HEWLETT hp PACKARD

MICROPROGRAM LISTING

30011A

EXTENDED PRECISION FLOATING POINT

(FOR THE HP 3000 COMPUTER SYSTEM)

Printed Circuit Assembly:

30011-60001 Series A-1339

30011-90007

HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD, CUPERTINO, CALIFORNIA, U.S.A.

PRINTED: JAN 1974

```
                                  &#@@@
                                  *
                                  *    EXTENDED PRECISION FLOATING-POINT ARITHMETIC (REV A - 10/9/73)
                                  *
                                  *         EADD, ESUB, EMPY, EDIV, ENEG, ECMP
                                  *
                                  *   EADD  %020400
                                  *   ESUB  %020401
                                  *   EMPY  %020402
                                  *   EDIV  %020403
                                  *   ENEG  %020404
                                  *   ECMP  %020405
                                  *   TRAP  %020406
                                  *   TRAP  %020407
                                  *
                                  *
                                  *         EADD, ESUB, EMPY, EDIV ENTERED WITH
                                  *                 (S)   = @V (DB-REL.)
                                  *                 (S-1) = @U (DB-REL.)
                                  *                 (S-2) = @W (DB-REL.)
                                  *         WHERE
                                  *                  W = U OP V
                                  *
                                  *         ENEG ENTERED WITH
                                  *                 (S)   = @U (DB-REL.)
                                  *         WHERE
                                  *                  U ← -U
                                  *
                                  *         ECMP ENTERED WITH
                                  *                 (S)   = @V (DB-REL.)
                                  *                 (S-1) = @U (DB-REL.)
                                  *         WHERE
                                  *                  CC ← U:V
                                  *
                                  *
                                  *
4000  37777777417   ESUB          ADD             SF2          IF SUBTRACT, SET F2=1.
4001  37762223425   EADD          JSB  PSHM            SR4     MAKE
4002  37777677777                 ADD                  SRL3      EXACTLY
4003  37762363425                 JSB  PSHM            UNC         TWO
4004  37762253420                 JSB  PUL1            SRL2          TOS
4005  37766254004                 JMP  *-1             SRL2            REGISTERS FULL.
4006  37762364152                 JSB  PULC            UNC
                                  *
                                  *    RETURN FROM GETN WITH
                                  *       V = V1,V2,V3 IN R0, R8, SP2
                                  *       U = U1,U2,U3 IN RA, SP1 OPND
                                  *
4007  26626174014       OPND JMP  EAD1 RC          NF2     RC ← U3.  IF ESUB, SET V= -V.
4010  30767377317        R0  CAD             HBF          F1 ← NOT SIGN(V).
4011  32773377770   R0   R8   IOR                        IF V = 0,
4012  35773007776   UBUS SP2  IOR                ZERO     DO NOT CHANGE SIGN.
4013  37617767330        R0   ADD       R0   FHB  UNC      ELSE SIGN(V) ← NOT SIGN(V).
4014  37777777310   EAD1 R0   ADD            HBF          F1 NOW CONTAINS SIGN(V).
4015  37777777333        RA   ADD            FHB          MAKE S(U)=S(V) FOR MAGNITUDE TEST.
```

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS
   56           4016  3052750?636           UBUS RD   SUB        SP3  CLO  CRRY      INSURE OVFLO IS CLEAR.
   57           4017  37766364026                    JMP  EXCH                UNC       IF MAG(U) < MAG(V), THEN
   58           4020  25766014034                SP3 JMP  NXCH               NZRO        INTERCHANGE U & V.
   59           4021  32527507774           SP1  RB   SUB        SP3         CRRY
   60           4022  37766364026                    JMP  EXCH               UNC
   61           4023  25766014034                SP3 JMP  NXCH               NZRO
   62           4024  35767517771           RC   SP2  SUB                    NCRY
   63           4025  37766364034                    JMP  NXCH               UNC
   64           4026  37617777773     EXCH  RA         ADD        RD                    SWAP U & V.
   65           4027  30677777777           RD   ADD        RA                     NOTE PIPE EFFECT
   66           4030  37657777774           SP1  ADD        RB                      ON REGISTER
   67           4031  32317777777           RB   ADD        SP1                      EXCHANGES.
   68           4032  37737777771           RC   ADD        SP2
   69           4033  35637777777           SP2  ADD        RC
   70           4034  37777527433     NXCH  RA         ADD             CF2  POS       F2 = RESULT SIGN.
   71           4035  37777777417                    ADD             SF2
   72           4036  32773377770           RD   RB   IOR                           IF V = 0, THEN
   73           4037  35773017776           UBUS SP2  IOR                NZRO        THE ANSWER
   74           4040  01526364142                SP1  JMP  EPK4 SP3       UNC         IS U.
   75           4041  37762364210                     JSB  UPAK           UNC       UNPACK U & V.
   76                                      *
   77                                      *     ON RETURN FROM UPAK WE HAVE
   78                                      *        RA  = U1 FRACTION    RD  = V1 FRACTION
   79                                      *        RB  = U2 FRACTION    SP1 = V2 FRACTION
   80                                      *        RC  = U3 FRACTION    SP2 = V3 FRACTION
   81                                      *        SP3 = EXP(U)         SP0 = EXP(V)
   82                                      *        U LEFT-SHIFTED 2 PLACES
   83                                      *        F1 = IF SIGN(U) = SIGN(V) THEN 0 ELSE 1
   84                                      *
   85           4042  37766154051                    JMP  NNEG               NF1       IF SIGN(U) NOT = SIGN(V),
   86           4043  30607377777                RD   CAD        RD                     THEN V = -V.
   87           4044  01307377777                SP1  CAD        SP1
   88           4045  35727507777                SP2  SUB        SP2         CRRY
   89           4046  37766364051                     JMP  NNEG              UNC
   90           4047  01316517777                SP1  INC        SP1         NCRY
   91           4050  30616777777                RD   INC        RD
   92           4051  25147777715     NNEG  SP0  SP3  SUB        BUSH OPND             OPND = - SHFT CNT FOR BIN PT ADJ
   93           4052  16771605000                UBUS ROM                 005000        UBUS = - SHIFT COUNT + 40.
   94           4053  16777527777                UBUS ADD                      POS      IF SHIFT > 40, THE ANS. IS U.
   95           4054  25326364224                SP3  JMP  EPK1 SP0            UNC       GO RE-PACK IT.
   96           4055  25731640000                SP3  ROM        SP2  040000          SP2 = EXP(U) + 256 FOR "EPAK".
   97           4056  35337777777                SP2  ADD        SP0                  PREPARE FOR SHIFT (NOTE PIPE).
   98           4057  30772767777                RD   REPN                 02        LINE UP
   99           4060  16600332275           SP0  UBUS TASL SL1  RD   INCT CTRM        V WITH
  100           4061  37737777765           RBUS      ADD        SP2                   U.
  101           4062  35337777777                SP2  ADD        SP0                  SP0 = XPON (NOTE PIPE).
  102           4063  01537777777                SP1  ADD        SP3                  PREPARE FOR BINARY POINT ADJUST.
  103           4064  26377417777                OPND ADD        CTRH       NZRO      CNTR = - SHFT CNT FOR BIN PT ADJ
  104           4065  37766364072                     JMP  EAD2              UNC       JMP IF ALREADY LINED UP
  105           4066  30772377777                RD   REPC                            ELSE
  106           4067  35600733276           UBUS SP2  TASR SR1  RD   INCT CTRM        LINE THEM UP (RIGHT-SHIFT V).
  107           4070  17737777777                SBUS ADD        SP2                  RD, SP3, SP2 = V
  108           4071  37777777777                     ADD                             ADD U + V. (PIPE ADJUST).
  109           4072  35637517771     EAD2  RC   SP2  ADD        RC         NCRY      RC = U3 + V3
  110           4073  25316767152           RB   SP3  INC        SP1  CTF   UNC       SP1 = U2 + V2 + COUT(3) OR
```

