```
                *       IBM-1130 INSTRUCTION SET EMULATOR.  I/O IS SIMULATED IN 620 MODE.
                *
                *       WCS ENTRIES: FIRST 31 CELLS OF PAGE.
                *
                *       ENTER AT 1 WITH THE A REGISTER (R 0) = 1130 EMULATED CORE BASE
                *       ADDRESS FOR LOAD MODE AND THE B REGISTER (R 1) = 1130 WRAPAROUND
                *       MASK.  THE EMULATOR WILL RETURN TO THE CELL FOLLOWING THE LOAD
                *       MODE CALL FOR ALL 620/F MODE SERVICES.  THE X REGISTER (R 2)
                *       WILL BE SET TO INDICATE THE NECESSARY ACTION.
                *           X = 0 -> AN 1130 WAIT INSTRUCTION WAS DECODED.
                *           X = 1 -> AN 1130 XIO INSTRUCTION WAS DECODED;  THE B REGISTER
                *                CONTAINS THE EMULATED ADDRESS FROM THE XIO INSTRUCTION.
                *           X = 2 -> AN INTERRUPT WAS DETECTED IN 1130 MODE AND 620/F
                *                MODE WAS ENTERED TO PROCESS THE INTERRUPT.  THE 620/F P
                *                REGISTER WAS SET TO POINT TO THE CELL AFTER THE LOAD MODE
                *                ENTRY, SO THAT 1130 EMULATOR 620 MODE SERVICE COULD SEND
                *                CONTROL BACK TO THE WCS.
                *           X = 3 -> AN ENTRY WAS MADE TO AN IMPROPER LOCATION (0 OR
                *                5 THROUGH 31) OF WCS PAGE 1.
                *           X = 4 -> A BOSC INSTRUCTION WITH BIT 6 SET WAS EXECUTED
                *                BY THE 1130 PROGRAM.  THIS INDICATES THAT THE INTERRUPT
                *                LEVEL IS TO BE RAISED.
                *           X = 5 -> THE STEP KEY WAS PUSHED.
                *
                *       ENTER AT 2 TO CONTINUE EMULATION AFTER A HALT, I/O INSTRUCTION
                *       OR INTERRUPT.  THE A REGISTER (R 0) MUST BE RESET TO THE BASE OF
                *       1130 EMULATED CORE.
                *
                *       ENTER AT 3 TO CONTINUE EMULATION AFTER A SENSE INSTRUCTION WITH
                *       B (R 1) EQUAL TO THE DATA TO BE PUT INTO THE 1130 ACCUMULATOR.
                *       THE A REGISTER (R 0) MUST BE RESET TO THE BASE ADDRESS OF
                *       EMULATED 1130 CORE.
                *
                *       ENTER AT 4 WITH B (R 1) EQUAL TO THE 1130 INTERRUPT LEVEL TO
                *       INTERRUPT THE 1130 PROGRAM.  THE A REGISTER (R 0) MUST BE RESET
                *       TO THE BASE ADDRESS OF 1130 EMULATED CORE.
                *
                *       ENTER AT 5 TO PUT 1130 A REGISTER INTO 620 A REGISTER.
                *
                *       ENTER AT 6 WITH A (R 0) = 1130 BASE ADDRESS, B (R 1) = CHARACTER
                *       AND X (R 2) = BUFFER ADDRESS TO PERFORM 1132 PRINTER SCAN
                *       X (R 2) IS DESTROYED.
                *
                *       ENTER AT X'C TO PERFORM MEMORY - TO - MEMORY BLOCK MOVE.
                *       CALLING SEQUENCE:
                *       BCS     PAGE1+C
                *       PZE/MZE FROM            MZE - ALL FROM ONE WORD
                *       PZE/MZE TO              MZE - ROTATED LEFT 4 BITS
                *       DATA    LENGTH
                *       RETURN HERE WITH A (R 0) = ZERO.
                *
                *       ENTER AT D TO SEARCH A TRANSLATE TABLE WITH A (R 0) = WORD TO
                *       SEARCH FOR, B (R 1) = TABLE ADDRESS.  THE FIRST WORD OF THE TABLE
                *       IS EXPECTED TO BE THE TABLE LENGTH.  AFTER THE SEARCH, B (R 1)
                *       IS SET TO THE RELATIVE LOCATION IN THE TABLE OF THE WORD (IF
                *       THE WORD IS FOUND), OR IS SET EQUAL TO THE TABLE LENGTH+1 IF
                *       THE WORD IS NOT FOUND.
