;37367          ALU_R[R15].ANDNOT.K[.FF],       ;  TEST HIGH BITS OF EXP
U 1BD5, 0018,0024,4980,FA78,0010,1BD6   ;37368          CLK.UBCC                        ;
                                ;37369
                                ;37370          ;-------------------------------;
U 1BD6, 0C01,003C,01E0,F908,0000,1BD7   ;37371          RC[T1]_D,Q_D,D_Q                ;  RC1,Q HAVE RESULT<L> D HAS RESULT<H>
                                ;37372
                                ;37373          ;-------------------------------;
U 1BD7, 0000,013C,0180,FA78,0010,1A18   ;37374          ALU_R[R15],CLK.UBCC,Z?          ;  TEST HIGH BITS OF EXP
                                ;37375
                                ;37376  =0      ;0-------------------------------;  ;  HIGH BITS NOT ZERO
U 1A18, 0000,1B3C,0180,F800,0000,1B63   ;37377          ALU?,J/CVTHD.4                  ;  FAULT FIND OUT WHICH ONE
                                ;37378
                                ;37379          ;1-------------------------------;  ;  HIGH BITS ZERO
U 1A19, 0000,013C,0180,F800,0000,1A2C   ;37380          Z?                              ;  TEST REST OF EXP
                                ;37381  =0
                                ;37382  CVTHD.2C:;0-------------------------------;  ;  FF>EXP>0
                                ;37383          ALU_D.ANDNOT.K[.1F],WORD,       ;
                                ;37384          N&Z_ALU,                        ;  SET CONDITION CODES
U 1A2C, F019,4027,8DF0,F847,0060,0300   ;37385          WRITE.DEST,J/WRD                ;  WRITE THE D-DEST
                                ;37386
                                ;37387          ;1-------------------------------;  ;  EXP=0
U 1A2D, 0000,003C,3DF0,2C00,0000,1B75   ;37388          Q_ID[PSL],J/UNFLHD              ;  UNDER FLOW
```

```
                                    ;37389  =0011
                                    ;37390  CVTHD.4::;0011----------------------------;  OVERFLOW
U 1B63, 0018,0038,0180,F9B8,0020,12B4 ;37391        SET.V,RC[T7]_K[.8],J/FLOAT.FAULT;  TAKE FAULT
                                    ;37392
                                    ;37393        ;0111----------------------------;  EXP=0,UNDERFLOW
U 1B67, 0000,003C,3DF0,2C00,0000,1B75 ;37394        Q_ID[PSL],J/UNFLHD               ;  ZERO OUT EXP,PREPARE TO TEST FU
                                    ;37395
                                    ;37396        ;1011----------------------------;  EXP<0,UNDERFLOW
U 1B6B, 0000,003C,3DF0,2C00,0000,1B75 ;37397        Q_ID[PSL],J/UNFLHD               ;  ZERO OUT EXP,PREPARE TO TEST FU
                                    ;37398  =;END
                                    ;37399  =0101
                                    ;37400  UNFLHD: ;0101----------------------------;  CALL SITE FOR PSL<FU> TEST
U 1B75, 0000,003D,0180,F800,0000,12AE ;37401        CALL[PSLFU.A]                    ;
                                    ;37402  =1101
                                    ;37403        ;1101----------------------------;  RETURN HERE IF FU=0,IGNORE UNDERFLOW
                                    ;37404        SC_K[ZERO],D_0,RC[T1]_K[ZERO],   ;  ZERO OUT ANSWER
U 1B7D, 0F18,0038,19F8,F988,0094,7A2C ;37405        Q_0,CLK.UBCC,J/CVTHD.2C          ;  CLOCK ALU CARRY
                                    ;37406
                                    ;37407        ;1111----------------------------;  FU IS SET,TAKE FAULT
U 1B7F, 0018,0038,F580,F9B8,0020,12B4 ;37408        SET.V,RC[T7]_K[.A],J/FLOAT.FAULT;
                                    ;37409
```

```
                                ;37410  .TOC    ''      G & H floating point  : CVTBG,CVTWG,CVTLG''
                                ;37411
                                ;37412  ;INPUT:
                                ;37413  ;               D = SRC
                                ;37414
                                ;37415  144F:   ;-------------------------------;
                                ;37416  CVTBG:  D_D.SXT[INST.DEP],               ;  SIGN EXTEND SRC TO ALL OF D
                                ;37417          SET.CC(INST),                   ;  SET CONDITON CODES
                                ;37418          Q_0,
                                ;37419          C[R.SD&SS,                      ;  INITIALIZE TO POSITIVE
U 144F, 0802,C03C,75FF,F800,00F4,7BD8   ;37420          SC_K[.20]                       ;  INITIAL EXPONENT
                                ;37421
                                ;37422          ;-------------------------------;
                                ;37423          ALU_K[.3FF], R[R15]_ALU,        ;  INITALIZE EXP TO BIAS
U 1BD8, 0018,0D38,8180,FAF8,0000,1990   ;37424          SIGNS?
                                ;37425          ;-------------------------------;
                                ;37426  =00     ;BRANCH ON D31 AND D NE 0
                                ;37427          ;00-----------------------------;
U 1990, 0F03,003C,0180,F988,0000,1BB0   ;37428          D_0,RC[T1]_0,J/WRITE.G          ;  RESULT IS 0
                                ;37429  =10     ;T0-----------------------------;
                                ;37430  CVTIG.1:
                                ;37431          D_DAL.NORM,                     ;  NORMALIZE FRACTION
                                ;37432          FE_SC-SHF.VAL,                  ;  ADJUST EXPONENT
                                ;37433          ALU_0+K[.4]+1,SC_ALU,           ;  GET LOW SHIFT COUNT
                                ;37434          LAB_R[R15],                     ;  LATCH EXPONENT
U 1992, 0E1B,0010,1180,FA78,018E,BBD9   ;37435          J/CVTIG.2
                                ;37436
                                ;37437          ;-------------------------------;
                                ;37438          D_0-D,                          ;  NEGATE INTEGER
                                ;37439          SD_NOT.SD,                      ;  SET SIGN BIT
U 1993, 081F,2000,0183,F800,0000,1992   ;37440          J/CVTIG.1
                                ;37441  =;END
                                ;37442          ;-------------------------------;
                                ;37443  CVTIG.2:
                                ;37444          SC_FE,                          ;  SC GETS UNBIASSED EXPONENT
                                ;37445          D_DAL.SC,                       ;  D GETS FRACTION <LS>
U 1BD9, 0D81,003C,0180,F990,0081,1BDC   ;37446          ALU_D, RC[T2]_ALU.RIGHT2        ;  START SHIFTING RIGHT
                                ;37447
                                ;37448          ;-------------------------------;
                                ;37449          ALU_LA+K[SC]+1,                 ;  GENERATE EXPONENT
                                ;37450          Q_ALU.LEFT3,                    ;  START SHIFTING IT LEFT
U 1BDC, 00B8,0010,1DC0,F910,0000,1BDD   ;37451          LC_RC[T2]                       ;  LATCH FRACTION <MS> IN LC
                                ;37452          ;-------------------------------;
                                ;37453          ALU_D.OXT[WORD], RC[T1]_ALU,    ;  STORE LONG FRAC IN RC1
U 1BDD, 0003,403C,01A8,F988,0000,1BDE   ;37454          Q_Q.LEFT                        ;  EXPONENT IS IN PLACE
                                ;37455          ;-------------------------------;
U 1BDE, 0850,0038,0180,F910,0000,1BE0   ;37456          ALU_RC[T2],D_ALU.RIGHT          ;  D GETS FRAC<MS>
                                ;37457          ;-------------------------------;
U 1BE0, 0808,0038,0180,F800,0000,1BE1   ;37458          D_PACK.FP                       ;  PACK FRACTION INTO D
                                ;37459          ;-------------------------------;
U 1BE1, 0819,0024,3980,F800,0000,1BE2   ;37460          D_D.ANDNOT.K[.7FF0]             ;  CLEAR EXPONENT BITS
                                ;37461          ;-------------------------------;
                                ;37462          D_D.OR.Q,                       ;  GENERATE LONG EXP
U 1BE2, F81D,0033,01F0,F847,0000,1400   ;37463          WRITE.G.DEST
                                ;37464          ;-------------------------------;
```

```
                                ;37465  .TOC    ""      G & H floating point  : CONVERT G FLOATING TO INTEGER''
                                ;37466
                                ;37467  ;INPUTS:
                                ;37468  ;               RC0 = SRC LONG EXP
                                ;37469  ;               D = SRC LONG FRAC
                                ;37470  1445:                                   ;  1400 + 45
                                ;37471          ;-------------------------------;
                                ;37472  CVTGI.00:
                                ;37473          Q_D,                            ;  Q GETS SRC LONG FRAC
                                ;37474          D_RC[TO],                       ;  D GETS SRC LONG EXP
U 1445, 0810,0039,01E0,F900,0000,19A0   ;37475          CALL, J/CVTGI                   ;  CALL CONVERT ROUTINE
                                ;37476
                                ;37477          ;-------------------------------;
                                ;37478  ;       RETURN WITH DST IN D
                                ;37479  1455:   ALU_D,
                                ;37480          N&Z_ALU,                        ;   THIS WORKS FOR LONG,WORD, AND BYTE
U 1455, 0001,DA3C,0180,F800,0060,10AC   ;37481          DT/INST.DEP,PSL.V?,J/GOUT.1     ;  USE GOUT TO CHECK FOR OVERFLOW
                                ;37482
                                ;37483  14C1:   ;-------------------------------;
                                ;37484  CVTRGL:
                                ;37485          Q_D,                            ;  Q GETS SRC LONG FRAC
                                ;37486          D_RC[TO],                       ;  D GETS SRC LONG EXP
U 14C1, 0810,0039,01E0,F900,0000,19A0   ;37487          CALL,J/CVTGI
                                ;37488          ;-------------------------------;
                                ;37489  14D1:
                                ;37490          ALU_0(A),Q_ALU.LEFT,SI/DIV,     ;  SHIFT HIGH BIT OF Q TO BIT 0
U 14D1, 0023,123C,02C0,F800,0000,041E   ;37491          SS?,J/CVTRDL.0
                                ;37492  =;END
                                ;37493  =00     ;00-------------------------------;
                                ;37494  CVTGI:  ; ENTER HERE FOR G-FORMAT SOURCE
                                ;37495  ;       < D , Q > = SRC
                                ;37496  CVTGI.1:
                                ;37497          Q_D,D_Q,                        ;  SWAP D AND Q
                                ;37498          ALU_K[.80],RC[T5]_ALU.LEFT3,    ;  LOAD BIAS IN RC5
                                ;37499          N&Z_ALU.V&C_0,                  ;  CLEAR PSL N& Z BITS
                                ;37500          SS_0&SD_0,                      ;  CLEAR SS AND SD
U 19A0, 0CB8,0039,41E7,F9A8,0050,1B93   ;37501          CALL, J7UNPKG                   ;  CALL ROUTINE TO UNPACK G-FORMAT SRC
                                ;37502
                                ;37503          ;01-------------------------------;
                                ;37504          D_0, Q_0, RC[T2]_0,             ;  SRC = 0
                                ;37505          N&Z_ALU.V&C_0,                  ;  SET PSL Z-BIT
                                ;37506          SC_R[ZERO],
U 19A1, 0F03,003E,19F8,F990,00D4,6010   ;37507          RETURN10
                                ;37508
                                ;37509          ;10-------------------------------;  SRC NE 0
                                ;37510  ;       SS = SD = SIGN
                                ;37511  ;       ID[T1] = SRC FRAC <MS>
                                ;37512  ;       D = SRC FRAC <LS>
                                ;37513  ;       RC4 = BIASSED EXPONENT
                                ;37514  ;       R15 = UNBIASSED EXPONENT
                                ;37515  ;       SC = LOW 10 BITS OF R15
                                ;37516  ;       ALU CC REFLECT R15
                                ;37517
U 19A2, 0000,003C,C5F4,2C00,0000,1BE3   ;37518          Q_ID[T1], SD_SS
                                ;37519  =;END
```

```
                                ;37520
                                ;37521          ;-------------------------------;
U 1BE3, 0C00,003C,01E0,F800,0000,1BE4   ;37522          Q_D,D_Q
                                ;37523
                                ;37524          ;-------------------------------;
                                ;37525          D_D.LEFT, SI/DIVD, Q_Q.LEFT,    ;  NORMALIZE FRACTION
                                ;37526          ALU_0+Q, RC[T2]_ALU.LEFT,       ;  RC2 GETS NORM.FRAC<LS>
                                ;37527          FE_SC,                          ;  BACK UP EXPONENT IN FE
U 1BE4, 053F,1B14,0028,F990,0100,1B83   ;37528          ALU?                            ;  TEST EXPONENT
                                ;37529          ;-------------------------------;
                                ;37530          ; D = FRACTION <MS>
                                ;37531          ; SC = LOW 10 BITS OF EXPONENT
                                ;37532          ; R15 = UNBIASSED EXPONENT
                                ;37533          ; RC[T2] = FRACTION <LS>
                                ;37534          ; SD = SIGN BIT OF DESTINATION
                                ;37535
                                ;37536          ; RETURNS:
                                ;37537          ; D = INTEGER
                                ;37538          ; Q<31> = ROUNDING BIT
                                ;37539          ; RC1 = FRACTION <MS>, RC2 = FRACTION <LS>
                                ;37540          ; EXCEPT FOR INTEGER SHIFT < 32.
                                ;37541          ; LOW 5 BITS OF SC = LOW 5 BITS OF EXPONENT
                                ;37542
                                ;37543          ;USES:
                                ;37544          ; FE FOR EXPONENT
                                ;37545
                                ;37546  =0011   ;BRANCH ON ALU Z AND N-BITS
                                ;37547          ;0011-----------------------------;  EXP > 0
                                ;37548  CVTGI.2:
                                ;37549          SC_SC-K[.20],FE_SC-K[.20],      ;  SUBTRACT 32. FROM EXPONENT
U 1B83, 0000,143C,75F8,F800,0184,B8A6   ;37550          Q_0,SC?, J/CVTGI.20             ;  SC CONTAINS ALL OF EXP, TEST IT
                                ;37551
                                ;37552          ;0111-----------------------------;  EXP = 0
                                ;37553          Q_D,                            ;  Q GETS FRACTION <MS>
                                ;37554          RC[T1]_D,                       ;  SAVE IT IN RC1 AS WELL
U 1B87, 0F01,003E,01E0,F988,0000,0010   ;37555          D_0, RETURN10                   ;  INT = 0, ROUND BIT = 1
                                ;37556
                                ;37557          ;1011-----------------------------;  EXP < 0
                                ;37558          RC[T1]_D,                       ;  SAVE FRAC <MS> IN RC1
                                ;37559          D_0,                            ;  INT = 0
U 1B8B, 0F01,003E,01F8,F988,0000,0010   ;37560          Q_0, RETURN10                   ;  ROUND BIT =0
                                ;37561  =;END
                                ;37562          ;-------------------------------;
                                ;37563  =110    ;BRANCH ON SC <9:5> NE 0
                                ;37564          ;110----------------------------;  1<= NUMBER < 2**31
                                ;37565  CVTGI.20:
                                ;37566          RC[T1]_D,                       ;  SAVE FRAC <MS> IN RC1
                                ;37567          Q_D,                            ;  AND IN Q
                                ;37568          D_DAL.SC,                       ;  D NOW GETS INTEGER
U 18A6, 0D01,123C,01E0,F988,0000,1B6E   ;37569          SS?, J/CVTGI.4                  ;  TEST SIGN
                                ;37570
                                ;37571          ;111----------------------------;  NUMBER >= 2**31
                                ;37572          Q_RC[T2],                       ;  RESTORE FRAC <LS>
U 18A7, 0010,1438,01C0,F910,0000,1886   ;37573          SC?                             ;  IS IT >= 2**63 ?
                                ;37574  =;END
```

```
                                ;37575
                                ;37576          ;--------------------------------;
                                ;37577  =110    ;BRANCH ON SC<9:5> NE 0
                                ;37578          ;10-------------------------------;  2**31 <= NUMBER < 2**63
                                ;37579          EALU_FE,
                                ;37580          D_DAL.SC,                          ;  CREATE INTEGER
                                ;37581          Q_D.OXT[BYTE].OR.PACK.FP,          ;  GENERATE FUNNY PACKED NUMBER
U 18B6, 0D0B,8030,01C0,F800,0000,7BE6   ;37582          J/CVTGI.3                          ;  IF INTEGER IS 80000000,SD=1,
                                ;37583                                             ;  THEN THIS NUMBER IS 00008000
                                ;37584          ;111------------------------------;  2**63<= NUMBER
U 18B7, 0000,003C,7580,F800,0084,BBE5   ;37585          SC_SC-K[.20]                       ;  SUBTRACT FOR LATER TEST
                                ;37586  =;END
                                ;37587          ;--------------------------------;
                                ;37588          RC[T2]_0,                          ;  CLEAR OUT FRACTION <LS>
                                ;37589          D_Q, Q_0,                          ;  SHIFT 32 TIMES LEFT
U 1BE5, 0C03,143C,01F8,F990,0000,18C6   ;37590          SC?                                ;  IS SC > 95 ?
                                ;37591
                                ;37592          ;--------------------------------;
                                ;37593  =110    ;BRANCH ON SC<9:5>
                                ;37594          ;110------------------------------;  2**63<= NUMBER < 2**95
                                ;37595          D_DAL.SC,                          ;  D GETS LOW BITS OF INTEGER
                                ;37596          RC[T1]_D,                          ;  SAVE MANTISSA
U 18C6, 0D01,123C,01F8,F988,0020,1B4E   ;37597          Q_0,SET.V,SS?,J/CVTGI.X            ;  SEE IF INT SHOULD BE PLUS OR MINUS
                                ;37598
                                ;37599          ;111------------------------------;
                                ;37600          D_0,Q_0, RC[T1]_0,                 ;  NUMBER >= 2**95
U 18C7, 0F03,003E,01F8,F988,0020,0010   ;37601          SET.V, RETURN10                    ; SET OVERFLOW BIT
                                ;37602
                                ;37603  =1110
                                ;37604  CVTGI.X:;1110-----------------------------;  POSITIVE
U 1B4E, 0000,003E,0180,F800,0000,0010   ;37605          RETURN10
                                ;37606
                                ;37607          ;1111-----------------------------;  MINUS
U 1B4F, 081F,2002,0180,F800,0000,0010   ;37608          D_0-D,RETURN10                     ;  COMPLIMENT INTEGER
                                ;37609
                                ;37610          ;--------------------------------;
                                ;37611  CVTGI.3:
                                ;37612          ALU_Q.XOR.K[.8000],CLK.UBCC,       ;  CHECK FOR MOST NEGATIVE NUMBER
U 1BE6, 0019,2020,45F8,F800,0010,1BE8   ;37613          Q_0
                                ;37614
                                ;37615          ;--------------------------------;
                                ;37616          RC[T1]_LC, Q_LC,SC_SC-K[.20],      ;  MOD SC SO ROUND BIT LINES UP
U 1BE8, 0010,0138,75C0,F988,0084,BA34   ;37617          Z?
                                ;37618
                                ;37619          ;--------------------------------;
                                ;37620  =0       ;BRANCH ON ALU Z-BIT
                                ;37621          ;0--------------------------------;
U 1A34, 0003,123C,0180,F990,0020,1B6E   ;37622          RC[T2]_0, SET.V, SS?, J/CVTGI.4
                                ;37623
                                ;37624          ;1--------------------------------;
U 1A35, 0003,123C,0180,F990,0000,1B6E   ;37625          RC[T2]_0, SS?, J/CVTGI.4
                                ;37626
```

```
                                ;37627          ;--------------------------------;
                                ;37628  =1110    ;BRANCH ON SS-BIT
                                ;37629          ;1110----------------------------;
                                ;37630  CVTGI.4:
                                ;37631          RC[T3]_D.SXT[INST.DEP],         ;  CONVERT LONGWORD TO TARGET DT
                                ;37632          Q_D, D_DAL.SC,                  ;  SAVE INTEGER IN Q, GET ROUND-BIT
U 1B6E, 0D02,C03C,01E0,F998,0000,1BEA   ;37633          J/CVTGI.5
                                ;37634
                                ;37635          ;1111----------------------------;  NEGATIVE
                                ;37636          Q_0-D,                          ;  NEGATE INTEGER
U 1B6F, 0D1F,2000,01C0,F800,0000,1BE9   ;37637          D_DAL.SC                        ;  D GETS ROUND BIT
                                ;37638
                                ;37639          ;--------------------------------;
U 1BE9, 0002,E03C,0180,F998,0000,1BEA   ;37640          RC[T3]_Q.SXT[INST.DEP]          ;  CONVERT LONGWORD TO TARGET DT
                                ;37641
                                ;37642          ;--------------------------------;
                                ;37643  CVTGI.5:
                                ;37644          LC_RC[T3], ALU_Q.XOR.LC,        ;  TEST UPPER BITS
                                ;37645          CLR.UBCC,                       ;  CLOCK UPPER BITS
U 1BEA, 0C11,2020,01EC,F918,0010,1BEB   ;37646          D_Q, Q_D                        ;  D GETS INTEGER, Q GETS ROUND BIT
                                ;37647
                                ;37648          ;--------------------------------;
U 1BEB, 0000,013C,0180,F800,0000,1A3C   ;37649          Z?                              ;  TEST FOR OVERFLOW DUE TO DATATYPE
                                ;37650
                                ;37651          ;--------------------------------;
                                ;37652  =0      ;BRANCH ON ALU Z-BIT
                                ;37653          ;0----------------------------;  INTEGER OVERFLOW
U 1A3C, 0000,003E,0180,F800,0020,0010   ;37654          SET.V, RETURN10
                                ;37655
                                ;37656          ;1----------------------------;  NO OVERFLOW
U 1A3D, 0000,003E,0180,F800,0000,0010   ;37657          RETURN10
                                ;37658          ;--------------------------------;
```

```
                                ;37659  .TOC    ''      G & H floating point  : ACBG''
                                ;37660
                                ;37661  ;INPUTS:
                                ;37662  ;               RC0 = LIMIT LONG EXP
                                ;37663  ;               Q = LIMIT LONG FRAC
                                ;37664  ;               RC2 = ADDEND LONG EXP
                                ;37665  ;               D = ADDEND LONG FRAC
                                ;37666
                                ;37667  ;TEMPORARIES:
                                ;37668  ;               RC6 = ADDEND LONG FRAC
                                ;37669  ;               T5 = LIMIT LONG EXP
                                ;37670  ;               T6 = LIMIT LONG EXP
                                ;37671  ;               T7 = INDEX ADDRESS
                                ;37672  ;               RC0 = INDEX LONG EXP
                                ;37673  ;               RC1 = INDEX LONG FRAC
                                ;37674  ;               RC7 = BRANCH ADDRESS
                                ;37675
                                ;37676
                                ;37677  16C5:   ;-------------------------------;  1600+C5
                                ;37678  ACBG:   TRAP.ACC[1],                    ;  RESET THE FPA
                                ;37679          RC[T6]_D,                       ;  SAVE ADDEND LONG FRAC IN RC6
                                ;37680          D_Q,                            ;  D GETS LIMIT LONG FRAC
                                ;37681          SC_K[ZERO],                     ;  CLEAR SC
                                ;37682          STATE_K[ZERO],                  ;  INITIALIZE STATE
U 16C5, 0C01,00BC,1880,F9B0,14F4,7BEC   ;37683          SET.CC(LONG)                    ;  DO THIS TO CLEAR V-BIT
                                ;37684
                                ;37685          ;-------------------------------;
                                ;37686          ID[T6]_D,                       ;  SAVE LIMIT LONG FRAC IN T6
                                ;37687          D_RC[T0],                       ;  READY TO SAVE LIMIT LONG EXP
U 1BEC, 0810,0038,D980,3D00,4000,1BED   ;37688          INTRPT.STROBE                   ;  SO MANY OPERANDS, STROBE INTRPT
                                ;37689
                                ;37690          ;-------------------------------;  C-BIT NOT SET
                                ;37691          ALU_D.AND.K[.7FF0],             ;  ISOLATE LIMIT EXPONENT
U 1BED, 0019,0034,3980,F800,0010,1840   ;37692          EALU_SC, CLK.UBCC               ;  CLOCK EXPONENT
                                ;37693  =10****0
                                ;37694          ;10****0--------------------------;
                                ;37695          ID[T5]_D,                       ;  SAVE LIMIT LONG EXP IN T5
                                ;37696          ALU_RC[T2], SS_ALU15,           ;  SS GETS SIGN OF ADDEND
                                ;37697          INTERRUPT.REQ?,                 ;  TEST FOR INTERRUPTS
U 1840, 0010,0E39,D581,3D10,0000,157E   ;37698          CALL,J/ASPCG                    ;  EVALUATE INDEX ADDRESS
                                ;37699
                                ;37700  =11****0;11****0-----------------------;  MEMORY OPERAND
                                ;37701          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;37702          ID[T7]_D,                       ;  SAVE INDEX ADDRESS IN T7
U 1860, 0000,00BC,DC80,3C00,0000,1BEE   ;37703          J/ACBG.2
                                ;37704
                                ;37705          ;11****1--------------------------;  REGISTER OPERAND
                                ;37706          TRAP.ACC[1],                    ;  RESET THE FPA
                                ;37707          Q_D,                            ;  SAVE INDEX LONG EXP IN RC0
                                ;37708          D_R(PRN+1),                     ;  GET INDEX LONG FRAC
U 1861, 0800,01BC,00E0,F860,0000,1A44   ;37709          Z?, J/ACBG.3                    ;  TEST FOR 0 LIMIT
                                ;37710          ;-------------------------------;
```

```
                                ;37711  ACBG.2:
                                ;37712          D[LONG]_CACHE.WCHK,              ;  READ INDEX LONG EXP
U 1BEE, 0000,003C,6580,5000,1404,7BF0   ;37713          STATE_K[.10]                    ;  SET MEMORY OPERAND FLAG IN STATE
                                ;37714
                                ;37715          ;--------------------------------;
U 1BF0, 0000,003C,01E0,F803,0000,1BF1   ;37716          Q_D, VA_VA+4                    ;  SAVE INDEX LONG EXP IN Q
                                ;37717
                                ;37718          ;--------------------------------;
                                ;37719          D[LONG]_CACHE.WCHK,             ;  READ INDEX LONG FRAC
U 1BF1, 0000,013C,0180,5000,0000,1A44   ;37720          Z?, J/ACBG.3                    ;  TEST FOR 0 LIMIT
                                ;37721
                                ;37722          ;--------------------------------;
                                ;37723  =0       ;BRANCH ON ALU Z-BIT
                                ;37724          ;0-------------------------------;
                                ;37725  ACBG.3: RC[T0]_Q,                       ;  SAVE INDEX LONG EXP IN T0
U 1A44, 0001,323C,0180,F980,0000,180C   ;37726          EALU?,J/ACBG.4                  ;
                                ;37727
                                ;37728          ;1-------------------------------;
                                ;37729          ALU_RC[T0],                     ;  GET LIMIT LONG EXP
U 1A45, 0010,0038,0180,F900,0800,1BF2   ;37730          CHK.FLT.OPR                     ;  CHECK FOR -0
                                ;37731  =;END
                                ;37732          ;--------------------------------;
                                ;37733          RC[T0]_Q,                       ;  SAVE INDEX LONG IN T0
U 1BF2, 0F01,203C,01E0,F980,0000,1BF3   ;37734          Q_D, D_0                        ;  Q GETS INDEX LONG FRAC
                                ;37735
                                ;37736          ;--------------------------------;
U 1BF3, 0000,003C,D580,3C00,0000,1BF4   ;37737          ID[T5]_D                        ;  CLEAR LIMIT LONG EXP
                                ;37738
                                ;37739          ;--------------------------------;
                                ;37740          ID[T6]_D,                       ;  CLEAR LIMIT LONG FRAC
U 1BF4, 0C00,123C,D980,3C00,0000,180C   ;37741          D_Q,EALU?                       ;
                                ;37742
                                ;37743          ;--------------------------------;
                                ;37744  =1*0     ;BRANCH ON SS
                                ;37745          ;1*0-------------------------------;
                                ;37746  ACBG.4:
                                ;37747          RC[T3]_D,                       ;  SAVE INEX LONG FRAC IN RC3
                                ;37748          Q_IB.BDEST, PC_PC+1,            ;  GET BRANCH DISPLACEMENT
U 180C, 7001,0B3C,01F0,F99C,0000,1988   ;37749          IB.TEST?, J/ACBG.5              ;  WAIT UNTIL IT ARRIVES
                                ;37750
                                ;37751          ;1*1-------------------------------;
                                ;37752          STATE_STATE.OR.K[.20],          ;  REMEMBER THAT ADDEND IS NEGATIVE
                                ;37753          RC[T3]_D,                       ;  SAVE INEX LONG FRAC IN RC3
                                ;37754          Q_IB.BDEST, PC_PC+1,            ;  GET BRANCH DISPLACEMENT
U 180D, 7001,0B3C,75F0,F99C,1404,3988   ;37755          IB.TEST?, J/ACBG.5              ;  WAIT UNTIL IT ARRIVES
                                ;37756  =;END
```

```
                                ;37757  =00     ;--------------------------------;
                                ;37758  ACBG.5:
U 19B8, 0000,003D,0180,F800,0000,0E64   ;37759          CALL,J/IB.TBM                   ;  REFILL TB
                                ;37760
                                ;37761          ;--------------------------------;
U 19B9, 0000,003D,0180,F800,0000,0B80   ;37762          CALL, J/IB.ERR                  ;  SERVE ERROR
                                ;37763
                                ;37764          ;--------------------------------;
                                ;37765          Q_IB.BDEST,                     ;  GET BRANCH DISPLACEMENT
U 19BA, 7000,0B3C,01F0,F800,0000,19B8   ;37766          IB.TEST?, J/ACBG.5
                                ;37767
                                ;37768          ;--------------------------------;
                                ;37769          RC[T7]_Q+PC+1,                  ;  CALCULATE BRANCH ADDRESS
                                ;37770          PC_PC+T,                        ;  PC GETS INCREMENTED TWICE
                                ;37771          Q_D,                            ;  COPY INDEX LONG FRAC FOR SWAP
                                ;37772          CLR.IB.SPEC,                    ;  CLEAR FIRST BYTE OF BDEST
U 19BB, D015,2010,65E0,F9BC,0084,7BF5   ;37773          SC_K[.10]                       ;  SHIFT COUNT FOR WORD SWAP
                                ;37774  =;END
                                ;37775          ;--------------------------------;
U 1BF5, 0000,003C,1180,F800,1404,3848   ;37776          STATE_STATE.OR.K[.4]            ;  THIS TELLS PACKG TO LEAVE C-BIT ALONE
                                ;37777  =0**00
                                ;37778          ;0**00----------------------------;
                                ;37779          ALU_K[.FFF9],SS_ALU15,          ;  SET SIGN-BIT
                                ;37780          D_DAL.SC,                       ;  SWAP INDEX LONG FRAC
                                ;37781          SC_K[.FFF9],                    ;  SC GETS -7
                                ;37782          CLR.IB.SPEC,                    ;  CLEAR 2. BYTE OF BDEST
U 1848, DD18,0039,BD81,F800,0084,7AF2   ;37783          CALL, J/UNPACKG                 ;  G FORMAT UNPACK ROUTINE
                                ;37784
                                ;37785          ;0**01----------------------------;  INDEX = 0
                                ;37786          Q_RC[T2],                       ;  Q GETS ADDEND LONG EXP
U 1849, 0010,0038,01C0,F910,0000,1BF6   ;37787          J7ACBG.6A                       ;
                                ;37788
                                ;37789          ;0**10----------------------------;  ADDEND = 0
                                ;37790          Q_RC[T0],                       ;  Q GETS INDEX LONG EXP
U 184A, 0010,0038,01C0,F900,0000,1BF8   ;37791          J7ACBG.6B                       ;
                                ;37792
                                ;37793  =0**11  ;0**11----------------------------;  BOTH OPERANDS NON-ZERO
                                ;37794          LC_RC[T4],Q_ID[T2],             ;  LATCH EXPONENT
U 184B, 0000,1B3D,C9F0,2D20,0000,19E3   ;37795          ALU?,CALL, J/ADDG.10            ;  CALL ADDG SUBROUTINE
                                ;37796
                                ;37797          ;1**11----------------------------;
                                ;37798  =1**11
                                ;37799          Q_ID[T7],                       ;  GET ADDRESS OF INDEX IF MEMORY
                                ;37800          LC_RC[T1],                      ;  LATCH INDEX LONG FRAC
U 185B, 0000,163C,DDF0,2D08,0000,1912   ;37801          STATE4?, J/ACBG.6               ;  MEMORY OR REGISTER ?
```