```
111          4074  25317777152          RB   SP3  ADD        SP1  CTF            SP1 + U2 + V2.
112          4075  37777557777               ADD                      NF1       F1 = COUT(2)
113          4076  30676767773          RA   RD   INC        RA        UNC       RA + U1 + V1 + COUT(2) OR
114          4077  30677777773          RA   RD   ADD        RA                  RA + U1 + V1.
115          4100  01773377771          RC   SP1  IOR                            IF RESULT FRACTION = 0,
116          4101  16773017773          RA   UBUS IOR                  NZRO      THEN ANSWER = 0.
117          4102  37666364234               JMP  ANSZ RA              UNC       U1 + 0.
118                                     *
119                                     *    THE FOLLOWING ROUTINE IS ENTERED WITH A (POSSIBLY
120                                     *    UNNORMALIZED) FRACTION IN RA, SP1, RC, AN EXCESS-512
121                                     *    EXPONENT IN SP0(8:9), AND A RESULT SIGN IN F2.  THE
122                                     *    FRACTION IS NORMALIZED, ROUNDED, PACKED WITH AN
123                                     *    EXCESS-256 EXPONENT, AND SIGNED.  CCA IS SET ON THE
124                                     *    RESULT & IT IS STORED IN THE DB-REL. ADDRESS  CONTAINED
125                                     *    IN S-2.   IF EXPONENT OVERFLOW OR UNDERFLOW HAS
126                                     *    OCCURED, A JUMP TO THE PROPER TRAP ROUTINE IS TAKEN.
127                                     *
128          4103  33766054107  EPAK      RA   JMP  ROND             BIT6      IF BIT6=1,NO NORMALIZATION REQUIRED
129          4104  33772377777               RA   REPC
130          4105  16660052271          RC   UBUS TASL SL1  RA   INCT BIT6      NORMALIZE TO BIT6.
131          4106  37637777765          RBUS      ADD       RC
132          4107  01537777777  ROND        SP1  ADD       SP3                  RA, SP3, RC + RESULT FRACTION.
133          4110  37771600004               ROM                 000004
134          4111  16637507771          RC   UBUS ADD       RC        CRRY      ROUND THE RESULT
135          4112  37766364115               JMP  EPK2                UNC        FRACTION TO
136          4113  25536517777               SP3  INC       SP3       NCRY        39 BITS
137          4114  33676777777               RA   INC       RA                    IN LENGTH.
138          4115  23117777777  EPK2        SM   ADD       BUSL RWA             GET #W1.
139          4116  15731640000               CTRM ROM       SP2  040000          SAVE SHIFT COUNT + 256 IN SP2.
140          4117  33772757777               RA   REPN                03        SHIFT RESULT FRAC. RIGHT 3 PLACES.
141          4120  31660733276          UBUS RC   TASR SR1  RA   INCT CTRM       LEADING 1 NOW IN BIT 8 OR 9.
142          4121  17637777777               SBUS ADD       RC                   SUBTRACT SHIFT COUNT + EXTRA 256
143          4122  35767777275          SP0  SP2  SUB             INCT           FROM EXPON. & PACK IN HI-WORD.
144          4123  16677527473          RA   UBUS ADD       RA   SF1  POS       IF RESULT = 10.. OR 11.. THEN
145          4124  22606364334               DB   JMP  EOVF RD              UNC        OVFLO OR UNFLO HAS OCCURRED.
146          4125  25773377453          RA   SP3  IOR                 CF1       IF PACKED RESULT = 0,
147          4126  31773017776          UBUS RC   IOR                  NZRO      THEN
148          4127  22606364334               DB   JMP  EOVF RD              UNC        UNDERFLOW.
149          4130  22617577777  EPK3        DB   ADD       RD        NF2       PUT DB IN RD AND
150          4131  33671300000               RA   ROMI      RA   100000          SET RESULT SIGN.
151          4132  26137777750          RD   OPND ADD       BSP0 CWA             SEND HI-BITS TO #1
152          4133  33177417757               RA   ADD       DATA CCA  NZRO        AND SET CCA
153          4134  25773377651          RC   SP3  IOR                 CCZ         ON RESULT
154          4135  37136777755          SP0       INC       BSP0 CWA             SEND MID-BITS
155          4136  25177777777               SP3  ADD       DATA                  TO #2.
156          4137  37116777755          SP0       INC       BUSL CWA             SEND LO-BITS
157          4140  23451777777               SM   ROM       SM   177777                  POP STACK BY 3 (S + S-3)
158          4141  31177757777               RC   ADD       DATA      NEXT        TO #3 & DONE.
159          4142  23117777777  EPK4        SM   ADD       BUSL RWA             GET #W1.
160          4143  37766364136               JMP  EPK3                UNC
161                                     *
162                                     *    BOTH PSBA AND PULC ARE ENTERED WITH SM AT S-2 AND SR=2.
163                                     *    PSBA PUSHES S AND S-1 INTO MEMORY.
164                                     *    PULC PULLS S-2 INTO RC.
165                                     *    THE GETN ROUTINE IS ENTERED WITH THE DB-REL. ADDRESS
```

```
166                          *    OF V1 IN RA, OF U1 IN RB, & OF W1 IN RC.  THE ABSOLUTE
167                          *    ADDR. OF V1, U1, & W1 IS CHECKED FOR >= DL, & V3, U3.
168                          *    & W3 FOR <= SM.  IF NO VIOLATION IS DETECTED, WE RETURN
169                          *    TO THE CALLING ROUTINE WITH
170                          *       RD, RB, SP2 = V (=V1, V2, V3)
171                          *       RA, SP1, OPND = U (= U1, U2, U3).
172                          *
173  4144  23136777757  PSBA      SM    INC        BSP0 C#A
174  4145  32177777777            RB    ADD        DATA                 PUSH B
175  4146  37136777755       SP0       INC        BSP0 CWA
176  4147  16767777142       Z    UBUS SUB             CTF
177  4150  33177777777            RA    ADD        DATA                 PUSH A
178  4151  37766152767                  JMP  STOV             NF1      IF Z < SM + 2  THEN STOV
179  4152  23117777777  PULC        SM    ADD        BUSL RWA           GET C
180  4153  23311777777            SM    ROM        SP1  177777         SP1 = S-3
181  4154  22767507776       UBUS DB    SUB                  CRRY
182  4155  37766262772                  JMP  STUN             NPRV     IF DB > S-3 & NPRV THEN STUN
183  4156  37536772777                  INC  SL1  SP3                   SP3 = 2.
184  4157  22617777772       RB   DB    ADD        RD                   RD = @U1
185  4160  22137777773  GETN RA   DB    ADD        BSP0 RWA             GET V1
186  4161  26637777777            OPND  ADD        RC                   RC = (S-2)
187  4162  34766777775       SP0  DL    BNDT                            IF @V1 < DL & USERMODE THEN BNDV.
188  4163  25777777775       SP0  SP3   ADD                             UBUS = @V3
189  4164  16766777774       SP1  UBUS  BNDT                            IF @V3 > S-3 & USERMODE, THEN BNDV.
190  4165  37136777775       SP0       INC        BSP0 RWA             GET V2.
191  4166  26617777777            OPND  ADD        RD                   RD = V1.
192  4167  34766777770       RD   DL    BNDT                            IF @U1 < DL & USERMODE, THEN BNDV.
193                          *                                            NOTE PIPE ON RD (=@U1)
194  4170  25777777765       RBUS SP3   ADD                             UBUS = @U3 (NOTE RBUS=@U1)
195  4171  16766777774       SP1  UBUS  BNDT                            IF @U3 > S-3 & USERMODE, THEN BNDV.
196  4172  37116777775       SP0       INC        BUSL RWA             GET V3.
197  4173  26317777777            OPND  ADD        SP1                  SP1 = V2.
198  4174  37677777774       SP1       ADD        RA                   RA = S-3 (NOTE PIPE)
199  4175  22337777771       RC   DB    ADD        SP0                  SP0 = @W1
200  4176  34766777776       UBUS DL    BNDT                            IF @W1 < DL & USERMODE, THEN BNDV.
201  4177  25537777775       SP0  SP3   ADD        SP3                  SP3 = @W3.
202  4200  22137777772       RB   DB    ADD        BSP0 RWA             GET U1.
203  4201  26737777777            OPND  ADD        SP2                  SP2 = V3.
204  4202  25766777773       RA   SP3   BNDT                            IF @W3 > S-3 & USERMODE, THEN BNDV.
205  4203  37136777775       SP0       INC        BSP0 RWA             GET U2.
206  4204  26677777777            OPND  ADD        RA                   RA = U1.
207  4205  01657777057            SP1   ADD        RB   SRA             RB = V2. POP STACK BY 2.
208  4206  37116777775       SP0       INC        BUSL RWA             GET U3.
209  4207  26317737777            OPND  ADD        SP1       RSB        SP1 = U2 & RETURN.
210                          *
211                          *    THE UPAK ROUTINE UNPACKS U & V, AND INSERTS THE
212                          *    LEADING 1 IN THE FRACTION PARTS. IT RETURNS WITH
213                          *       RA  = U1 FRACTION      RD  = V1 FRACTION
214                          *       RB  = U2 FRACTION      SP1 = V2 FRACTION
215                          *       RC  = U3 FRACTION      SP2 = V3 FRACTION
216                          *       SP3 = EXP(U)           SP0 = EXP(V)
217                          *       U LEFT-SHIFTED 2 PLACES
218                          *       F1 = IF SIGN(U) = SIGN(V) THEN 0 ELSE 1.
219                          *
220  4210  30763377313  UPAK RA   RD    XOR             HBF              F1=0 IF S(U) = S(V), ELSE =1.
```