```

```
                    59  *       REGISTERS USED BY THE 1130 EMULATOR:                                    059
                    60  *           0 = 1130 MEMORY BASE ADDRESS (SUPPLIED ON ALL ENTRIES).             060
                    61  *               NOTE:  R0 MUST BE THE SAME ON ALL ENTRIES (UNLESS THE           061
                    62  *               EMULATED CORE AREA IS RELOCATED), AND MUST BE EVEN UNDER         062
                    63  *               ALL CONDITIONS FOR THE DOUBLE-WORD REFERENCE INSTRUCTIONS        063
                    64  *               TO WORK PROPERLY.                                               064
                    65  *           1 = EMULATED 1130 ADDRESS: 1130 ADDRESS + R(0).                     065
                    66  *               NOTE:  R1 IS NOT ALWAYS THE 1130 EMULATED ADDRESS, BUT IT       066
                    67  *               IS ALWAYS SO WHEN IN 620/F MODE.                                067
                    68  *           3 = ALL ZEROES, AS IN 620 EMULATION.                                068
                    69  *           5 = ALL ONES, AS IN 620 EMULATION.                                  069
                    70  *           8 = STATUS REGISTER SAVE.                                           070
                    71  *           9 = 1130 WRAPAROUND MASK SUPPLIED BY THE 620 B REGISTER (R 1)       071
                    72  *               ON THE LOAD MODE ENTRY.                                         072
                    73  *           A = 1130 ACCUMULATOR.                                               073
                    74  *           B = 1130 EXTENSION.                                                 074
                    75  *           C = P SAVE FOR 620/F MODE.                                          075
                    76  *           D = P SAVE FOR 1130 MODE DURING 620/F MODE SUPPORT.                 076
0000                77  BASE    EQU     0    1130 BASE ADDRESS                                         077
0001                78  ADDR    EQU     1    EMULATED 1130 ADDRESS                                     078
0008                79  STAT    EQU     8    STATUS REGISTER SAVE AREA                                 079
0009                80  WRAP    EQU     9    1130 WRAPAROUND MASDK                                     080
000A                81  ACC     EQU     X'A 1130 ACCUMULATOR                                          081
000B                82  EXT     EQU     X'B 1130 EXTENSION                                            082
000C                83  P620    EQU     X'C P SAVE FOR 620/F MODE                                     083
000D                84  P1130   EQU     X'D P SAVE FOR 1130 MODE DURING 620/F MODE SUPPORT             084
000E                85  W1      EQU     X'E USED FREELY AS WORK REGISTERS.  HARDWARE REQUIRES         085
000F                86  W2      EQU     X'F W1=X'E AND W2=X'F FOR MULTIPLY AND DIVIDE.                086
```

```
0040   88  DJUMP  EQU     O'00100        DECODER JUMP BASE ADDRESS                          088
0200   89  PAGE1  EQU     X'200                                                             089
0400   90  FFP    EQU     X'400          FORTRAN FIRMWARE PAGE.                             090
00D1   91  VINT1  EQU     X'D1           FIRST STEP OF 620/F INTERRUPT SERVICE.             091
00B4   92  VSS2M  EQU     X'B4           620 MODE STANDARD STATE SS2.                       092
002D   93  VSS3M  EQU     X'2D           620 MODE STANDARD STATE SS3M.                      093
       94  *                                                                               094
       95  *      CONDITION TEST ROUTINE CALL MACRO                                        095
       96  *                                                                               096
       97  CTCALL MAC                                                                     097
       98         GMSK    /F(CTEST5),2(1),FS4,GF2,LB3,FF7,        FS(0),                  098
       99               CMK0100,AK8                               R8.X'100,SAMPLE         099
      100         EMAC                                                                    100
      101  *                                                                             101
      102  *      GET TAG BITS TO OPERAND REGISTER, SAMPLE ALU.                           102
      103  *                                                                             103
      104  GETTAG MAC                                                                    104
      105         GMSK    /N(P(1)),GF2,LB2,FFA,RF3,MKFCFF   TAG -> OPR, SAMPLE ALU       105
      106         EMAC                                                                    106
      107  *                                                                             107
      108  *      GET TAG BITS TO OPERAND REGISTER, SAMPLE ALU, IF(P), INCP                108
      109  *                                                                             109
      110  GETTGL MAC                                                                    110
      111         GMSK    /N(P(1)),SF2,GF2,IM8,LB2,FFA,RF7,MKFCFF                         111
      112         EMAC                                                                    112
      113  *                                                                             113
      114  *      ORIGIN AT NEXT MULTIPLE OF P(1).                                         114
      115  *                                                                             115
      116  MORG   MAC                                                                    116
      117         ORG     *+P(1)-1                                                        117
      118         ORG     */P(1)*P(1)                                                     118
      119         EMAC                                                                    119
      120  *      SPECIAL CASE OF MORG.                                                    120
      121  EVEN   MAC                                                                    121
      122         MORG    2                       ORIGIN AT NEXT EVEN LOCATION.           122
      123         EMAC                                                                    123