```
                                ;37802
                                ;37803          ;--------------------------------;
                                ;37804  ACBG.6A:
                                ;37805          ALU_Q.ANDNOT.K[.F],             ;  ISOLATE SIGN AND EXPONENT BITS
U 1BF6, 0019,6024,6180,F800,0070,1BF7   ;37806          SET.CC(WORD)                    ;  SET CC
                                ;37807
                                ;37808          ;--------------------------------;
                                ;37809          Q_ID[T7],LC_RC[T1],             ;  LOAD INDEX ADDRESS AND ADDEND FRAC
U 1BF7, 0000,163C,DDF0,2D08,0000,1912   ;37810          STATE4?,J/ACBG.6                ;  MEM OR REG?
                                ;37811
                                ;37812  ACBG.6B:
                                ;37813          ;--------------------------------;
                                ;37814          ALU_Q.ANDNOT.K[.F],             ;  ISOLATE SIGN AND EXP
U 1BF8, 0019,6024,6180,F800,0070,1BF9   ;37815          SET.CC(WORD)                    ;  SET CC
                                ;37816
                                ;37817          ;--------------------------------;
U 1BF9, 0000,003C,DDF0,2D18,0000,1BFA   ;37818          Q_ID[T7],LC_RC[T3]              ;  LOAD INDEX ADDRESS AND LONG FRAC
                                ;37819
                                ;37820          ;--------------------------------;
                                ;37821          RC[T1]_LC,                      ;  PUT NEW INDEX FRAC INTO RC1
U 1BFA, 0010,1638,0180,F988,0000,1912   ;37822          STATE4?,J/ACBG.6                ;  MEM OR REG?
                                ;37823
                                ;37824          ;--------------------------------;
                                ;37825  =10      ;BRANCH ON BIT 4 OF STATE
                                ;37826          ;10--------------------------------;
                                ;37827  ACBG.6: R(PRN)_D, SS_ALU15,             ;  STORE INDEX LONG EXP
                                ;37828          ID[T0]_D,                       ;  SAVE INDEX LONG EXP IN T0
U 1912, 0001,003C,C181,3CD8,0000,1543   ;37829          J/ACBG.7
                                ;37830
                                ;37831          ;11--------------------------------;
                                ;37832          VA_Q,                           ;  RELOAD ADDRESS OF INDEX
U 1913, 0001,203C,C180,3C00,0200,1BFB   ;37833          ID[T0]_D                        ;  SAVE INDEX LONG EXP IN T0
                                ;37834  =;END
                                ;37835          ;--------------------------------;
                                ;37836          CACHE_D[INST.DEP],              ;  STORE INDEX LONG EXP
U 1BFB, 0001,C03C,0181,3000,0000,1BFC   ;37837          ALU_D,SS_ALU15                  ;  SAVE SIGN
                                ;37838
                                ;37839          ;--------------------------------;
                                ;37840          VA_VA+4,                        ;  ADVANCE ADDRESS
                                ;37841          D_LC,                           ;  D GETS INDEX LONG FRAC
U 1BFC, 0810,0038,D5F0,2C03,0000,1BFD   ;37842          Q_ID[T5]                        ;  GET LIMIT LONG EXP FOR COMAPRE
                                ;37843
                                ;37844          ;--------------------------------;
                                ;37845          CACHE_D[LONG],                  ;  STORE INDEX LONG FRAC
U 1BFD, 0000,163C,0180,3000,0000,1851   ;37846          STATE5?, J/ACBG.8               ;  TEST ADDEND SIGN
```

```
                                ;37847
                                ;37848          ;--------------------------------;
                                ;37849  ACBG.7: R(PRN+1)_LC,                      ;  STORE INDEX LONG FRAC
                                ;37850          Q_ID[T5],                       ;  Q GETS LIMIT LONG EXP
U 1543, 0010,1638,D5F0,2CE0,0000,1851   ;37851          STATE5?                         ;  TEST ADDEND SIGN
                                ;37852
                                ;37853          ;--------------------------------;
                                ;37854  =01     ;BRANCH ON BIT 5 OF STATE
                                ;37855          ;01------------------------------;
                                ;37856  ACBG.8:
                                ;37857          RC[T3] Q,                       ;  PUT LIMIT WHERE ACBF WANTS IT
                                ;37858          Q_ID[T0],                       ;  GET INDEX TO
U 1851, 0001,203C,C1F0,2D98,0000,1544   ;37859          J/ACBG.60                       ;  JOIN ACBF ROUTINE
                                ;37860
                                ;37861          ;11------------------------------;
                                ;37862          ALU_K[.8000],                   ;  SET ALU15
                                ;37863          SS_SS.XOR.ALU15&SD_ALU15,       ;  COMPLEMENT INDEX SIGN IN SS
U 1853, 0018,0038,4585,F800,0000,1851   ;37864          J/ACBG.8
                                ;37865  ;END
                                ;37866          ;--------------------------------;
U 1544, 0000,1A3C,1180,F800,1404,5B8D   ;37867  ACBG.60:STATE_STATE.ANDNOT.K[.4],PSL.V? ;  CLEAR STATE2
                                ;37868
                                ;37869          ;--------------------------------;
                                ;37870  =1101   ;BRANCH ON PSL V-BIT
                                ;37871          ;1101----------------------------;
                                ;37872          D_Q.XOR.RC[T3],                 ;  D GETS DIFF. BETWEEN INDEX AND LIMIT
                                ;37873          WORD, CLK.UBCC,                 ;  TESTING ONLY SIGN, EXP, AND MSB
                                ;37874          SD_SS,                          ;  SD GETS INDEX SIGN
                                ;37875          FE_K[.1], CLK.UBCC,             ;  CLEAR EALU CC
U 1B8D, 0811,6020,0584,F918,0114,7057   ;37876          J/ACBF.00                       ;  JOIN ACBF-ECO
                                ;37877
                                ;37878          ;1111----------------------------;
U 1B8F, C000,003C,0180,F804,4000,0062   ;37879          CLR.IB.OPC, PC_PC+1, J/IRD      ;  DO NOT BRANCH
                                ;37880  =;END   ;--------------------------------;
```

```
;37881  .TOC    ""      G & H floating point  : HUGE FLOATING NOTATION"'
;37882
;37883  ; THE FOLLOWING NOTATION IS USED IN COMMENTS BELONGING TO THE MICROCODE FOR THE
;37884  ; IMPLEMENTATION OF THE HUGE INSTRUCTION SET ON THE VAX11/780
;37885  ;  IN ORDER OF SIGNIFICANCE, THE PACKED LONGWORDS ARE NAMED:
;37886  ;       OP LONG EXP
;37887  ;       OP LONG FRAC1
;37888  ;       OP LONG FRAC2
;37889  ;       OP LONG FRAC3
;37890
;37891  ; AFTER EVALUATION OF TWO OPERANDS,
;37892  ; THE VARIOUS LONGWORDS ARE STORED AS FOLLOWS, IN ORDER OF SIGNIFICANCE:
;37893
;37894  ;  OP1: RC0, ID[T0], RC1, ID[T1]
;37895  ;  OP2: RC2, ID[T2], RC3, ID[T3]
;37896  ; IN ADDITION,
;37897  ;       D = OP1 LONG EXP
;37898
;37899  ;AFTER UNPACKING, THE OPERANDS ARE STORED AS FOLLOWS:
;37900  ;  OP1:
;37901  ;       EXPONENT IN RC4
;37902  ;       RC0 = OP1 FRAC <VMS> (VERY MOST SIGNIFICANT)
;37903  ;       T0 = OP1 FRAC <AMS> (ALMOST MOST SIGNIFICANT)
;37904  ;       RC1 = OP1 FRAC <ALS> (ALMOST LEAST SIGNIFICANT)
;37905  ;       T1 = OP1 FRAC <VLS> (VERY LEAST SIGNIFICANT)
;37906
;37907  ;  OP2:
;37908  ;       EXPONENT IN RC6
;37909  ;       RC2 = OP2 FRAC <VMS>
;37910  ;       T2 = OP2 FRAC <AMS>
;37911  ;       RC3 = OP2 FRAC <ALS>
;37912  ;       T3 = OP2 FRAC <VLS>
;37913
;37914  ;WHEN WRITING DESTINATION, THE PACKED OPERANDS SHOULD BE AS FOLLOWS:
;37915  ;  DST:
;37916  ;       RC[T0] = DST LONG EXP
;37917  ;       ID[T0] = DST LONG FRAC1
;37918  ;       RC[T1] = DST LONG FRAC2
;37919  ;       ID[T1] = DST LONG FRAC3
;37920
;37921  ;STATE REGISTER USAGE:
;37922  ;       UNPACK ROUTINE USES STATE<4> FOR LOOP COUNT.
;37923  ;       STATE<1> SIGNALS EMOD
```

```
                                ;37924  .TOC    "       G & H floating point  : FLOATING H FORMAT COMPARE"
                                ;37925
                                ;37926  ;INPUTS:
                                ;37927  ;       RC0-ID[T1] CONTAINS OPERAND 1
                                ;37928  ;       RC2-ID[T3] CONTAINS OPERAND 2
                                ;37929
                                ;37930  1482:
                                ;37931  CMPH:   ;-------------------------------;
                                ;37932          D_RC[T0],                       ;  GET OP1 LONG EXP
U 1482, 0810,0038,0181,F900,0000,1546   ;37933          SS_ALU15                        ;  SS GETS SIGN
                                ;37934
                                ;37935          ;-------------------------------;
                                ;37936          ALU_D.ANDNOT.K[.8000],          ;  ISOLATE EXPONENT
U 1546, 0019,4024,4580,F800,0010,19D0   ;37937          WORD, CLK.UBCC                  ;  CLOCK EXPONENT
                                ;37938
                                ;37939          ;00-------------------------------;
U 19D0, 0000,003D,0180,F800,0000,1548   ;37940  =00     CALL,J/CMPH.S                   ;  CALL SUBROUTINE
                                ;37941
                                ;37942          ;01-------------------------------;
                                ;37943  CMPH.A: ALU_RC[T0],                     ;  SET PSL CC ON OP1
                                ;37944          N&Z_ALU.V&C_0,WORD,             ;  CLOCK IT ON SIGN AND EXP
U 19D1, C010,4038,0180,F904,4050,0062   ;37945          CLR.IB.OPC,PC_PC+1,J/IRD        ;  FINISHED
                                ;37946
                                ;37947          ;10-------------------------------;
U 19D2, 0000,123C,0182,F800,0000,1B9E   ;37948          SS_SD.SS?,J/CMPH.C              ;  ABSOLUTE VALUES ARE EQUAL,TEST SIGNS
                                ;37949
                                ;37950  CMPH.B: ;11-------------------------------;
                                ;37951          ALU_R[R15].XOR.K[.8000],        ;  SET CC ON -OP2
                                ;37952          N&Z_ALU.V&C_0, WORD,            ;  CLOCK PSL CC
U 19D3, C018,4020,4580,FA7C,4050,0062   ;37953          CLR.IB.OPC, PC_PC+1, J/IRD
                                ;37954  =;END
                                ;37955  =1110
                                ;37956  CMPH.C: ;1110-------------------------------;  EQUAL
                                ;37957          ALU_0(A),N&Z_ALU.V&C_0,         ;  EQUALITY
U 1B9E, C003,003C,0180,F804,4050,0062   ;37958          CLR.IB.OPC,PC_PC+1, J/IRD
                                ;37959
                                ;37960          ;1111-------------------------------;  SIGNS ARE DIFFERENT
U 1B9F, 0000,123C,0180,F800,0000,1BAE   ;37961          SS?                             ;  FIND OUT WHICH ONE IS POSITIVE
                                ;37962
                                ;37963  =1110   ;1110-------------------------------;
U 1BAE, 0000,003C,0180,F800,0000,19D3   ;37964          J/CMPH.B                        ;  OP2 IS POS
                                ;37965
                                ;37966          ;1111-------------------------------;
U 1BAF, 0000,003C,0180,F800,0000,19D1   ;37967          J/CMPH.A                        ;  OP1 IS POS
```

```
                                ;37968  CMPH.S: ;-------------------------------;
                                ;37969          STATE_STATE.ANDNOT.K[.F],       ;  CLEAR FIRST FOUR BITS OF STATE
                                ;37970          Q_RC[T2],                       ;  Q GETS OP2 LONG EXP
                                ;37971          SS_SS.XOR.ALU15&SD_ALU15,       ;  TAKE XOR OF SIGN BITS
U 1548, 0010,0138,61C5,F910,1404,5A4C   ;37972          Z?                              ;  IS OP1 = 0?
                                ;37973
                                ;37974          ;-------------------------------;
                                ;37975  =0      ;BRANCH ON ALU Z-BIT
                                ;37976          ;0------------------------------;
                                ;37977  CMPH0:  ALU_Q.ANDNOT.K[.8000],          ;  ISOLATE OP2 EXPONENT
                                ;37978          CLK.UBCC,WORD,                  ;  CLOCK OP2 EXPONENT
U 1A4C, 0019,6024,4580,F800,0010,1549   ;37979          J/CMPH01
                                ;37980
                                ;37981          ;1------------------------------;
                                ;37982          ALU_D.AND.K[.8000],             ;  ISOLATE SIGN BIT
                                ;37983          CHK.FLT.OPR,                    ;  CHECK FOR -0
U 1A4D, 0019,0034,4580,F800,0800,1A4C   ;37984          J/CMPH0
                                ;37985  =;END
                                ;37986          ;-------------------------------;
U 1549, 0001,203C,0180,FAF8,0000,154A   ;37987  CMPH01: ALU_Q,R[R15]_ALU                ;  SAVE OP2 LONG EXP IN R15
                                ;37988
                                ;37989          ;-------------------------------;
                                ;37990          D_D.ANDNOT.K[.8000],            ;  CLEAR SIGN BIT
U 154A, 0819,0124,4580,F800,0000,1A54   ;37991          Z?                              ;  TEST FOR OP2 = 0
                                ;37992
                                ;37993          ;-------------------------------;
                                ;37994  =0      ;BRANCH ON ALU Z-BIT
                                ;37995          ;0------------------------------;
                                ;37996          Q_Q.ANDNOT.K[.8000],            ;  CLEAR SIGN BIT OF LIMIT
U 1A54, 0019,2024,45C0,F800,0000,154B   ;37997          J/CMPH1                         ;  BRANCH ON SS REMOVED
                                ;37998
                                ;37999          ;1------------------------------;
                                ;38000          ALU_Q.AND.K[.8000],             ;  ISOLATE SIGN BIT
                                ;38001          CHK.FLT.OPR,                    ;  CHECK FOR -0
U 1A55, 0019,2036,4580,F800,0800,0001   ;38002          RETURN1                         ;
                                ;38003  =;END
                                ;38004          ;-------------------------------;
U 154B, 001D,4000,0180,F800,0010,154C   ;38005  CMPH1:  ALU_D-Q, WORD, CLK.UBCC        ;  COMPARE EXPONENTS, OP1 - OP2
                                ;38006
                                ;38007          ;-------------------------------;
                                ;38008          STATE_STATE.OR.K[.1],           ;  STATE GETS 1, TO DO ID[TO] NEXT
                                ;38009          ALU_Q-D,LONG, CLK.UBCC,         ;  COMPARE HIGH FRACTION BITS,OP2-OP1
U 154C, 001D,3B00,0580,F800,1414,3883   ;38010          ALU?, J/CMPH2                   ;  TEST EXPONENT-DIFFERENCE, OP1-OP2
                                ;38011
                                ;38012          ;-------------------------------;
U 154E, 0000,003E,0180,F800,0000,0003   ;38013          RETURN3
                                ;38014
```

```
                                ;38015          ;-------------------------------;
                                ;38016  =0011   ;BRANCH ON ALU N&Z BITS
                                ;38017          ;0011---------------------------;
U 1BB3, 0000,003E,0180,F800,0000,0001   ;38018  CMPH2:  RETURN1                  ;  SET PSL ON OP1
                                ;38019
                                ;38020          ;0111---------------------------;
                                ;38021          SC_K[.30],                      ;  START WITH ID[TO]
U 1BB7, 0000,1B3C,7980,F800,0084,7BDA   ;38022  ALU?,J/CMPHL.3                   ;  BRANCH ON OP2-OP1
                                ;38023
                                ;38024          ;1011---------------------------;
U 1BBB, 0000,003E,0180,F800,0000,0003   ;38025  RETURN3                         ;  SET PSL ON -OP2
                                ;38026  =;END
                                ;38027          ;-------------------------------;
                                ;38028  =010    ;BRANCH ON STATE BITS 0 AND 2
                                ;38029          ;010----------------------------;
                                ;38030  CMPHL.0:ALU_D.0XT[WORD], D_ALU,         ;  ISOLATE WORD PART
                                ;38031          ID[T9] ,                        ;  SAVE LONGWORD IN T9
U 1942, 0803,403C,E580,3C00,0000, ›51   ;38032  J/CMPHL.02
                                ;38033
                                ;38034          ;011----------------------------;
                                ;38035          ALU_Q.0XT[WORD],D_ALU,          ;  ISOLATE WORD PART
                                ;38036          Q_ID(SC),                       ;  Q GETS OP2 FRACTION PART
                                ;38037          SC_SC-K[.2],                    ;  POINT TO OP1 AGAIN
U 1943, 0803,603C,09F0,2400,0084,B550   ;38038  J/CMPHL.01
                                ;38039
                                ;38040          ;110----------------------------;
U 1946, 0000,003E,0180,F800,0000,0002   ;38041  RETURN2                         ;  EQUALITY
                                ;38042  =;END
                                ;38043          ;-------------------------------;
                                ;38044  CMPHL.01:ALU_Q.0XT[WORD]-D,             ;  COMPARE LOW WORD FIRST, OP2-OP1
                                ;38045          CLK.UBCC,WORD,                  ;  CLOCK DIFFERENCE
                                ;38046          D_Q,Q_ID(SC),                   ;  RETRIEVE ORIGINAL OP1, D GETS OP2
                                ;38047          SC_SC+1,                        ;  POINT TO NEXT OPERANDS
U 1550, 0C1F,6000,01F0,2400,0090,D553   ;38048  J/CMPHL.2
                                ;38049
                                ;38050          ;-------------------------------;
U 1551, 0010,0038,09C0,F830,0084,B552   ;38051  CMPHL.02:Q_RC(SC),SC_SC-K[.2]     ;  Q GETS OP2 FRACTION PART
                                ;38052
                                ;38053          ;-------------------------------;
                                ;38054          ALU_Q.0XT[WORD]-D,              ;  COMPARE WORD PARTS, OP2-OP1
                                ;38055          CLK.UBCC,WORD,                  ;  CLOCK DIFFERENCE
                                ;38056          D_Q,Q_ID[T9],                   ;  RETRIEVE WHOLE OP1 LONGWORD
U 1552, 0C1F,6000,E5F0,2C00,0010,1553   ;38057  J7CMPHL.2
                                ;38058
```

```
                                ;38059          ;-------------------------------;
                                ;38060  CMPHL.2:ALU_D-Q,LONG,CLK.UBCC,          ;  CLOCK NEXT WORD DIFFERENCE, OP2-OP1
                                ;38061          STATE_STATE+1,                  ;  GET TO NEXT STATE
U 1553, 001D,1B00,0180,F800,1410,DBCA   ;38062          ALU?                    ;  BRANCH ON WORD DIFF, OP1-OP2
                                ;38063
                                ;38064          ;-------------------------------;
                                ;38065  =1010   ;BRANCH ON ALU Z AND C BITS
                                ;38066          ;1010---------------------------; BORROW ON OP2-OP1
U 1BCA, 0000,003E,0180,F800,0000,0001   ;38067          RETURN1                 ;  SET PSL CC ON OP1
                                ;38068
                                ;38069          ;1011---------------------------; NO BORROW ON OP2-OP1
U 1BCB, 0000,003E,0180,F800,0000,0003   ;38070          RETURN3                 ;  SET PSL ON -OP2
                                ;38071
                                ;38072  =1111   ;1111---------------------------;
U 1BCF, 0000,1B3C,0180,F800,0000,1BDA   ;38073          ALU?                    ;  TEST NEXT WORD DIFFERENCE
                                ;38074  =;END
                                ;38075          ;-------------------------------;
                                ;38076  =1010   ;BRANCH ON ALU Z AND C BITS
                                ;38077          ;1010---------------------------;  BORROW ON OP2-OP1
U 1BDA, 0000,003E,0180,F800,0000,0001   ;38078  CMPHL.3:RETURN1                  ;  SET PSL CC ON OP1
                                ;38079
                                ;38080          ;1011---------------------------; NO BORROW ON OP2-OP1
U 1BDB, 0000,003E,0180,F800,0000,0003   ;38081          RETURN3                 ;  SET PSL CC ON -OP2
                                ;38082  =1111
                                ;38083          ;1111---------------------------; EQUALITY
                                ;38084          D_RC(SC),                       ;  GET NEXT OP1 FRACTION PART
                                ;38085          Q_ID(SC),                       ;  FROM RC OR ID
                                ;38086          SC_SC+K[.2],                    ;  POINT TO OP2 FRACTION
U 1BDF, 0810,1738,09F0,2430,0084,9942   ;38087          STATE3-0?,J/CMPHL.0
                                ;38088  =;END   ;-------------------------------;
```

```
                                ;38089
                                ;38090  .TOC    ""      G & H floating point  : MOVE H FORMAT FLOATING POINT NUMBERS""
                                ;38091
                                ;38092
                                ;38093  ; INPUTS:
                                ;38094  ;               RC[TO] = OP LONG EXP
                                ;38095  ;               ID[TO] = OP LONG FRAC1
                                ;38096  ;               RC[T1] = OP LONG FRAC2
                                ;38097  ;               ID[T1] = OP LONG FRAC3
                                ;38098
                                ;38099  ; OUTPUTS:
                                ;38100  ;               D = RC0 =DST LONG EXP
                                ;38101  ;               ID[TO] = DST LONG FRAC1
                                ;38102  ;               RC1 = DST LONG FRAC2
                                ;38103  ;               ID[T1] = DST LONG FRAC3
                                ;38104
                                ;38105          ;-------------------------------;
                                ;38106  MOVH:   ; ENTER HERE FROM 1441 ON G-FORK
                                ;38107          ALU_RC[TO], Q_ALU,              ;  GET OP LONG EXP
                                ;38108          SET.CC(INST),                   ;  NEED TO CLEAR PSL-V
U 1558, 0F10,C038,01C0,F900,0070,1559   ;38109          D_0                             ;  IN CASE WE NEED TO CLEAR FRACTION
                                ;38110
                                ;38111          ;-------------------------------;
                                ;38112          ALU_Q.ANDNOT.K[.8000],          ;  ISOLATE EXPONENT
U 1559, 0019,6024,4580,F800,0010,155A   ;38113          CLK.UBCC, WORD                  ;  CLOCK ALU Z-BIT
                                ;38114
                                ;38115          ;-------------------------------;
                                ;38116          D_Q[INST.DEP]K[.8000],          ;  AND OR XOR, MOV OR MNEG
                                ;38117          RC[TO] ALU,                     ;  NEW LONG EXP IN RC0
                                ;38118          N&Z_ALU,WORD,                   ;  CLOCK PSL N AND Z
U 155A, 0819,610C,4580,F980,0060,1A58   ;38119          Z?                              ;  IS IT 0 ?
                                ;38120
                                ;38121          ;-------------------------------;
                                ;38122  =0      ;BRANCH ON ALU Z-BIT
                                ;38123          ;0------------------------------;
U 1A58, F000,003F,01F0,F847,0000,1400   ;38124          WRITE.G.DEST                    ;  NON-ZERO, WRITE DESTINATION
                                ;38125
                                ;38126          ;1------------------------------;
                                ;38127          RC[T1] Q.OXT[WORD],             ;  CLEAR RC1
                                ;38128          CHK.FLT.OPR,                    ;  CHECK FOR -0
                                ;38129          D_0,
                                ;38130          N&Z_ALU,WORD,                   ;  CLOCK PSL N AND Z
U 1A59, 0F03,603C,0180,F988,0860,190E   ;38131          J/WRITE.DEST.0
                                ;38132  =;END   ;-------------------------------;
```

```
                                    ;38133  NEWMOVO:
                                    ;38134          ;-------------------------------;
                                    ;38135          ALU_Q,D_ALU,Q_ID[T3],           ; MOVE DATA AROUND
U 155B, 0801,203C,CDF0,2D18,0030,155C ;38136        LC_RC[T3],N_AMX.Z_TST           ; LATCH LONG FRAC2
                                    ;38137
                                    ;38138          ;-------------------------------;
                                    ;38139          ID[T0]_D,                       ; STORE LONG FRAC1 IN T0
                                    ;38140          RC[T1]_LC,N_AMX.Z_TST,          ; RC1 GETS LONG FRAC2
U 155C, 0C10,0038,C180,3D88,0030,155E ;38141        D_Q                             ; D GETS LONG FRAC3
                                    ;38142
                                    ;38143          ;-------------------------------;
U 155E, 0001,003C,C580,3C00,0030,1560 ;38144        ID[T1]_D,ALU_D,N_AMX.Z_TST      ;  FINALLY STOR LONG FRAC3
                                    ;38145
                                    ;38146          ;-------------------------------;
U 1560, 0810,0039,0180,F900,0000,147E ;38147        D_RC[T0],CALL,J/SPECG           ;  D GETS LONG EXP
```

```
                                ;38148  .TOC    "      G & H floating point  : CONVERT INTEGER TO H-FLOATING"
                                ;38149
                                ;38150  14CC:   ;  CONVERT BYTE,WORD, OR LONG TO H-FORMAT FLOATING
                                ;38151          ;  EXPECTS SRC OPERAND IN D
                                ;38152  CVTIH:
                                ;38153          ;---------------------------------;
                                ;38154          D_D.SXT[INST.DEP],              ;  SIGN EXTEND SRC TO ALL OF D
                                ;38155          SET.CC(INST),                   ;  SET CONDITION CODES
                                ;38156          Q_0,                            ;  FOR NORMALIZING
                                ;38157          C[R.SD&SS,                      ;  INITIALIZE TO POSITIVE
U 14CC, 0802,C03C,75FF,F800,00F4,7561   ;38158          SC_K[.20]                       ;  MAXIMAL EXPONENT
                                ;38159
                                ;38160          ;---------------------------------;
                                ;38161          ALU_K[.4000], R[R15]_ALU,       ;  STORE BIAS IN R15
U 1561, 0018,0D38,B180,FAF8,0000,19D8   ;38162          SIGNS?                          ;  BRANCH ON SIGN AND ZERONESS
                                ;38163
                                ;38164          ;---------------------------------;
                                ;38165  =00     ; BRANCH ON D31 AND D NE 0
                                ;38166          ;00-------------------------------;
                                ;38167          D_0,RC[T1]_0,                   ;  SRC IS 0
U 19D8, 0F03,003C,0180,F988,0000,190E   ;38168          J/WRITE.DEST.0
                                ;38169  =10     ;10-------------------------------;
                                ;38170  CVTIH.1:D_DAL.NORM,                     ;  NORMALIZE SRC
                                ;38171          SC_SC-SHF.VAL,                  ;  ADJUST EXPONENT FOR LEADING 0'S
                                ;38172          LAB_R[R15],
U 19DA, 0E00,003C,0180,FA78,008C,B563   ;38173          J/CVTIH.2
                                ;38174
                                ;38175          ;---------------------------------;
                                ;38176          D_0-D,                          ;  NEGATE INTEGER
U 19DB, 081F,2000,0180,FA78,0000,1562   ;38177          LAB_R[R15]                      ;  LATCH EXPONENT
                                ;38178  =;END
                                ;38179          ;---------------------------------;
                                ;38180          ALU_LA.OR.K[.8000],             ;  SET SIGN BIT IN EXPONENT
U 1562, 0018,0030,4580,FAF8,0000,19DA   ;38181          R[R15]_ALU,J/CVTIH.1
                                ;38182
                                ;38183          ;---------------------------------;
                                ;38184  CVTIH.2:ALU_LA.OR.K[SC],R[R15]_ALU,     ;  R15 GETS EXPONENT
U 1563, 0518,0030,1D80,FAF8,0000,1564   ;38185          DK/LEFT                         ;  GET RID IF HIDDEN BIT
                                ;38186
                                ;38187          ;---------------------------------;
                                ;38188          ALU_D.OXT[WORD],D_ALU,          ;  D GETS LONG FRAC1
                                ;38189          Q_D,                            ;  Q GETS INTEGER
U 1564, 0803,403C,01E0,FA78,0000,1566   ;38190          LAB_R[R15]                      ;  LATCH EXPONENT
                                ;38191
                                ;38192          ;---------------------------------;
                                ;38193          ID[TO]_D,                       ;  WRITE ID[TO] WITH LONG FRAC1
U 1566, 081D,2000,C180,3C00,0000,1568   ;38194          ALU_Q-D,D_ALU                   ;  D GETS HIGH WORD OF FRACTION
                                ;38195
                                ;38196          ;---------------------------------;
                                ;38197          ALU_D+LB,Q_ALU,D_0,             ;  Q GETS LONG EXP
U 1568, 0F0D,0014,01C0,F980,0000,156A   ;38198          RC[TO]_ALU                      ;  AND SO DOES RC0
                                ;38199
                                ;38200          ;---------------------------------;
                                ;38201  CVTIG.3:ID[T1]_D,RC[T1]_D,D_Q,
U 156A, 0C01,003C,C580,3D88,0000,1BB0   ;38202          J/WRITE.G
```