```
221        4211  33521677700            RA   ROMN       SP3  077700        SP3 ← EXP(U).
222        4212  33761600077            RA   ROMN            000077        RA ← U1 FRACTION
223        4213  16771200100            UBUS ROMI            000100         WITH LEADING 1 INSERTED.
224        4214  16772767777            UBUS REPN                 02        SHIFT U LEFT
225        4215  16660332271       RC   UBUS TASL SL1  RA   INCT CTRM       2 PLACES TO
226        4216  37637777765       RBUS      ADD       RC                     PROVIDE GUARD BITS.
227        4217  30321677700            RD   ROMN       SP0  077700        SP0 ← EXP(V).
228        4220  30761600077            RD   ROMN            000077        RD ← V1 FRACTION.
229        4221  16611200100            UBUS ROMI       RD   000100         WITH LEADING 1 INSERTED.
230        4222  01657777777            SP1  ADD        RB                  SWAP U2, V2
231        4223  32317707777            RB   ADD        SP1       RSB        (NOKE PIPE EFFECT).
232                                 *
233                                 *    THE EPK1 ROUTINE RE-PACKS U AND TREATS IT
234                                 *    AS THE RESULT. IT IS ENTERED WITH
235                                 *       RA ← U1      RC ← U3
236                                 *       RB ← U2      SP0 ← EXP(U)
237                                 *          F2 ← RESULT SIGN
238                                 *          U LEFT-SHIFTED 2 PLACES
239        4224  23117777777       EPK1      SM   ADD        BUSL RWA           GET MW1.
240        4225  32537777777            RB   ADD        SP3                     PREPARE U FOR RIGHT-SHIFT.
241        4226  33772767777            RA   REPN                 02
242        4227  31660733276       UBUS RC   TASR SR1  RA   INCT CTRM       RESTORE U FRAC. TO ORIG. POSITION.
243        4230  17637777777            SBUS ADD        RC
244        4231  33761600077            RA   ROMN            000077         DELETE IMPLIED 1
245        4232  16677777775       SP0  UBUS ADD        RA                      & INSERT EXPONENT.
246        4233  37766364130                 JMP  EPK3             UNC        GO EXIT.
247                                 *
248                                 *    THE ANSZ ROUTINE SETS THE RESULT TO ZERO. IT IS
249                                 *    ENTERED WITH RA ← 0.
250                                 *
251        4234  23117777777       ANSZ      SM   ADD        BUSL RWA           GET MW1.
252        4235  37537777437                 ADD        SP3  CF2               SET MID-WORD, SIGN = 0.
253        4236  37626364130                 JMP  EPK3 RC        UNC        SET LO-WORD = 0 & GO FINISH.
254                                 *
255                                 *
256        4237  37762223425       EMPY           JSB  PSHM             SR4    MAKE
257        4240  37777677777                 ADD                      SRL3     EXACTLY
258        4241  37762363425                 JSB  PSHM             UNC        TWO
259        4242  37762253420                 JSB  PUL1             SRL2         TOS
260        4243  37766254242                 JMP  *-1              SRL2           REGISTERS FULL.
261        4244  37762364144                 JSB  PSBA             UNC
262                                 *
263                                 *    RETURN FROM GETN WITH
264                                 *      V = V1, V2, V3, IN RD, RB, SP2
265                                 *      U = U1, U2, U3 IN RA, SP1, OPND
266                                 *
267        4245  01773377633            RA   SP1  IOR             CLO          INSURE OVFLO IS CLEAR. IF
268        4246  26773017776            UBUS OPND IOR                  NZRO     U = 0, THE
269        4247  37666364234                 JMP  ANSZ RA             UNC        ANSWER = 0.
270        4250  32773377770            RD   RB   IOR                          IF V = 0,
271        4251  35773017776            UBUS SP2  IOR                  NZRO     THE ANSWER = 0.
272        4252  37666364234                 JMP  ANSZ RA             UNC
273        4253  26622364210                 OPND JSB  UPAK RC        UNC        RC ← U3.  GO UNPACK U & V.
274                                 *
275                                 *    RETURN FROM UPAK WITH
```