      124  *                                                                             124
      125  *      ILLEGAL INSTRUCTIONS ALL BECOME WAITS.                                   125
      126  *                                                                             126
      127  WAITM  MAC                                                                    127
      128         GEN     /N(E620M),FF3,MF1,WR1,AA2          ZERO -> 620 X.           128
      129         EMAC                                                                    129
```

```
                          131  *       BCS ENTRIES.                                                                  131
                          132  *                                                                                     132
0000 0570000063AFFFC0     133          GMSK    /N(E620MV),LB3,RF3,FFA,MKFFFC     3 -> OPERAND REGISTER         133
                          134  ************************************************************************************ 134
                          135  *                                                                                  *  135
                          136  *       1130 LOAD MODE ENTRY POINT.                                                *  136
                          137  *                                                                                  *  137
0001 199000018801000C     138  LMODE1 GEN     /P(LMODE2+PAGE1),IM3,LA1,WR1,AAC  620(P)->RC                      *  138
                          139  ************************************************************************************ 139
                          140  *                                                                                  *  140
                          141  *       ENTRY POINT TO CONTINUE 1130 PROGRAM.                                      *  141
                          142  *                                                                                  *  142
0002 1848090201900000D    143  RETURN GEN     /P(ENTR+PAGE1),SF2,GF4,IM4,RF1,FF9,AAD R0+RD->P,IF(ALU)           *  143
                          144  ************************************************************************************ 144
                          145  *                                                                                  *  145
                          146  *       ENTRY POINT TO CONTINUE THE 1130 PROGRAM WITH NEW ACCUMULATOR               *  146
                          147  *                                                                                  *  147
0003 0010000000A9001A     148  RETRNA GEN     /N(RETURN),FFA,MF1,WR1,BB1,AAA          R1->RA                    *  148
                          149  ************************************************************************************ 149
                          150  *                                                                                  *  150
                          151  *       ENTRY POINT TO INTERRUPT 1130 TO LEVEL IN 620 B.                           *  151
                          152  *                                                                                  *  152
0004 1870000180910001     153  INTEN  GEN     /P(INTEN0+PAGE1),IM3,FF9,WR1,AA1  R0+R1->R1                       *  153
                          154  ************************************************************************************ 154
                          155  *                                                                                  *  155
                          156  *       ENTRY POINT TO PUT 1130 A REGISTER INTO 620 A REGISTER.                    *  156
                          157  *                                                                                  *  157
                          158  GETREG GEN     /P(VSS3M),SF2,GF4,IM8,RF4,FFA,    RA->R0, INCP, IF(P)            *  158
0005 0168090404A900A0     159                 CMF1,WR1,BBA,AA0                                                  *  159
                          160  ************************************************************************************ 160
                          161  *                                                                                  *  161
                          162  *       ENTRY POINT FOR PRINTER SCAN.                                              *  162
                          163  *                                                                                  *  163
                          164  PSCAN1 GMSK    /N(PSCAN2),SF1,IM5,LB3,RF3,FF9,   BASE+32->OPR, OF(ALU)           *  164
0006 0AB80402E39FFDF0     165                 CMKFFDF                                                           *  165
                          166  ************************************************************************************ 166
                          167  *                                                                                  *  167
                          168  *       ILLEGAL ENTRANCES.                                                         *  168
                          169  *                                                                                  *  169
0007 0570000063AFFFC0     170          GMSK    /N(E620MV),LB3,RF3,FFA,MKFFFC     3 -> OPERAND REGISTER       *  170
0008 0570000063AFFFC0     171          GMSK    /N(E620MV),LB3,RF3,FFA,MKFFFC     3 -> OPERAND REGISTER       *  171
0009 0570000063AFFFC0     172          GMSK    /N(E620MV),LB3,RF3,FFA,MKFFFC     3 -> OPERAND REGISTER       *  172
000A 0570000063AFFFC0     173          GMSK    /N(E620MV),LB3,RF3,FFA,MKFFFC     3 -> OPERAND REGISTER       *  173
000B 0570000063AFFFC0     174          GMSK    /N(E620MV),LB3,RF3,FFA,MKFFFC     3 -> OPERAND REGISTER       *  174
                          175  ************************************************************************************ 175
                          176  *                                                                                  *  176
                          177  *       ENTRY POINT FOR MEMORY - TO - MEMORY BLOCK MOVE.                           *  177
                          178  *                                                                                  *  178
000C 09A8040484000000     179  MBM    GEN     /N(MBM1),SF1,IM9,RF4                AF(P),INCP                    *  179