```
                                ;38203  .TOC    ''      G & H floating point  : CONVERT H FORMAT FLOATING POINT''
                                ;38204
                                ;38205  ;       ENTER HERE FROM CVTFG-INSTRUCTION,
                                ;38206  ;       FOR CVTFH, WITH OPCODE 98FD, WE HAVE:
                                ;38207  ;       WITH Q=UNPACKED OPERAND, SC = EXPONENT,
                                ;38208  ;       SS = SIGN
                                ;38209  ;       FOR CVTDH, WITH OPCODE 32FD, WE HAVE:
                                ;38210  ;       RC2 = LONG EXP
                                ;38211  ;       RC3 = LONG FRAC OF SRC
                                ;38212
                                ;38213  =10     ;  BRANCH ON IR<1>
                                ;38214          ;10-----------------------------;  CVTFH
                                ;38215  CVTFH:  QK/LEFT2,SI/ZERO,               ;  GET RID OF OVERFLOW AND HIDDEN BIT
                                ;38216          FE_SC-K[.80],                   ;  SUBTRACT F-BIAS OF .80
                                ;38217          CLR.UBCC,                       ;  CHECK FOR NEGATIVE EXPONENT
                                ;38218          ALU_D,RC[T1]_ALU,               ;  CLEAR RC[T1]
U 19B2, 0001,003C,418C,F988,0114,B56E   ;38219          SD_SS,J/CVTFH.00                ;  SD GETS SIGN
                                ;38220
                                ;38221          ;11-----------------------------;  CVTDH
                                ;38222          Q_RC[T2],                       ;  Q GETS LONG EXP
U 19B3, 0010,0038,01C0,F910,0000,19F8   ;38223          J/CVTDH
                                ;38224  =;END
                                ;38225          ;-------------------------------;
                                ;38226  CVTFH.00:ID[T1]_D,                      ;  CLEAR OUT ID1
U 156E, 0803,603C,C580,3C00,0000,1570   ;38227          ALU_Q.OXT[WORD],D_ALU           ;  D GETS FRAC1
                                ;38228
                                ;38229          ;-------------------------------;
                                ;38230          ALU_Q.XOR.D,Q_ALU,              ;  Q GETS HIGH WORD OF FRACTION PART
                                ;38231          ID[TO]_D,                       ;  STORE FRAC1 IN TO
U 1570, 0F1D,2020,C1C0,3C00,0000,1571   ;38232          D_0
                                ;38233
                                ;38234          ;-------------------------------;
                                ;38235          Q_Q.OR.K[.4000],                ;  ADD IN BIAS
                                ;38236          SC_FE,                          ;  GET UNBIASSED EXPONENT
U 1571, 0019,2C30,B1C0,F800,0081,1A11   ;38237          SC.NE.0?                        ;  IS SRC = 0?
                                ;38238
                                ;38239          ;-------------------------------;
                                ;38240  =0**    ;BRANCH ON SC NE 0
                                ;38241          ;0**----------------------------;
                                ;38242          ALU_RC[T3],CHK.FLT.OPR,         ;  CHECK FOR -0
U 1A11, 0010,0038,C1F8,3D18,0880,BA15   ;38243          ID[TO]_D,SC_SC-FE,Q_0           ;  MAKE IT A CLEAN 0
                                ;38244
                                ;38245          ;1**----------------------------;
U 1A15, 0818,0038,1D80,F800,0000,1572   ;38246          D_K[SC]                         ;  GET EXPONENT
                                ;38247  =;END
                                ;38248          ;-------------------------------;
                                ;38249          ALU_D.SXT[BYTE]+Q,              ;  ADD IN EXPONENT
                                ;38250          RC[TO]_ALU,                     ;  STORE LONG EXP
U 1572, 081E,9214,0180,F980,0000,1BBE   ;38251          D_ALU,EALU?                     ;  TEST SIGN
                                ;38252
```

```
                                ;38253          ;--------------------------------;
                                ;38254  =1110   ;BRANCH ON SS
                                ;38255          ;1110----------------------------;
                                ;38256  CVTFH.2:ALU_RC[T0],N&Z_ALU,WORD,        ;  SET CONDITION CODES
U 1BBE, 0010,4038,0180,F900,0060,1B80   ;38257          J/WRITE.G
                                ;38258
                                ;38259          ;1111----------------------------;
                                ;38260          D_D.OR.K[.8000],RC[T0]_ALU,     ;  STORE LONG EXP IN RC0
U 1BBF, 0819,0030,4580,F980,0000,1BBE   ;38261          J7CVTFH.2
                                ;38262  =;END
                                ;38263  =00     ;00----------------------------;
                                ;38264  CVTDH:  CLR.SD&SS,
                                ;38265          D_RC[T3],                       ;  GET LONG FRAC
U 19F8, 0810,0039,0187,F918,0000,08BE   ;38266          CALL,J/UNPK                     ;  UNPACK THE DOUBLE OPERAND IN <Q,D>
                                ;38267
                                ;38268  =01     ;01----------------------------;  SRC = 0
                                ;38269          D_0,Q_0,                        ;  WRITE ZERO
U 19F9, 0F00,003C,01F8,F800,0000,190E   ;38270          J7WRITE.DEST.0
                                ;38271
                                ;38272  ; RETURN WITH:
                                ;38273  ;       SC = BIASSED EXP
                                ;38274  ;       RC5 = UNPACKED FRAC <MS>
                                ;38275  ;       D = UNPACKED FRAC <LS>
                                ;38276  ;       SS= SD = SIGN
                                ;38277
                                ;38278  =10     ;10----------------------------;  SRC IS NON-ZERO
                                ;38279          ID[T0]_D,                       ;  STORE FRAC <AMS>
U 19FA, 0F10,0038,C1C0,3D28,0000,1573   ;38280          Q_RC[T5],D_0
                                ;38281  =;END
                                ;38282          ;--------------------------------;
                                ;38283          RC[T0]_Q,
U 1573, 0001,203C,4180,F980,0084,B578   ;38284          SC_SC-R[.80]                    ;  UNBIAS THE EXP
                                ;38285
                                ;38286          ;--------------------------------;
U 1578, 0018,0038,B1C0,F800,0000,1579   ;38287          Q_K[.4000]                      ;  H-FORMAT BIAS
                                ;38288
                                ;38289          ;--------------------------------;
U 1579, 0818,0038,1D80,F800,0000,157A   ;38290          D_K[SC]                         ;  D GETS OLD UNBIASSED EXP
                                ;38291
                                ;38292          ;--------------------------------;
                                ;38293          ALU_D.SXT[BYTE]+Q,              ;  NEW BIASSED EXP
                                ;38294          RC[T6]_ALU,                     ;  STORE IT IN RC6
U 157A, 0F1E,8014,01F8,F9B0,0000,19C9   ;38295          D_0,Q_0,J/CVTGH.W
                                ;38296          ;--------------------------------;
```

```
                                        ;38297  .TOC    ""      G & H floating point  : H FORMAT UNPACK ROUTINE"
                                        ;38298
                                        ;38299  ;ROUTINE TO UNPACK H FORMAT FLOATING OPERANDS.
                                        ;38300
                                        ;38301  ;INPUTS:
                                        ;38302  ;               RC0-T3 CONTAIN THE TWO INPUT OPERANDS
                                        ;38303  ;               IR<2:1> = TYPE OF OPERATION BEING PERFORMED
                                        ;38304  ;               FOR EMODH, LOW 15 BITS OF RC6 = EXTENDER
                                        ;38305  ;               STATE<1> SIGNALS EMODH(WHEN=1) OR NOT
                                        ;38306  ;OUTPUTS:
                                        ;38307  ;               RC0-ID[T3] = FRACTION PARTS
                                        ;38308  ;       RC4 = OP1 EXPONENT
                                        ;38309  ;       RC6 = OP2 EXPONENT
                                        ;38310  ;       SS = SIGN1.XOR.SIGN2.XOR.IR<1>
                                        ;38311  ;       R15 = OP2 EXP -OP1 EXP
                                        ;38312  ;       FE = NABS OF LOW BITS OF EXPONENT DIFFERENCE
                                        ;38313  ;       IF ONE OPERAND IS 0, THE OTHER IS LEFT UNCHANGED
                                        ;38314  ;TEMPORARIES:
                                        ;38315  ;       STATE<4> IS USED FOR LOOP FLAG
                                        ;38316  ;       D,Q,SC ARE USED FOR A VARIETY OF OPERATIONS.
                                        ;38317  ;       SD IS USED
                                        ;38318  ;RETURNS:
                                        ;38319  ;               RETURN @ 1 IF OP1 = 0 , OP2 <> 0
                                        ;38320  ;               RETURN @ 2 IF OP1 <> 0, OP2 = 0
                                        ;38321  ;               RETURN @ 3 IF BOTH OPERANDS ARE NON ZERO
                                        ;38322  ;               RETURN @ 4 IF BOTH OPERANDS ARE 0
                                        ;38323
                                        ;38324  ;IN ADDTION TO UNPACKING, BOTH OPERANDS ARE CHECKED FOR -0
                                        ;38325
                                        ;38326  ; NOTATION:
                                        ;38327  ;       BELOW, THE PIECES OF THE NUMBER IS REFERRED TO AS:
                                        ;38328  ;
                                        ;38329  ;       +-------+-------+
                                        ;38330  ;       |  0H   |  EXP  |
                                        ;38331  ;       +-------+-------+
                                        ;38332  ;       |  1H   |  0L   |
                                        ;38333  ;       +-------+-------+
                                        ;38334  ;       |  2H   |  1L   |
                                        ;38335  ;       +-------+-------+
                                        ;38336  ;       |  3H   |  2L   |
                                        ;38337  ;       +-------+-------+
                                        ;38338
                                        ;38339  UNPACKH:
                                        ;38340          SC_K[.30],                      ;  POINTER, X, TO OPERAND
U 157B, 0810,0038,7980,F910,0084,7580   ;38341          ALU_RC[T2],D_ALU
                                        ;38342
                                        ;38343          ;-------------------------------;
                                        ;33344          ALU_D.ANDNOT.K[.8000],
U 1580, 0019,4024,4580,F800,0010,1581   ;38345          CLK.UBCC,WORD
                                        ;38346
                                        ;38347          ;-------------------------------;
                                        ;38348          Q_RC[T0],                       ;  GET LONG EXP
                                        ;38349          SS_SS.XOR.ALU15&SD_ALU15,       ;  USE SIGN TO COMPLEMENT SS
U 1581, 0010,0038,C1C5,F900,0104,78CC   ;38350          FE_K[.FFFF]                     ;  SHIFT COUNT OF -1
```

```
                                ;38351  =01100  ;01100-------------------------;
                                ;38352          ALU_Q,OXT[WORD].ANDNOT.K[.8000], ; ISOLATE EXPONENT PART
                                ;38353          RC[T4]_ALU,                     ;  SAVE IT
                                ;38354          CLK.UBCC,                       ;  CLOCK IT
                                ;38355          D_Q,                            ;  D GETS OHEXP
                                ;38356          Q_ID(SC),                       ;  Q GETS 1H0L
                                ;38357          SC_FE,FE_SC,                    ;  FE GETS X, SC GETS -1
                                ;38358          Z?,                             ;  IS OP2 = 0 ?
U 18CC, 0C1B,6125,45F0,25A0,0191,1A64   ;38359          CALL,J/UNP.SUBR.2               ;  CALL UNPACK SUBROUITNE
                                ;38360
                                ;38361  =11100  ;11100-------------------------;
U 18DC, 0000,003E,0180,F800,0000,C001   ;38362          RETURN1                         ;  OP1 = 0
                                ;38363
                                ;38364  ;RETURN WITH:
                                ;38365  ;BRANCH ON EMOD-BIT
                                ;38366
                                ;38367  =11101  ;11101-------------------------;  NOT EMODH
                                ;38368  UNPACKH.1:SC_SC+1,                      ;  POINT TO NEXT LONG EXP
                                ;38369          ID(SC)_D,                       ;  STORE LAST FRACTION WORD
                                ;38370          Q_RC[T2],                       ;  Q GETS LONG EXP
                                ;38371          SGN/ADD.SUB,                    ;  SD_ALU15,SS_SS.XOR.ALU15.XOR.IR1
U 18DD, 0010,0038,01C6,3510,0080,D8EE   ;38372          J/UNPH1
                                ;38373
                                ;38374          ;11111-------------------------;  EMODH
                                ;38375  =11111  STATE_STATE.ANDNOT.K[.80],      ;  DON'T NORMALIZE 2. OPERAND
                                ;38376          LC_RC[T6],                      ;  LATCH LAST PIECE OF MULTIPLIER
                                ;38377          ALU_D.OR.LC,D_ALU,              ;  OR IN LAST 15 BI S OF M'PLIER
U 18DF, 0811,0030,4180,F930,1404,58DD   ;38378          J/UNPACKH.1
                                ;38379  =;END
                                ;38380  =01110
                                ;38381  UNPH1:  ;01110-------------------------;
                                ;38382          ALU_Q,OXT[WORD].ANDNOT.K[.8000],
                                ;38383          RC[T6]_ALU,
                                ;38384          D_Q,                            ;  D GETS OLEXP
                                ;38385          CLK.UBCC,                       ;  ACTUALLY NON ZERO
                                ;38386          Q_ID(SC),                       ;  Q GETS 1H0L
                                ;38387          FE_SC,                          ;  FE GETS X, THE NEW OP-POINTER
                                ;38388          SC_FE,                          ;  SC GETS -1
U 18EE, 0C1B,6025,45F0,25B0,0191,1A64   ;38389          CALL,J/UNP.SUBR.2
                                ;38390
                                ;38391  =11110  ;11110-------------------------;
U 18FE, 0000,003E,0180,F800,0000,0002   ;38392          RETURN2
                                ;38393
                                ;38394  ;RETURN AFTER UNPACKING ALL THE OPERANDS
                                ;38395  =11111  ;11111-------------------------;
                                ;38396          ID(SC)_D,                       ;  STORE LOW FRACTION IN ID[T3]
U 18FF, 0810,0038,0180,3530,0000,1582   ;38397          D_RC[T6]                        ;  D GETS EXPONENT 2
                                ;38398  =;END
                                ;38399          ;-------------------------------;
                                ;38400          LC_RC[T4],                      ;  LATCH UP EXP1
                                ;38401          FE_K[ZERO],                     ;  NEEDED FOR NABS-FUNCTION
U 1582, 0000,003E,1980,F920,0104,6003   ;38402          RETURN3
```

```
                                ;38403          ;--------------------------------;
                                ;38404  =0      ;BRANCH ON ALU Z-BIT
                                ;38405          ;0--------------------------------;  OP2 IS NOT ZERO
                                ;38406  UNP.SUBR.2:
                                ;38407  ;       EXPECTS:
                                ;38408  ;               D = 0HEXP
                                ;38409  ;               Q = 1H0L
                                ;38410  ;               FE = X = ARGUMENT POINTER (.30 OR .32)
                                ;38411  ;               SC = -1
                                ;38412          ALU_D.ANDNOT.K[.FFFF],D_ALU,    ;  ISOLATE UPPER 16 BITS OF FRACTION
                                ;38413          FE_SC,                          ;  FE GETS -1
                                ;38414          SC_FE,                          ;  SC GETS X
                                ;38415          Z?,                             ;  WAS EXPONENT 0 ?
U 1A64, 0819,0124,C180,F800,0181,1A6C   ;38416          J/UNP.SUBR.3
                                ;38417
                                ;38418          ;1--------------------------------;  OP2 = 0
                                ;38419          ALU_RC[T2],CHK.FLT.OPR,         ;  CHECK FOR -0
                                ;38420          Z?,Q_ID[USTACK],                ;  POP STACK ONCE
U 1A65, 0010,0138,81F0,2D10,0800,1A74   ;38421          J/UNP.SUBR.7                    ;  TEST OP1 FOR 0
                                ;38422  =;END
                                ;38423          ;--------------------------------;
                                ;38424  =0      ;BRANCH ON ALU Z-BIT
                                ;38425          ;0--------------------------------;
                                ;38426          ;  ENTER HERE FROM UNPACK1-ROUTINE
                                ;38427          ;  EXPECTS: FE = -1
                                ;38428          ;           SC = X
                                ;38429  UNP.SUBR.3:
                                ;38430          ALU_Q.OXT[WORD].OR.D,           ;  ASSEMBLE 0H0L
                                ;38431          RC(SC)_ALU.RIGHT,SI/MUL-,       ;  SHIFT IN HIDDEN BIT
                                ;38432          Q_ALU.RIGHT,                    ;  IN CASE OF DOUBLE SHIFT
                                ;38433          D_Q,                            ;  D GETS 1H0L
                                ;38434          STATE7-4?,                      ;  TEST NORMALIZE-BIT
U 1A6C, 0C5F,7630,03C0,F838,0000,1BE7   ;38435          J/UNP.SUBR.4                    ;  FIRST OR SECOND PASS ?
                                ;38436
                                ;38437          ;1--------------------------------;  OP IS 0
                                ;38438          ALU_RC(SC),                     ;  GET OP1 LONG EXP
                                ;38439          CHK.FLT.OPR,                    ;  CHECK FOR -0
U 1A6D, 0010,003A,0180,F830,0800,0010   ;38440          RETURN[10]
                                ;38441  =;END
                                ;38442          ;--------------------------------;
                                ;38443  =0111   ;BRANCH ON STATE<7>
                                ;38444          ;0111-----------------------------;
                                ;38445  UNP.SUBR.4:ALU_Q,RC(SC)_ALU.RIGHT,      ;  STORE 0H0L (SHIFTED TWICE)
                                ;38446          SI/MUL+,                        ;  SHIFT IN OVERFLOW-BIT
                                ;38447          SC_FE,                          ;  SC GETS -1
                                ;38448          FE_SC+1,                        ;  FE GETS X+1
U 1BE7, 0041,203C,0300,F838,0181,D583   ;38449          J/UNP.SUBR.5
                                ;38450
                                ;38451          ;1111-----------------------------;
                                ;38452          SC_SC+1,                        ;  SC GETS X+1, FE STAYS AT -1
                                ;38453          ALU_D.ANDNOT.K[.FFFF],D_ALU,    ;  D GETS 1H
U 1BEF, 0819,0024,C180,F800,0080,D584   ;38454          J/UNP.SUBR.6
                                ;38455  =;END
```

```
                                ;38456          ;--------------------------------;
                                ;38457  UNP.SUBR.5:SC_FE,                       ;  SC GETS X+1
                                ;38458          FE_SC+K[.FFFF],                 ;  FE GETS -2
U 1583, 0819,0024,C180,F800,0185,9584   ;38459          ALU_D.ANDNOT.K[.FFFF],D_ALU     ;  D GETS 1H
                                ;38460
                                ;38461          ;--------------------------------;
                                ;38462  UNP.SUBR.6:Q_RC(SC),                     ;  Q GETS 2H1L
U 1584, 0010,0038,05C0,F830,0084,B585   ;38463          SC_SC-K[.1]                     ;  SC GETS X
                                ;38464
                                ;38465          ;--------------------------------;
                                ;38466          ALU_Q.OXT[WORD].OR.D,            ;  ASSEMBLE 1L WITH 1H
                                ;38467          D_ALU,                          ;  D GETS 1H1L
                                ;38468          Q_ID(SC),                       ;  Q GETS 1H0L
                                ;38469          FE_SC,                          ;  FE GETS X
U 1585, 081F,6030,01F0,2400,0181,1586   ;38470          SC_FE                           ;  SC GETS -1 OR -2
                                ;38471
                                ;38472          ;--------------------------------;
                                ;38473          D_DAL.SC,                       ;  SHIFT 1H1L RIGHT
                                ;38474          ID[T9]_D,                       ;  SAVE 1H1L IN T9
                                ;38475          Q_LC,                           ;  Q GETS 2H1L
U 1586, 0D10,0038,E5C0,3C00,0181,1588   ;38476          SC_FE,FE_SC                     ;  SC GETS X, FE GETS -1 OR -2
                                ;38477
                                ;38478          ;--------------------------------;
                                ;38479          ID(SC)_D,                       ;  STORE 1H1L (SHIFTED)
                                ;38480          ALU_Q.ANDNOT.K[.FFFF],          ;  ISOLATE 2H
                                ;38481          D_ALU,                          ;  D GETS 2H
U 1588, 0819,2024,C180,3400,0080,D58A   ;38482          SC_SC+1                         ;  SC GETS X+1
                                ;38483
                                ;38484          ;--------------------------------;
U 158A, 0000,003C,01F0,2400,0000,158C   ;38485          Q_ID(SC)                        ;  Q GETS 3H2L
                                ;38486
                                ;38487          ;--------------------------------;
                                ;38488          ALU_Q.OXT[WORD].OR.D,D_ALU,
                                ;38489          Q_ID[T9],                       ;  Q GETS 1H1L
U 158C, 081F,6030,E5F0,2C00,0181,158E   ;38490          SC_FE,FE_SC                     ;  SC GETS X+1, FE GETS -1 OR -2
                                ;38491
                                ;38492          ;--------------------------------;
                                ;38493          ID[T9]_D,                       ;  SAVE 2H2L IN T9
                                ;38494          D_DAL.SC,                       ;  D GETS 2H2L (SHIFTED)
                                ;38495          Q_D,                            ;  Q GETS 2H2L
U 158E, 0D00,003C,E5E0,3C00,0181,1590   ;38496          SC_FE,FE_SC                     ;  SC GETS -1 OR -2, FE GETS X+1
                                ;38497
                                ;38498          ;--------------------------------;
                                ;38499          RC(SC)_D,                       ;  STORE 2H2L (SHIFTED)
U 1590, 0C01,003C,01F0,2438,0000,1591   ;38500          D_Q,Q_ID(SC)                    ;  Q GETS 3H2L
                                ;38501
                                ;38502          ;--------------------------------;
                                ;38503          FE_SC,SC_FE,                    ;  SC GETS -1 OR -2
                                ;38504          ALU_Q.ANDNOT.K[.FFFF],D_ALU,
U 1591, 0819,2024,C1E0,F800,0181,1592   ;38505          Q_D                             ;  Q GETS 2H2L
                                ;38506
```

```
                                        ;38507          ;--------------------------------;
                                        ;38508          D_DAL.SC,
                                        ;38509          SC_FE,                          ;  SC GETS X+1
                                        ;38510          FE_K[.FFFF],                    ;  FE GETS -1
                                        ;38511          STATE1?,                        ;  BRANCH ON EMODH-BIT
U 1592, 0D00,173E,C180,F800,0185,6011   ;38512          RETURN[11]
                                        ;38513
                                        ;38514          ;--------------------------------;
                                        ;38515  =0       ; BRANCH ON ALU Z-BIT
                                        ;38516          ;0--------------------------------;
U 1A74, 0000,003E,0180,F800,0000,0002   ;38517  UNP.SUBR.7:RETURN2                       ;  OP2 = 0, OP1 NE 0
                                        ;38518
                                        ;38519          ;1--------------------------------;
                                        ;38520          ALU_RC[TO],CHK.FLT.OPR,         ;  CHECK FOR -0
U 1A75, 0010,00>A,0180,F900,0800,0004   ;38521          RETURN4                         ;  BOTH OPERANDS ARE 0
                                        ;38522  =;END    ;--------------------------------;
```

```
                                        ;38523  .TOC    ""      G & H floating point  : H-FORMAT FLOATING PACK-ROUTINE''
                                        ;38524
                                        ;38525  ;INPUTS:
                                        ;38526  ;               RC0-IDT1 = DST, ROUNDED AND SHIFTED LEFT ONF BIT BEYOND
                                        ;38527  ;                          NORMALIZATION I.E. THE HIDDEN BIT IS ALREADY
                                        ;38528  ;                          LEFT OF THE MSB.
                                        ;38529  ;                          ALU CARRY TELLS WHETHER THIS
                                        ;38530  ;                          RESULTED IN A CARRY OUT OF THE LEADING DIGIT
                                        ;38531  ;               LC = RC6 = DST EXPONENT, BIASSED
                                        ;38532  ;               ENTER WHILE BRANCHING ON ALU CARRY
                                        ;38533  ;OUTPUTS:
                                        ;38534  ;               RC0-ID1 = PACKED H FORMAT EXPONENT
                                        ;38535  ;               RC6 = DST EXPONENT, BIASSED
                                        ;38536  ;               CHECKS FOR OVERFLOW AND UNDERFLOW
                                        ;38537  ;RETURN:
                                        ;38538  ;               RETURN10
                                        ;38539
                                        ;38540          ;-------------------------------;
                                        ;38541  =0*     ;BRANCH ON C31
                                        ;38542          ;0*-----------------------------;  NO CARRY
                                        ;38543  PACKH:  D_RC[T1],                       ;  D GETS 2H2L
                                        ;38544          Q_ID[T1],                       ;  Q GETS 3H3L
U 1A80, 0810,0038,C5F0,2D08,0000,1593   ;38545          J7PACKH.1
                                        ;38546
                                        ;38547          ;1*-----------------------------;  CARRY PROPAGATED ALL THE WAY
                                        ;38548          ALU_0+LC+1,RC[T6]_ALU,          ;  INCREMENT EXPONENT
U 1A82, 0013,0010,0180,F9B0,0010,1A80   ;38549          CLK.UBCC,J/PACKH
                                        ;38550  =;END
                                        ;38551  PACKH.1:;-------------------------------;  NOW WE PACK FRACTION WORDS TOGETHER.
                                        ;38552          ALU_Q.ANDNOT.K[.FFFF],          ;  ISOLATE HIGH 16 FRACTION BITS
U 1593, 0019,2024,C1C0,F800,0000,1594   ;38553          Q_ALU                           ;  Q GETS 3H00
                                        ;38554
                                        ;38555          ;-------------------------------;
                                        ;38556          ALU_D.OXT[WORD].OR.Q,            ;  ASSEMBLE LONG FRAC1
                                        ;38557          D_ALU,                          ;  D GETS 3H2L
U 1594, 081F,4030,C1F0,2C00,0000,1595   ;38558          Q_ID[T0]                        ;  Q GETS 1H1L
                                        ;38559
                                        ;38560          ;-------------------------------;
                                        ;38561          ID[T1]_D,                       ;  WRITE LONG FRAC3, 3H2L
                                        ;38562          D_Q,                            ;  D GETS 1H1L
U 1595, 0C10,0038,C5C0,3D08,0000,1596   ;38563          Q_RC[T1]                        ;  Q GETS 2H2L
                                        ;38564
                                        ;38565          ;-------------------------------;
U 1596, 0019,2024,C1C0,F800,0000,1598   ;38566          ALU_Q.ANDNOT.K[.FFFF],Q_ALU     ;  Q GETS 2H00
                                        ;38567
                                        ;38568          ;-------------------------------;
                                        ;38569          ALU_D.OXT[WORD].OR.Q,            ;  ASSEMBLE 2H1L
U 1598, 001F,4030,0180,F988,0000,1599   ;38570          RC[T1]_ALU                      ;  STORE IT IN RC1
                                        ;38571
                                        ;38572          ;-------------------------------;
                                        ;38573          Q_D,                            ;  Q GETS 1H1L
U 1599, 0810,0038,01E0,F900,0000,159A   ;38574          D_RC[T0]                        ;  D GETS 0H0L
                                        ;38575
                                        ;38576          ;-------------------------------;
U 159A, 0019,2024,C1C0,F800,0000,159B   ;38577          ALU_Q.ANDNOT.K[.FFFF],Q_ALU     ;  Q GETS 1H00
```

```
                                ;38578          ;-------------------------------;
                                ;38579          ALU_D.OXT[WORD].OR.Q,D_ALU,     ;  D GETS 1HOL
U 159B, 081F,4030,01E0,F800,0000,159C   ;38580          Q_D                             ;  Q GETS 0HOL
                                ;38581
                                ;38582          ;-------------------------------;
                                ;38583          ID[TO]_D,
                                ;38584          D_RC[T6],                       ;  D GETS BIASSED EXPONENT
                                ;38585          C[K.UBCC,WORD,                  ;  CLOCK ALU-N-BIT ON BIT15
U 159C, 0810,5238,C180,3D30,0010,1BFE   ;38586          EALU?                           ;  TEST SIGN-BIT
                                ;38587
                                ;38588          ;-------------------------------;
                                ;38589  =1110   ;  BRANCH ON SS-FF
                                ;38590          ;1110---------------------------;
                                ;38591  PACKH.2:ALU_Q.ANDNOT.K[.FFFF],Q_ALU,    ;  Q GETS 0H00
U 1BFE, 0019,3724,C1C0,F800,0000,1597   ;38592          STATE3?,J/PACKH.3               ;  BRANCH ON STATE3
                                ;38593
                                ;38594          ;1111---------------------------;  NEGATIVE
                                ;38595          ALU_D.OR.K[.8000],D_ALU,        ;  OR SIGN-BIT INTO EXP
U 1BFF, 0819,0030,4580,F800,0000,1BFE   ;38596          J/PACKH.2
                                ;38597  =;END
                                ;38598          ;-------------------------------;
                                ;38599  =0111   ;  BRANCH ON STATE3
                                ;38600          ;1110---------------------------;  NOT POLYH,CONTINUE
U 1597, 0000,0D3C,0180,F800,0000,18E6   ;38601  PACKH.3:SIGNS?,J/PACKH.3A          ;  BRANCH ON EXP<31>
                                ;38602
                                ;38603          ;1111---------------------------;  POLYH
U 159F, 0000,0D3C,0180,F800,0000,19D6   ;38604          SIGNS?,J/POLYH.X                ;
                                ;38605
                                ;38606          ;-------------------------------;
                                ;38607          ;  BRANCH ON D<31>
                                ;38608  =110    ;110----------------------------;
                                ;38609  PACKH.3A:ALU_D.OXT[WORD].OR.Q,          ;  ASSEMBLE LONG EXP
                                ;38610          RC[TO]_ALU,D_ALU,               ;  STORE IT IN RC0
U 18E6, 081F,5B30,0180,F980,0000,15A3   ;38611          ALU?,J/PACKH.4                  ;  TEST FOR OVERFLOW
                                ;38612
                                ;38613          ;111----------------------------;  UNDERFLOW
                                ;38614          Q_ID[PSL],                      ;  GET PSL TO TEST FU-BIT
                                ;38615          D_0,ALU_0(A),RC[TO]_ALU,
U 18E7, 0F03,003C,3DF0,2D80,0000,15B5   ;38616          J/FL.UNDER
                                ;38617  =;END
```