```
276                          *      RA ← U1 FRACTION      RD  ← V1 FRACTION
277                          *      RB ← U2 FRACTION      SP1 ← V2 FRACTION
278                          *      RC ← U3 FRACTION      SP2 ← V3 FRACTION
279                          *      SP3 ← EXP(U)          SP0 ← EXP(V)
280                          *      U LEFT-SHIFTED 2 PLACES
281                          *      F1 ← RESULT SIGN
282                          *
283        4254  25157777715       SP0  SP3  ADD        BUSH OPND              SAVE RESULT XPON + 512 IN OPND REG.
284        4255  32537777777            RB   ADD        SP3                    SP3 ← U2 (= MPYR)
285        4256  35337777777            SP2  ADD        SP0                    SP0 ← V3 (= MPCND)
286        4257  37772607777                 REPN                    20
287        4260  16774333275       SP0  UBUS MPAD SR1         INCT CTRM      DO U2*V3. DISCARD LO-BITS (IN SP3).
288        4261  17337777777            SBUS ADD        SP0                    SAVE HI-BITS IN SP0.
289        4262  01537777777            SP1  ADD        SP3                    SP3 ← V2 (= MPYR)
290        4263  37772607777                 REPN                    20
291        4264  16774333271       RC   UBUS MPAD SR1         INCT CTRM      DO U3*V2. DISCARD LO-BITS.
292        4265  17337517435       SP0  SBUS ADD        SP0  CF2  NCRY      SP0 ← U2*V3 + U3*V2 HI-BITS (= K1).
293        4266  37777777417                 ADD                 SF2            SAVE CARRY IN F2.
294        4267  35537777777            SP2  ADD        SP3                    SP3 ← V2 (=MPYR).
295        4270  37772607775       SP0       REPN                    20       UBUS ← K1 (ADD TO NEXT PRODUCT).
296        4271  16774333273       RA   UBUS MPAD SR1         INCT CTRM      DO U1*V3 + K1. PUT RESULT IN
297        4272  17337577777            SBUS ADD        SP0       NF2           SP0, SP3. ADD IN PREVIOUS CARRY
298        4273  16336777777            UBUS INC        SP0                      TO HI-BITS. LET LO-BITS = K2.
299        4274  31537777777            RC   ADD        SP3                    SP3 ← U3 (=MPYR).
300        4275  25772607777            SP3  REPN                    20       UBUS ← K2 (NOTE PIPE EFFECT).
301        4276  16774333270       RD   UBUS MPAD SR1         INCT CTRM      DO U3*V1 + K2. ADD PREVIOUS HI-BITS
302        4277  17337777775       SP3  SBUS ADD        SP0                    TO PRODUCT HI-BITS. RESULT=K4,K3.
303        4300  32537777777            RB   ADD        SP3                    SP3 ← U2 (= MPYR).
304        4301  25772607777            SP3  REPN                    20       UBUS ← K3 (NOTE PIPE EFFECT).
305        4302  16774333274       SP1  UBUS MPAD SR1         INCT CTRM      DO U2*V2 + K3.
306        4303  17337517435       SP0  SBUS ADD        SP0  CF2  NCRY      ADD IN K4 TO HI-BITS & SAVE
307        4304  37777777417                 ADD                 SF2            CARRY IN F2. RESULT = K6,K5.
308        4305  25637777777            SP3  ADD        RC                     SAVE FINAL RESULT LO-BITS IN RC.
309        4306  01537777777            SP1  ADD        SP3                    SP3 ← V2 (= MPYR).
310        4307  37772607775       SP0       REPN                    20       UBUS ← K6.
311        4310  16774333273       RA   UBUS MPAD SR1         INCT CTRM      DO U1*V2 + K6
312        4311  17337577777            SBUS ADD        SP0       NF2           ADD IN PREVIOUS CARRY
313        4312  16336777777            UBUS INC        SP0                      TO HI-BITS. RESULT = K8,K7.
314        4313  37766324737                 JMP  RIP                   TEST    RESET ON INTERRUPT
315        4314  32537777777            RB   ADD        SP3                    SP3 ← U2 (= MPYR).
316        4315  25772607777            SP3  REPN                    20       UBUS ← K7 (NOTE PIPE EFFECT).
317        4316  16774333270       RD   UBUS MPAD SR1         INCT CTRM      DO U2*V1 + K7. ADD K8 TO PRODUCT
318        4317  17337777775       SP0  SBUS ADD        SP0                    HI-BITS. RESULT = K10,K9.
319        4320  25657777777            SP3  ADD        RB                     SAVE FINAL RESULT MID-BITS IN RB.
320        4321  30537777777            RD   ADD        SP3                    SP3 ← V1 (= MPYR).
321        4322  37772607775       SP0       REPN                    20       UBUS ← K10.
322        4323  16774333273       RA   UBUS MPAD SR1         INCT CTRM      DO U1*V1 + K10. FINAL RESULT HI-
323        4324  32537777777            RB   ADD        SP3                    BITS ARE NOW IN SP3.  SET UP FOR
324        4325  25772727777            SP3  REPN                    06         RIGHT-SHIFT (NOTE PIPE EFFECT).
325        4326  31660733276       UBUS RC   TASR SR1   RA   INCT CTRM      SET UP RESULT FRACTION & SIGN
326        4327  17637557437            SBUS ADD        RC   CF2  NF1         FOR EPAK.
327        4330  37777777417                 ADD                 SF2            NOTE THAT ARITH. SHIFT MAY HAVE
328        4331  33661601777            RA   ROMN       RA   001777            COPIED 1'S INTO RA(0:5).
329        4332  25317777777            SP3  ADD        SP1                    SP1 ← W2.
330        4333  26326364103            OPND JMP  EPAK  SP0       UNC         SP0 ← XPON & GO PACK.
```

```
331                                *
332                                *
333                                *    THE EOVF ROUTINE IS ENTERED IF AN EXPONENT OVERFLOW OR
334                                *    UNDERFLOW HAS BEEN DETECTED IN EPAK.  NOTE THAT CCE IS
335                                *    NOT POSSIBLE HERE (ZERO IS A VALID ANSWER & HAS BEEN
336                                *    PREVIOUSLY CHECKED FOR).  HENCE, WITH CNTR SET NOT = 0, WE
337                                *    FORCE CCL OR CCG ON THE RESULT.  IT IS ENTERED WITH
338                                *       RA = PACKED U1      RC = U3          F1 = OVFLO/UNFLO
339                                *       SP3 = U2            F2 = RESULT SIGN    INDICATOR
340                                *    THE POINTER TO W IS LEFT IN TOS IF STA(1)=1.
341                                *
342  4334  33661677777  EOVF       RA   ROMN       RA   #77777        REMOVE EXP. OVF. FROM SIGN BIT
343  4335  23117577777             SM   ADD        BUSL R=A  MF2       GET W1,  SET
344  4336  33671300006             RA   ROMI       RA   100000         RESULT SIGN.
345  4337  37777777777                  ADD
346  4340  14773377753        RA   CTRL IOR             CCA           SET CCL OR CCG
347  4341  26137777750        RD   OPND ADD        BSP0 C=A           SEND HI-WORD
348  4342  33177777777             RA   ADD        DATA                TO W1.
349  4343  37136777755        SP0       INC        BSP0 C=A           SEND MIDDLE-WORD
350  4344  25177777777             SP3  ADD        DATA                TO W2.
351  4345  37116777755        SP0       INC        BUSL C=A           SEND LO-WORD
352  4346  23337777777             SM   ADD        SP0                TO W3. IF F1=0,
353  4347  31177777777             RC   ADD        DATA                 OR IF W(1)=1,
354  4350  24777772777             STA  ADD  SL1                         THEN UNDERFLOW HAS OCCURRED
355  4351  16777532777             UBUS ADD  SL1                 NEG   ELSE OVERFLOW
356  4352  37447357615        SP0       CAD        SM   SOV  NEXT    IF STATUS(2)=0,THEN POP BY 3 & SOV
357  4353  33777532777             RA   ADD  SL1                 NEG
358  4354  37777547777                  ADD                      F1
359  4355  37776777777                  INC                           MAKE SP1=%11 IF W(1)=1 OR F1=0
360  4356  16311600010             UBUS ROM        SP1  000010        SP1 = %10 OR %11
361  4357  37766362757                  JMP  TRPE                 UNC
362                                *
363                                *
364                                &4374
365  4374  37766364001                  JMP  EADD                 UNC   ENTRY POINT FOR EADD
366  4375  37766364000                  JMP  ESUB                 UNC   ENTRY POINT FOR ESUB
367  4376  37766364237                  JMP  EMPY                 UNC   ENTRY POINT FOR EMPY
368                                *
369  4377  37762223425  EDIV            JSB  PSHM                 SR4   MAKE
370  4400  37777677777                  ADD                       SRL3    EXACTLY
371  4401  37762363425                  JSB  PSHM                 UNC       TWO
372  4402  37762253423                  JSB  PUL1                 SRL2        TOS
373  4403  37766254402                  JMP  *-1                  SRL2          REGISTERS FULL.
374  4404  37762364144                  JSB  PSBA                 UNC
375                                *
376                                *    RETURN FROM GETN WITH
377                                *       V = V1, V2, V3 IN RD, RB, SP2
378                                *       U = U1, U2, U3 IN RA, SP1, OPND
379                                *
380  4405  32773377630        RD   RB   IOR             CLO           INSURE OVFLO IS CLEAR. IF V = 0,
381  4406  35773017776        UBUS SP2  IOR                 NZRO       GO TO "DIVIDE-BY-ZERO"
382  4407  26626364614             OPND JMP  EDVZ RC         UNC        ERROR ROUTINE.
383  4410  01773377773        RA   SP1  IOR                           IF U = 0,
384  4411  26773017776        UBUS OPND IOR                 NZRO       GO TO
385  4412  37666364234                  JMP  ANSZ RA              UNC    "ANS.= ZERO" ROUTINE.
```