                          180  ************************************************************************************ 180
                          181  *                                                                                  *  181
                          182  *       ENTRY POINT FOR TABLE SEARCH.                                              *  182
                          183  *                                                                                  *  183
000D 0C7800000801000F     184  XLATE  GEN     /N(XLATE1),LA1,WR1,24(W2)           P->W2                     *  184
                          185  ************************************************************************************ 185
                          186  *                                                                                  *  186
                          187  *       ILLEGAL ENTRANCES                                                          *  187
```

```
                          188  *                                                                                *      188
000E 0570000063AFFFC0     189          GMSK     /N(E620MV),LB3,RF3,FFA,MKFFFC      3 -> OPERAND REGISTER *      189
000F 0570000063AFFFC0     190          GMSK     /N(E620MV),LB3,RF3,FFA,MKFFFC      3 -> OPERAND REGISTER *      190
0010 0570000063AFFFC0     191          GMSK     /N(E620MV),LB3,RF3,FFA,MKFFFC      3 -> OPERAND REGISTER *      191
0011 0570000063AFFFC0     192          GMSK     /N(E620MV),LB3,RF3,FFA,MKFFFC      3 -> OPERAND REGISTER *      192
0012 0570000063AFFFC0     193          GMSK     /N(E620MV),LB3,RF3,FFA,MKFFFC      3 -> OPERAND REGISTER *      193
0013 0570000063AFFFC0     194          GMSK     /N(E620MV),LB3,RF3,FFA,MKFFFC      3 -> OPERAND REGISTER *      194
0014 0570000063AFFFC0     195          GMSK     /N(E620MV),LB3,RF3,FFA,MKFFFC      3 -> OPERAND REGISTER *      195
0015 0570000063AFFFC0     196          GMSK     /N(E620MV),LB3,RF3,FFA,MKFFFC      3 -> OPERAND REGISTER *      196
0016 0570000063AFFFC0     197          GMSK     /N(E620MV),LB3,RF3,FFA,MKFFFC      3 -> OPERAND REGISTER *      197
                          198  ********************************************************************************      198
                          199  *                                                                                *      199
                          200  *       FORTRAN DO INCREMENT (1)                                                 *      200
                          201  *                                                                                *      201
0017 20B8000180000000     202          GEN      /P(FFP+X'17),IM3                   PAGE JUMP                  *      202
                          203  ********************************************************************************      203
                          204  *                                                                                *      204
                          205  *       BYTE STRING FIRMWARE (CBS, MBS)                                          *      205
                          206  *                                                                                *      206
0018 20C0000180000000     207          GEN      /P(FFP+X'18),IM3                   PAGE JUMP                  *      207
                          208  ********************************************************************************      208
                          209  *                                                                                *      209
                          210  *       STACK FIRMWARE                                                           *      210
                          211  *                                                                                *      211
0019 20C8000180000000     212          GEN      /P(FFP+X'19),IM3                   PAGE JUMP                  *      212
                          213  ********************************************************************************      213
                          214  *                                                                                *      214
                          215  *       DOUBLE LOAD                                                              *      215
                          216  *                                                                                *      216
001A 20D0000180000000     217          GEN      /P(FFP+X'1A),IM3                   PAGE JUMP                  *      217
                          218  ********************************************************************************      218
                          219  *                                                                                *      219
                          220  *       DOUBLE STORE                                                             *      220
                          221  *                                                                                *      221
001B 20D8000180000000     222          GEN      /P(FFP+X'1B),IM3                   PAGE JUMP                  *      222
                          223  ********************************************************************************      223
                          224  *                                                                                *      224
                          225  *       FLOATING POINT AND DOUBLE INTEGER ARITHMETIC                              *      225
                          226  *                                                                                *      226
001C 20E0000180000000     227          GEN      /P(FFP+X'1C),IM3                   PAGE JUMP                  *      227
                          228  ********************************************************************************      228
                          229  *                                                                                *      229
                          230  *       FORTRAN DO INCREMENT (GENERAL)                                           *      230
                          231  *                                                                                *      231
001D 20E8000180000000     232          GEN      /P(FFP+X'1D),IM3                   PAGE JUMP                  *      232