```
                                        ;38618          ;--------------------------------;
                                        ;38619  =0011   ;  BRANCH ON ALU N-BIT
                                        ;38620          ;0011----------------------------;  NO OVERFLOW
                                        ;38621  PACKH.4:ALU_RC[T0],N&Z_ALU,             ;
                                        ;38622          WORD,                           ;  CLOCK N AMD Z ON RESULT
U 15A3, 0010,403A,0180,F900,0060,0010   ;38623          RETURN10
                                        ;38624
                                        ;38625          ;0111----------------------------;  UNDERFLOW
                                        ;38626          Q_ID[PSL],                      ;  GET PSL TO TEST FU-BIT
                                        ;38627          D_0,ALU_0(A),RC[T0]_ALU,        ;  MAKE THINGS ZERO
U 15A7, 0F03,003C,3DF0,2D80,0000,15B5   ;38628          J/FL.UNDER                      ;  TAKE FAULT
                                        ;38629
                                        ;38630          ;1011----------------------------;  OVERFLOW
                                        ;38631          RC[T7]_K[.8],                   ;  RCT7 GETS OVERFLOW FAULT CODE
U 15AB, 0018,0038,0180,F9B8,0000,12B4   ;38632          J/FLOAT.FAULT
                                        ;38633  =;END
                                        ;38634  =0101   ;0101----------------------------;
                                        ;38635  FL.UNDER:RC[T1]_ALU,ALU_0(A),ID[T0]_D,  ;  MAKE MORE THINGS ZERO
U 15B5, 0003,003D,C180,3D88,0000,12AE   ;38636          CALL[PS[FU.A]                   ;  SEE IF FU-BIT IS SET
                                        ;38637
                                        ;38638  =1101   ;1101----------------------------;  FU NOT SET IGNORE UNDERFLOW
                                        ;38639          ALU_0(A),N&Z_ALU,               ;
U 15BD, 0003,003E,C5F8,3C00,0060,0010   ;38640          ID[T1]_D,Q_0,RETURN10           ;  SET THINGS TO ZERO
                                        ;38641
                                        ;38642          ;1111----------------------------;  FU SET TAKE FAULT
U 15BF, 0018,0038,F580,F9B8,0000,12B4   ;38643          RC[T7]_K[.A],J/FLOAT.FAULT      ;
```

```
                                    ;38644  .TOC    ""      G & H floating point  : H-FORMAT FLOATING ADD AND SUBTRACT""
                                    ;38645
                                    ;38646  ;HERE FOR *-*-* AND *-MEMORY
                                    ;38647  ;ENTER HERE WITH SRC OPND IN RC0-ID1
                                    ;38648  ;AND DST OPND IN RC2-ID3
                                    ;38649
                                    ;38650  148A:
                                    ;38651          STATE_K[ZERO],                  ;  FIND SOMETHING BETTER TO DO
U 148A, 0000,003D,1980,F800,1404,7950 ;38652        CALL,J/ADDH.ROUTINE             ;  CALL ADD H-FORMAT SUBR.
                                    ;38653
                                    ;38654  149A:   ;-------------------------------;
U 149A, F000,003F,01F0,F847,0000,1400 ;38655        WRITE.G.DEST,J/WRG
                                    ;38656  =;END
                                    ;38657
                                    ;38658  ;HERE FOR *-REG
                                    ;38659  ;ENTER WITH SRC-OPERAND IN RC0-ID1
                                    ;38660  ;AND WITH DST-OPND IN LA,R(PRN),ETC.
                                    ;38661
                                    ;38662          ;-------------------------------;
                                    ;38663  142E:   RC[T2]_LA,                      ;  RC2 GET OP2 LONG EXP
                                    ;38664          K[.80],                         ;  CONSTANT FOR FINDING REG #
U 142E, 0F00,003C,4180,F990,0000,186E ;38665        D_0                             ;  NEEDED FOR REG-ROUTINE
                                    ;38666
                                    ;38667          ;0*11*-------------------------;
U 186E, 0000,003D,0180,F800,0000,1862 ;38668 =0*11* CALL,J/HUGE.REG.READ.1          ;  READ 2. OPERAND
                                    ;38669
                                    ;38670          ;-------------------------------;
U 187E, 0000,003C,1980,F800,1404,7A21 ;38671        STATE_K[ZERO]                   ;  NOTHING TO DO
                                    ;38672  =;END
                                    ;38673          ;0****-------------------------;
U 1A21, 0000,003D,0180,F800,0000,1950 ;38674 =0**** CALL,J/ADDH.ROUTINE             ;  CALL ROUTINE FOR ADDING
                                    ;38675
                                    ;38676          ;1****-------------------------;
                                    ;38677          TRAP.ACC[1],                    ;  RESET THE ACCELLARATOR
                                    ;38678          D_RC[T0],                       ;  D GETS RESULT LONG EXP
U 1A31, 0810,00B8,0080,F900,0000,18D4 ;38679        J7HUGE.REG.WRITE.1              ;  STORE RESULT
                                    ;38680  =;END   ;-------------------------------;
```

```
                                ;38681  =000    ;000----------------------------;
                                ;38682  ADDH.ROUTINE:
                                ;38683          CLR.SD&SS,                      ;  INITIALIZE SS AND SD
                                ;38684          ALU_RC[T0],N&Z_ALU.V&C_0,WORD,  ;  CLEAR V AND C OF PSL
U 1950, 0010,4039,0187,F900,0050,157B   ;38685          CAL[,J/UNPACKH
                                ;38686
                                ;38687          ; RETURN WITH RC2-ID3 CONTAINING DESTINATION
                                ;38688          ; IN PACKED FORM
                                ;38689  =001    ;001----------------------------;  SRC = 0
                                ;38690          ALU_RC[T2],N&Z_ALU.V&C_0,WORD,  ;  SET CONDITON CODES ON DST
                                ;38691          D_ALU,Q_ID[T2],                 ;  D GETS LONG EXP, Q GETS LONG FRAC1
U 1951, 0810,4038,C9F0,2D10,0050,1636   ;38692          J7SHIFT.H
                                ;38693
                                ;38694          ;RETURN WITH RC0-ID1 CONTAINING SRC IN PACKED FORM.
                                ;38695          ;010----------------------------;  DST = 0
                                ;38696  =010    D_RC[T0],                       ;  D GETS SRC LONG EXP
U 1952, 0810,0938,0180,F900,0000,1822   ;38697          IR1?, J/ADDG.8                  ;  TEST FOR ADD OR SUBTRACT
                                ;38698
                                ;38699  ;RETURN WITH:
                                ;38700  ;       D=EXP2
                                ;38701  ;       LC=EXP1
                                ;38702
                                ;38703  =011    ;011----------------------------;  BOTH OPERANDS NON ZERO
                                ;38704  ADDH.00:                                ;  ENTER HERE FROM POLYH
                                ;38705          ALU_D-LC,R[R15]_ALU,            ;  GENERATE EXP2-EXP1
                                ;38706          CLK.UBCC,                       ;  CLOCK IT
                                ;38707          SC_ALU,                         ;
                                ;38708          FE_K[ZERO],                     ;  USE IN NABS-FUNCTION BELOW
U 1953, 0011,0000,1980,FAF8,0196,759D   ;38709          J/ADDH.01
                                ;38710
                                ;38711          ; BOTH OPERANDS ARE 0
                                ;38712          ;  CLEAR OUT DIRTY 0'S
                                ;38713  =100    ;100----------------------------;
                                ;38714          ALU_0(A),N&Z_ALU.V&C_0,         ;  SET PSL Z-BIT
                                ;38715          D_0,Q_0,SC_K[.30],
U 1954, 0F03,003C,79F8,F800,00D4,7632   ;38716          J7CLEAR.DIRTY.0
                                ;38717  =;END
                                ;38718          ;----------------------------;
                                ;38719  ADDH.01:FE_NABS(SC-FE),                 ;  FE GETS NABS OF EXP DIFFERENCE
                                ;38720          ALU_K[.30],SC_ALU,              ;  FOR POINTING TO OPERANDS
                                ;38721          LAB_R[R15],                     ;  LATCH EXPONENT DIFFERENCE
U 159D, 0018,1B38,7980,FA78,0182,F5C3   ;38722          ALU?,J/ADDH.10                  ;  TEST RESULT OF EXP2-EXP1
                                ;38723          ;----------------------------;
```

```
                                ;38724  =0011   ;  BRANCH ON ALU Z AND N-BITS
                                ;38725          ;0011-------------------------;
                                ;38726  ADDH.10:SC_SC.OR.K[.1],                 ;  ALIGN SRC-OPERAND
                                ;38727          D_0,                            ;  INITIALIZE FILLER
                                ;38728          EALU?,
U 15C3, 0F00,123C,0580,F800,0084,388E   ;38729          J/ADDH.100
                                ;38730
                                ;38731          ;0111-------------------------;
                                ;38732          SC_SC.OR.K[.1],                 ;  NEGATE SRC IF ANY
                                ;38733          ALU -1,D_ALU,                   ;  D GETS FILLER
U 15C7, 0803,1228,0580,F800,0084,3846   ;38734          EALU?,J/ADDH.EQUAL              ;  EXPONENTS ARE EQUAL
                                ;38735
                                ;38736          ;1011-------------------------;
                                ;38737          ALU_0-LB,R[R15]_ALU,            ;  NEGATE EXP DIFFERENCE
                                ;38738          D_0,                            ;  INITALIZE FILLER
                                ;38739          SC_SC.OR.K[.3],                 ;  ALIGN DST-OPERAND
U 15CB, 0F0F,1200,0D80,FAF8,0084,348E   ;38740          EALU?                           ;  TEST FOR ADD OR SUBTRACT
                                ;38741  =;END
                                ;38742          ;-----------------------------;
                                ;38743  =1110   ;  BRANCH ON SS-FLIP FLOP
                                ;38744          ;1110-------------------------;
                                ;38745          ALU_LC,RC[T6]_ALU,              ;  RC6 GETS LARGEST EXP
U 148E, 0010,0038,0180,F9B0,0000,188E   ;38746          J/ADDH.100
                                ;38747
                                ;38748          ;1111-------------------------;
                                ;38749          ALU_LC,RC[T6]_ALU,              ;  RC6 GETS LARGEST EXP
                                ;38750          SGN/NOT.SD,                     ;  SUBTRACT, SO COMPLEMENT SIGN
U 148F, 0010,0038,0183,F9B0,0000,188F   ;38751          J/ADDH.101
                                ;38752  =;END
```

```
                                ;38753          ;--------------------------------;
                                ;38754  =001110 ;BRANCH ON SS
                                ;38755          ;001110--------------------------;  ADD
                                ;38756  ADDH.100:ID[T9] D,                        ;  T9 GETS 0 OR -1
                                ;38757          ALU_D,RC[T4] ALU,
U 188E, 0001,003D,E580,3DA0,0000,15C4   ;38758          CALL,J/ALNPOSH
                                ;38759
                                ;38760  =001111 ;001111--------------------------;  SUBTRACT
                                ;38761  ADDH.101:Q_ID(SC),                        ;  GET LOWEST FRACTION BITS
                                ;38762          LC_RC(SC),                        ;  AND THE NEXT LOWEST ONES
                                ;38763          SC_SC-K[.1],                      ;  DECREMENT POINTER
U 188F, 0000,003D,05F0,2430,0084,B5C1   ;38764          CALL,J/ALNNEGH                    ;  DOES A RETURN10
                                ;38765
                                ;38766  ;RETURN WITH :
                                ;38767  ;                RC0-ID3 ARE THE ALIGNED OPERANDS
                                ;38768  ;
                                ;38769  =011110 ;011110--------------------------;
                                ;38770  ADDH.SUBR.0:SC_SC.OR.K[.3],               ;  SC GETS .33
                                ;38771          ALU_0+[C,CLK.UBCC,                ;  CLEAR ALU CARRY
U 189E, 0013,0015,0D80,F800,0094,359E   ;38772          CALL,J/ADDH.SUBR                  ;  DOES A RETURN20
                                ;38773
                                ;38774          ; RETURN FROM NEGATING SMALLEST NUMBER
                                ;38775          ;  WITH: D = -1, THE FILLER CHARACTER
                                ;38776          ;  SC = POINTER TO MS FRACTION
                                ;38777  =011111 ;011111--------------------------;
                                ;38778          SC_SC.OR.K[.1],                   ;  POINT TO FRACTIONS <LS>
U 189F, 0000,003C,0580,F800,0084,388E   ;38779          J/ADDH.100                        ;  ALIGN FRACTION
                                ;38780
                                ;38781  =111110 ;  RETURN FROM ADDH.SUBR
                                ;38782          ;111110--------------------------;
                                ;38783          SC_K[ZERO],                       ;  INITIALIZE SHIFT-COUNT
U 18BE, 0000,003C,1980,F800,0084,75A5   ;38784          J/ADDH.310
                                ;38785  =;END   ;--------------------------------;
```

```
                                ;38786  ; ENTER HERE IF EXPONENTS ARE EQUAL
                                ;38787  ; NO ALIGNMENT IS NECESSARY
                                ;38788
                                ;38789  =00*110 ; BRANCH ON SS-FF
                                ;38790          ;00*110-------------------------;
U 1846, 0000,003C,E580,3C00,0000,189E   ;38791  ADDH.EQUAL:ID[T9]_D,J/ADDH.SUBR.0
                                ;38792
                                ;38793          ;00*111-------------------------;
                                ;38794          Q_ID(SC),                       ;  GET LOWEST FRACTION BITS
                                ;38795          LC_RC(SC),                      ;  AND THE NEXT LOWEST ONES
                                ;38796          SC_SC-K[.1],                    ;  DECREMENT POINTER
U 1847, 0000,003D,05F0,2430,0084,B5C1   ;38797          CALL,J/ALNNEGH
                                ;38798
                                ;38799          ; RETURN WITH RC[T0]-ID[T1] NEGATED
                                ;38800          ; SC = .30
                                ;38801  =01*111 ;01*111-------------------------;
                                ;38802          SC_SC.OR.K[.3],                 ;  MAKE SC = .33
                                ;38803          ALU_0+LC,CLK.UBCC,              ;  CLEAR ALU CARRY
U 1857, 0013,0015,0D80,F800,0094,359E   ;38804          CALL,J/ADDH.SUBR
                                ;38805
                                ;38806          ;11*111-------------------------;
                                ;38807  =11*111 SC_K[.30],                      ;  POINT TO RC0-ID1
                                ;38808          LC_RC[T1],
U 1877, 0000,033C,7980,F908,0084,78E0   ;38809          C3T?                            ;  TEST FOR BORROW
                                ;38810  =;END
                                ;38811          ;0**0*-------------------------;
                                ;38812  =0**0*  LC_RC[T1],
                                ;38813          SGN/NOT.SD,                     ;  COMPLEMENT DST-SIGN
                                ;38814          Q_ID[T1],
U 18E0, 0000,003D,C5F3,2D08,0000,15C1   ;38815          CALL,J/ALNNEGH
                                ;38816
                                ;38817          ;0**1*-------------------------;
                                ;38818  =0**1*  SC_K[ZERO],                     ;  INITIALIZE SHIFT-COUNT
U 18E2, 0000,003C,1980,F800,0084,75A5   ;38819          J/ADDH.310
                                ;38820
                                ;38821          ;1**0*-------------------------;
U 18F0, 0000,003C,1980,F800,0084,75A5   ;38822  =1**0*  SC_K[ZERO],J/ADDH.310     ;RETURN10 FROM ALNNEGH-ROUTINE
                                ;38823  =;END   ;-------------------------------;
```

```
                                ;38824  ADDH.SUBR:
                                ;38825  ;EXPECTS:
                                ;38826  ;       RC0-ID3 ARE THE ALIGNED OPERANDS
                                ;38827  ;       SC = .33 OR .31
                                ;38828  ;       ALU CARRY MUST BE CLEAR
                                ;38829  ;
                                ;38830  ;  RETURN20
                                ;38831
                                ;38832          ;-------------------------------;
U 159E, 0000,003C,3180,F800,1404.55A0   ;38833          STATE_STATE.ANDNOT.K[.40]       ;  CLEAR BIT 6 OF STATE
                                ;38834
                                ;38835          ;-------------------------------;
                                ;38836  ADDH.SUBR.1:
                                ;38837          Q_ID(SC),                       ;  GET LOWER FRACTION (1 OP 3)
                                ;38838          D_RC(SC),                       ;  NEXT HIGHER FRACTION
U 15A0, 0810,0038,09F0,2430,0084,B5A1   ;38839          SC_SC-K[.2]                     ;  POINT TO OP1
                                ;38840
                                ;38841          ;-------------------------------;
                                ;38842          R[R15]_D,                       ;  STORE IT IN R15
                                ;38843          D_Q,                            ;  D GETS OP2 FRACTION
                                ;38844          Q_ID(SC),                       ;  Q GETS OP1 LOWER FRACTION
U 15A1, 0C01,033C,01F0,26F8,0000,1A81   ;38845          C31?                            ;  CARRY ?
                                ;38846
                                ;38847          ;-------------------------------;
                                ;38848  =0*     ;BRANCH ON ALU C31
                                ;38849          ;0*-----------------------------;
                                ;38850          D_D+Q,                          ;  ADD LOW FRACTION PARTS
                                ;38851          LAB_R[R15],                     ;  LATCH FRACTION PART
                                ;38852          CLK.UBCC,                       ;  CLOCK CARRY IF ANY
U 1A81, 081D,0014,0180,FA78,0010,15A2   ;38853          J/ADDH.13
                                ;38854
                                ;38855          ;1*-----------------------------;
                                ;38856          D_D+Q+1,                        ;  ADD LOW FRACTION PARTS
                                ;38857          LAB_R[R15],                     ;  LATCH FRACTION PART
U 1A83, 081D,0010,0180,FA78,0010,15A2   ;38858          CLK.UBCC                        ;  CLOCK CARRY IF ANY
                                ;38859  =;END   ;-------------------------------;
```

```
                                    ;38860  ADDH.13:
                                    ;38861          FE_K[.60],                      ;  FOR USE LATER ON
                                    ;38862          ID(SC)_D,                       ;  STORE RESULT IN OP1 LOCATION
                                    ;38863          LC_RC(SC),                      ;  LATCH UP HIGHER FRACTION
U 15A2, 0000,033C,A580,3430,0104,7A88 ;38864        C3T?                            ;  WAS THERE A CARRY ?
                                    ;38865
                                    ;38866          ;-------------------------------;
                                    ;38867  =0*     ;BRANCH ON ALU CARRY
                                    ;38868          ;0*-----------------------------;
                                    ;38869          ALU_LA+LC, RC(SC)_ALU,          ;  ADD HIGHER FRACTIONS TOGETHER
                                    ;38870          CLK.UBCC,                       ;  CLOCK CARRY IF ANY
                                    ;38871          STATE_STATE.OR.K[.40],          ;  SET FIRST PASS BIT
U 1A88, 0010,1614,3180,F838,1414,35DB ;38872        STATE7-4?,J/ADDH.14
                                    ;38873
                                    ;38874          ;1*-----------------------------;
                                    ;38875          ALU_LA+LC+1, RC(SC)_ALU,        ;  ADD HIGHER FRACTIONS TOGETHER
                                    ;38876          CLK.UBCC,                       ;  CLOCK CARRY IF ANY
                                    ;38877          STATE_STATE.OR.K[.40],          ;  SET FIRST PASS BIT
U 1A8A, 0010,1610,3180,F838,1414,35DB ;38878        STATE7-4?,J/ADDH.14
                                    ;38879
                                    ;38880          ;-------------------------------;
                                    ;36881  =1011   ;BRANCH ON STATE<6>
                                    ;38882          ;1011---------------------------;
U 15DB, 0000,003C,0180,F800,0080,D5A0 ;38883 ADDH.14:SC_SC+1,J/ADDH.SUBR.1          ;  NOW ADD TWO MOST SIGNIF. FRACS
                                    ;38884
                                    ;38885          ;1111---------------------------;  ADDITION COMPLETE
U 15DF, 0000,003C,3180,F800,1404,55A4 ;38886        STATE_STATE.ANDNOT.K[.40]       ;  CLEAR BIT 6 OF STATE
                                    ;38887
                                    ;38888          ;-------------------------------;
U 15A4, 0000,003E,1980,F800,0084,6020 ;38889        SC_K[ZERO],RETURN[20]           ;
                                    ;38890          ;-------------------------------;
```

```
                                ;38891  ADDH.310:
                                ;38892          ;  ENTER HERE TO NORMALIZE, ROUND AND PACK H-FORMAT
                                ;38893          ; EXPECTS:
                                ;38894          ; RC0-ID1 = NEW FRACTION, MAY BE UNNORMALIZED
                                ;38895          ; RC[T6] = BIASSED EXPONENT
                                ;38896          ; SC = 0
                                ;38897          ;  SD = DST-SIGN
                                ;38898          ;-------------------------------;
                                ;38899          FE_K[.60],
                                ;38900          SGN/SS.FROM.SD,                 ;  LOAD SS WITH DEST-SIGN
U 15A5, 0810,0038,A582,F900,0114,75A6   ;38901          D_RC[T7],CLK.UBCC               ;  CLOCK FRAC 0
                                ;38902
                                ;38903          ;-------------------------------;
                                ;38904  ADDH.311:LC_RC[T6],                     ;  GET READY TO ADJUST EXP
                                ;38905          Q_ID[T0],                       ;  GET FRAC1
                                ;38906          EALU_SC-FE,CLK.UBCC,            ;  CLOCK NORMALIZE COUNT
U 15A6, 0000,013C,C1F0,2D30,0010,BA7C   ;38907          Z?                              ;  FIRST LONGWORD 0 ?
                                ;38908
                                ;38909          ;-------------------------------;
                                ;38910  =0       ;BRANCH ON ALU Z-BIT
                                ;38911          ;0------------------------------;
                                ;38912          SC_SC+SHF.VAL,                  ;  SC GETS NORMALIZE COUNT
U 1A7C, 0000,003C,0180,F800,008C,95AA   ;38913          J/ADDH.36
                                ;38914
                                ;38915          ;1------------------------------;
                                ;38916          SC_SC+K[.20],                   ;  ONE MORE LONGWORD
                                ;38917          RC[T0]_Q,CLK.UBCC,              ;  CLOCK LEADING FRACTION BIT
U 1A7D, 0001,323C,7580,F980,0094,95EB   ;38918          EALU?                           ;  IS IT ALL 0 ?
                                ;38919  =;END
                                ;38920          ;-------------------------------;
                                ;38921  =1011    ;BRANCH ON EALU Z-BIT
                                ;38922          ;1011---------------------------;
                                ;38923          D_RC[T1],Q_ID[T1],              ;  GET NEXT TWO FRACTION WORDS
U 15EB, 0810,0038,C5F0,2D08,C000,15A8   ;38924          J/ADDH.312
                                ;38925
                                ;38926          ;1111---------------------------;  ALL FRACTION WORDS ARE 0
                                ;38927          ALU_0(A),N&Z_ALU,               ;
                                ;38928          RC[T0]_ALU,D_0,Q_0,             ;  RESULT IS 0
                                ;38929          SC_K[.30],
U 15EF, 0F03,003C,79F8,F980,00E4,7632   ;38930          J/CLEAR_DIRTY.0                 ;  RETURN10 FROM THAT ROUTINE
                                ;38931  =;END
                                ;38932          ;-------------------------------;
                                ;38933  ADDH.312:
                                ;38934          ID[T0]_D,                       ;  WRITE FRAC1
                                ;38935          RC[T1]_Q,                       ;  WRITE FRAC2
U 15A8, 0F01,203C,C180,3D88,0000,15A9   ;38936          D_0
                                ;38937
                                ;38938          ;-------------------------------;
                                ;38939          ID[T1]_D,                       ;  WRITE FRAC3
                                ;38940          D_RC[T0],                       ;  GET FRAC1
U 15A9, 0810,0038,C580,3D00,0000,15A6   ;38941          J/ADDH.311
                                ;38942          ;-------------------------------;
```

```
                                ;38943  ; ENTER HERE AFTER FINDING NON-ZERO LONGWORD IN RESULT.
                                ;38944  ;  RC[T0] CONTAINS THAT LONGWORD.
                                ;38945  ;  SC CONTAINS TOTAL NUMBER OF LEADING 0-BITS IN RESULT.
                                ;38946  ;  THIS ROUTINE SHIFTS THIS FRACTION ONE BIT BEYOND NORMALIZING,
                                ;38947  ;  I.E. THE HIDDEN BIT IS NO LONGER WITH US AFTER THIS.
                                ;38948  ;  EXPONENT IN RC6 IS ADJUSTED BY SC, TO REFLECT NORMALIZE-PROCESS
                                ;38949          ;-------------------------------;
                                ;38950  ADDH.36:SC_SC+1,                       ;  WE WANT TO ADJUST BEYOND HIDDEN BIT
                                ;38951          ALU_0-K[SC],Q_ALU,             ;  Q GETS NEGATIVE ADJUSTMENT
U 15AA, 001B,0000,1DC0,F930,0080,D5AD ;38952    LC_RC[T6]                      ;  LATCH EXP
                                ;38953
                                ;38954          ;-------------------------------;
                                ;38955          EALU_SC.ANDNOT.K[.FFE0],       ;  ISOLATE LOW FIVE BITS
U 15AD, 0000,003C,A180,F800,0014,55B0 ;38956    CLK.UBCC                       ;  TEST FOR ALIGNMENT OF 32.
                                ;38957
                                ;38958          ;-------------------------------;
                                ;38959          ALU_Q+LC+1,RC[T6]_ALU,         ;  RC6 HAS CORRECT BIASSED EXP,
                                ;38960                                         ;  EXCEPT FOR ROUND
                                ;38961          Q_ID[T0],                      ;  Q GETS FRAC <AMS>
U 15B0, C011,3210,C1F0,2DB0,0000,15FB ;38962    EALU?                          ;  TEST FOR ALIGNMENT OF 32.
                                ;38963
                                ;38964          ;-------------------------------;
                                ;38965  =1011   ;  BRANCH ON EALU Z-BIT
                                ;38966          ;1011---------------------------;
U 15FB, 0D00,003C,0180,F800,0000,15B1 ;38967    D_DAL.SC,J/ADDH.361            ;  NORMALIZE FIRST PART
                                ;38968
                                ;38969          ;1111---------------------------;
                                ;38970          SC_SC.OR.K[.FF00],             ;  SHIFTCOUNT OF -32.
U 15FF, 0C00,003C,4D80,F800,0084,35B1 ;38971    D_Q
                                ;38972
                                ;38973          ;-------------------------------;
U 15B1, 0C01,003C,0180,F980,0000,15B2 ;38974  ADDH.361:RC[T0]_D,D_Q
                                ;38975
                                ;38976          ;-------------------------------;
U 15B2, 0010,0038,01C0,F908,0000,15B3 ;38977    Q_RC[T1]
                                ;38978
                                ;38979          ;-------------------------------;
U 15B3, 0D00,003C,C5F0,2C00,0000,15B4 ;38980    D_DAL.SC,Q_ID[T1]
                                ;38981
                                ;38982          ;-------------------------------;
U 15B4, 0810,0038,C180,3D08,0000,15B8 ;38983    ID[T0]_D,D_RC[T1]
                                ;38984
                                ;38985          ;-------------------------------;
U 15B8, 0D00,003C,0180,F800,0000,15B9 ;38986    D_DAL.SC
                                ;38987
                                ;38988          ;-------------------------------;
U 15B9, 0C01,003C,01F8,F988,0000,15BA ;38989    RC[T1]_D,D_Q,Q_0
                                ;38990
                                ;38991          ;-------------------------------;
                                ;38992  ADDH.37:D_DAL.SC,                      ;  LEFT ADJUST LOW FRACTION PART
                                ;38993          Q_RC[T1],                      ;  Q GETS FRAC2
U 15BA, 0D10,0038,45C0,F908,0000,15BB ;38994    K[.8000]
                                ;38995          ;-------------------------------;
```

```
                                ;38996  ; ENTER HERE FROM MULH-ROUTINE.
                                ;38997  ;  THIS ROUTINE ROUNDS THE ALREADY NORMALIZED FRACTION.
                                ;38998  ;  EXITS TO PACK H-FORMAT ROUTINE, WHICH DOES A RETURN10
                                ;38999  ; EXPECTS:
                                ;39000  ;       RC0-ID1 = FRACTION LEFT-ADJUSTED, ONE BIT BEYOND NORMALIZATION
                                ;39001  ;       RC6 = BIASSED EXPONENT
                                ;39002  ;       D = DST FRAC3 (UNROUNDED)
                                ;39003  ;       Q = DST FRAC2
                                ;39004  ;       SLOW CONSTANT .8000 HAS BEEN SELECTED ONCE ALREADY.
                                ;39005  ;       SD = SS = DEST-SIGN
                                ;39006
                                ;39007  ADDH.380:;-------------------------------;
                                ;39008          D_D+K[.8000],                   ;  ROUND LOW FRACTION PART
                                ;39009          LC_RC[T0],                      ;
U 15BB, 0819,0014,4580,F900,0010,15BC   ;39010          CLR.UBCC                        ;
                                ;39011
                                ;39012  ADDH.39:;-------------------------------;
                                ;39013          ID[T1]_D,                       ;
                                ;39014          ALU_0+Q+1,Q_ALU,                ;  ADD TO NEXT FRACTION WORD
                                ;39015          CLK.UBCC,
U 15BC, 001F,0310,C5C0,3C00,0010,1A89   ;39016          C31?
                                ;39017
                                ;39018          ;-------------------------------;
                                ;39019  =0*     ;BRANCH ON ALU C31
                                ;39020          ;0*-----------------------------;
U 1A89, 0000,003C,0180,F930,0000,1A80   ;39021          LC_RC[T6],J/PACKH
                                ;39022
                                ;39023          ;1*-----------------------------;
                                ;39024          ALU_Q,RC[T1]_ALU,
                                ;39025          Q_ID[T0],
U 1A8B, 0001,233C,C1F0,2D88,0000,1A90   ;39026          C31?
                                ;39027  =;END
                                ;39028          ;-------------------------------;
                                ;39029  =0*     ;BRANCH ON ALU C31
                                ;39030          ;0*-----------------------------;
U 1A90, 0000,003C,0180,F930,0000,1A80   ;39031          LC_RC[T6],J/PACKH
                                ;39032
                                ;39033          ;1*-----------------------------;
                                ;39034          ALU_0+Q+1,D_ALU,
U 1A92, 081F,0010,0180,F800,0010,15BE   ;39035          CLK.UBCC
                                ;39036  =;END
```

```
                                        ;39037          ;--------------------------------;
                                        ;39038          ID[T0]_D,                        ;  STORE ROUNDED FRAC1
                                        ;39039          ALU_0+[C+1,
                                        ;39040          Q_D,
                                        ;39041          RC[T0]_ALU,D_ALU,
                                        ;39042          CLK.UBCC,
U 15BE, 0813,0310,C1E0,3D80,0010,1A91   ;39043          C31?
                                        ;39044
                                        ;39045          ;--------------------------------;
                                        ;39046  =0*     ;BRANCH ON ALU C31
                                        ;39047          ;0*------------------------------;
                                        ;39048          RC[T0]_LC,D_ALU,
U 1A91, 0810,0038,0180,F980,0000,15C0   ;39049          J/ADDH.390
                                        ;39050
                                        ;39051          ;1*------------------------------;
                                        ;39052          Q_RC[T6],LC_RC[T6],              ;  GET EXPONENT
U 1A93, 0010,0338,01C0,F930,0000,1A80   ;39053          C31?,J/PACKH                     ;  TEST FOR LEADING CARRY
                                        ;39054  =;END
                                        ;39055  ADDH.390:;-----------------------------;
U 15C0, 0000,003C,0180,F930,0000,1A80   ;39056          LC_RC[T6],J/PACKH                ;
```