```
                OPND JSB  UPAK RC        UNC       RC ← U3, GO UNPACK U & V.
                *
                *    RETURN FROM UPAK WITH
                *       RA ← U1 FRACTION     RB  ← V1 FRACTION
                *       RB ← U2 FRACTION     SP1 ← V2 FRACTION
                *       RC ← U3 FRACTION     SP2 ← V3 FRACTION
                *       SP3 ← EXP(U)         SP0 ← EXP(V)
                *       U LEFT-SHIFTED 2 PLACES
                *       F1 ← RESULT SIGN
                *
                SP0       ADD        BUSH OPND          SAVE EXP(V) IN OPND REG.
                          ROM        000003
                UBUS SM   ADD        BUSL CWA           SAVE EXP(U) - EXP(V) + 511
                     SP3  ROM        077700              IN MEMORY
                UBUS OPND SUB        DATA                AT SM+3
                     SP2  ADD        SP0                SP0 ← V3 FOR SHIFT
                     RD   REPN                  11      SHIFT V LEFT 9 PLACES & RESTORE
                SP0  UBUS TASL SL1   RD   INCT CTRM      LEADING 1 LOST DUE TO ARITH.
                RBUS      ADD        SP2                 SHIFT. (THIS INSURES
                     RD   ROMI       RD   100000          V1 >= 2**16/2).
                     SP1  ADD        RB                 SAVE V2 IN RB.  PREPARE FOR
                     RB   ADD        SP1                 FIRST DIVIDE (NOTE PIPE ON RB).
                RA        REPN                 CF2  21   DO U1,U2/V1.
                UBUS RD   DVSB SL1             INCT CTRM  SP1 ← Q'1; F2,RBUS ← 2*REM.
                RBUS      ADD  SR1  RA              NF2  PUT REMAINDER
                     UBUS ROMI      RA         100000     IN RA.
                          ADD       SP0  FHB           SAVE RESULT SIGN IN SP0(0).
                     SP1  ADD       SP3                SP3 ← MPYR (= Q'1).
                          REPN                  20      DO Q'1*V2.
                RB   UBUS MPAD SR1             INCT CTRM  SBUS,SP3 ← PRODUCT.
                     SBUS ADD       BUSH OPND           PUT HI-BITS IN OPND REG.
                RC   SP3  SUB       RC             NCRY  IF REM,U3 - Q'1*V2 >= 0,
                RA   OPND SUB       RA   CTF  UNC       THEN Q'1-1 <= Q1 = Q'1.
                RA   OPND CAD       RA   CTF
                          JMP  EDV1                - F1    (F1=1 MEANS REM,U3 - Q'1*V2 >= 0)
                SP1       CAD       SP1                 ELSE Q'1-2 <= Q1 < Q'1.
                RC   RB   ADD       RC             NCRY   SET Q'1 ← Q'1 - 1,
                RA   RD   INC       RA             UNC     R11,R12 ← REM,U3-Q'1*V2+V1,V2.
                RA   RD   ADD       RA
EDV1            SP2  ADD            SP3                SP3 ← MPYR (=V3)
                          REPN                  20      DO Q'1*V3
                SP1  UBUS MPAD SR1             INCT CTRM  SBUS,SP3 ← PRODUCT HI,LO-BITS.
                     SBUS ADD       BUSH OPND           SAVE HI-BITS IN OPND REG.
                     SP3  SUB       SP3            NCRY  IF R11,R12,0 - 0,Q'1*V3 HI,LO >= 0,
                RC   OPND SUB       RC   CTF  UNC       THEN
                RC   OPND CAD       RC   CTF             Q1 = Q'1
                          ADD                F1           & TRUE REM. = R11,R12,R13
                RA        CAD       RA   CTF              IS IN RA,RC,SP3.
                          JMP  EDV2                F1    DONE WITH Q1.
                SP1       CAD       SP1                 ELSE
                     SP3  ADD                            Q1 ← Q'1 - 1,
                UBUS SP2  ADD       SP3            NCRY   R11,R12,R13 ← R11,R12,0 - 0,
                RC   RB   INC       RC   CTF  UNC         Q'1*V3 HI,LO ← V1,V2,V3.
                RC   RB   ADD       RC   CTF
                          ADD                F1
```

```
                ADDRESS   CONTENTS      LABL RBUS SBUS FUNC SHFT STOR SPEC SKIP      COMMENTS          HP 3000 MICRO-ASSEMBLER X1.00