                          233  ********************************************************************************      233
                          234  *                                                                                *      234
                          235  *       UNTANGLE CALLING SEQUENCE ($FSE)                                         *      235
                          236  *                                                                                *      236
001E 20F0000180000000     237          GEN      /P(FFP+X'1E),IM3                   PAGE JUMP                  *      237
                          238  ********************************************************************************      238
                          239  *                                                                                *      239
                          240  *       DOUBLE WORD MOVE                                                         *      240
001F 20F8000180000000     241          GEN      /P(FFP+X'1F),IM3                   PAGE JUMP                  *      241
                          242  ********************************************************************************      242
```

```
                          245  *                                                                                    245
                          246  *       CONTROL REACHES HERE VIA FIELD SELECT ADDRESSING FROM ADDRESS                 246
                          247  *       COMPUTATION.                                                                 247
                          248  *                                                                                    248
                          249          MORG    32                                                                  249
                     0020 250  OP00    EQU     *                                                                   250
                          251  *       SS2MX IS IDENTICAL TO SS2M BUT IS PLACED HERE BECAUSE IT APPEARS             251
                          252  *       IN A TEST ADDRESS WITH BSC1.                                                 252
0020 0530040404000000     253  SS2MX   GEN     /N(SS3M),SF1,IM8,RF4                INCP, IF(P)                      253
0021 0CC0000063AFFFE0     254  XIO1    GMSK    /N(XIO2),LB3,RF3,FFA,MKFFFE         1 -> OPERAND REGISTER            254
0025                      255          ORG     XIO1+4                                                              255
                          256  STS7    GMSK    /N(STS8),SF1,GF4,IM3,LB3,FFB,       OVR(BS), RE.3->ALU           256
0025 0B900501E0BFFFCE     257                  CMKFFFC,AKE                         RESET OVERFLOW                   257
0028                      258          ORG     STS7+3                                                              258
0028 0790040108660000     259  BSI1    GEN     /N(BSI2),SF1,IM2,LA1,FF6,CF3        OVR(OS), P-R0 -> ALU             259
0029 0541C00001000001     260  BSC1    GEN     /F(SS1M),2(8),FS7,RF1,AA1           FS(6),R1->P                      260
0030                      261          ORG     BSC1+7                                                              261
0030 0740000080000000     262  A1      GEN     /N(A2),IM1                          WAIT FOR OPERAND                 262
0031 0748008060BFFFE1     263  AD1     GMSK    /N(ADSD2),GF2,LB3,FFB,MKFFFE,AK1    R1.1, SAMPLE ALU                 263
0032 0B100000E3AF7FF0     264  S1      GMSK    /N(S2),IM1,LB3,RF3,FFA,MKF7FF       WAIT(M),X'800->OPR               264
0033 0748008060BFFFE1     265  SD1     GMSK    /N(ADSD2),GF2,LB3,FFB,MKFFFE,AK1    R1.1, SAMPLE ALU                 265
0034 0A6000000300400A     266  M1      GEN     /N(MUL2),RF3,VF1,AAA                A->OPR,SIGN->DSB                 266
0035 07A00000E3AFEFF0     267  D1      GMSK    /N(DIV1),IM1,LB3,RF3,FFA,MKFEFF     X'100->OPR, WAIT(MEM)            267
0038                      268          ORG     D1+3                                                                268
0038 0898040404000000     269  LDA1    GEN     /N(LDA2),SF1,IM8,RF4                IF(P),INCP                       269
0039 08A0008060BFFFE1     270  LDD1    GMSK    /N(LDD2),GF2,LB3,FFB,MKFFFE,AK1     R1.1, SAMPLE                     270
003A 0B4004010000000A     271  STO1    GEN     /N(STO2),SF1,IM2,AAA                OS(OVR),HOLD RA FOR OS           271
003B 0B2804010000000A     272  STD1    GEN     /N(STD2),SF1,IM2,AAA                OS(OVR),HOLD RA FOR OS           272
003C 0788040404000000     273  AND1    GEN     /N(AND2),SF1,IM8,RF4                IF(P), INCP                      273
003D 0AB0040404000000     274  OR1     GEN     /N(OR2),SF1,IM8,RF4                 IF(P), INCP                      274
003E 0868040404000000     275  EOR1    GEN     /N(EOR2),SF1,IM8,RF4                IF(P), INCP                      275
```

```
                         277  *
                         278  *       DECODER JUMPS TO THE NEXT 64 WORDS, EXCEPT THOSE INDICATED.
                         279  *
0040                     280          ORG     DJUMP
                         281  OP00S   WAITM                           OP = 0
0040 0560000000390002
                         282  OP00L   WAITM                           OP = 0
0041 0560000000390002
                         283  XIOS    GETTAG  SHORT                   EXECUTE I/O SHORT
0042 04C8008043AFCFF0
                         284  XIOL    GETTGL  LONG                    EXECUTE I/O LONG
0043 04A0088447AFCFF0
                         285  SHIFT   GETTAG  SHIFT1                  SHIFT INSTRUCTIONS.  TAG -> OPR.
0044 0630008043AFCFF0
                         286  *       THE NEXT 3 WORDS ARE NOT ACCESSED BY DECODER ADDRESSING.
0048                     287          ORG     SHIFT+4
0048 5889400000000000    288  LDSS    GEN     /F(LDSRC),2(1),FS5      LOAD STATUS SHORT   FSEL(1)
0049 5889400000000000    289  LDSL    GEN     /F(LDSRC),2(1),FS5      LOAD STATUS LONG    FSEL(1)
004A 0B4800000300000B    290  STSS    GEN     /N(STS1),RF3,AA8        STORE STATUS SHORT  R8 -> OPR
004B 0B4800000300000B    291  STSL    GEN     /N(STS1),RF3,AA8        STORE STATUS LONG   R8 -> OPR
                         292  WAIT67  WAITM                           OP = 6 OR 7 (ILLEGAL)
004C 0560000000390002
                         293  *       THE NEXT 3 WORDS ARE NOT ACCESSED BY DECODER ADDRESSING.