```
                                ;39057  ;  ROUTINE TO NEGATE 4 LONGWORDS, STORED IN RC(SC-),ID(SC),RC(SC+1),ID(SC+1)
                                ;39058  ;EXPECTS:
                                ;39059  ;               LC = RC(SC+1)
                                ;39060  ;               Q = ID(SC+1)
                                ;39061
                                ;39062  ;THIS ROUTINE RELIES ON STEVE JENKINS' STATEMENT THAT
                                ;39063  ;THE ALU OPERATION A-B-1 WITH A=0,B=FFFFFFFF, WILL CAUSE CARRY TO BE CLEARED.
                                ;39064
                                ;39065  ALNNEGH:
                                ;39066          ;-------------------------------;
                                ;39067          D_0-Q,                          ;  NEGATE FRAC <VLS>
                                ;39068          Q_ID(SC),                       ;  GET FRAC <AMS>
                                ;39069          SC_SC+1,                        ;  INCREMENT POINTER AGAIN
U 15C1, 081F,0000,01F0,2400,0090,D5C2   ;39070          CLR.UBCC
                                ;39071
                                ;39072          ;-------------------------------;
                                ;39073          ID(SC)_D,                       ;  STORE NEGATED FRAC <VLS>
                                ;39074          RC(SC)_0-LC,                    ;  NEGATE FRAC <ALS>
                                ;39075          CLK.UBCC,
U 15C2, 0013,0300,0180,3438,0010,1A94   ;39076          C31?                            ;  WAS THERE A BORROW?
                                ;39077
                                ;39078          ;-------------------------------;
                                ;39079  =0*     ;BRANCH ON ALU C31
                                ;39080          ;0*-----------------------------;  BORROW
                                ;39081          ALU_0-LC-1,RC(SC)_ALU,          ;  STORE FRAC <ALS>
U 1A94, 0013,0008,0180,F838,0010,1A96   ;39082          CLK.UBCC                        ;  WE COULD HAVE A BORROW
                                ;39083
                                ;39084          ;1*-----------------------------;  NO BORROW
                                ;39085          D_0-Q,CLK.UBCC,                 ;  NEGATE FRAC <ALS>
                                ;39086          SC_SC-K[.1],                    ;  POINT TO <MS> FRACTIONS
U 1A96, 081F,0300,0580,F800,0094,BA98   ;39087          C31?
                                ;39088  =;END
                                ;39089          ;-------------------------------;
                                ;39090  =0*     ;BRANCH ON ALU C31
                                ;39091          ;0*-----------------------------;
                                ;39092          D_0-Q-1,                        ;  SUBTRACT WITH BORROW
                                ;39093          LC_RC(SC),                      ;  LATCH UP FRAC <VMS>
U 1A98, 081F,0008,0180,F830,0010,1A9A   ;39094          CLR.UBCC
                                ;39095
                                ;39096          ;1*-----------------------------;
                                ;39097          ALU_-1,D_ALU,                   ;  GET EXTENDER
                                ;39098          ID(SC)_D,                       ;  STORE FRAC <ALS>
                                ;39099          LC_RC(SC),                      ;  LATCH UP FRAC <VMS>
U 1A9A, 0803,0328,0180,3430,0000,1A99   ;39100          C31?
                                ;39101  =;END
                                ;39102          ;-------------------------------;
                                ;39103  =0*     ;BRANCH ON ALU 31
                                ;39104          ;0*-----------------------------;
                                ;39105          ALU_0-LC-1,RC(SC)_ALU,          ;  STORE FRAC <VMS>
U 1A99, 0013,000A,0180,F838,0000,0010   ;39106          RETURN[10]
                                ;39107
                                ;39108          ;1*-----------------------------;
                                ;39109          ALU_0-LC,RC(SC)_ALU,            ;  STORE FRAC <VMS>
U 1A9B, 0013,0002,0180,F838,0000,0010   ;39110          RETURN[10]
                                ;39111  =;END    ;-------------------------------;
```

```
                                ;39112  ; ROUTINE TO ALIGN ONE OF THE TWO ADDENDS, TO PREPARE FOR ADD
                                ;39113  ;INPUTS:
                                ;39114  ;       SC = ARGUMENT POINTER,(31 OR 33)
                                ;39115  ;       R15 IS POSITVE EXPONENT DIFFERENCE
                                ;39116  ;       FE = NEGATIVE EXPONENT DIFFERENCE (LOW BITS ONLY)
                                ;39117  ;       ID9 = FILLER, EITHER 0 OR -1
                                ;39118  ;       IN COMMENTS BELOW, X REFERS TO SC-1, I.E. 30 OR 31
                                ;39119  ;
                                ;39120  ;OUTPUTS:
                                ;39121  ;       SC = X, RC(X),ID(X),RC(X+1),ID(X+1) HAVE ALL BEEN SHIFTED RIGHT
                                ;39122  ;
                                ;39123  ; RETURN10
                                ;39124
                                ;39125
                                ;39126  ALNPOSH:
                                ;39127          ;-------------------------------;
                                ;39128          LAB_R[R15],ALU_LA.ANDNOT.K[.1F],;
U 15C4, 0018,0024,8D80,FA78,0010,15C5   ;39129          CLK.UBCC                        ;  CLOCK ALL BUT LOW 4 BITS
                                ;39130
                                ;39131          ;-------------------------------;
                                ;39132          LAB_R[R15],                     ;  LATCH POSITIVE EXP DIFF
                                ;39133          ALU_LA.ANDNOT.K[.7F],           ;  CLOCK ALL BUT LOW 7 BITS
                                ;39134          CLK.UBCC,                       ;
                                ;39135          Q_ID(SC),                       ;  GET LOW LONGWORD
U 15C5, 0018,0124,59F0,2678,0010,1A84   ;39136          Z?                              ;  CHECK SIZE OF SHIFT
                                ;39137
                                ;39138          ;-------------------------------;
                                ;39139  =0       ;BRANCH ON ALU Z-BIT
                                ;39140          ;0------------------------------;
                                ;39141          ALU_LA-K[.20],R[R15]_ALU,       ;  SUBTRACT 32 FROM EXPONENT DIFFERENCE
                                ;39142          CLK.UBCC,LONG,                  ;  CLOCK ALU Z-BIT ON NEW DIFFERENCE
U 1A84, 0018,0100,7580,FAF8,0010,1A8C   ;39143          Z?,J/ALNPOSH.1                  ;  TEST FOR SHIFTS > 127
                                ;39144
                                ;39145          ;1------------------------------;
                                ;39146          D_Q,Q_RC(SC),                   ;  GET LOW TWO FRACTION PARTS
U 1A85, 0C10,0038,01C0,F830,0181,15C6   ;39147          SC_FE,FE_SC                     ;  GET SHIFT COUNT
                                ;39148  =;END
                                ;39149          ;-------------------------------;
                                ;39150          D_DAL.SC,                       ;  SHIFT INTO FRAC<VLS>
U 15C6, 0D00,003C,0180,F800,0181,15C8   ;39151          SC_FE,FE_SC                     ;  SC GETS X+1, FE GETS SHIFT-COUNT
                                ;39152
                                ;39153          ;-------------------------------;
                                ;39154          ID(SC)_D,                       ;  STORE FRAC<VLS>
                                ;39155          D_Q,                            ;  D GETS OLD FRAC <ALS>
U 15C8, 0C00,003C,0580,3400,0084,B5C9   ;39156          SC_SC-K[.1]                     ;  POINT TO MS FRACTIONS INSTEAD
                                ;39157
                                ;39158          ;-------------------------------;
                                ;39159          Q_ID(SC),                       ;  Q GETS FRAC<AMS>
                                ;39160          LC_RC(SC),                      ;  LATCH FRAC <VMS>
U 15C9, 0000,003C,01F0,2430,0181,D5CA   ;39161          SC_FE,FE_SC+1                   ;  SC GETS SHIFT-COUNT, FE GETS X+1
                                ;39162
```

```
                                ;39163          ;-------------------------------;
                                ;39164          D_DAL.SC,                       ; SHIFT INTO FRAC <ALS>
U 15CA, 0D00,003C,0180,F800,0181,15CC ;39165    SC_FE,FE_SC                     ;  SC GETS X+1, FE GETS SHIFT-COUNT
                                ;39166
                                ;39167          ;-------------------------------;
                                ;39168          RC(SC)_D,                       ;  STORE FRAC<ALS>
                                ;39169          D_Q,                            ;  D GETS FRAC<AMS>
                                ;39170          FE_SC-K[.1],                    ;  FE GETS X
U 15CC, 0C01,003C,0580,F838,0185,B5CE ;39171    SC_FE                           ;  SC GETS SHIFT-COUNT
                                ;39172
                                ;39173          ;-------------------------------;
U 15CE, 0010,0038,01C0,F800,0000,15D0 ;39174    Q_LC                            ;  Q GETS FRAC<VMS>
                                ;39175
                                ;39176          ;-------------------------------;
                                ;39177          D_DAL.SC,                       ;  SHIFT INTO FRAC<AMS>
                                ;39178          SC_FE,FE_SC,                    ;  SC GETS X, FE GETS SHIFT-COUNT
U 15D0, 0D00,003C,E5F0,2C00,0181,15D1 ;39179    Q_ID[T9]                        ;  GET 0 OR -1
                                ;39180
                                ;39181          ;-------------------------------;
                                ;39182          ID(SC)_D,                       ;  STORE FRAC<AMS>
                                ;39183          D_LC,                           ;  D GETS FRAC<VMS>
U 15D1, 0810,0038,0180,3400,0181,15D2 ;39184    SC_FE,FE_SC                     ;  SC GETS SHIFT-COUNT, FE GETS X
                                ;39185
                                ;39186          ;-------------------------------;
                                ;39187          D_DAL.SC,                       ;  SHIFT INTO FRAC<VMS>
U 15D2, 0D00,003C,0180,F800,0181,15D3 ;39188    SC_FE,FE_SC                     ;  SC GETS X,FE GETS SHIFT-COUNT
                                ;39189
                                ;39190          ;-------------------------------;
U 15D3, 0001,003E,0180,F838,0000,0010 ;39191    RC(SC)_D,RETURN10               ;  STORE FRAC<VMS> IN RC(X)
                                ;39192
                                ;39193          ;-------------------------------;
                                ;39194 =0       ;  BRANCH ON ALU Z-BIT
                                ;39195          ;0------------------------------;
                                ;39196 ALNPOSH.1:                               ;  ALIGNMENT > 127
                                ;39197          Q_ID[T9],                       ;  GET FILLER, -1 OR 0
U 1A8C, 0000,003C,E5F0,2C00,0000,15D9 ;39198    J/ALNPOSH.2
                                ;39199
                                ;39200          ;1------------------------------; 31 < ALIGMENT < 128
U 1A8D, 0810,0038,0580,F830,0084,B5D4 ;39201    D_RC(SC),SC_SC-K[.1]
                                ;39202 =;END
                                ;39203          ;-------------------------------;
U 15D4, 0000,003C,01F0,2430,0080,D5D6 ;39204    Q_ID(SC),SC_SC+1,LC_RC(SC)
                                ;39205
                                ;39206          ;-------------------------------;
                                ;39207          ID(SC)_D,RC(SC)_Q,              ;  WRITE FRAC3 AND FRAC2
U 15D6, 0001,203C,0580,3438,0084,B5D8 ;39208    SC_SC-K[.1]
                                ;39209
```

```
                                    ;39210          ;---------------------------------;
                                    ;39211          D_LC,                             ;  D GETS FRAC0
                                    ;39212          Q_ID[T9],                         ;  Q GETS FILLER
U 15D8, 0810,0138,E5F0,2C00,0000,1A9C ;39213        Z?                                ;  TEST NEW ALIGNMENT SHIFT
                                    ;39214
                                    ;39215          ;---------------------------------;
                                    ;39216  =0      ;  BRANCH ON ALU Z-BIT
                                    ;39217          ;0--------------------------------;
                                    ;39218          ID(SC)_D,RC(SC)_Q,                ;  WRITE FRAC0 AND FRAC1
                                    ;39219          SC_SC+T,                          ;  RESTORE ORIGINAL POINTER
U 1A9C, 0001,203C,0180,3438,0080,D5C4 ;39220        J/ALNPOSH
                                    ;39221
                                    ;39222          ;1--------------------------------;
                                    ;39223          ID(SC)_D,RC(SC)_Q,                ;  WRITE FRAC0 AND FRAC1
U 1A9D, 0001,203E,0180,3438,0000,0010 ;39224        RETURN10
                                    ;39225  =;END
                                    ;39226          ;---------------------------------;
                                    ;39227  ALNPOSH.2:D_Q,                            ;  D GETS FILLER
U 15D9, 0C01,203C,0180,F838,0000,15DA ;39228        RC(SC)_Q                          ;  SO DOES RC(SC)
                                    ;39229
                                    ;39230          ;---------------------------------;
                                    ;39231          ID(SC)_D,                         ;  FILL WHOLE FRACTION WITH FILLER
U 15DA, 0000,003C,0580,3400,0084,B5DC ;39232        SC_SC-R[.1]                       ;  POINT TO MS FRACTIONS
                                    ;39233
                                    ;39234          ;---------------------------------;
                                    ;39235          RC(SC)_D,
U 15DC, 0001,003E,0180,3438,0000,0010 ;39236        ID(SC)_D,RETURN10
                                    ;39237          ;---------------------------------;
```

```
                                        ;39238  .TOC    ""      G & H floating point  : H-FORMAT FLOATING DIVIDE''
                                        ;39239
                                        ;39240  ; DIVH2,  *-R
                                        ;39241
                                        ;39242  ;INPUTS:
                                        ;39243  ;       RC0-ID1 = DIVISOR
                                        ;39244  ;       LA = LB = DIVIDEND LONG EXP
                                        ;39245
                                        ;39246  1429:   ;-------------------------------;
                                        ;39247          RC[T2]_LA,                      ;  GET DIVIDEND LONG EXP
                                        ;39248          K[.80],                         ;  FOR READ HUGE ROUTINE
                                        ;39249          D_0,                            ;
U 1429, 0F00,003D,4180,F990,0000,1862   ;39250          CALL,J/HUGE.REG.READ.1          ;
                                        ;39251
                                        ;39252  143F:   ;-------------------------------;
U 143F, 0000,003C,1980,F800,1404,7AA0   ;39253          STATE_K[ZERO]                   ;  NOTHING TO DO
                                        ;39254
                                        ;39255  =0****  ;0******************************;
U 1AA0, 0000,003D,0180,F800,0000,1960   ;39256          CALL,J/DIVH.SUBR
                                        ;39257
                                        ;39258          ;1******************************;
                                        ;39259          TRAP.ACC[1],                    ;  RESET THE ACCEL
U 1AB0, 0000,00BC,0080,F800,0000,18D4   ;39260          J/HUGE.REG.WRITE.1
                                        ;39261  =;END   ;-------------------------------;
                                        ;39262
                                        ;39263
                                        ;39264  ; DIVIDE H FORMAT FLOATING
                                        ;39265  ; ENTER HERE FOR *-MEMORY, AND *-*-*
                                        ;39266
                                        ;39267  ;INPUTS:
                                        ;39268  ;               RC0-ID1 = DIVISOR
                                        ;39269  ;               RC2-ID3 = DIVIDEND
                                        ;39270
                                        ;39271  148B:   ;-------------------------------;
                                        ;39272          STATE_K[ZERO],                  ;  NEED THIS FOR UNPACKH-ROUTINE
U 148B, 0000,003D,1980,F800,1404,7960   ;39273          CALL,J/DIVH.SUBR
                                        ;39274
                                        ;39275          ;-------------------------------;
U 149B, F000,003F,01F0,F847,0000,1400   ;39276  149B:   WRITE.G.DEST
                                        ;39277          ;-------------------------------;
```

```
                                ;39278  ;DIVIDE H FORMAT SUBROUTINE
                                ;39279
                                ;39280  ;INPUTS:
                                ;39281  ;               RC0-ID1 = DIVIDOR
                                ;39282  ;               RC2-ID3 = DIVIDEND
                                ;39283
                                ;39284  =000
                                ;39285  DIVH.SUBR:;000-------------------------;
                                ;39286          CLR.SD&SS,                      ;  CLEAR SIGN-FLIP-FLOPS
                                ;39287          ALU_RC[T2],N&Z_ALU.V&C_0,WORD,
U 1960, 0010,4039,0187,F910,0050,157B   ;39288          CALL,J/UNPACKH
                                ;39289
                                ;39290  DIVH.20:;001-------------------------;  DIVISOR = 0
U 1961, 0018,0038,D980,F9B8,0000,12B4   ;39291          RC[T7]_K[.9],J/FLOAT.FAULT
                                ;39292
                                ;39293          ;010-------------------------;  DIVIDEND = 0, DIVISOR NE 0
                                ;39294          D_0,Q_0,RC[T0]_0,
                                ;39295          SC_K[.30],
                                ;39296          N&Z_ALU.V&C_0,
U 1962, 0F03,003C,79F8,F980,00D4,7632   ;39297          J/CLEAR.DIRTY.0
                                ;39298
                                ;39299  =011    ;011-------------------------;  BOTH OPERANDS NON-ZERO
                                ;39300
                                ;39301  ;RETURNS WITH:
                                ;39302  ;       D= RC[T6] = DIVIDEND EXPONENT
                                ;39303  ;       RC0-ID1 = DIVISOR FRACTION
                                ;39304  ;       RC4 = DIVISOR EXPONENT (BIASSED)
                                ;39305  ;       RC2-ID3 = DIVIDEND FRACTION
                                ;39306  ;       RC6 = DIVIDEND EXPONENT(BIASSED)
                                ;39307  ;       SS = -SIGN
                                ;39308
                                ;39309          LC_RC[T4],                      ;  LATCH EXPONENT OF DIVISOR
                                ;39310          Q_ID[T1],                       ;  Q GETS FRAC1
U 1963, 0000,003C,C5F0,2D20,0000,15DD   ;39311          J7DIVH.S.1
                                ;39312
                                ;39313          ;100-------------------------;
U 1964, 0000,003C,0180,F800,0000,1961   ;39314  =100    J/DIVH.20                       ;  BOTH OPERANDS ARE 0
                                ;39315  =;END
                                ;39316          ;-------------------------------;
                                ;39317  DIVH.S.1:SD_SS,                         ;  NEED TO COMPLEMENT SIGN-FF
                                ;39318          ALU_D-LC,D_ALU,                 ;  EXP2-EXP1
                                ;39319          LAB_R[R1],                      ;  NEED TO SAVE R1
U 15DD, 0811,0000,B184,FA08,0000,15DE   ;39320          K[.4000]                        ;  BIAS
                                ;39321
                                ;39322          ;-------------------------------;
                                ;39323          SD_NOT.SD,                      ;  COMPLEMENT SIGN
                                ;39324          ALU_D+K[.4000],                 ;  BIASSED QUOTIENT EXPONENT
                                ;39325          RC[T6]_ALU,                     ;  STORE EXPONENT IN RC6
U 15DE, 0F19,0014,B183,F980,0000,15E0   ;39326          D_0
                                ;39327
                                ;39328          ;-------------------------------;
                                ;39329          ID[T1]_D,                       ;  CLEAR QUOTIENT
                                ;39330          D_LA,                           ;  D GETS R1
U 15E0, 0800,003C,C580,3D00,0000,15E1   ;39331          LC_RC[T0]                       ;  LATCH FRAC 1
```

```
                                    ;39332          ;-------------------------------;
                                    ;39333          ID[T8]_D,                       ;  SAVE R1 IN T8
U 15E1, 0010,0038,E180,3EF8,0000,15E2 ;39334        ALU_LC,R[R15]_ALU               ;  R15 GETS DI'END FRAC0
                                    ;39335
                                    ;39336          ;-------------------------------;
                                    ;39337          ALU_Q,RC[T5]_ALU,               ;  STORE FRAC3 IN RC5
                                    ;39338          D_Q,                            ;  D GETS FRAC3 AS WELL
U 15E2, 0C01,203C,C1F0,2DA8,0000,15E4 ;39339        Q_ID[T0]                        ;  Q GETS FRAC1
                                    ;39340
                                    ;39341          ;-------------------------------;
U 15E4, 0001,203C,0180,F9A0,0000,15E5 ;39342        RC[T4]_Q
                                    ;39343
                                    ;39344          ;-------------------------------;
U 15E5, 0000,003C,0180,F908,0000,15E6 ;39345        LC_RC[T1]
                                    ;39346
                                    ;39347          ;-------------------------------;
                                    ;39348          Q_ID[T3],                       ;  Q GETS D'END FRAC3
U 15E6, 0010,0038,CDF0,2E88,0000,1AA2 ;39349        R[R1]_LC                        ;  R1 GETS D'SOR FRAC2
                                    ;39350
                                    ;39351  =0      ;0-------------------------------;  CALL INNER DIVIDE LOOP
                                    ;39352          SC_K[.20],                      ;  INITIAL LOOP-COUNT
                                    ;39353          ALU_Q-D,                        ;  SUBTRACT
                                    ;39354          D_ALU.LEFT,                     ;  SHIFT RESULT LEFT
                                    ;39355          SET.CC(LONG),                   ;  CLOCK BORROW
                                    ;39356          SI/ZERO,                        ;
                                    ;39357          LAB_R[R1],                      ;
U 1AA2, 083D,2001,7580,FA08,00F4,75F1 ;39358        CALL,J/DIVH.LOOP.1              ;  START WITH A SUBTRACT
                                    ;39359
                                    ;39360          ;1-------------------------------;
                                    ;39361          RC[T0]_D,D_0,                   ;  SAVE HIGH QUOTIENT IN RC0
U 1AA3, 0F01,003C,7580,F980,0184,7AA4 ;39362        SC_K[.20],FE_K[.20]
                                    ;39363  =;END
                                    ;39364  =0      ;0-------------------------------;  CALL INNER DIVIDE LOOP
                                    ;39365          ID[T1]_D,                       ;  CLEAR QUOTIENT
                                    ;39366          D_RC[T5],                       ;  GET D'SOR FRAC3
U 1AA4, 0810,0339,C580,3D28,0000,1AAC ;39367        C31?,CALL,J/DIVH.LOOP           ;  DO A RETURN1
                                    ;39368
                                    ;39369          ;1-------------------------------;
U 1AA5, 0F00,003C,C180,3C00,0000,1AA8 ;39370        ID[T0]_D,D_0                    ;
                                    ;39371  =;END
                                    ;39372  =0      ;0-------------------------------;  CALL INNER DIVIDE LOOP
                                    ;39373          SC_FE,                          ;
                                    ;39374          D_RC[T5],                       ;
                                    ;39375          ID[T1]_D,                       ;  CLEAR QUOTIENT
U 1AA8, 0810,0339,C580,3D28,0081,1AAC ;39376        C31?,CALL,J/DIVH.LOOP           ;
                                    ;39377
                                    ;39378          ;1-------------------------------;
                                    ;39379          RC[T1]_D,                       ;  STORE QUOTIENT FRAC2
                                    ;39380          SC_K[.T4],                      ;  FEWER PASSES IN LAST PASS
U 1AA9, 0F01,003C,2180,F988,0084,7AAA ;39381        D_0
                                    ;39382  =;END
```

```
                                    ;39383  =0      ;0-------------------------------;  CALL INNER DIVIDE LOOP
                                    ;39384          D_RC[T5],                       ;  D GETS DIVISOR FRAC3
                                    ;39385          ID[T1]_D,                       ;  CLEAR INITIAL QUOTIENT
U 1AAA, 0810,0339,C580,3D28,0000,1AAC ;39386        C31?,CALL,J/DIVH.LOOP
                                    ;39387
                                    ;39388          ;1-------------------------------;
U 1AAB, 0000,003C,E1F0,2C00,0000,15E8 ;39389        Q_ID[T8]                        ;  RETRIEVE ORIGINAL R1
                                    ;39390
                                    ;39391          ;-------------------------------;
U 15E8, 0001,203C,3DF0,2E88,0000,15E9 ;39392        R[R1]_Q,Q_ID[PSL]               ;  RESTORE OLD R1,GET PSL
                                    ;39393
                                    ;39394          ;-------------------------------;
U 15E9, 0000,003C,8580,F800,0084,75EA ;39395        SC_K[.C]                        ;  NEED TO LEFT-ADJUST LOW FRAC
                                    ;39396
                                    ;39397          ;-------------------------------;
                                    ;39398          Q_Q.ANDNOT.K[.1],               ;  CLEAR C BIT
U 15EA, 0D19,2024,05C0,F800,0000,15EC ;39399        D_DAL.SC                        ;
                                    ;39400
                                    ;39401          ;-------------------------------;
U 15EC, 0000,003C,1980,F800,0084,75ED ;39402        SC_K[ZERO]                      ;
                                    ;39403
                                    ;39404          ;-------------------------------;
U 15ED, 0C00,003C,C580,3C00,0000,15EE ;39405        ID[T1]_D,D_Q                    ;  STORE QUOTIENT FRAC <VLS>
                                    ;39406
                                    ;39407          ;-------------------------------;
U 15EE, 0000,003C,3D80,3C00,0000,15A5 ;39408        ID[PSL]_D,J/ADDH.310            ;  GO TO ROUND AND PACK RESULT
                                    ;39409  =;END   ;-------------------------------;
```

```
                                ;39410  ; INNER LOOP FOR DIVH
                                ;39411  ;EXPECTS:
                                ;39412  ;       DIVISOR IN R15,RC4,R1 (=LA),RC5
                                ;39413  ;       DIVIDEND IN RC2-ID3
                                ;39414  ;
                                ;39415  ;       QUOTIENT IN ID1
                                ;39416
                                ;39417  ;       LOOP COUNT IS IN SC
                                ;39418  ;       ALU CARRY DETERMINES WHETHER WE DO ADD OR SUBTRACT
                                ;39419  ;       STATE<4> TELLS US OF FIRST TIME PASS
                                ;39420
                                ;39421  DIVH.LOOP.0:
                                ;39422          ;-------------------------------;
                                ;39423          D_RC[T5],                       ;  D GETS D'SOR FRAC3
                                ;39424          Q_ID[T3],                       ;  Q GETS D'END FRAC3
U 15F0, 0810,0338,CDF0,2D28,0000,1AAC   ;39425          C31?                            ;  ADD OR SUBTRACT
                                ;39426
                                ;39427          ;-------------------------------;
                                ;39428  =0*     ;BRANCH ON ALU C31
                                ;39429          ;0*-----------------------------;  BORROW
                                ;39430  DIVH.LOOP:ALU_D+Q,D_ALU.LEFT,           ;  ADD AND SHIFT LEFT
                                ;39431          SI/ZERO,                        ;  LOW BITS
                                ;39432          SET.CC(LONG),                   ;  CLOCK C AND N OF PSL
                                ;39433          LC_RC[T3],                      ;  LATCH D'END FRAC 2
                                ;39434          Q_ID[T2],                       ;  READ D'END FRAC 1
U 1AAC, 083D,0014,C9F0,2D18,0070,15F9   ;39435          J/DIVH.LOOP.ADD.2
                                ;39436
                                ;39437          ;1*-----------------------------;  NO BORROW, SUBTRACT
                                ;39438          STATE_STATE.ANDNOT.K[.10],      ;  CLEAR FIRST TIME FLAG
                                ;39439          ALU_Q-D,                        ;  SUBTRACT WITH BORROW
                                ;39440          D_ALU.LEFT,                     ;  SHIFT LEFT
                                ;39441          SET.CC(LONG),                   ;  CLOCK C AND N OF PSL
                                ;39442          SI/ZERO,                        ;  LOW BITS
U 1AAE, 083D,2000,6580,FA08,1474,55F1   ;39443          LAB_R[R1]                       ;  LATCH D'SOR FRAC2
                                ;39444  =:END
                                ;39445  DIVH.LOOP.1:;-------------------------------;
                                ;39446          ID[T3]_D,                       ;  STORE IN T3
U 15F1, 0810,0038,CD80,3D18,0000,15F2   ;39447          D_RC[T3]                        ;  GET D'END FRAC2
                                ;39448
                                ;39449          ;-------------------------------;
                                ;39450          ALU_D[INST.DEP]LB,              ;  SUBTRACT WITH BORROW
                                ;39451          SET.CC(LONG),                   ;  CLOCK PSL-C FOR POSSIBLE BORROW
                                ;39452          RC[T3]_ALU.LEFT,                ;  RESTORE D'END FRAC2
                                ;39453          Q_ID[T2],                       ;  Q GETS D'END FRAC1
U 15F2, 002D,000C,C870,2D98,0070,15F3   ;39454          SI/DIVD                         ;  SHIFT IN PSL N-BIT
                                ;39455
                                ;39456          ;-------------------------------;
                                ;39457          LC_RC[T4],                      ;  LATCH D'SOR FRAC1
                                ;39458          D_Q,                            ;  D GETS D'END FRAC1
U 15F3, 0C00,003C,C5F0,2D20,0000,15F4   ;39459          Q_ID[T1]                        ;  Q GETS QUOTIENT
                                ;39460
```

```
                                ;39461          ;--------------------------------;
                                ;39462          SC_SC-K[.1],                     ;  DECREMENT COUNT
                                ;39463          ALU_D[INST.DEP]LC,               ;  SUBTRACT WITH BORROW
                                ;39464          D_ALU.LEFT,                      ;  D GETS RESULT*2
                                ;39465          SI/DIVD,                         ;  SHIFT IN PSL N-BIT
                                ;39466          SET.CC(LONG),                    ;  CLOCK C AND N OF PSL
U 15F4, 0831,000C,0400,FA78,00F4,B5F5   ;39467          LAB_R[R15]                       ;  LATCH D'SOR FRAC0
                                ;39468
                                ;39469          ;--------------------------------;
                                ;39470          ID[T2]_D,                        ;  RESTORE D'END FRAC1
U 15F5, 0810,0038,C980,3D10,0000,15F8   ;39471          D_RC[T2]                         ;  GET D'END FRAC0
                                ;39472
                                ;39473          ;--------------------------------;
                                ;39474          ALU_D[INST.DEP]LB,               ;  SUBTRACT WITH BORROW
                                ;39475          CLK.UBCC,                        ;  CLOCK ALU BORROW
                                ;39476          RC[T2]_ALU.LEFT,
                                ;39477          QK/LEFT,                         ;  SHIFT QUOTIENT LEFT
                                ;39478          SI/DIVD,                         ;  SHIFT IN PSL N-BIT
                                ;39479          SC.GT.0?,                        ;  COUNTER STILL POSITVE ?
U 15F8, 002D,140C,0028,F990,0010,1AB1   ;39480          J/DIVH.LOOP.ADD3
                                ;39481
                                ;39482  DIVH.LOOP.ADD.2:;--------------------------;
                                ;39483          ID[T3]_D,                        ;  REWRITE NEW D'END
                                ;39484          ALU_LA+LC+PSL.C,                 ;  NOTE THAT LA = R1
                                ;39485          RC[T3]_ALU.LEFT,                 ;  ADD AND SHIFT
                                ;39486          SET.CC(LONG),                    ;  CLOCK C AND N OF PSL
U 15F9, 0030,002C,CC00,3D98,0070,15FA   ;39487          SI/DIVD                          ;  SHIFT IN PSL N-BIT
                                ;39488
                                ;39489          ;--------------------------------;
                                ;39490          SC_SC-K[.1],                     ;  DECREMENT COUNT
U 15FA, 0810,0038,0580,F920,0084,B5FC   ;39491          D_RC[T4]                         ;  GET D'SOR FRAC1
                                ;39492
                                ;39493  DIVH.ADD.LOOP.20:;--------------------------;
                                ;39494          ALU_D+Q+PSL.C,D_ALU.LEFT,        ;  ADD TO GET NEW D'END FRAC1
                                ;39495          SI/DIVD,                         ;  SHIFT IN PSL N-BIT
                                ;39496          SET.CC(LONG),                    ;  CLOCK C AND N OF PSL
                                ;39497          LC_RC[T2],                       ;  LATCH D'END FRAC0
U 15FC, 083D,002C,C470,2D10,0070,15FD   ;39498          Q_ID[T1]                         ;  GET QUOTIENT
                                ;39499
                                ;39500          ;--------------------------------;
U 15FD, 0000,003C,C980,3E78,0000,15FE   ;39501          LAB_R[R15],ID[T2]_D              ;  TOO BAD, WE NEED THIS STATE
                                ;39502
                                ;39503          ;--------------------------------;
                                ;39504          ALU_LA+LC+PSL.C,                 ;  ADD FRAC0'S
                                ;39505          RC[T2]_ALU.LEFT,                 ;  STORE IN RC2
                                ;39506          SI/DIVD,                         ;  SHIFT IN PSL N-BIT
                                ;39507          CLK.UBCC,                        ;  USE ALU CARRY FOR ADD/SUB
                                ;39508          QK/LEFT,                         ;  SHIFT QUOTIENT
U 15FE, 0030,142C,0028,F990,0010,1AB1   ;39509          SC.GT.0?                         ;  END OF COUNT ?
                                ;39510
```