 441            4471   30677767773           RA   RD   ADD       RA        UNC
 442            4472   30676777773           RA   RD   INC       RA
 443            4473   37771600004    EDV2                ROM          000004
 444            4474   23117777756           UBUS SM   ADD       BUSL CWA              SAVE QI IN MEMORY
 445            4475   01177777777                SP1  ADD       DATA                   AT SM+4.
 446                                  *
 447                                  *     AT THIS POINT WE HAVE QI IN MEMORY AT SM+2, AND
 448                                  *     A TRUE FIRST REMAINDER IN RA,RC,SP3.  THIS WILL
 449                                  *     BE THE DIVIDEND FOR Q2.
 450                                  *
 451            4476   37777777315           SP0       ADD            MBF              PUT RESULT SIGN BACK IN F1 FOR NOW.
 452            4477   37317777771           RC        ADD       SP1                   PREPARE FOR SECOND DIVIDE.
 453            4500   33767417430           RD   RA   SUB            CF2  NZRO        IF V1 = R11
 454            4501   37336767417                     INC       SP0  SF2  UNC         THEN Q'21 = 1 & R'21 = 0
 455            4502   37337777777                     ADD       SP0                    ELSE Q'21 = 0 & R'21 = R11
 456            4503   37157577712           RB        ADD       BUSH OPND NF2        SAVE V2 IN OPND REG.  SECOND DIV.
 457            4504   37677777777                     ADD       RA                     WILL BE 0,R12/V1 OR R11,R12/V1,
 458            4505   32537777777                RB   ADD       SP3                    DEPENDING ON Q'21.  SWAP R13,
 459            4506   25657777777                SP3  ADD       RB                      V2 FOR FOLLOWING MULTIPLY.
 460            4507   37772577433           RA        REPN           CF2  21          DO R11,R12/V1 (R11 MAY = 0).
 461            4510   30764332276           UBUS RD   DVSH SL1       INCT CTRM        SP1 + Q'22; F2,RBUS + 2*REM.
 462            4511   37677573765           RBUS      ADD  SR1  RA        NF2         PUT REMAINDER
 463            4512   16671300000                UBUS MOMI      RA   100000          IN RA.
 464            4513   37772607777                     REPN                 20         DO Q'22*V2.
 465            4514   16774333274           SP1  UBUS MPAD SR1       INCT CTRM        SBUS,SP3 + PRODUCT HI,LO-BITS.
 466            4515   17637557437                SBUS ADD       RC   CF2  NF1       RC + Q'22*V2 HI-BITS.
 467            4516   37777777417                     ADD            SF2              F2 + RESULT SIGN.
 468            4517   37777407775           SP0       ADD                 ZERO        ADD Q'21*V2 TO ABOVE
 469            4520   26637777771           RC   OPND ADD       RC                    PRODUCT HI-BITS.
 470                                  *
 471                                  *     TO RECAPITULATE, WE NOW HAVE
 472                                  *        SP0,SP1 + Q'21,Q'22      RD,OPND,SP2 + V1,V2,V3
 473                                  *        RA + REM.                RC,SP3 + Q'21*V2 + Q'22*V2 = Q'2*V2
 474                                  *        RB + R13                 F2 + RESULT SIGN.
 475                                  *
 476            4521   37766324737                JMP  RIP                 TEST        RESET ON INTERRUPT
 477                                  *
 478            4522   25647517772           RB   SP3  SUB       RB        NCRY        IF R21,R22 + REM,R13 - Q'2*V2 >= 0
 479            4523   31667767153           RA   RC   SUB       RA   CTF  UNC          THEN
 480            4524   31667377153           RA   RC   CAD       RA   CTF               Q'2-1 <= Q2 = Q'2.
 481            4525   37766144534                     JMP  EDV4                F1      F1=1 MEANS R21,R22 >= 0.
 482            4526   37307107774    EDV3   SP1       CAD       SP1       CRRY        ELSE Q'2 + Q'2 - 1,
 483            4527   37327377775           SP0       CAD       SP0                    R21,R22 + R21,R22 + V1,V2.
 484            4530   26657517772           RB   OPND ADD       RB        NCRY         IF R21,R22 STILL < 0, GO
 485            4531   30676767153           RA   RD   INC       RA   CTF  UNC           BACK TO EDV3 (THIS LOOP
 486            4532   30677777153           RA   RD   ADD       RA   CTF                WILL BE EXECUTED
 487            4533   37766154526                     JMP  EDV3                NF1      0,1, OR 2 TIMES).
 488            4534   35537777777    EDV4        SP2  ADD       SP3                   SP3 + MPYR (= V3).
 489            4535   37772607777                     REPN                 20         DO Q'22*V3
 490            4536   16774333274           SP1  UBUS MPAD SR1       INCT CTRM        SBUS,SP3 + PRODUCT HI,LO-BITS.
 491            4537   17637777777                SBUS ADD       RC                    RC + PRODUCT HIBITS.
 492            4540   37777407775           SP0       ADD                 ZERO        ADD Q'21*V3 TO
 493            4541   35637777771           RC   SP2  ADD       RC                    PRODUCT HI-BITS (NOW = Q'2*V3).
 494            4542   25527517777                SP3  SUB       SP3       NCRY        IF R21,R22,R23 + R21,R22,0
 495            4543   31647767152           RB   RC   SUB       RB   CTF  UNC          - 0,Q'2*V3HI,Q'2*V3LO >= 0
```

Computer Museum

```
 496        4544  31647377152          RB   RC   CAD       RB   CTF              THEN Q2 = Q'2
 497        4545  25637547777               SP3  ADD       RC        F1          & TRUE REM. = R21,R22,R23
 498        4546  37667377153          RA        CAD       RA   CTF                ( IN RA,RB,RC )
 499        4547  37766144557                    JMP  EDV5              F1       F1 = 1 MEANS R21,R22,R23>0
 500        4550  37307377774          SP1       CAD       SP1                   ELSE Q2 = Q'2 - 1 (NOTE Q'21 NOW
 501        4551  35637517771          RC   SP2  ADD       RC        NCRY       MUST = 0), AND
 502        4552  26656767152          RB   OPND INC       RB   CTF  UNC        TRUE REM = R21,R22,R23 +
 503        4553  26657777152          RB   OPND ADD       RB   CTF             R21,R22,R23 + V1,V2,V3.
 504        4554  37777557777                    ADD                 NF1
 505        4555  30676767773          RA   RD   INC       RA        UNC
 506        4556  30677777773          RA   RD   ADD       RA
 507                                  *
 508                                  *    AT THIS POINT WE HAVE SP1 = Q2 & THE TRUE
 509                                  *    REMAINDER IN RA,RB,RC.   THE RESULT SIGN IS
 510                                  *    IN F2, & V1,V2,V3 IS IN RD,OPND,SP2.  TO PREVENT
 511                                  *    THE OVERFLOW CONDITION THAT OCCURRED IN COM-
 512                                  *    PUTING Q2, WE WILL COMPUTE Q3/2 & THEN MULTIPLY
 513                                  *    THIS RESULT BY 2.
 514                                  *
 515        4557  37537577452    EDV5 RB        ADD       SP3  CF1  NF2        SP3 = R22 FOR DIV. BY 2.
 516        4560  37777777477               ADD                 SF1          F1 = RESULT SIGN
 517        4561  37772377777               REPC
 518        4562  31660763773          RA   RC   TASR SR1  RA        UNC       RA,SP3,RC = REM/2.
 519        4563  17637777777               SBUS ADD       RC
 520        4564  33661677777               RA   ROMN      RA   077777         GET RID OF MSB FROM ARITH. SHIFT.
 521        4565  01657777777               SP1  ADD       RB                  SAVE Q2 IN RB
 522        4566  25317777777               SP3  ADD       SP1                 PREPARE FOR THIRD DIVIDE.
 523        4567  37772577433          RA        REPN                CF2  21    DO R21,R22/V1.
 524        4570  30764332276          UBUS RD   DVSH  SL1            INCT CTRM   SP1 = Q'31 F2,RBUS = 2*REM.
 525        4571  37617573765          RBUS      ADD   SR1  RD             NF2  PUT REMAINDER
 526        4572  16611300000               UBUS ROMI       RD   100000        IN RD (V1 NO LONGER NEEDED).
 527        4573  26537777777               OPND ADD        SP3                 SP3 = MPYR (= V2).
 528        4574  37771600003                    ROM                 000003
 529        4575  23137777776          UBUS SM   ADD       BSP0 R=A            GET XPON FROM MEMORY
 530        4576  37772607777               REPN                      22       DO Q'3*V2
 531        4577  16774333274          SP1  UBUS MPAD  SR1       INCT CTRM      SBUS,SP3 = PRODUCT HI,LO-BITS.
 532        4600  17737777777               SBUS ADD        SP2                 SP2 = Q'3*V2 HI-BITS.
 533        4601  37116777775          SP0       INC       BUSL R=A            GET Q1 FROM MEMORY.
 534        4602  26337557437               OPND ADD        SP0  CF2  NF1       SP0 = XPON.
 535        4603  37777777417                    ADD                 SF2          F2 = RESULT SIGN.
 536        4604  25767517771          RC   SP3  SUB                     NCRY   IF REM,R23 - Q'3*V2 >= 0,
 537        4605  35767767150          RD   SP2  SUB                CTF  UNC     THEN
 538        4606  35767377150          RD   SP2  CAD                CTF           Q'3 = Q'3
 539        4607  37777547777                    ADD                 F1           ELSE
 540        4610  37307377774          SP1       CAD       SP1                     Q'3 = Q'3 - 1.
 541        4611  26677777777               OPND ADD        RA                  RA = Q1.
 542        4612  01637772777               SPI  ADD  SL1   RC                  RC = Q3 = Q'3 *2.
 543        4613  32306364103               RB   JMP  EPAK  SP1       UNC       SP1 = Q2 & GO PACK.
 544                                  *
 545                                  *    THE FOLLOWING ROUTINE IS ENTERED IF A DIVIDE-BY-ZERO
 546                                  *    ERROR IS DETECTED.  THE DIVIDEND IS PLACED IN THE RESULT
 547                                  *    LOCATION (W), CCA IS SET ON IT, & THE POINTER TO W
 548                                  *    IS LEFT ON TOS IF STA(1)=1.
 549                                  *
 550        4614  23117777777    EDVZ      SM   ADD       BUSL R=A            GET W1
```