0050                     294          ORG     WAIT67+4
                         295  BSIS    GETTAG  SHORT                   BRANCH & STORE IAR SHORT
0050 04C8008043AFCFF0
                         296  BSIL    CTCALL                          BSI LONG
0051 1409008060701008
                         297  BSCS    CTCALL                          BSC SHORT
0052 1409008060701008
                         298  BSCL    CTCALL                          BSC LONG
0053 1409008060701008
                         299  WAITAB  WAITM                           OP = A OR B (ILLEGAL)
0054 0560000000390002
                         300  *       THE NEXT 3 WORDS ARE NOT ACCESSED BY DECODER ADDRESSING.
0058                     301          ORG     WAITAB+4
                         302  LDXS    GETTAG  LDXS1                   LOAD INDEX SHORT.
0058 0938008043AFCFF0
                         303  LDXL    GETTAG  LDXL1                   LOAD INDEX LONG.
0059 0958008043AFCFF0
                         304  STXS    GETTAG  STXS1                   STORE INDEX SHORT.
005A 0C28008043AFCFF0
                         305  STXL    GETTGL  STXL1                   STORE INDEX LONG.
005B 0B98088447AFCFF0
                         306  MDXS    GETTAG  MDX                     MDX SHORT
005C 05B8008043AFCFF0
                         307  MDXL    GETTAG  MDX                     MDX LONG
005D 05B8008043AFCFF0
                         308  WAITFS  WAITM                           OP = F (ILLEGAL) SHORT
005E 0560000000390002
                         309  WAITFL  WAITM                           OP = F (ILLEGAL) LONG
005F 0560000000390002
                         310  AS      GETTAG  SHORT                   ADD SHORT
0060 04C8008043AFCFF0
                         311  AL      GETTGL  LONG                    ADD LONG
0061 04A0088447AFCFF0
                         312  ADS     GETTAG  SHORT                   ADD DOUBLE SHORT
```

```
0062 04C8008043AFCFF0
                     313  ADL     GETTGL  LONG                    ADD DOUBLE LONG                  313
0063 04A0088447AFCFF0
                     314  SS      GETTAG  SHORT                   SUBTRACT SHORT                   314
0064 04C8008043AFCFF0
                     315  SL      GETTGL  LONG                    SUBTRACT LONG                    315
0065 04A0088447AFCFF0
                     316  SDS     GETTAG  SHORT                   SUBTRACT DOUBLE SHORT            316
0066 04C8008043AFCFF0
                     317  SDL     GETTGL  LONG                    SUBTRACT DOUBLE LONG             317
0067 04A0088447AFCFF0
                     318  MS      GETTAG  SHORT                   MULTIPLY SHORT                   318
0068 04C8008043AFCFF0
                     319  ML      GETTGL  LONG                    MULTIPLY LONG                    319
0069 04A0088447AFCFF0
                     320  DS      GETTAG  SHORT                   DIVIDE SHORT                     320
006A 04C8008043AFCFF0
                     321  DL      GETTGL  LONG                    DIVIDE LONG                      321
006B 04A0088447AFCFF0
                     322  WT1617  WAITM                           OP = 16 OR 17 (ILLEGAL)          322
006C 0560000000390002
                     323  *       THE NEXT 3 WORDS ARE NOT ACCESSED BY DECODER ADDRESSING.         323
0070                 324          ORG     WT1617+4                                                 324
                     325  LDAS    GETTAG  SHORT                   LOAD ACCUMULATOR SHORT           325
0070 04C8008043AFCFF0
                     326  LDAL    GETTGL  LONG                    LOAD ACCUMULATOR LONG            326
0071 04A0088447AFCFF0
                     327  LDDS    GETTAG  SHORT                   LOAD DOUBLE SHORT                327
0072 04C8008043AFCFF0
                     328  LDDL    GETTGL  LONG                    LOAD DOUBLE LONG                 328
0073 04A0088447AFCFF0
                     329  STOS    GETTAG  SHORT                   STORE SHORT                      329
0074 04C8008043AFCFF0
                     330  STOL    GETTGL  LONG                    STORE LONG                       330
0075 04A0088447AFCFF0
                     331  STDS    GETTAG  SHORT                   STORE DOUBLE SHORT               331
0076 04C8008043AFCFF0
                     332  STDL    GETTGL  LONG                    STORE DOUBLE LONG                332
0077 04A0088447AFCFF0
                     333  ANDS    GETTAG  SHORT                   AND SHORT                        333
0078 04C8008043AFCFF0
                     334  ANDL    GETTGL  LONG                    AND LONG                         334
0079 04A0088447AFCFF0
                     335  ORS     GETTAG  SHORT                   OR SHORT                         335
007A 04C8008043AFCFF0
                     336  ORL     GETTGL  LONG                    OR LONG                          336
007B 04A0088447AFCFF0
                     337  EORS    GETTAG  SHORT                   EXCLUSIVE OR SHORT               337
007C 04C8008043AFCFF0
                     338  EORL    GETTGL  LONG                    EXCLUSIVE OR LONG                338