```
                                    ;39511          ;--------------------------------;
                                    ;39512  =*0*    ;BRANCH ON SC GT 0
                                    ;39513          ;*0*-----------------------------;
                                    ;39514  DIVH.LOOP.ADD3:LAB_R[R1],                ;  LATCH DIVISOR FRAC 2
U 1AB1, 0C00,003E,CDF0,2E08,0000,0001 ;39515        D_Q,Q_ID[T3],RETURN1             ;
                                    ;39516
                                    ;39517          ;*1*-----------------------------;
                                    ;39518          D_Q,Q_ID[T3],                    ;  GET D'END FRAC3
U 1AB3, 0C00,003C,CDF0,2E08,0000,1601 ;39519        LAB_R[R1]                        ;  LATCH DIVISOR FRAC 2
                                    ;39520  =;END
                                    ;39521          ;--------------------------------;
                                    ;39522          D_RC[T5],                        ;  GET D'SOR FRAC3
                                    ;39523          ID[T1]_D,                        ;  STORE QUOTIENT
                                    ;39524          C31?,                            ;  ADD AOR SUBTRACT NEXT ?
U 1601, 0810,0338,C580,3D28,0000,1AAC ;39525        J/DIVH.LOOP
                                    ;39526          ;--------------------------------;
```

```
                                ;39527  .TOC    ''      G & H floating point  : MULTIPLY H FLOATING''
                                ;39528  ; MULH
                                ;39529  ; HERE FOR *-R
                                ;39530
                                ;39531  ;INPUTS:
                                ;39532  ;       RC0-ID1 = M'PLIER
                                ;39533  ;       LA = LB = M'CAND LONG EXP
                                ;39534
                                ;39535  1428:   ;-------------------------------;
                                ;39536          RC[T2] LA,                      ;  GET DIVIDEND LONG EXP
                                ;39537          N&Z_ALU.V&C_0,WORD,             ;  CLEAR C AND V OF PSL
                                ;39538          K[.80],                         ;  FOR READ HUGE ROUTINE
                                ;39539          D_0,                            ;
U 1428, 0F00,403D,4180,F990,0050,1862   ;39540          CALL,J/HUGE.REG.READ.1          ;
                                ;39541
                                ;39542  143E:   ;-------------------------------;
U 143E, 0000,003C,1980,F800,1404,7AA6   ;39543          STATE_K[ZERO]                   ;  NOTHING TO DO
                                ;39544
                                ;39545  =0****  ;0****-------------------------;
U 1AA6, 0000,003D,0180,F800,0000,1970   ;39546          CALL,J/MULH.00                  ;  MULTIPLY HUGE SUBROUTINE
                                ;39547
                                ;39548          ;1****-------------------------;
                                ;39549          TRAP.ACC[1],                    ;  RESET THE ACCEL
U 1AB6, 0000,00BC,0080,F800,0000,18D4   ;39550          J/HUGE.REG.WRITE.1
                                ;39551          ;-------------------------------;
                                ;39552
                                ;39553  ; HERE FOR MULH *-M, AND *-*-*
                                ;39554
                                ;39555  ; INPUTS:
                                ;39556  ;       RC0-ID1 = M'PLIER
                                ;39557  ;       RC2-ID3 = M'CAND
                                ;39558
                                ;39559          ;-------------------------------;
U 148C, 0000,093C,0180,F800,0000,1842   ;39560  148C:   IR2-1?                          ;  IS THIS MULH OR TSTH ?
                                ;39561
                                ;39562          ;-------------------------------;
                                ;39563  =0**10  ;  BRANCH ON IR<2:1>             ;
                                ;39564          ;0**10-------------------------;
                                ;39565          STATE_K[ZERO],                  ;
                                ;39566          ALU_RC[T0],N&Z_ALU.V&C_0,WORD,  ;  CLEAR C AND V OF PSL
U 1842, 0010,4039,1980,F900,1454,7970   ;39567          CAL[,J/MULH.00                  ;
                                ;39568
                                ;39569          ;0**10-------------------------;
U 1843, 0010,0038,01C0,F900,0000,181D   ;39570          ALU_RC[T0],Q_ALU,J/TSTH.00      ;  TSTH, M-FORMAT
                                ;39571
                                ;39572          ;1**10-------------------------;
U 1852, F000,003F,01F0,F847,0000,1400   ;39573  =1**10  WRITE.G.DEST                    ;
                                ;39574  =;END   ;-------------------------------;
                                ;39575
```

```
                                ;39576  ; MULTIPLY-ROUTINE
                                ;39577  ; USED BY POLYH AND EMODH AS  WELL
                                ;39578
                                ;39579  ;INPUTS:
                                ;39580  ;               RC0-ID1 = MULTIPLIER (REFERRED TO AS A0-A3)
                                ;39581  ;               RC2-ID3= MULTIPLICAND (= B0-B3)
                                ;39582  ;OUTPUTS:
                                ;39583  ;               RC0-ID1 = PRODUCT FRACTION
                                ;39584  ;TEMPORARIES:
                                ;39585  ;               T6-T9 TEMPORARILY HOLDS PRODUCT
                                ;39586  ;               RC4-RC6 USED FOR EXPONENT AND TEMP. STORAGE
                                ;39587
                                ;39588  =000
                                ;39589  MULH.00:;000-------------------------;
                                ;39590          CLR.SD&SS,                      ;  CLEAR SIGN-FLIP-FLOPS
U 1970, 0000,003D,0187,F800,0000,157B   ;39591          CALL,J/UNPACKH
                                ;39592
                                ;39593          ;001-------------------------;  MULTIPLIER = 0
                                ;39594          SC_K[.30],                      ;  POINTER FOR DEST
                                ;39595          D_0,Q_0,                        ;  PRODUCT IS 0
                                ;39596          RC[T1]_0,                       ;  STORE 0
                                ;39597          N&Z_ALU.V&C_0,                  ;  SET Z, CLEAR N
U 1971, 0F03,003C,79F8,F988,00D4,7632   ;39598          J/CLEAR.DIRTY.0
                                ;39599
                                ;39600          ;010-------------------------;  MULTIPLICAND = 0
                                ;39601          SC_K[.30],                      ;  POINTER FOR DEST
                                ;39602          D_0,Q_0,                        ;  PRODUCT IS 0
                                ;39603          RC[T1]_0,                       ;  STORE 0
                                ;39604          N&Z_ALU.V&C_0,                  ;  SET Z, CLEAR N
U 1972, 0F03,003C,79F8,F988,00D4,7632   ;39605          J/CLEAR.DIRTY.0
                                ;39606
                                ;39607          ;011-------------------------;  BOTH OPERANDS NON-ZERO
                                ;39608  ;RETURN WITH:
                                ;39609  ;       RC0-ID1 = MULTIPLIER FRACTION = A0,A1,A2,A3
                                ;39610  ;       RC2-ID3 = MULTIPLICAND FRACTION = B0,B1,B2,B3
                                ;39611  ;       RC4 = MULTIPLIER EXPONENT (BIASSED)
                                ;39612  ;       RC6 = MULTIPLICAND EXPONENT (BIASSED)
                                ;39613  ;       D = 'CAND EXPONENT
                                ;39614  ;       SS = SIGN, FOR MULH AT LEAST
                                ;39615  ;
                                ;39616  ; THE PROBLEM IS THAT THE MULTIPLICAND FRACTION IS NOT ALIGNED QUITE
                                ;39617  ; THE WAY WE WANT IT, NAMELY WITH A LEADING ZERO IN EACH LONGWORD.
                                ;39618
                                ;39619          SD_SS,                          ;  DUPLICATE SIGN FLIP-FLOP
                                ;39620          Q_ID[T2],                       ;  GET LOW 'CAND FRAC1
                                ;39621          LC_RC[T4],                      ;  LATCH 'PLIER EXPONENT
U 1973, 0000,003C,C9F4,2D20,0000,1602   ;39622          J/MULH.01
                                ;39623
                                ;39624  =100    ;100-------------------------;  BOTH ARE 0
                                ;39625          SC_K[.30],                      ;  POINTER FOR DEST
                                ;39626          D_0,Q_0,                        ;  PRODUCT IS 0
                                ;39627          RC[T1]_0,                       ;  STORE 0
                                ;39628          N&Z_ALU.V&C_0,                  ;  SET Z, CLEAR N
U 1974, 0F03,003C,79F8,F988,00D4,7632   ;39629          J/CLEAR.DIRTY.0
                                ;39630  =;END   ;-------------------------;
```

```
                                ;39631  MULH.01::;--------------------------------;
                                ;39632          ALU_D+LC,RC[T6]_ALU,            ;  SUM THE TWO EXPONENTS
U 1602, 0C11,0014,01B0,F9B0,0000,1603   ;39633          QK/RIGHT,SI/ZERO,D_Q            ;  M'CAND FRAC1 GETS LEADING 0
                                ;39634
                                ;39635          ;--------------------------------;
                                ;39636          D_Q,Q_D,                        ;  D GETS FRAC1/2
                                ;39637          ALU_NOT.K[.FFFC],R[R15]_ALU,    ;  R15 GETS 3
U 1603, 0C1B,001C,F1E0,FAF8,0084,7605   ;39638          SC_R[.FFFC]                     ;  SC GETS -4
                                ;39639
                                ;39640          ;--------------------------------;
                                ;39641          ID[T2]_D,                       ;  RESTORE FRAC1
U 1605, 0810,0038,C980,3D18,0000,160E   ;39642          D_RC[T3]                        ;  D GETS FRAC2
                                ;39643
                                ;39644          ;--------------------------------;
                                ;39645          DK/RIGHT,SI/ASHR,               ;  SHIFT IN ONE BIT FROM FRAC1
U 160E, 0600,003C,CCF0,2C00,0000,1610   ;39646          Q_ID[T3]                        ;  GET FRAC3
                                ;39647
                                ;39648          ;--------------------------------;
                                ;39649          ALU_D,RC[T3]_ALU.RIGHT,SI/ZERO, ;  RESTORE SHIFTED FRAC2
U 1610, 0C41,003C,0180,F998,0000,1611   ;39650          D_Q                             ;
                                ;39651
                                ;39652          ;--------------------------------;
                                ;39653          LAB_R[R15],                     ;  LATCH 3
                                ;39654          ALU_LA.AND.LC,Q_ALU,            ;  ISOLATE LOW 2 BITS OF FRAC2
U 1611, 0010,0034,01C0,FA78,0080,D612   ;39655          SC_SC+1                         ;  SC NOW HAS -3
                                ;39656
                                ;39657          ;--------------------------------;
                                ;39658          D_DAL.SC,                       ;  GENERATE NEW FRAC3
U 1612, 0D00,003C,C5F0,2C00,0000,1AB8   ;39659          Q_ID[T1]                        ;  GET READY TO MULTIPLY BY A3
                                ;39660
                                ;39661          ;0*-------------------------------;
                                ;39662  =0*     ALU_D,RC[T4]_ALU.LEFT,          ;  RC 4 GETS B3*2
                                ;39663          ID[T3]_D,SI/ZERO,               ;  ID3 GETS B3
                                ;39664          D_Q,Q_D,                        ;  D GETS A3, Q GETS B3
U 1AB8, 0C21,003D,CDE0,3DA0,0000,1947   ;39665          CALL,J/MPY.HUGE.1
                                ;39666
                                ;39667          ;1*-------------------------------;  RETURN WITH:  <Q,D> = A3*B3
                                ;39668          LC_RC[T3],                      ;  LATCH B2
U 1ABA, 0000,003C,0128,F918,0000,1613   ;39669          QK/LEFT, SI/ASHL                ;  NEED TO SAVE HIGH 32 BITS ONLY
                                ;39670
                                ;39671          ;--------------------------------;
                                ;39672          ALU_LC,RC[T4]_ALU.LEFT,SI/ZERO, ;  RC4 GETS B2*2
U 1613, 0C30,0038,C5F0,2DA0,0000,1614   ;39673          D_Q,Q_ID[T1]                    ;  Q GETS A3
                                ;39674
                                ;39675          ;--------------------------------;
                                ;39676          ID[T9]_D,D_0,                   ;  STORE PARTIAL PRODUCT <LS>
U 1614, 0F10,0038,E580,3EF8,0000,1A00   ;39677          ALU_LC,R[RT5]_ALU               ;  R15 GETS B2
                                ;39678
```

```
                                        ;39679          ;00------------------------------;
                                        ;39680  =00     ID[T8]_D,                       ;  CLEAR PARTIAL PROD <MS>
                                        ;39681          D_LC,                           ;  D GETS B2
U 1A00, 0810,0039,E180,3C00,0000,1956   ;39682          CALL,J/MULTIPLY.HUGE.SUBR
                                        ;39683
                                        ;39684          ;10------------------------------;  RETURN WITH:  <Q,D> = A3*B2
                                        ;39685  =10     SC_K[.FFFF],                    ;  SC GETS -1
U 1A02, 0000,003D,C180,F800,0084,7626   ;39686          CALL,J/HUGE.MUL.ADD.SHF         ;  ADD TO PARTIAL PRODUCT AND SHIFT
                                        ;39687
                                        ;39688  =11     ;11------------------------------;
                                        ;39689          Q_ID[T3],                       ;  Q GETS B3
U 1A03, 0810,0038,CDF0,2D08,0000,1A04   ;39690          D_RC[T1]                        ;  D GETS A2
                                        ;39691  =;END
                                        ;39692  =00     ;00------------------------------;
                                        ;39693          ALU_Q,RC[T4]_ALU.LEFT,          ;  RC4 GETS B3*2
                                        ;39694          SI/ZERO,
U 1A04, 0021,203D,0180,F9A0,0000,1947   ;39695          CALL,J/MPY.HUGE.1
                                        ;39696
                                        ;39697          ;10------------------------------;   RETURN WITH :  <Q,D> = A2*B3
                                        ;39698  =10     SC_K[.2],                       ;  RIGHT SHIFT OF 32.
U 1A06, 0000,003D,0980,F800,0084,7626   ;39699          CALL,J/HUGE.MUL.ADD.SHF         ;
                                        ;39700
                                        ;3970;          ;11------------------------------;
U 1A07, 0000,003C,C9F0,2C00,0000,1A08   ;39702  =11     Q_ID[T2]                        ;  Q GETS B1
                                        ;39703  =;END
                                        ;39704          ;00------------------------------;
                                        ;39705  =00     D_Q,                            ;  D GETS B1
                                        ;39706          Q_ID[T1],                       ;  Q GETS A3
U 1A08, 0C00,003D,C5F0,2C00,0000,1956   ;39707          CALL,J/MULTIPLY.HUGE.SUBR
                                        ;39708
                                        ;39709          ;10------------------------------;  RETURN WITH:  <Q,D> = A3*B1
                                        ;39710  =10     SC_K[.FFFF],                    ;  SC GETS -1 FOR RIGHT SHIFT
U 1A0A, 0000,003D,C180,F800,0084,7626   ;39711          CALL,J/HUGE.MUL.ADD.SHF
                                        ;39712
                                        ;39713          ;11------------------------------;
                                        ;39714  =11     Q_RC[T3],                       ;  Q GETS B2
U 1A0B, 0F10,0038,01C0,F918,0000,1A10   ;39715          D_0                             ;  NEED TO CLEAR OUT T7
                                        ;39716  =;END
                                        ;39717          ;00------------------------------;
                                        ;39718  =00     ID[T7]_D,                       ;  CLEAR OUT PARTIAL PRODUCT FRAC1
                                        ;39719          D_Q,                            ;  D GETS B2
                                        ;39720          Q_RC[T1],                       ;  Q GETS A2
U 1A10, 0C10,0039,DDC0,3D08,0000,1956   ;39721          CALL, J/MULTIPLY.HUGE.SUBR
                                        ;39722
                                        ;39723          ;10------------------------------;  RETURN WITH:   <Q,D> = A2*B2
                                        ;39724  =10     SC_K[.FFFF],                    ;  SC GETS -1
U 1A12, 0000,003D,C180,F800,0084,7626   ;39725          CALL,J/HUGE.MUL.ADD.SHF
                                        ;39726
                                        ;39727          ;11------------------------------;
U 1A13, 0000,003C,CDF0,2C00,0000,1A14   ;39728  =11     Q_ID[T3]                        ;  Q GETS B3
                                        ;39729  =;END
```

```
                                ;39730          ;00------------------------------;
                                ;39731  =00     D_Q,Q_ID[T0],                    ;  D GETS B3, Q GETS A1
U 1A14, 0C00,003D,C1F0,2C00,0000,1956   ;39732          CALL,J/MULTIPLY.HUGE.SUBR
                                ;39733
                                ;39734          ;10------------------------------;  RETURN WITH:  <Q,D> = A1*B3
                                ;39735  =10     SC_K[.3],                        ;  LOAD SC FOR RIGHT SHIFT OF 29.
U 1A16, 0000,003D,0D80,F800,0084,7626   ;39736          CALL,J/HUGE.MUL.ADD.SHF
                                ;39737
                                ;39738          ;11------------------------------;
                                ;39739  =11     Q_ID[T1],                        ;  Q GETS A3
U 1A17, 0810,0038,C5F0,2D10,0000,1A20   ;39740          D_RC[T2]                         ;  D GETS B0
                                ;39741  =;END
                                ;39742          ;00------------------------------;
                                ;39743  =00     ALU_D,RC[T4]_ALU.LEFT,           ;  RC4 GETS B0*2
                                ;39744          SI/ZERO,
                                ;39745          D_Q,Q_D,                         ;  SWAP A3 AND B0
U 1A20, 0C21,003D,01E0,F9A0,0000,1947   ;39746          CALL,J/MPY.HUGE.1
                                ;39747
                                ;39748          ;10------------------------------;  RETURN WITH:  <Q,D> = A3*B0
                                ;39749  =10     SC_K[.FFFF],                     ;  SC GETS -1
U 1A22, 0000,003D,C180,F800,0084,7626   ;39750          CALL,J/HUGE.MUL.ADD.SHF
                                ;39751
                                ;39752          ;11------------------------------;
                                ;39753  =11     Q_ID[T2],                        ;  Q GETS B1
U 1A23, 0810,0038,C9F0,2D08,0000,1A24   ;39754          D_RC[T1]                         ;  D GETS A2
                                ;39755  =;END
                                ;39756          ;00------------------------------;
                                ;39757  =00     ALU_Q,RC[T4]_ALU.LEFT,           ;  RC4 GETS B1*2
                                ;39758          SI/ZERO,
U 1A24, 0021,203D,0180,F9A0,0000,1947   ;39759          CALL,J/MPY.HUGE.1
                                ;39760
                                ;39761          ;10------------------------------;  RETURN WITH :  <Q,D> = A2*B1
                                ;39762  =10     SC_K[.FFFF],                     ;  SC GETS -1
U 1A26, 0000,003D,C180,F800,0084,7626   ;39763          CALL,J/HUGE.MUL.ADD.SHF
                                ;39764
                                ;39765          ;11------------------------------;
                                ;39766  =11     Q_ID[T0],                        ;  Q GETS A1
U 1A27, 0810,0038,C1F0,2D18,0000,1A28   ;39767          D_RC[T3]                         ;  D GETS B2
                                ;39768  =;END
                                ;39769          ;00------------------------------;
                                ;39770  =00     ALU_D,RC[T4]_ALU.LEFT,           ;  RC4 GETS B2*2
                                ;39771          SI/ZERO,
                                ;39772          D_Q,Q_D,                         ;  SWAP A1 WITH B2
U 1A28, 0C21,003D,01E0,F9A0,0000,1947   ;39773          CALL,J/MPY.HUGE.1
                                ;39774
                                ;39775          ;10------------------------------;  RETURN WITH:  <Q,D> = A1*B2
                                ;39776  =10     SC_K[.FFFF],                     ;  SC GETS -1
U 1A2A, 0000,003D,C180,F800,0084,7626   ;39777          CALL,J/HUGE.MUL.ADD.SHF
                                ;39778
                                ;39779          ;11------------------------------;
                                ;39780  =11     D_RC[T0],                        ;  D GETS A0
U 1A2B, 0810,0038,CDF0,2D00,0000,1A30   ;39781          Q_ID[T3]                         ;  Q GETS B3
                                ;39782  =;END
```

```
                                        ;39783          ;00----------------------------;
                                        ;39784  =00     ALU_Q,RC[T4]_ALU.LEFT,          ;  RC4 GETS B3*2
                                        ;39785          SI/ZERO,                        ;
U 1A30, 0021,203D,0180,F9A0,0000,1947   ;39786          CALL,J/MPY.HUGE.1               ;
                                        ;39787
                                        ;39788          ;10----------------------------;  RETURN WITH:  <Q,D> = A0*B3
                                        ;39789  =10     SC_K[.4],                       ;  SHIFT 28. TO KEEP 3 GUARD BITS
U 1A32, 0000,003D,1180,F800,0084,7626   ;39790          CALL,J/HUGE.MUL.ADD.SHF
                                        ;39791
                                        ;39792          ;11----------------------------;
                                        ;39793  =11     SC_K[.30],                      ;  BASE FOR ID-REGISTERS
U 1A33, 0810,0038,7980,F910,0084,7920   ;39794          D_RC[T2]                        ;  GET B0
                                        ;39795
                                        ;39796          ;00*---------------------------;
                                        ;39797  =00*    FE_SC+K[.9],                    ;  FE GETS .39 = ADDRESS OF T9
                                        ;39798          Q_RC[T1],                       ;  Q GETS A2
U 1920, 0010,0039,D9C0,F908,0104,9956   ;39799          CALL, J/MULTIPLY.HUGE.SUBR
                                        ;39800
                                        ;39801          ;01*---------------------------;  RETURN WITH:  <Q,D> = A2*B0
                                        ;39802  =01*                                    ;                       FE = .39
                                        ;39803          SC_K[.1],                       ;  SHIFT COUNT OF 31.
U 1922, 0000,003D,0580,F800,0084,7AC0   ;39804          CALL,J/MULH.ADD                 ;  ADD IN NEW PRODUCT
                                        ;39805
                                        ;39806          ;11*---------------------------;
                                        ;39807  =11*    ID(SC)_D,                       ;  WRITE BACK PARTIAL PROD <MS>
U 1926, 0000,003C,0980,3400,0104,9615   ;39808          FE_SC+K[.2]                     ;  FE GETS .39
                                        ;39809  =;END
                                        ;39810          ;---------------------------;
U 1615, 0000,003C,C9F0,2C00,0000,1930   ;39811          Q_ID[T2]                        ;  Q GETS B1
                                        ;39812
                                        ;39813          ;00*---------------------------;
                                        ;39814  =00*    D_Q,Q_ID[T0],                   ;  D GETS B1, Q GETS A1
U 1930, 0C00,003D,C1F0,2C00,0000,1956   ;39815          CALL,J/MULTIPLY.HUGE.SUBR       ;
                                        ;39816
                                        ;39817          ;01*---------------------------;  RETURN WITH:  < Q,D> = A1*B1
                                        ;39818  =01*    SC_K[.2],                       ;  OFFSET OF 30.
U 1932, 0000,003D,0980,F800,0084,7AC0   ;39819          CALL,J/MULH.ADD
                                        ;39820
                                        ;39821          ;11*---------------------------;
                                        ;39822  =11*    FE_SC+K[.2],                    ;  SC GETS .39
                                        ;39823          ID(SC)_D,                       ;  WRITE PARTIAL PROD <MS>
                                        ;39824          D_0,                            ;  NEEDED FOR INITIALIZATION OT T6
U 1936, 0F10,0038,09C0,3518,0104,99A8   ;39825          Q_RC[T3]                        ;  Q GETS B2
                                        ;39826  =;END
```

```
                                ;39827          ;00*-------------------------------;
                                ;39828  =00*     ID[T6]_D,                       ;  CLEAR OUT PARTIAL PROD FRACO
                                ;39829          D_Q,                            ;  D GETS B2
                                ;39830          Q_RC[T0],                       ;  Q GETS A0
U 19A8, 0C10,0039,D9C0,3D00,0000,1956   ;39831          CALL,J/MULTIPLY.HUGE.SUBR
                                ;39832
                                ;39833          ;01*-------------------------------;  RETURN WITH:  <Q,D> = A0*B2
                                ;39834  =01*                                    ;                 FE = .39
                                ;39835          SC_K[.3],                       ;  OFFSET OF 29.
U 19AA, 0000,003D,0D80,F800,0084,7AC0   ;39836          CALL,J/MULH.ADD
                                ;39837
                                ;39838          ;11*-------------------------------;
                                ;39839  =11*     ID(SC)_D,                       ;  REWRITE PARTIAL PROD <MS>
                                ;39840          FE_SC+K[.1],                    ;  FE GETS .38
U 19AE, 0810,0038,0580,3510,0104,99C0   ;39841          D_RC[T2]                        ;  D GETS B0
                                ;39842  =;END
                                ;39843          ;00*-------------------------------;
                                ;39844  =00*     Q_ID[T0],                       ;  Q GETS A1
U 19C0, 0000,003D,C1F0,2C00,0000,1956   ;39845          CALL,J/MULTIPLY.HUGE.SUBR
                                ;39846
                                ;39847  =01*     ;01*-------------------------------;  RETURN WITH:  <Q,D> = A1*B0
                                ;39848                                          ;                 FE = .38
                                ;39849          SC_K[.1],                       ;  OFFSET OF 31.
U 19C2, 0000,003D,0580,F800,0084,7AC0   ;39850          CALL,J/MULH.ADD
                                ;39851
                                ;39852          ;11*-------------------------------;
                                ;39853  =11*     ID(SC)_D,                       ;  REWRITE LAST PARTIAL PRODUCT
                                ;39854          D_RC[T0],                       ;  D GETS A0
U 19C6, 0810,0038,0980,3500,0104,9616   ;39855          FE_SC+K[.2]                     ;  FE GETS .38
                                ;39856  =;END
                                ;39857          ;-------------------------------;
U 1616, 0000,003C,C9F0,2C00,0000,19E0   ;39858          Q_ID[T2]                        ;  Q GETS B1
                                ;39859
                                ;39860          ;00*-------------------------------;
                                ;39861  =00*     ALU_Q,RC[T4]_ALU.LEFT,SI/ZERO,  ;  RC4 GETS B1*2
U 19E0, 0021,203D,0180,F9A0,0000,1947   ;39862          CALL,J/MPY.HUGE.1
                                ;39863
                                ;39864          ;01*-------------------------------;  RETURN WITH:<Q,D> = A0*B1,FE = .38
                                ;39865          SC_K[.2],                       ;  OFFSET OF 30.
U 19E2, 0000,003D,0980,F800,0084,7AC0   ;39866          CALL,J/MULH.ADD
                                ;39867
                                ;39868          ;11*-------------------------------;
                                ;39869  =11*     ID(SC)_D,                       ;  REWRITE LAST PARTIAL PROD
                                ;39870          Q_RC[T0],                       ;  Q GETS A0
                                ;39871          D_0,                            ;  NEED TO CLEAR ID5
U 19E6, 0F10,0038,05C0,3500,0104,99F0   ;39872          FE_SC+K[.1]                     ;  FE GETS .37
                                ;39873  =;END
```

```
                                ;39874          ;00*-------------------------------;
                                ;39875  =00*    ID[T5]_D,                       ;  CLEAR ID5
                                ;39876          D_RC[T2],                       ;  D GETS B0
U 19F0, 0810,0039,D580,3D10,0000,1956   ;39877          CALL,J/MULTIPLY.HUGE.SUBR
                                ;39878
                                ;39879          ;01*-------------------------------;  RETURN WITH <Q,D> = A0*B0
                                ;39880  =01*    SC_K[.1],                       ;  OFFSET OF 31.
U 19F2, 0000,003D,0580,F800,0084,7AC0   ;39881          CALL,J/MULH.ADD
                                ;39882
                                ;39883          ;11*-------------------------------;  RETURN ID6-ID9 = PRODUCT-FRACTION
                                ;39884  =11*                                    ;       Q = ID6
                                ;39885          D_Q,                            ;  D GETS FRAC 0
U 19F6, 0C00,003C,DDF0,2C00,0000,1618   ;39886          Q_ID[T7]                        ;  Q GETS FRAC 1
                                ;39887  =;END
                                ;39888          ;-------------------------------;
                                ;39889          FE_SHF.VAL,                     ;  FE GETS SHIFT-VALUE
                                ;39890          SC_SHF.VAL,                     ;  SC GETS SHIFT-VALUE AS WELL
U 1618, 0000,173C,0180,F800,018C,790C   ;39891          STATE3-0?                       ;  TEST FOR POLY OR EMOD
                                ;39892
                                ;39893          ;-------------------------------;
                                ;39894  =01100  ;  BRANCH ON STATE<1:0>
                                ;39895          ;01100----------------------------;
                                ;39896          SC_SC+1,                        ;  MULH, SHIFT OUT HIDDEN BIT
U 190C, 0000,003D,0180,F800,0080,D90F   ;39897          CALL,J/SHIFT.MULH.ID            ;  SHIFT ID5-8 INTO RC0-ID1
                                ;39898
                                ;39899          ;01101----------------------------;
                                ;39900          SC_SC-K[.1],                    ;  POLYH,  LEAVE OVERFLOW-BIT
U 190D, 0000,003C,0580,F800,0084,B90F   ;39901          J/SHIFT.MULH.ID
                                ;39902
                                ;39903  =01111  ;01111----------------------------;
                                ;39904  SHIFT.MULH.ID:
                                ;39905          D_DAL.SC,                       ;  D GETS FRAC <VMS>
U 190F, 0D00,003C,0180,F800,0000,161A   ;39906          J7SHIFT.MULH.ID.1               ;  EMODH, JUST NORMALIZE
                                ;39907
                                ;39908  =11110  ;11110----------------------------;
                                ;39909          ;  D = LEAST FRACTION
                                ;39910          ;  FE = LEADING 0S IN FRACTION (BEFORE THE RECENT NORMALIZING)
                                ;39911          ;  Q = BIASSED EXPONENT
                                ;39912          ;  RC6 = BIASSED EXP
                                ;39913          K[.8000],                       ;  ROUNDING CONSTANT
                                ;39914          Q_RC[T1],                       ;  NEED FRAC <ALS>
U 191E, 0010,0038,45C0,F908,0000,15BB   ;39915          J7ADDH.380                      ;  GO TO ROUND AND PACK RESULT
                                ;39916  =;END   ;-------------------------------;
```