```
551          4615  33777417757                RA   ADD                CCA  NZRO   SET CCA
552          4616  31773177654           SP1  PC   IOR                CCZ         ON DIVIDEND.
553          4617  22777777777                DB   ADD
554          4620  26137777756           UBUS OPND ADD          BSP? C=A         PUT DVND
555          4621  33177777777                RA   ADD          DATA               IN RESULT
556          4622  37136777755           SP0       INC          BSP? C=A           LOCATION.
557          4623  01177777777                SP1  ADD          DATA
558          4624  37116777755           SP0       INC          BUSL C=A
559          4625  23337777777                SM   ADD          SP0
560          4626  31177777777                RC   ADD          DATA
561          4627  24777772777                STA  ADD  SL1
562          4630  16777532777                UBUS ADD  SL1                NEG
563          4631  37447357615           SP0       CAD               SM   SOV  NEXT   IF STATUS(2)=0,THEN POP BY 3 & SOV
564          4632  37311600012                     ROM               SP1  0000012      SP1 ← DIV-BY-ZERO PARAM.
565          4633  37766362757                     JMP  TRPE                    UNC
566                                *
567                                *
568          4634  37762223425   ENEG              JSB  PSHM                    SR4    MAKE
569          4635  37777677777                     ADD                          SRL3     EXACTLY
570          4636  37762363425                     JSB  PSHM                    UNC        ONE
571          4637  37777657777                     ADD                          SRL2         TOS
572          4640  37762363425                     JSB  PSHM                    UNC            REGISTER
573          4641  37762203420                     JSB  PUL1                    SR2              FULL.
574          4642  22137777773           RA   DB   ADD          BSP? R=A           GET U1.   SP0 ← @U1
575          4643  34766777776           UBUS DL   BNDT                         IF @U1 < DL & USERMODE, THEN BNDV.
576          4644  37136777775           SP0       INC          BSP? R=A           GET U2.   SP0 ← @U2
577          4645  26317777577                OPND ADD          SP1  POP           SP1 ← U1.  POP STACK
578          4646  16767377317                UBUS CAD               4BF         F1 ← COMPLEMENT OF SIGN(U).
579          4647  23677777777                SM   ADD          RA                 RA ← SM
580          4650  37776777775           SP0       INC                          UBUS ← @U3
581          4651  16766777773           RA   UBUS BNDT                         IF SM < @U3 & USERMODE, THEN BNDV.
582          4652  37116777775           SP0       INC          BUSL R=A           GET U3.
583          4653  26773377777                OPND IOR                          U2 NOW IN OPND REG.
584          4654  26533377776           UBUS OPND IOR          SP3                SP3 ← U2 IOR U3.
585          4655  16773017774           SP1  UBUS IOR                    NZRO   IF U = 0,
586          4656  37777757737                     ADD                     CCE  NEXT    SET CCE & EXIT.
587          4657  37107377755           SP0       CAD          BUSL C=A             ELSE NEGATE U,
588          4660  01177777337                SP1  ADD          DATA F=B               RETURN U1 TO MEMORY,
589          4661  16777417757                UBUS ADD                     CCA  NZRO      & SET CCA
590          4662  25777777657                SP3  ADD                     CCZ             ON RESULT.
591          4663  37777757777                     ADD                          NEXT   DONE.
592                                *
593                                *
594                                *
595          4664  37762223425   ECMP              JSB  PSHM                    SR4    MAKE
596          4665  37777677777                     ADD                          SRL3     EXACTLY
597          4666  37762363425                     JSB  PSHM                    UNC        TWO
598          4667  37762253420                     JSB  PUL1                    SRL2         TOS
599          4670  37766254667                     JMP  *-1                     SRL2           REGISTERS FULL.
600          4671  22117777772           RB   DB   ADD          BUSL R=A           GET U1
601          4672  34766777776           UBUS DL   BNDT                         IF @U1 < DL & USERMODE, THEN BNDV.
602          4673  23311777776                SM   ROM          SP1  177776        SP1 ← S-4
603          4674  22617777772           RB   DB   ADD          RD                 RD ← @U1
604          4675  22737777773           RA   DB   ADD          SP2                SP2 ← @V1
605          4676  30766777774           SP1  RD   BNDT                         IF S-4 < @U1 & USERMODE,THEN BNDV.
```

```
                4677  22137777773          RA   DB   ADD        BSP2 RWA          GET V1 & SP0 ← #V1
                4700  26317777777          OPND ADD        SP1                    SP1 ← U1.
                4701  35766777774          SP1  SP2  BNDT                         IF S-4 < #V1 & USERMODE,THEN BNDV.
                                           *                                      NOTE PIPE ON SP1
                4702  34766777775          SP0  DL   BNDT                         IF #V1 < DL & USERMODE,THEN BNDV..
                4703  22136777772          RB   DB   INC        BSP0 RWA          GET U2.
                4704  26617777777          OPND ADD        RD                     RD ← V1.
                4705  16763137774          SP1  UBUS XOR                  NEG     SIGN(U) = SIGN(V)?
                4706  37766364712                    JMP  ECM1           UNC     YES. GO GET THE REST OF U & V.
                4707  01777527717          SP1  ADD             CCG  POS          NO. CC ← CCG IF U >= 0,
                4710  37777777677               ADD             CCL               ELSE CC ← CCL.
                4711  37777757057               ADD             SR0  NEXT           POP STACK & DONE.
                4712  37116777775          ECM1 SP0  INC        BUSL RWA          GET U3.
                4713  26637777457          OPND ADD        RC   SR0               RC ← U2. POP STACK.
                4714  30607377774          SP1  RD   CAD        RD                RD ← W1' = U1 - V1 - 1.
                4715  22136777773          RA   DB   INC        BSP0 RWA          GET V2.
                4716  26657777777          OPND ADD        RB                     RB ← U3.
                4717  37116777775          SP0  INC        BUSL RWA          GET V3.
                4720  26627117771          RC   OPND CAD        RC        NCRY    RC ← W2' = U2 - V2 - 1.
                4721  30616777777               RD   INC        RD                CARRY OUT OF MIDDLE WORD
                4722  26647517772          RB   OPND SUB        RB        NCRY    RB ← W3 = U3 - V3.
                4723  31636507777               RC   INC        RC        CRRY     CHECK FOR
                4724  37777767777                    ADD                  UNC       BORROWS.
                4725  30616777777               RD   INC        RD                W ← U - V.
                4726  16766014732                    UBUS JMP  ECM2           NZRO    W1 = 0?
                4727  32773017771          RC   RB   IOR                  NZRO    YES. W2, W3 = 0?
                4730  37777757737                    ADD             CCE  NEXT    YES. SET CCE & DONE.
                4731  37616777777                    INC        RD                NO. SET W1 = 1 TO FORCE CCG.
                4732  01777537777          ECM2      SP1  ADD             NEG     U < 0?
                4733  30777757757               RD   ADD             CCA  NEXT    NO. SET CCL OR CCG ON W1 & DONE.
                4734  30607017757               RD   CAD        RD   CCA  NZRO    YES. SET CCA ON NOT(W1).
                4735  37777777717                    ADD             CCG          IF NOT(W1) = 0, FORCE CCG.
                4736  37777757777                    ADD                  NEXT    DONE.
                                           *
                                           *
                4737  23451677777          RIP       SM   ROM        SM   177777
                4740  37766362400                    JMP  IRD                 UNC
                                           *
                                           *
                                           &4774
                4774  37766364634                    JMP  ENEG                UNC     ENTRY POINT FOR ENEG
                4775  37766364664                    JMP  ECMP                UNC     ENTRY POINT FOR ECMP
                4776  37766362765                    JMP  TRP7                UNC     UNIMPLEMENTED
                4777  37766362765                    JMP  TRP7                UNC      INSTRUCTION
                                           % TRPE  2757
                                           % STOV  2767
                                           % STUN  2772
                                           % IRD   2400
                                           % TRP7  2765
                                           % PUL1  3420
                                           % PSHM  3425
                                           #