007D 04A0088447AFCFF0
                     339  WT1FS   WAITM                           OP = 1F (ILLEGAL) SHORT          339
007E 0560000000390002
                     340  WT1FL   WAITM                           OP = 1F (ILLEGAL) LONG           340
007F 0560000000390002
```

```
                         342  *       CONDITION TEST ROUTINE.  ON EXIT, DSB=1 IF ANY SELECTED TEST          342
                         343  *       PASSES, AND DSB=0 IF ALL SELECTED TESTS FAIL.                         343
                         344  *                                                                             344
                         345          MORG    16                                                           345
0080 0148228008070001    346  CBS     GEN     /T(BSC1,SS2MX),TF2,GFA,LA1,CF3,WR1,AA1 P+1->R1,DSB=>BSC1     346
0081 25483E8404000000    347  CBL     GEN     /T(CBL1,SS2M),TF3,SF3,GFA,IM8,RF4      INCP,DSB=0=>CBL1;    347
                         348  *                                                      DSB=>IF(P),SS2M       348
                         349          EVEN                                                                 349
0082 240A40800000000A    350  CTEST5  GEN     /F(CTEST4),2(1),FS9,GF2,AAA            FS(5),A->ALU,SAMP     350
                         351  CTST0A  GMSK    /N(CTST0B),SF1,GF4,LB3,RF3,FFB,        RESET OVERFLOW,       351
0083 0470050063B01008    352                 CMK0100,AK8                            R8&FEFF->OPR          352
                         353          EVEN                                                                 353
0084 340A000000004005    354  CTEST4  GEN     /F(CTEST3),2(1),FS8,VF1,AA5            1->DSB,FS(4)          354
                         355  CTST5A  GEN     /S(CBS,CTEST4),2(1),FSE,TF2,GF9,       1->DSB,               355
0085 240BA24000004005    356                 CVF1,AA5                               NOT ALUZ => CTEST4    356
                         357          EVEN                                                                 357
0086 4409C00000000000    358  CTEST3  GEN     /F(CTEST2),2(1),FS7                    FS(3)                 358
0087 340BA1C000000000    359  CTST4A  GEN     /S(CBS,CTEST3),2(1),FSE,TF2,GF7        NOT ALUS=>CTEST3      359
                         360          EVEN                                                                 360
                         361  CTEST2  GMSK    /F(CTEST1),2(1),FS6,GF2,LB3,FF7,       FS(2),                361
0088 540980806070001A    362                 CMK0001,AKA                            RA.1, SAMPLE          362
0089 4478324000000000    363  CTST3A  GEN     /T(CTST3B,CTEST2),TF3,GF9              NOT ALUZ=>CTST3B      363
                         364          EVEN                                                                 364
                         365  CTEST1  GMSK    /F(CTESTF),2(1),FS5,GF2,LB3,FF7,       FS(1),                365
008A 6409408060708008    366                 CMK0800,AK8                            R8.X'800,SAMPLE       366
008B 540BA24000000000    367  CTST2A  GEN     /S(CBS,CTEST1),2(1),FSE,TF2,GF9        A ODD => CTEST1       367
                         368          EVEN                                                                 368
008C 040B800000004003    369  CTESTF  GEN     /F(CBS),2(1),FSE,VF1,AA3               FS(10), 0->DSB        369
008D 640BA24000000000    370  CTST1A  GEN     /S(CBS,CTESTF),2(1),FSE,TF2,GF9        CARRY => CTEST0       370
                         371  CTST0B  GEN     /T(CTST5A,CTEST5),TF2,GF9,             OPR->R8,              371
008E 1428224020A90008    372                 CLB1,FFA,MF1,WR1,AA8                   OVFL => CTEST5        372
008F 440BB1C000000000    373  CTST3B  GEN     /S(CBS,CTEST2),2(1),FSE,TF3,GF7        ALUS => CTEST2        373
```

```
                        375  *       ADDRESS COMPUTATION SUBROUTINES                                          375
                        376  *                                                                                376
                        377  *                                                                                377
                        378  *       COMPUTE EFFECTIVE ADDRESS AND FETCH OPERAND,  LONG FORMAT                 378
                        379  *                                                                                379
                        380          MORG    16                              ADDREX, LNGNXI REF BY CND FSEL   380
                        381  ADDREX  GEN      /F(OP00),2(X'F),MT1,FSF,SF1,IM5,          R1+R0->R1, OF(ALU)   381
0090 017FC40280910001   382                   CFF9,WR1,AA1                                                     382
0091 0490040280910001   383  LNGNXI  GEN      /N(LGNXI1),SF1,IM5,FF9,WR1,AA1            R1+R0->R1, OF(ALU)   383
0092 0498000080A90091   384  LGNXI1  GEN      /N(LGNXI2),IM1,FFA,MF1,WR1,BB9,AA1        WAIT(MEM), R9->R1    384
0093 0480000020B90011   385  LGNXI2  GEN      /N(ADDREX),LB1,FFB,MF1,WR1,BB1,AA1        MIR.WRAPAROUND->R1   385