```
                                ;39917  ;  ROUTINE TO NORMALIZE FRACTION AFTER MULTIPLYING.
                                ;39918  ;  SC HAS NUMBER OF POSITIONS TO SHIFT LEFT.
                                ;39919  ;  FOR MULH, THE SHIFT GOES ONE BIT BEYOND THE HIDDEN BIT.
                                ;39920  ;  FOR POLYH, THE SHIFT LEAVES AN OVERFLOW BIT FOR THE ADD.
                                ;39921  ;  FOR EMODH, THE NUBER IS NORMALIZED, FOR THE CONVERSION TO INTEGER.
                                ;39922
                                ;39923  ;  RETURN12
                                ;39924
                                ;39925  SHIFT.MULH.ID.1:
                                ;39926          ;-------------------------------;
                                ;39927          RC[T0]_D,                       ;  STORE IT IN RC0
U 161A, 0C01,003C,E1F0,2D80,0000,161C   ;39928          D_Q,Q_ID[T8]                    ;
                                ;39929
                                ;39930          ;-------------------------------;
                                ;39931          D_DAL.SC,                       ;
U 161C, 0D01,203C,E5F0,2D88,0000,161D   ;39932          RC[T1]_Q,Q_ID[T9]               ;
                                ;39933
                                ;39934          ;-------------------------------;
                                ;39935          ID[T0]_D,                       ;
U 161D, 0810,0038,C180,3D08,0000,161E   ;39936          D_RC[T1]                        ;
                                ;39937
                                ;39938          ;-------------------------------;
U 161E, 0D00,003C,0180,F800,0000,1620   ;39939          D_DAL.SC                        ;
                                ;39940
                                ;39941          ;-------------------------------;
                                ;39942          RC[T1]_D,                       ;
U 1620, 0C01,003C,01F8,F988,0000,1621   ;39943          D_Q,Q_0                         ;
                                ;39944
                                ;39945          ;-------------------------------;
                                ;39946          Q_RC[T6],                       ;  Q GETS EXPONENT+4000
                                ;39947          D_DAL.SC,                       ;
U 1621, 0D10,0038,B1C0,F930,0000,1622   ;39948          K[.4000]                        ;  EXPONENT IS DOUBLY BIASSED
                                ;39949
                                ;39950          ;-------------------------------;
                                ;39951          ALU_Q-K[.4000],RC[T6]_ALU,      ;  RC6 GETS BIASSED EXP
                                ;39952          SC_FE,                          ;  SC GETS # OF LEADING 0'S
U 1622, 0019,2000,B1C0,F9B0,0081,1624   ;39953          Q_ALU                           ;  Q GETS BIASSED EXP AS WELL
                                ;39954
                                ;39955          ;-------------------------------;
                                ;39956          ID[T1]_D,                       ;
                                ;39957          ALU_0+Q+1,                      ;  ADJUST EXP FOR ONE LEADING 0
U 1624, 001F,0010,C5C0,3C00,0000,1625   ;39958          Q_ALU                           ;
                                ;39959
                                ;39960          ;-------------------------------;
                                ;39961          ALU_Q-K[SC],RC[T6]_ALU,         ;  RC6 GETS ADJUSTED EXP (PRE ROUNDING)
                                ;39962          Q_ALU,                          ;  LEAVE IT HERE FOR EMODH-ADJUSTMENT
U 1625, 0019,2002,1DC0,F9B0,0000,0012   ;39963          RETURN12                        ;  RETURN TO POLY,EMOD OR MULH ROUTINES
                                ;39964          ;-------------------------------;
```

```
                                ;39965  =110
                                ;39966  MULTIPLY.HUGE.SUBR:
                                ;39967  ; EXPECTS:
                                ;39968  ;               D = 31 BITS OF MULTIPLICAND = B-OPERAND
                                ;39969  ;               Q = 32 BITS OF MULTIPLIER = A-OPERAND
                                ;39970
                                ;39971          ;  BRANCH ON PENDING INTERRUPTS
                                ;39972          ;110----------------------------;
                                ;39973          ALU_D,RC[T4]_ALU.LEFT,          ;  RC4 GETS M'CAND*2
                                ;39974          SI/ZERO,                        ;
                                ;39975          D_Q,Q_D,                        ;  SWAP M'CAND WITH M'PLIER
                                ;39976          D.NE.0?,                        ;  TEST FOR MULTIPLICAND = 0?
U 1956, 0C21,0D3C,01E0,F9A0,0000,1945   ;39977          J/MULH.SUBR.0
                                ;39978
                                ;39979  H.FORMAT.INTRPT:
                                ;39980          ;111----------------------------;
U 1957, 0000,003C,0180,F800,0000,04FA   ;39981          J/INT.I
                                ;39982  =;END
                                ;39983          ;----------------------------;
                                ;39984  =101    ;  BRANCH ON D.NE.0
                                ;39985          ;101----------------------------;
                                ;39986  MULH.SUBR.0:
U 1945, 0F00,003E,01F8,F800,0000,0002   ;39987          D_0,Q_0,RETURN2                 ;  PRODUCT IS 0
                                ;39988
                                ;39989          ;111----------------------------;
                                ;39990  MPY.HUGE.1:
                                ;39991  ;  EXPECTS:
                                ;39992  ;               D = A-OPERAND
                                ;39993  ;               Q = 31-BITS B-OPERAND
                                ;39994  ;               RC4 = B-OPERAND*2
                                ;39995          ALU_Q,R[R15]_ALU,               ;  R15 GETS M'CAND
                                ;39996          CLK.UBCC,                       ;  CLOCK ALU Z-BIT
                                ;39997          Q_0,                            ;  CLEAR PRODUCT
U 1947, 0001,2D3C,01F8,FAF8,0010,1965   ;39998          D.NE.0?                         ;  IS MULTIPLIER = 0?
                                ;39999
                                ;40000          ;----------------------------;
                                ;40001  =101    ;BRANCH ON D NE 0
                                ;40002          ;101----------------------------;
U 1965, 0F00,003E,01F8,F800,0000,0002   ;40003          D_0,Q_0,RETURN2                 ;  PRODUCT IS 0
                                ;40004
                                ;40005          ;----------------------------;
                                ;40006          LC_RC[T4],                      ;  LATCH M'CAND*2
U 1967, 0000,013C,0180,F920,0000,10E8   ;40007          Z?,J/MULDMPY                    ;  BRANCH ON M'CAND =0
                                ;40008  =;END   ;----------------------------;
```

```
                                ;40009  HUGE.MUL.ADD.SHF:
                                ;40010          ;ROUTINE TO ADD NEW PRODUCT TO RUNNING PARTIAL PRODUCT.
                                ;40011          ;AND SHIFT AND STORE RESULT.
                                ;40012  ; INPUTS:
                                ;40013  ;               <Q,D> = NEW PRODUCT
                                ;40014  ;               SC = SHIFT COUNT (BIT 9 NEED NOT HAVE BEEN SET)
                                ;40015  ;               <T8,T9> = PARTIAL PRODUCT
                                ;40016
                                ;40017          ;-------------------------------;
                                ;40018          RC[T5]_Q,                       ;  SAVE HIGH PART TEMPORARILY
U 1626, 0001,203C,E5F0,2DA8,0000,1628   ;40019          Q_ID[T9]                        ;  GET LOW PART OF PARTIAL PRODUCT
                                ;40020
                                ;40021          ;-------------------------------;
                                ;40022          ALU_D+Q,CLK.UBCC,               ;  ADD LOW PARTS, CLOCK CARRY
                                ;40023          D_ALU,                          ;  D GETS RESULT
                                ;40024          LC_RC[T5],                      ;  LATCH HIGH PART
U 1628, 081D,0014,E1F0,2D28,0010,1629   ;40025          Q_ID[T8]                        ;  Q GETS PARTIAL PROD <MS>
                                ;40026
                                ;40027          ;-------------------------------;
                                ;40028          SC_SC.OR.K[.FF00],              ;  SET BIT 9 FOR RIGHT SHIFT
U 1629, 0000,033C,4D80,F800,0084,3AB9   ;40029          C3T?                            ;  WAS THERE A CARRY FROM <LS> ADD
                                ;40030
                                ;40031  =0*     ;0*-----------------------------;
                                ;40032          ALU_Q+LC,Q_ALU,
                                ;40033          CLK.UBCC,                       ;  CLOCK CARRY
U 1AB9, 0011,2014,01C0,F800,0010,162A   ;40034          J/HUGE.ASH.1
                                ;40035
                                ;40036          ;1*-----------------------------;
                                ;40037          ALU_Q+LC+1,Q_ALU,
                                ;40038          CLK.UBCC,                       ;  CLOCK CARRY
U 1ABB, 0011,2010,01C0,F800,0010,162A   ;40039          J/HUGE.ASH.1
                                ;40040
                                ;40041  HUGE.ASH.1:
                                ;40042          ;-------------------------------;
                                ;40043          D_DAL.SC,                       ;  D GETS NEW PARTIAL PRODUCT
U 162A, 0D00,033C,0180,F800,0000,1ABC   ;40044          C31?                            ;  ANOTHER CARRY ?
                                ;40045
                                ;40046  =0*     ;0*-----------------------------;
                                ;40047          ID[T9]_D,D_Q,Q_0,
U 1ABC, 0C00,003C,E5F8,3C00,0000,162C   ;40048          J/HUGE.ASH.2
                                ;40049
                                ;40050          ;1*-----------------------------;
                                ;40051          ID[T9]_D,D_Q,
U 1ABE, 0C23,003C,E7C0,3C00,0000,162C   ;40052          ALU_0(A),Q_ALU.LEFT,SI/MUL-     ;  Q GETS 1
                                ;40053
                                ;40054  HUGE.ASH.2:
                                ;40055          ;-------------------------------;
                                ;40056          D_DAL.SC,                       ;  SHIFT <MS> INTO PLACE
U 162C, 0D00,003C,E5F0,2C00,0000,162D   ;40057          Q_ID[T9]                        ;  NOTHING BETTER TO DO
                                ;40058
                                ;40059          ;-------------------------------;
U 162D, 0000,003E,E180,3C00,0000,0001   ;40060          ID[T8]_D,RETURN1                ;  STORE <LS> IN T8
```

```
                                ;40061  ; ROUTINE TO ADD NEWLY FORMED PRODUCT INTO PARTIAL PRODUCT.
                                ;40062  ; ALL THE BITS ARE CONSIDERED SIGNIFICANT
                                ;40063
                                ;40064  ; EXPECTS:
                                ;40065  ;       <Q,D> = NEWLY FORMED PRODUCT
                                ;40066  ;       FE = STARTING ID-REGISTER ADDRESS
                                ;40067  ;       SC = OFFSET INTO THAT REGISTER
                                ;40068
                                ;40069  ; OUTPUTS:
                                ;40070  ;       D = MOST SIGNIFICANT 32 BITS OF PARTIAL PRODUCT
                                ;40071  ;       Q = ID[T6]
                                ;40072  ;       SC = ID-ADDRESS OF MOST SIGNIF PARTIAL PRODUCT
                                ;40073  ;       ALU CARRY REFLECTS RESULT OF LAST ADD
                                ;40074  ;       AND WILL BE CLEARED IN ALL CASES EXCEPT THE LAST ONE
                                ;40075  ;       WHEN IT INDICATES THAT A RIGHT SHIFT IS REQUIRED
                                ;40076
                                ;40077  ; TEMPORARIES:
                                ;40078  ;       RC5 AND R15 ARE USED FOR STORAGE
                                ;40079
                                ;40080  =0**    ;0**-----------------------------;
                                ;40081  MULH.ADD:
                                ;40082          SC_SC.OR.K[.FF00],              ;  SET SIGN BIT FOR RIGHT SHIFT
                                ;40083          ALU_0+Q,CLK.UBCC,               ;  CLEAR ALU CARRY-BIT
                                ;40084          RC[T5]_ALU,                     ;  SAVE NEW PROD <MS> IN RC5
                                ;40085          Q_D,D_0,                        ;  SHIFT IN 0'S ON RIGHT
U 1AC0, 0F1F,0015,4DE0,F9A8,0094,3630   ;40086          CALL,J/MULH.ADD.SUBR
                                ;40087
                                ;40088          ;1**----------------------------;
                                ;40089          ID(SC)_D,                       ;  STORE NEW PARTIAL PRODUCT <LS>
                                ;40090          D_RC[T5],                       ;  D GETS NEW PROD <LS>
U 1AC4, 0800,003C,0580,3678,0084,BAC1   ;40091          SC_SC-K[.1]                     ;  DECREMENT ID-POINTER
                                ;40092  =;END
                                ;40093  =0**    ;0**-----------------------------;
                                ;40094          Q_RC[T5],                       ;  Q GETS NEW PROD <MS>
                                ;40095          SC_FE,FE_SC,                    ;
U 1AC1, 0010,0039,01C0,F928,0181,1630   ;40096          CALL,J/MULH.ADD.SUBR            ;
                                ;40097
                                ;40098          ;1**----------------------------;
                                ;40099          ID(SC)_D,                       ;  STORE NEW PARTIAL PROD
                                ;40100          D_RC[T5],                       ;  D GETS NEW PROD <MS>
                                ;40101          Q_0,                            ;  SHIFT IN 0'S
                                ;40102          SC_FE,FE_SC-K[.1],
U 1AC5, 0810,0038,05F8,3528,0185,B630   ;40103          J/MULH.ADD.SUBR                 ;  NOT A CALL THIS TIME
                                ;40104  =;END   ;-----------------------------;
                                ;40105
```

```
                                ;40106  MULH.ADD.SUBR:
                                ;40107  ; SUBROUTINE USED IN THE MULH.ADD ROUTINE
                                ;40108
                                ;40109          ;-------------------------------;
                                ;40110          R[R15]_Q,                       ;  SAVE NEW PRODUCT <LS> IN R15
                                ;40111          D_DAL.SC,                       ;  SHIFT THE NEW PRODUCT
U 1630, 0D01,203C,0180,FAF8,0181,1631   ;40112          SC_FE,FE_SC                     ;  SC GETS ID-ADDRESS, FE GETS SHIFT
                                ;40113
                                ;40114          ;-------------------------------;
U 1631, 0000,033C,01F0,2400,0000,1AC8   ;40115          Q_ID(SC),C31?                   ;  Q GETS PARTIAL PRODUCT
                                ;40116
                                ;40117          ;-------------------------------;
                                ;40118  =0*     ;BRANCH ON ALU CARRY
                                ;40119          ;0*-----------------------------;
                                ;40120          ALU_D+Q,D_ALU,CLK.UBCC,         ;  ADD IN NEW PRODUCT
U 1AC8, 081D,0016,D9F0,2C00,0010,0004   ;40121          Q_ID[T6],RETURN4
                                ;40122
                                ;40123          ;1*-----------------------------;
                                ;40124          ALU_D+Q+1,CLK.UBCC,D_ALU,       ;  ADD WITH CARRY
U 1ACA, 081D,0012,D9F0,2C00,0010,0004   ;40125          Q_ID[T6],RETURN4
                                ;40126  =;      ;-------------------------------;
                                ;40127
                                ;40128  ; ROUTINE TO CLEAR OUT DIRTY 0 H-FORMAT NUMBERS
                                ;40129
                                ;40130  ;       EXPECTS:
                                ;40131  ;               SC POINTS TO BASE REGISTER IN RC (TYPICALLY =.30)
                                ;40132  ;               D = 0, Q = 0
                                ;40133  ;
                                ;40134  ;       RETURNS:
                                ;40135  ;               RC(SC),ID(SC),RC(SC+1),ID(SC+1) = 0
                                ;40136  ;               SC = SC +1
                                ;40137  ;               RETURN1
                                ;40138
                                ;40139          ;-------------------------------;
                                ;40140  CLEAR.DIRTY.0:
                                ;40141          ID(SC)_D,                       ;  CLEAR OUT FRACTION 1
                                ;40142          RC(SC)_Q,                       ;  CLEAR OUT LONG EXP
U 1632, 0001,203C,0180,3438,0080,D634   ;40143          SC_SC+1
                                ;40144
                                ;40145          ;-------------------------------;
                                ;40146          ID(SC)_D,                       ;  CLEAR OUT FRACTION 3
                                ;40147          RC(SC)_Q,                       ;  CLEAR OUT FRACTION 2
U 1634, 0001,203E,0180,3438,0000,001C   ;40148          RETURN10
                                ;40149  144D:
                                ;40150  CLRH:   ;-------------------------------;
                                ;40151          ;  ENTER HERE FOR CLR0 AND CLRH
                                ;40152          D_0,Q_0,ALU_0(A),               ;  CLEAR EVERYTHING
                                ;40153          SET.CC(LONG),
U 144D, 0F03,003C,01F8,F800,0070,190E   ;40154          J/WRITE.DEST.0
                                ;40155          ;-------------------------------;
```

```
                                ;40156  ;       ROUTINE TO MOVE H-FORMAT OPERAND FROM RC2-ID3 TO RC0-ID1
                                ;40157  ;
                                ;40158  ;       EXPECTS:
                                ;40159  ;               D = RC2
                                ;40160  ;               Q = ID2
                                ;40161
                                ;40162  ;       RETURNS:
                                ;40163  ;               RC0-ID1 = RC2-ID3
                                ;40164  ;               RETURN10
                                ;40165
                                ;40166  SHIFT.H:
                                ;40167          RC[T0]_D,                       ;  RC[T0] GETS LONG EXP
U 1636, 0C01,003C,CDF0,2D80,0000,1638   ;40168          D_Q,Q_ID[T3]                    ;  D GETS LONG FRAC1
                                ;40169
                                ;40170          ;-------------------------------;
                                ;40171          ID[T0]_D,                       ;  T0 GETS LONG FRAC1
                                ;40172          D_Q,                            ;  D GETS LONG FRAC3
U 1638, 0C00,003C,C180,3D18,0000,1639   ;40173          LC_RC[T3]                       ;  LATCH LONG FRAC2
                                ;40174
                                ;40175          ;-------------------------------;
                                ;40176          ID[T1]_D,                       ;  STORE LONG FRAC3 IN T1
                                ;40177          RC[T1]_LC,                      ;  STORE LONG EXP IN RC0
U 1639, 0010,003A,C580,3D88,0000,0010   ;40178          RETURN10
                                ;40179          ;-------------------------------;
```

```
                                ;40180          ;ROUTINE TO UNPACK SINGLE H-FORMAT OPERAND.
                                ;40181
                                ;40182          ;INPUTS:
                                ;40183          ;       RC(SC),ID(SC),RC(SC+1),ID(SC+1) CONTAIN OPERAND
                                ;40184          ;       FE = SC POINTS TO OPERAND
                                ;40185          ;       STATE<7> DETERMINES WHETHER TO NORMALIZE OR NOT
                                ;40186
                                ;40187          ;OUTPUTS:
                                ;40188          ;       RC(SC),ID(SC),RC(SC+1),ID(SC+1) CONTAIN FRACTION
                                ;40189          ;       RC(SC+4) CONTAIN EXPONENT
                                ;40190          ;       R[R15] CONTAINS UNBIASSED EXPONENT
                                ;40191          ;       SC CONTAINS LOW 10 BITS OF UNBIASSED EXP
                                ;40192
                                ;40193          ;TEMPORARIES:
                                ;40194
                                ;40195          ;RETURNS:
                                ;40196          ;       3 IF NON-ZERO
                                ;40197          ;       1 IF 0
                                ;40198
                                ;40199          ;-------------------------------;
                                ;40200  UNPACK1H:
                                ;40201          Q_RC(SC),                       ;  Q GETS LONG EXP
                                ;40202          SS_SS.XOR.ALU15&SD_ALU15,       ;  STORE SIGN IN SS AND SD
U 163A, 0010,0038,11C5,F830,0084,963B   ;40203          SC_SC+K[.4]                     ;  POINT TO EXPONENT ADDRESS
                                ;40204
                                ;40205          ;-------------------------------;
                                ;40206          ALU_Q.OXT[WORD].ANDNOT.K[.8000], ; CLEAR SIGNBIT
                                ;40207          RC(SC)_ALU,                     ;  STORE EXPONENT IN RC(SC+4)
                                ;40208          CLK.UBCC,WORD,                  ;  CLOCK ALU Z-BIT ON EXPONENT
                                ;40209          D_Q,                            ;  D GETS OLEXP
U 163B, 0C1B,6024,4580,F838,0091,192E   ;40210          SC_FE                           ;  SC GETS BASE ADDRESS BACK
                                ;40211
                                ;40212  =01110  ;01110-------------------------;
                                ;40213          ALU_Q.ANDNOT.K[.FFFF],D_ALU,    ;  D GETS HIGH FRACTION WORD
                                ;40214          FE_K[.FFFF],                    ;  FE GETS -1
                                ;40215          Q_ID(SC),                       ;  Q GETS LONG FRAC1
                                ;40216          Z?,                             ;  IS OPERAND 0 ?
U 192E, 0819,2125,C1F0,2400,0104,7A6C   ;40217          CALL,J/UNP.SUBR.3
                                ;40218
                                ;40219  =11110  ;11110-------------------------;
                                ;40220          D_RC[T4],                       ;  NEEDED FOR POLYH
U 193E, 0810,003A,0180,F920,0000,0001   ;40221          RETURN1
                                ;40222
                                ;40223          ;11111-------------------------;
                                ;40224  =11111
                                ;40225          ;RETURN WITH:
                                ;40226          ; RC(SC),ID(SC), RC(SC+1) ARE LOADED WITH 3 MS FRACTION LONGWORDS
                                ;40227          ; D = VLS FRACTION
                                ;40228          ; SC = SC+1 (ORIGINAL SC)
                                ;40229          ; RC(SC+4) = BIASSED EXPONENT
                                ;40230
                                ;40231          ID(SC)_D,                       ;  WRITE LAST FRACTION WORD
U 193F, 0000,003C,0D80,3400,0084,963C   ;40232          SC_SC+K[.3]                     ;  POINT TO EXPONENT
                                ;40233  =;END
```



```
                                ;40234          ;-------------------------------;
```

```
                        ;40235          D_RC(SC),                       ;  D GETS BIASSED EXPONENT
U 163C, 0810,0038,B180,F830,0000,163E   ;40236          K[.4000]                        ;  SET UP SLOW CONSTANT
                        ;40237
                        ;40238          ;-------------------------------;
                        ;40239          ALU_D-K[.4000],R[R15]_ALU,      ;  R15 GETS UNBIASSED EXPONENT
                        ;40240          SC_ALU,                         ;  SC GETS 10 LOW ORDER BITS
                        ;40241          CLR.UBCC,                       ;  SET ALU CC ON EXPONENT
U 163E, 0019,0002,B180,FAF8,0092,0003   ;40242          RETURN3
                        ;40243          ;-------------------------------;
```

```
                                ;40244  ;  ROUTINE TO CONVERT H-FORMAT FLOATING FORMATS TO INTEGERS.
                                ;40245
                                ;40246          ;-------------------------------;
                                ;40247  1443:
                                ;40248  CVTHB:   ; EXPECTS SRC IN RC0-ID1
                                ;40249          SGN/CLR.SD+SS,                  ;  CLEAR SIGN-FLIPFLOPS
                                ;40250          SC_K[.30],FE_K[.30],            ;  LOAD BASE OPERAND POINTER
                                ;40251          D_RC[TO],                       ;  D GETS LONG EXP
                                ;40252          N&Z_ALU.V&C_0,                  ;  CLEAR C AND V-BITS OF PSL
U 1443, 0810,0039,7987,F900,01D4,7A38   ;40253          CAL[,J/CVTHI
                                ;40254
                                ;40255          ;-------------------------------;
                                ;40256  1453:
                                ;40257          ALU_D,N&Z_ALU,                  ;  CLOCK PSL N & Z
                                ;40258          DT/INST.DEP,                    ;  WORKS FOR BYTE,WORD,LONG
U 1453, 0001,DA3C,0180,F800,0060,15AC   ;40259          PSL.V?                          ;  CHECK FOR INT OVERFLOW
                                ;40260
                                ;40261  =1100   ;1100-------------------------;
U 15AC, F000,003F,01F0,F847,0000,0300   ;40262          WRITE.DEST                      ;  V-BIT NOT SET,NORMAL
                                ;40263
                                ;40264  =1110   ;1110-------------------------;
U 15AE, 000C,003D,31F0,2C00,0000,0DFC   ;40265          Q_ID[CES],CALL[INOVFL]          ;  V-BIT SET,OVERFLOW
                                ;40266
                                ;40267  =1111   ;1111-------------------------;
U 15AF, F000,003F,01F0,F847,0000,0300   ;40268          WRITE.DEST                      ;
                                ;40269          ;-------------------------------;
                                ;40270
                                ;40271
                                ;40272  14C5:   ;-------------------------------;
                                ;40273  CVTRHL: ;  CONVERT H TO LONGWORD AND ROUND
                                ;40274          ;  EXPECTS RC0-ID1 = SRC
                                ;40275          SGN/CLR.SD+SS,                  ;  CLEAR SIGN-FLIPFLOPS
                                ;40276          SC_K[.30],FE_K[.30],            ;  POINTERS TO SRC-OPERAND
                                ;40277          D_RC[TO],                       ;  D GETS LONG EXP
                                ;40278          N&Z_ALU.V&C_0,                  ;  USED ONLY TO CLEAR V BIT
U 14C5, 0810,0039,7987,F900,01D4,7A38   ;40279          CAL[,J/CVTHI                    ;  CALL CONVERSION ROUTINE
                                ;40280
                                ;40281          ;-------------------------------;
                                ;40282  14D5:   ;  RETURN WITH: D = INTEGER
                                ;40283          ;    Q<31>=ROUNDING BIT
                                ;40284          ;    SS = SIGN OF SRC-OPERAND
                                ;40285
                                ;40286          ALU_0(A),Q_ALU.LEFT,SI/DIV,     ;  SHIFT Q<31> INTO Q<0>
U 14D5, 0023,123C,02C0,F800,0000,041E   ;40287          SS?,J/CVTRDL.0                  ;  TEST SIGN-BIT, AND JOIN D-FORMAT
                                ;40288          ;-------------------------------;
```