ROM COUNT=473     ERRORS=0
```

```
ANSZ  4234  <=  4102  4247  4252  4412
EAD1  4014  <=  4007
EAD2  4072  <=  4065
EADD  4001  <=  4374
ECM1  4712  <=  4706
ECM2  4732  <=  4726
ECMP  4664  <=  4775
EDIV  4377
EDV1  4451  <=  4444
EDV2  4473  <=  4462
EDV3  4526  <=  4533
EDV4  4534  <=  4525
EDV5  4557  <=  4547
EDVZ  4614  <=  4407
EMPY  4237  <=  4376
ENEG  4634  <=  4774
EOVF  4334  <=  4124  4127
EPAK  4103  <=  4333  4613
EPK1  4224  <=  4054
EPK2  4115  <=  4112
EPK3  4130  <=  4143  4233  4236
EPK4  4142  <=  4040
ESUB  4000  <=  4375
EXCH  4026  <=  4017  4022
GETN  4160
IRD   2400  <=  4740
NNEG  4051  <=  4042  4046
NXCH  4034  <=  4020  4023  4025
```

PSBA 4144 <= 4244 4404

PSHM 3425 <= 4001 4003 4237 4241 4377 4401 4634 4636 4640 4664 4666

PUC1 3420 <= 4004 4242 4402 4641 4667

PULC 4152 <= 4006

RIP 4737 <= 4313 4521

ROND 4107 <= 4103

STOV 2767 <= 4151

STUN 2772 <= 4155

TRP7 2765 <= 4776 4777

TRPE 2757 <= 4357 4633

URAK 4210 <= 4041 4253 4413

<<<<<<<<<RBUS>>>>>>>>>

	0017	280	MREG	0003	0	PADD	0004	0	PL	0000	0	RA	0013	40	RB	0012	17
RBUS	0005	8	RC	0011	23	RD	0010	14	SP0	0015	43	SP1	0014	21	SR	0001	0
UBUS	0016	26	X	0006	0	XC	0007	0	Z	0002	1						

<<<<<<<<<SBUS>>>>>>>>>

	0037	165	CC	0027	0	CIR	0000	0	CPX1	0004	0	CPX2	0006	0	CTRH	0015	1
CTRL	0014	1	DB	0022	16	DL	0034	6	IOA	0011	0	IOD	0012	0	MASK	0013	0
MOD	0005	0	OPND	0026	38	P	0020	0	PADD	0002	0	PB	0036	0	Q	0021	0
QDWN	0010	0	RA	0033	20	RB	0032	20	RC	0031	15	RD	0030	32	SBUS	0017	17
SM	0023	18	SP1	0001	21	SP2	0035	23	SP3	0025	35	STA	0024	2	SWCH	0007	0
UBUS	0016	43															

<<<<<<<FUNCTION>>>>>>>

ADD	0037	199	ADDO	0033	0	AND	0007	0	BNDT	0015	12	CAD	0016	23	CADO	0012	0
CAND	0005	0	CRS	0032	0	CTSD	0027	0	CTSS	0034	0	DVSB	0010	3	INC	0035	41
INCO	0031	0	IOR	0026	23	JMP	0014	50	JSB	0004	22	MPAD	0030	13	PNLR	0020	0
PNLS	0021	0	REPC	0024	3	REPN	0025	22	ROM	0023	14	ROMI	0022	8	ROMN	0003	8
ROMX	0002	0	RPTY	0036	0	SUB	0017	21	SUBO	0013	0	TASL	0000	4	TASR	0001	5
UBNT	0011	0	XOR	0006	2												

<<<<<<<<<STORE>>>>>>>>>

	0037	208	BSP0	0005	18	BUSH	0006	6	BUSL	0004	20	CTRH	0017	1	CTRL	0016	0
DATA	0007	14	DB	0023	0	DL	0034	0	IOA	0001	0	IOD	0002	0	MASK	0000	0
MREG	0003	0	P	0020	0	PB	0036	0	PL	0011	0	PUSH	0010	0	Q	0021	0
QUP	0013	0	RA	0033	44	RAR	0027	0	RB	0032	15	RC	0031	32	RD	0030	23
SM	0022	4	SP0	0015	24	SP1	0014	24	SP2	0035	10	SP3	0025	30	STA	0024	0
X	0026	0	Z	0012	0												

<<<<<<<<<SKIP>>>>>>>>>

	0037	221	BIT6	0005	2	BIT8	0006	2	CRRY	0010	7	CTRM	0033	23	EVEN	0002	0
F1	0014	9	F2	0016	0	F3	0034	0	INDR	0024	0	JLUI	0031	0	NCRY	0011	17
NEG	0013	5	NEXT	0035	9	NF1	0015	8	NF2	0017	10	NOFL	0007	0	NPRV	0026	1
NSME	0004	0	NZRO	0001	18	ODD	0003	0	POS	0012	4	RSB	0030	2	SR4	0022	5
SRL2	0025	9	SRL3	0027	5	SRN4	0023	0	SRNZ	0021	0	SRZ	0020	1	TEST	0032	2
UNC	0036	60	ZERO	0000	3												

<<<<<<<<SPECIAL>>>>>>>

	0037	298	CCA	0036	6	CCB	0000	0	CCE	0035	2	CCG	0034	2	CCL	0033	1
CCPX	0001	0	CCRY	0025	0	CCZ	0032	3	CF1	0022	2	CF2	0021	11	CF3	0007	0
CLIB	0016	0	CLO	0031	3	CTF	0006	21	DCSR	0011	0	FHB	0015	4	HALT	0003	0
HRF	0014	5	INCN	0012	0	INCT	0013	25	INSR	0010	0	LBF	0017	0	POP	0027	1
POPA	0026	0	SCRY	0024	0	SDFG	0005	0	SF1	0023	2	SF2	0020	8	SIFG	0004	0
SOV	0030	2	SR0	0002	3												

<<<<<<<<<MCU>>>>>>>>>>

CMD	0030	0	CRL	0031	0	CWA	0036	14	NIR	0032	0	OPND	0034	6	RNWA	0035	0
RWA	0037	24	RWAN	0024	0	RWN	0033	2									

<<<<<<<<<SHIFT>>>>>>>>

	0007	366	LLZ	0001	0	LRZ	0000	0	RLZ	0005	0	ROT	0006	0	RRZ	0004	0
SL1	0002	14	SR1	0003	21												

HEWLETT hp PACKARD

CERTIFICATION

The Hewlett-Packard Company certifies that this instrument was thoroughly tested and inspected and found to meet its published specifications when it was shipped from the factory. The Hewlett-Packard Company further certifies that its calibration measurements are traceable to the U.S. National Bureau of Standards to the extent allowed by the Bureau's calibration facility.

HEWLETT hp PACKARD

MANUAL PART NO. 30011-90007
MICROFICHE PART NO. 30011-90008

PRINTED IN U.S.A.