```
                                ;40289          ; SUBROUTINE TO CONVERT FROM H-FORMAT TO INTEGER
                                ;40290
                                ;40291          ;EXPECTS:
                                ;40292          ; RC0-ID1 = SRC H-FORMAT NUMBER
                                ;40293          ; EXPECTS SC TO POINT TO RC[T0], I.E. SC =.30
                                ;40294          ; FE MUST BE .30 AS WELL
                                ;40295          ;RETURNS:
                                ;40296          ; D = INTEGER
                                ;40297          ; Q<31> = ROUND BIT
                                ;40298          ; RC0 - ID1 = UNPACKED FRACTION PART
                                ;40299          ; R15 = EXPONENT OF FRACTION (<=0)
                                ;40300          ; SC = LOW 5 BITS OF EXPONENT
                                ;40301          ; FE = # OF LONGWORDS OF FRACTION TO BE SHIFTED LEFT
                                ;40302
                                ;40303          ; RETURN10 IF ALL DATA IS CORRECT
                                ;40304          ; RETURN11 IF FRACTION MAY NEED CLEARING
                                ;40305  =00
                                ;40306  CVTHI:  ;00------------------------------;
                                ;40307          STATE_K[.80],                   ;  SET NORMALIZE-BIT OF STATE
U 1A38, 0000,003D,4180,F800,1404,763A   ;40308          CALL,J/UNPACK1H                 ;  UNPACK SRC IN RC0-ID1
                                ;40309
                                ;40310          ;01------------------------------;
                                ;40311  =01     D_0,Q_0,                        ;  EVERYTHING IS 0
U 1A39, 0F00,003E,01F8,F800,0000,0011   ;40312          RETURN[11]
                                ;40313
                                ;40314          ;11------------------------------;
                                ;40315  =11     ;RETURN WITH:
                                ;40316          ; RC0-ID1 = UNPACKED FRACTION (NORMALIZED)
                                ;40317          ; SS=SD= SIGN
                                ;40318          ; R15 = UNBIASSED EXPONENT
                                ;40319          ; SC = LOW 10 BITS OF R15
                                ;40320          ; ALU CC REFLECT R15
                                ;40321
                                ;40322  CVTHI.EMODH:    ;ENTER HERE FOR EMODH
                                ;40323          LAB_R[R15],                     ;  LATCH UNBIASSED EXPONENT
                                ;40324          ALU_LA.ANDNOT.K[.FF],           ;  TEST HIGH ORDER BITS
                                ;40325          CLK.UBCC,WORD,                  ;  CLOCK ALU CC ON HIGH BITS
U 1A3B, 0018,5B24,4980,FA78,0010,1623   ;40326          ALU?                            ;  TEST UNBIASSED EXP
                                ;40327
                                ;40328          ;------------------------------;
                                ;40329  =0011   ;BRANCH ON ALU N&Z BITS
                                ;40330          ;0011----------------------------;  POSITIVE EXPONENT, INTEGER PORTION
                                ;40331          ALU_LA-K[.A0]-1,CLK.UBCC,       ;
U 1623, 0018,0008,2580,F800,0010,1640   ;40332          J/CVTHI.20                      ;  TEST FOR EXP >=160
                                ;40333
                                ;40334          ;0111----------------------------;
                                ;40335          FE_K[ZERO],                     ;  NO LONGWORDS TO ALIGN
                                ;40336          Q_RC[T0],                       ;  Q GETS FRAC <VMS> FOR ROUNDING
                                ;40337          D_0,                            ;  INTEGER = 0
U 1627, 0F10,003A,19C0,F900,0104,6010   ;40338          RETURN10
                                ;40339
                                ;40340          ;1011----------------------------;
                                ;40341          FE_K[ZERO],                     ;  NO LONGWORDS TO ALIGN
U 162B, 0F00,003E,19F8,F800,0104,6010   ;40342          Q_0,D_0,RETURN10                ;  NEGATIVE EXPONENT
                                ;40343  =;END
```

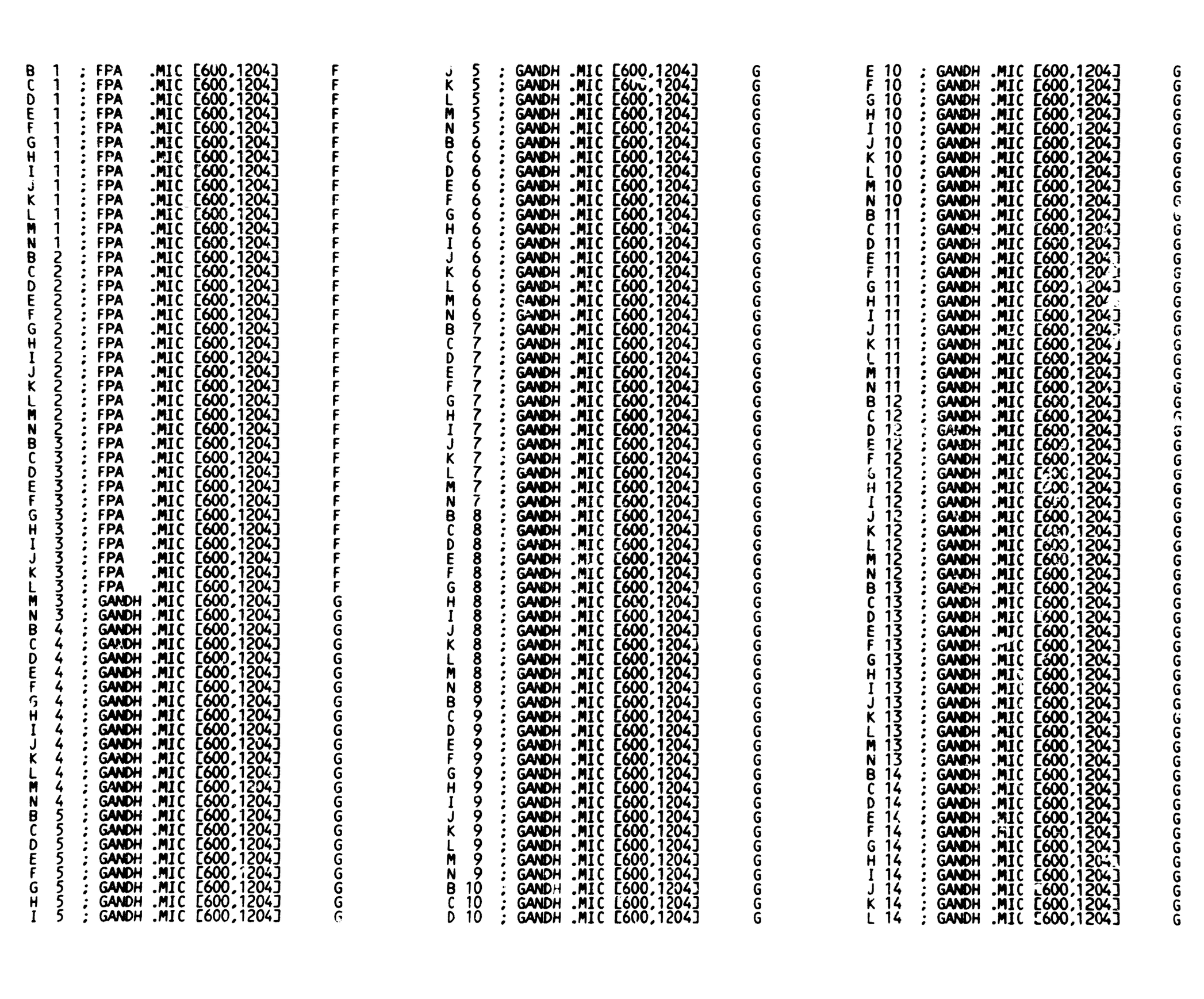
```
                    ;40344          ;--------------------------------;
                    ;40345  CVTHI.20:
                    ;40346          FE_K[.20],
                    ;40347          D_RC[T0],                       ;  D GETS FRAC<VMS>
U 1640, 0810,1B38,75F8,F900,0104,7617 ;40348          Q_0,ALU?                        ;  SHIFT IN 0'S
                    ;40349
                    ;40350  =0111   ;0111----------------------------;
                    ;40351          D_0,Q_0,SET.V,                  ;  NUMBER GT 2**160
U 1617, 0F03,003E,01F8,F980,0020,0011 ;40352          RC[T0]_0,RETURN[11]             ;  REMEMBER TO CLEAR FRACTION
                    ;40353
                    ;40354          ;1111---------------------------;
                    ;40355          SC_SC-K[.20],                   ;  ALL OF EXP IS IN SC
U 161F, 0000,143C,7580,F800,0084,B976 ;40356          SC?                             ;  TEST SC<9:5>
                    ;40357  =;END
                    ;40358          ;--------------------------------;
                    ;40359  =110    ;BRANCH ON SC<9:5> NE 0
                    ;40360          ;110----------------------------;  NUMBER < 2**31
                    ;40361          FE_K[ZERO],                     ;  FE GETS # OF LONG SHIFTS
                    ;40362          D_DAL.SC,                       ;  D GETS INTEGER, SHIFTING RIGHT
                    ;40363          Q_D,                            ;  Q GETS ROUND BIT
U 1976, 0D00,123C,19E0,F800,0104,7B6E ;40364          SS?, J/CVTGI.4                  ;  JOIN G-CODE
                    ;40365
                    ;40366          ;111----------------------------;  2**31<=NUMBER
                    ;40367          Q_ID[T0],                       ;  Q GETS FRACTION <AMS>
U 1977, 0000,143C,C1F0,2C00,0000,1986 ;40368          SC?                             ;  TEST FOR SC NEGATIVE
                    ;40369  =;END
                    ;40370          ;--------------------------------;
                    ;40371  =110    ;BRANCH ON SC<9:5> NE 0
                    ;40372          ;110----------------------------;  2**31 <= NUMBER < 2**63
                    ;40373          EALU_SC,                        ;  GET EXPONENT-32. FOR OVERFLOW CHK
                    ;40374          Q_D.OXT[BYTE].OR.PACK.FP,       ;  GENERATE 8000 IF
                    ;40375                                          ;  INTEGER IS 80000000 AND SD = 1
                    ;40376          D_DAL.SC,                       ;  D GETS INTEGER STILL
U 1986, 0D0B,8030,01C0,F800,0000,1645 ;40377          J/CVTHI.3
                    ;40378
                    ;40379          ;111----------------------------;  2**63 <= NUMBER
                    ;40380          SC_SC-FE,                       ;  SUBTRACT 32. FROM EXPONENT
U 1987, 0000,003C,0180,F800,00A0,B641 ;40381          SET.V                           ;  DEFINITELY OVERFLOW
                    ;40382  =;END
                    ;40383          ;--------------------------------;
                    ;40384  ;REMINDER:
                    ;40385          ; D = FRAC <VMS> = RC0
                    ;40386          ; Q = FRAC <AMS> = ID0
                    ;40387          ; SC = EXPONENT-64.
                    ;40388          ; FE = 32.
                    ;40389          D_RC[T1],                       ;  <Q,D> NOW EQUALS FRAC2,FRAC3
                    ;40390          SC_SC-FE,                       ;  SUBTRACT 32. MORE FROM EXP
U 1641, 0810,1438,0180,F908,0080,B996 ;40391          SC?
                    ;40392
```

```
                                ;40393          ;-------------------------------;
                                ;40394  =110    ;BRANCH ON SC<9:5> NE 0
                                ;40395          ;110----------------------------;  2**63 <= NUMBER < 2**95
                                ;40396          FE_K[.2],                       ;  NUMBER OF LONG WORDS
                                ;40397          D_DAL.SC,                       ;  D GETS INTEGER
                                ;40398          Q_D,                            ;  NEEDED FOR ROUND BIT
U 1996, 0D00,123C,09E0,F800,0104,7B6E   ;40399  SS?,J/CVTGI.4                    ;  TEST SIGN, JOIN G-CODE
                                ;40400
                                ;40401          ;111----------------------------;  2**95 <= NUMBER
                                ;40402          Q_ID[T1],                       ;  Q GETS FRAC4
                                ;40403          SC_SC-FE,                       ;  SUBTRACT 32. MORE FROM EXP
U 1997, 0000,143C,C5F0,2C00,0080,B9A6   ;40404  SC?                             ;  TEST IT
                                ;40405  =;END   ;-------------------------------;
                                ;40406  =110    ;BRANCH ON SC<9:5> NE 0
                                ;40407          ;-------------------------------;  2**95 <= NUMBER < 2**127
                                ;40408          D_Q,Q_D,                        ;  <Q,D> NOW EQUALS FRAC3,FRAC4
U 19A6, 0C00,003C,01E0,F800,0000,1644   ;40409  J7CVTHI.4
                                ;40410          ;-------------------------------;  2**127 <= NUMBER < 2**160
                                ;40411          SC_SC-FE,
U 19A7, 0F00,003C,0180,F800,0080,B642   ;40412  D_0
                                ;40413  =;END   ;-------------------------------;
                                ;40414          FE_K[.4],                       ;  NO MORE FRACTION BITS
                                ;40415          D_DAL.SC,                       ;  D GETS INTEGER
                                ;40416          Q_0,                            ;  DO THIS FOR ROUND
U 1642, 0D00,123C,11F8,F800,0104,7B6E   ;40417  SS?,J/CVTGI.4                    ;  TEST SIGN, JOIN G-CODE
                                ;40418          ;-------------------------------;
                                ;40419  CVTHI.4:
                                ;40420          FE_K[.3],                       ;  CLEAR 3 LOW LONG WORDS OF FRACTION
                                ;40421          D_DAL.SC,                       ;  D GETS INTEGER
                                ;40422          Q_D,                            ;  NEEDED TO ROUND
U 1644, 0D00,123C,0DE0,F800,0104,7B6E   ;40423  SS?,J/CVTGI.4                    ;  TEST SIGN, JOIN G-CODE
                                ;40424
                                ;40425          ;-------------------------------;
                                ;40426  CVTHI.3:
                                ;40427          ALU_Q.XOR.K[.8000],             ;  COMPARE WITH MOST NEG #
U 1645, 0019,2020,4580,F800,0010,1646   ;40428  CLK.UBCC,LONG
                                ;40429
                                ;40430          ;-------------------------------;
U 1646, 0000,003C,7580,F800,0084,B648   ;40431  SC_SC-K[.20]                     ;  LOWER EXP SO WE CAN GET ROUND BIT
                                ;40432
                                ;40433          ;-------------------------------;
                                ;40434          FE_K[.1],                       ;  NUMBER OF WHOLE LONGWORDS TO SHIFT
                                ;40435                                          ;  FRACTION
U 1648, 0000,013C,0580,F800,0104,7AD0   ;40436  Z?                              ;  INTEGER = MOST NEG NUMBER ?
                                ;40437
                                ;40438          ;0------------------------------;
                                ;40439  =0       Q_ID[TO],SET.V,                 ;  NO THERE IS OVERFLOW
U 1AD0, 0000,123C,C1F0,2C00,0020,1B6E   ;40440  SS?,J/CVTGI.4                    ;  JOIN G-CODE
                                ;40441
                                ;40442          ;1------------------------------;
                                ;40443          Q_ID[TO],                       ;  YES, NO OVERFLOW
U 1AD1, 0000,123C,C1F0,2C00,0000,1B6E   ;40444  SS?,J/CVTGI.4
                                ;40445  =;END   ;-------------------------------;
```

```
                                    ;40446  .TOC     "       G & H floating point  : H-FORMAT POLYNOMIAL EVALUATION"
                                    ;40447
                                    ;40448          ; THIS INSTRUCTION HAS TWO ENTRY POINTS, DEPENDING ON FPD.
                                    ;40449
                                    ;40450  14C3:   ;-------------------------------;
U 14C3, 0800,003C,0180,FA08,0000,1649 ;40451        D_R[R1]                         ;  THE NEXT SIX WORDS RELOAD THE PARTIAL
                                    ;40452
                                    ;40453          ;-------------------------------;  PRODUCT FROM R0-R3 TO RC0-ID1
U 1649, 0000,003C,C180,3C00,0000,164A ;40454        ID[T0]_D                        ;
                                    ;40455
                                    ;40456          ;-------------------------------;
U 164A, 0000,003C,0180,FA00,0000,164C ;40457        LAB_R[R0]
                                    ;40458
                                    ;40459          ;-------------------------------;
U 164C, 0000,003C,0180,F980,0000,164D ;40460        RC[T0]_LA                       ;
                                    ;40461
                                    ;40462          ;-------------------------------;
U 164D, 0000,003C,0180,FA10,0000,1650 ;40463        LAB_R[R2]                       ;
                                    ;40464
                                    ;40465          ;-------------------------------;
U 1650, 0000,003C,0180,F988,0000,1651 ;40466        RC[T1]_LA                       ;
                                    ;40467
                                    ;40468          ;-------------------------------;
                                    ;40469          D_R[R3],                        ;
U 1651, 0800,003C,7180,FA18,0084,7652 ;40470        SC_K[.FFF8]                     ;  GET READY TO SHIFT IT
                                    ;40471
                                    ;40472          ;-------------------------------;
U 1652, 0800,003C,C580,3E20,0000,1653 ;40473        ID[T1]_D,D_R[R4]                ;  SETUP TO GET PC DELTA
                                    ;40474
                                    ;40475          ;-------------------------------;
U 1653, 0D00,003C,0180,F800,0000,1654 ;40476        D_DAL.SC                        ;  PC DELTA NOW IN D<07:00>
                                    ;40477
                                    ;40478          ;-------------------------------;
U 1654, 0819,0000,0580,F800,0000,1A40 ;40479        D_D-K[.1]                       ;  ADJUST FOR THE ESCD
                                    ;40480
                                    ;40481          ;00-------------------------------;
                                    ;40482  =00     PC&VA_D.OXT[BYTE]+PC,           ;  BYPASS SPECIFIERS
U 1A40, 0017,8015,0180,F801,0200,0E16 ;40483        CALL,J/SETFPD
                                    ;40484
                                    ;40485          ;10-------------------------------;
U 1A42, 0014,0038,01CC,F800,0000,0EB8 ;40486  =10    Q_PC,J/BAKUP.PC                 ;  ENTER HERE ON READ ERRORS
                                    ;40487
                                    ;40488          ;11-------------------------------;
U 1A43, 0000,003C,0180,F800,0000,18C0 ;40489        J/POLYH.REEN                    ;  REENTER MAIN LOOP BY READING ARGUMENT
                                    ;40490  =;END
```

```
                                    ;40491          ;--------------------------------;
                                    ;40492          ;  SUBROUTINE TO READ 4 LONGWORDS IN MEMORY
                                    ;40493          ;STARTING AT VA, AND STORING THEM IN RC(SC),ID(SC),RC(SC+1),ID(SC+1)
                                    ;40494          ;  RETURNS SC = SC.OR.K[.30]+1
                                    ;40495          ;  VA = VA+K[.10]
                                    ;40496  =0**    ;0**-----------------------------;
                                    ;40497  READ.H.SUBR:
                                    ;40498          D[LONG]_CACHE,                  ;  READ FIRST LONGWORD
                                    ;40499          SC_SC.OR.K[.30],                ;  MAKE IT POINT TO TEMP AREA ON ID-BUS
U 1AC2, 0000,003D,7980,4000,0084,38F2 ;40500        CALL,J/READH.SUBR.SUB           ;  SUBROUTINE WITHIN THIS ROUTINE
                                    ;40501
                                    ;40502          ;1**----------------------------;
                                    ;40503          D[LONG]_CACHE,                  ;  READ 3. LONGWORD
                                    ;40504          SC_SC+1,
U 1AC6, 0000,003C,0180,4000,0080,D8F2 ;40505        J/READH.SUBR.SUB
                                    ;40506  =;END
                                    ;40507  =0***   ;0***----------------------------;
                                    ;40508  READH.SUBR.SUB:
                                    ;40509          RC(SC)_D,                       ;  STORE FIRST LONGWORD IN RC(SC)
                                    ;40510          VA_VA+4,                        ;  INCREMENT MEMORY POINTER
U 18F2, 0001,003D,0180,F83B,0000,1656 ;40511        CALL,J/READ.CACHE.LONG          ;  SINGLE LINE SUBROUTINE
                                    ;40512
                                    ;40513          ;1***---------------------------;
                                    ;40514          ID(SC)_D,                       ;  STORE 2. LONGWORD IN ID(SC)
                                    ;40515          VA_VA+4,                        ;  INCREMENT MEMORY POINTER
U 18FA, 0000,003E,0180,3403,0000,0004 ;40516        RETURN4
                                    ;40517  =;END
                                    ;40518          ;--------------------------------;
                                    ;40519  READ.CACHE.LONG:
                                    ;40520          D[LONG]_CACHE,
U 1656, 0000,003E,0180,4000,0000,0008 ;40521        RETURN[8]
                                    ;40522          ;--------------------------------;
```

```
                                ;40523  1685:
                                ;40524          ;  ENTER HERE FOR POLYH WITHOUT FPD SET.
                                ;40525          ;  EXPECTS : RC2-ID3 = PACKED ARGUMENT
                                ;40526          ;       D = DEGREE
                                ;40527
                                ;40528          ;-------------------------------;
                                ;40529          D_D.0XT[WORD],RC[T6]_ALU,       ;  STORE DEGREE IN RC6 TEMPORARILY
U 1685, 0803,403C,0580,F9B0,1404,7825   ;40530          STATE_K[.1]                     ;  INITIALIZE STATE TO POLY-FLAG
                                ;40531
                                ;40532  =01****1;01****1----------------------;
                                ;40533          ALU_D.ANDNOT.K[.1F],            ;  ISOLATE RESERVED BITS
                                ;40534          CLK.UBCC,WORD,
U 1825, 0019,4025,8D80,F800,0010,047E   ;40535          CALL,J/ASPC                     ;  EVALUATE TABLE ADDRESS
                                ;40536
                                ;40537  =11****1;11****1----------------------;
                                ;40538          RC[T5]_D,                       ;  SAVE TABLE ADDRESS IN RC5
                                ;40539          Q_ID[T2],                       ;  GET ARG LONG FRAC 1
U 1865, 0001,013C,C9F0,2DA8,0000,1AD2   ;40540          Z?                              ;  TEST FOR ILLEGAL DEGREE (>31.)
                                ;40541
                                ;40542          ;-------------------------------;
                                ;40543  =0       ;  BRANCH ON ALU Z-BIT
                                ;40544          ;0------------------------------;
U 1AD2, 0000,003C,0180,F800,0000,0106   ;40545          J/RSVOPR
                                ;40546
                                ;40547          ;1------------------------------;
                                ;40548          D_RC[T2],                       ;  D GETS ARG LONG EXP
U 1AD3, 0810,0038,0180,F910,0000,1658   ;40549          J/POLYH.I.1
                                ;40550
                                ;40551          ;-------------------------------;
                                ;40552          ;  ENTER HERE WITH ARGUMENT IN RC2-ID3
                                ;40553          ;  TABLE ADDRESS IN RC5
                                ;40554          ;  RC6 = DEGREE
                                ;40555          ;  EXPECT D = RC[T2]
                                ;40556          ;  EXPECTS Q = ID[T2]
                                ;40557  POLYH.I.1:
U 1658, 0000,003C,6580,FA70,0000,1659   ;40558          LAB_R[SP],K[.10]                ;  ALLOCATE STORAGE ON STACK FOR ARG
                                ;40559
                                ;40560          ;-------------------------------;
                                ;40561          ALU_LA-K[.10],
U 1659, 0018,0000,6580,F800,0200,165A   ;40562          VA_ALU
                                ;40563
                                ;40564          ;-------------------------------;
U 165A, 0000,003C,0180,3000,0000,165C   ;40565          CACHE_D[LONG]                   ;  WRITE LONG EXP ON STACK
                                ;40566
                                ;40567          ;-------------------------------;
                                ;40568          VA_VA+4,                        ;  INCREMENT STACK ADDRESS
                                ;40569          ALU_D.ANDNOT.K[.8000],          ;  ISOLATE EXPONENT
                                ;40570          CLK.UBCC,WORD,                  ;  CLOCK ALU Z-BIT ON EXPONENT
U 165C, 0C19,4024,45E0,F803,0010,165D   ;40571          D_Q,Q_D                         ;  D GETS LONG FRAC 0
                                ;40572
                                ;40573          ;-------------------------------;
                                ;40574          CACHE_D[LONG],                  ;  WRITE LONG FRAC 0 ON STACK
U 165D, 0000,013C,0180,3000,0000,1AD4   ;40575          Z?                              ;  TEST FOR 0 EXPONENT
                                ;40576
```


B 1 : FPA .MIC [600,1204] F
C 1 : FPA .MIC [600,1204] F
D 1 : FPA .MIC [600,1204] F
E 1 : FPA .MIC [600,1204] F
F 1 : FPA .MIC [600,1204] F
G 1 : FPA .MIC [600,1204] F
H 1 : FPA .MIC [600,1204] F
I 1 : FPA .MIC [600,1204] F
J 1 : FPA .MIC [600,1204] F
K 1 : FPA .MIC [600,1204] F
L 1 : FPA .MIC [600,1204] F
M 1 : FPA .MIC [600,1204] F
N 1 : FPA .MIC [600,1204] F
B 2 : FPA .MIC [600,1204] F
C 2 : FPA .MIC [600,1204] F
D 2 : FPA .MIC [600,1204] F
E 2 : FPA .MIC [600,1204] F
F 2 : FPA .MIC [600,1204] F
G 2 : FPA .MIC [600,1204] F
H 2 : FPA .MIC [600,1204] F
I 2 : FPA .MIC [600,1204] F
J 2 : FPA .MIC [600,1204] F
K 2 : FPA .MIC [600,1204] F
L 2 : FPA .MIC [600,1204] F
M 2 : FPA .MIC [600,1204] F
N 2 : FPA .MIC [600,1204] F
B 3 : FPA .MIC [600,1204] F
C 3 : FPA .MIC [600,1204] F
D 3 : FPA .MIC [600,1204] F
E 3 : FPA .MIC [600,1204] F
F 3 : FPA .MIC [600,1204] F
G 3 : FPA .MIC [600,1204] F
H 3 : FPA .MIC [600,1204] F
I 3 : FPA .MIC [600,1204] F
J 3 : FPA .MIC [600,1204] F
K 3 : FPA .MIC [600,1204] F
L 3 : FPA .MIC [600,1204] F
M 3 : GANDH .MIC [600,1204] G
N 3 : GANDH .MIC [600,1204] G
B 4 : GANDH .MIC [600,1204] G
C 4 : GANDH .MIC [600,1204] G
D 4 : GANDH .MIC [600,1204] G
E 4 : GANDH .MIC [600,1204] G
F 4 : GANDH .MIC [600,1204] G
G 4 : GANDH .MIC [600,1204] G
H 4 : GANDH .MIC [600,1204] G
I 4 : GANDH .MIC [600,1204] G
J 4 : GANDH .MIC [600,1204] G
K 4 : GANDH .MIC [600,1204] G
L 4 : GANDH .MIC [600,1204] G
M 4 : GANDH .MIC [600,1204] G
N 4 : GANDH .MIC [600,1204] G
B 5 : GANDH .MIC [600,1204] G
C 5 : GANDH .MIC [600,1204] G
D 5 : GANDH .MIC [600,1204] G
E 5 : GANDH .MIC [600,1204] G
F 5 : GANDH .MIC [600,1204] G
G 5 : GANDH .MIC [600,1204] G
H 5 : GANDH .MIC [600,1204] G
I 5 : GANDH .MIC [600,1204] G
J 5 : GANDH .MIC [600,1204] G
K 5 : GANDH .MIC [600,1204] G
L 5 : GANDH .MIC [600,1204] G
M 5 : GANDH .MIC [600,1204] G
N 5 : GANDH .MIC [600,1204] G
B 6 : GANDH .MIC [600,1204] G
C 6 : GANDH .MIC [600,1204] G
D 6 : GANDH .MIC [600,1204] G
E 6 : GANDH .MIC [600,1204] G
F 6 : GANDH .MIC [600,1204] G
G 6 : GANDH .MIC [600,1204] G
H 6 : GANDH .MIC [600,1204] G
I 6 : GANDH .MIC [600,1204] G
J 6 : GANDH .MIC [600,1204] G
K 6 : GANDH .MIC [600,1204] G
L 6 : GANDH .MIC [600,1204] G
M 6 : GANDH .MIC [600,1204] G
N 6 : GANDH .MIC [600,1204] G
B 7 : GANDH .MIC [600,1204] G
C 7 : GANDH .MIC [600,1204] G
D 7 : GANDH .MIC [600,1204] G
E 7 : GANDH .MIC [600,1204] G
F 7 : GANDH .MIC [600,1204] G
G 7 : GANDH .MIC [600,1204] G
H 7 : GANDH .MIC [600,1204] G
I 7 : GANDH .MIC [600,1204] G
J 7 : GANDH .MIC [600,1204] G
K 7 : GANDH .MIC [600,1204] G
L 7 : GANDH .MIC [600,1204] G
M 7 : GANDH .MIC [600,1204] G
N 7 : GANDH .MIC [600,1204] G
B 8 : GANDH .MIC [600,1204] G
C 8 : GANDH .MIC [600,1204] G
D 8 : GANDH .MIC [600,1204] G
E 8 : GANDH .MIC [600,1204] G
F 8 : GANDH .MIC [600,1204] G
G 8 : GANDH .MIC [600,1204] G
H 8 : GANDH .MIC [600,1204] G
I 8 : GANDH .MIC [600,1204] G
J 8 : GANDH .MIC [600,1204] G
K 8 : GANDH .MIC [600,1204] G
L 8 : GANDH .MIC [600,1204] G
M 8 : GANDH .MIC [600,1204] G
N 8 : GANDH .MIC [600,1204] G
B 9 : GANDH .MIC [600,1204] G
C 9 : GANDH .MIC [600,1204] G
D 9 : GANDH .MIC [600,1204] G
E 9 : GANDH .MIC [600,1204] G
F 9 : GANDH .MIC [600,1204] G
G 9 : GANDH .MIC [600,1204] G
H 9 : GANDH .MIC [600,1204] G
I 9 : GANDH .MIC [600,1204] G
J 9 : GANDH .MIC [600,1204] G
K 9 : GANDH .MIC [600,1204] G
L 9 : GANDH .MIC [600,1204] G
M 9 : GANDH .MIC [600,1204] G
N 9 : GANDH .MIC [600,1204] G
B 10 : GANDH .MIC [600,1204] G
C 10 : GANDH .MIC [600,1204] G
D 10 : GANDH .MIC [600,1204] G
E 10 : GANDH .MIC [600,1204] G
F 10 : GANDH .MIC [600,1204] G
G 10 : GANDH .MIC [600,1204] G
H 10 : GANDH .MIC [600,1204] G
I 10 : GANDH .MIC [600,1204] G
J 10 : GANDH .MIC [600,1204] G
K 10 : GANDH .MIC [600,1204] G
L 10 : GANDH .MIC [600,1204] G
M 10 : GANDH .MIC [600,1204] G
N 10 : GANDH .MIC [600,1204] G
B 11 : GANDH .MIC [600,1204] G
C 11 : GANDH .MIC [600,1204] G
D 11 : GANDH .MIC [600,1204] G
E 11 : GANDH .MIC [600,1204] G
F 11 : GANDH .MIC [600,1204] G
G 11 : GANDH .MIC [600,1204] G
H 11 : GANDH .MIC [600,1204] G
I 11 : GANDH .MIC [600,1204] G
J 11 : GANDH .MIC [600,1204] G
K 11 : GANDH .MIC [600,1204] G
L 11 : GANDH .MIC [600,1204] G
M 11 : GANDH .MIC [600,1204] G
N 11 : GANDH .MIC [600,1204] G
B 12 : GANDH .MIC [600,1204] G
C 12 : GANDH .MIC [600,1204] G
D 12 : GANDH .MIC [600,1204] G
E 12 : GANDH .MIC [600,1204] G
F 12 : GANDH .MIC [600,1204] G
G 12 : GANDH .MIC [600,1204] G
H 12 : GANDH .MIC [600,1204] G
I 12 : GANDH .MIC [600,1204] G
J 12 : GANDH .MIC [600,1204] G
K 12 : GANDH .MIC [600,1204] G
L 12 : GANDH .MIC [600,1204] G
M 12 : GANDH .MIC [600,1204] G
N 12 : GANDH .MIC [600,1204] G
B 13 : GANDH .MIC [600,1204] G
C 13 : GANDH .MIC [600,1204] G
D 13 : GANDH .MIC [600,1204] G
E 13 : GANDH .MIC [600,1204] G
F 13 : GANDH .MIC [600,1204] G
G 13 : GANDH .MIC [600,1204] G
H 13 : GANDH .MIC [600,1204] G
I 13 : GANDH .MIC [600,1204] G
J 13 : GANDH .MIC [600,1204] G
K 13 : GANDH .MIC [600,1204] G
L 13 : GANDH .MIC [600,1204] G
M 13 : GANDH .MIC [600,1204] G
N 13 : GANDH .MIC [600,1204] G
B 14 : GANDH .MIC [600,1204] G
C 14 : GANDH .MIC [600,1204] G
D 14 : GANDH .MIC [600,1204] G
E 14 : GANDH .MIC [600,1204] G
F 14 : GANDH .MIC [600,1204] G
G 14 : GANDH .MIC [600,1204] G
H 14 : GANDH .MIC [600,1204] G
I 14 : GANDH .MIC [600,1204] G
J 14 : GANDH .MIC [600,1204] G
K 14 : GANDH .MIC [600,1204] G
L 14 : GANDH .MIC [600,1204] G

```
M 14  ; GANDH .MIC [600,1204]      G
N 14  ; GANDH .MIC [600,1204]      G
B 15  ; GANDH .MIC [600,1204]      G
C 15  ; GANDH .MIC [600,1204]      G
D 15  ; GANDH .MIC [600,1204]      G
E 15  ; GANDH .MIC [600,1204]      G
F 15  ; GANDH .MIC [600,1204]      G
G 15  ; GANDH .MIC [600,1204]      G
H 15  ; GANDH .MIC [600,1204]      G
I 15  ; GANDH .MIC [600,1204]      G
J 15  ; GANDH .MIC [600,1204]      G
K 15  ; GANDH .MIC [600,1204]      G
L 15  ; GANDH .MIC [600,1204]      G
M 15  ; GANDH .MIC [600,1204]      G
N 15  ; GANDH .MIC [600,1204]      G
B 16  ; GANDH .MIC [600,1204]      G
C 16  ; GANDH .MIC [600,1204]      G
D 16  ; GANDH .MIC [600,1204]      G
E 16  U 163C, 0810,0038,B180,F830,00
F 16  ; GANDH .MIC [600,1204]      G
G 16  ; GANDH .MIC [600,1204]      G
H 16
I 16  ; GANDH .MIC [600,1204]      G
J 16  ; GANDH .MIC [600,1204]      G
K 16  ; GANDH .MIC [600,1204]      G
L 16  ; GANDH .MIC [600,1204]      G
```