0094 04A8000020A90011   386  LONG    GEN      /N(LONG1),LB1,FFA,MF1,WR1,BB1,AA1         MIR(=Y) -> R1        386
                        387  LONG1   GEN      /S(ADDREX,LONGX),2(1),FSB,TF2,GF9,         WRAPAROUND           387
0095 348AE24000B90091   388                   CFFB,MF1,WR1,BB9,AA1                                             388
                        389          EVEN                                                                     389
0096 04B80402A0900050   390  LONGX   GEN      /N(LONGX1),SF1,IM5,LB1,FF9,BB5            R0+OLSE, OF(ALU)     390
0097 04C0008083000100   391  LONGX1  GEN      /N(LONGX2),GF2,IM1,RF3,SH1                WAIT(MEM), SAMPLE,   391
                        392  *                                                          0->OPR.              392
0098 04A8000020910011   393  LONGX2  GEN      /N(LONG1),LB1,FF9,WR1,BB1,AA1             R1+MIR->R1           393
                        394  *                                                                               394
                        395  *       COMPUTE EFFECTIVE ADDRESS AND FETCH OPERAND, SHORT FORMAT                 395
                        396  *                                                                                396
0099 04D0000043AFC000   397  SHORT   GMSK     /N(SHORT1),LB2,RF3,FFA,MKFC00             TAG & DISP -> OPR    397
                        398  SHORT1  GEN      /T(SHORTP,SHORTX),TF2,SF2,GF9,IM5,         TEST ALUZ:           398
009A 64D82A42A0900050   399                   CLB1,FF9,BB5                              T => SHORTP;         399
                        400  *                 F => OF(ALU = TAG+BASE), SHORTX                                 400
009B 04F0000008670001   401  SHORTP  GEN      /N(SHRTP1),LA1,FF6,CF3,WR1,AA1            P-R0 -> R1           401
                        402          EVEN                                                                     402
009C 04E80000A0A90041   403  SHORTX  GEN      /N(SHRTX1),IM1,LB1,FFA,MF1,WR1,BB4,AA1    WAIT(MEM),ORSE->R1   403
009D 04F8000020910011   404  SHRTX1  GEN      /N(SHRTR1),LB1,FF9,WR1,BB1,AA1            R1+MIR->R1           404
009E 04F8000020910041   405  SHRTP1  GEN      /N(SHRTR1),LB1,FF9,WR1,BB4,AA1            R1+ORSE->R1          405
009F 0480000000B90091   406  SHRTR1  GEN      /N(ADDREX),FFB,MF1,WR1,BB9,AA1            WRAPAROUND           406
```

```
                         408  *       INTERRUPT SERVICE.                                                                  408
                         409  *                                                                                           409
                         410  *       THE STATES WHICH SELECT THE DECODER ALSO ENABLE INTERRUPTS.  IF                      410
                         411  *       AN INTERRUPT IS DETECTED, AND AN INTERRUPT IS NOT BEING PROCESSED                    411
                         412  *       (CINTF=0), CONTROL GOES TO INTBAS+7.                                                412
                         413  *       AT INTBAS+7, THE I/O CONTROL STORE MICROPROGRAM IS INITIATED,                        413
                         414  *       AND THE INTERRUPT IS CHECKED AGAIN.  IF THE INTERRUPT SERVICE                       414
                         415  *       CAN PROCEED, CONTROL GOES TO INTBAS+1.  IF THE INTERRUPT IS                         415
                         416  *       SUPRESSED BY DMA ACTIVITY, CONTROL GOES TO INTBAS AND THE                           416
                         417  *       INTERRUPT IS TRIED AGAIN AFTER THE NEXT INSTRUCTION.                                417
                         418  *                                                                                           418
                         419          MORG    16                                                                         419
                   00A0  420  INTBAS EQU     *                                                                          420
00A0 0520040400000000    421  SS1M    GEN     /N(SS2M),SF1,IM8                        IF(P)                              421
00A1 0510000000A900CE    422  INT1    GEN     /N(INT2),FFA,MF1,WR1,BBC,AAE            RC->RE                             422
00A2 0518000000670000    423  INT2    GEN     /N(INT3),FF6,CF3,WR1,BB0,AAD            RD-R0->RD                          423
00A3 0558000063AFFFD0    424  INT3    GMSK    /N(E620I),RF3,LB3,FFA,MKFFFD            2->OPERAND                         424
                         425  *       STANDARD STATE INSTRUCTIONS.                                                       425
                         426  *       SS2M AND SS3M MUST HAVE THE SAME 5 HIGH ORDER BITS:  THEY                          426
                         427  *       APPEAR TOGETHER IN A TEST ADDRESSING MICRO.                                        427
                         428          EVEN                                                                               428
00A4 0530040424A80000    429  SS2M    GEN     /N(SS3M),SF1,IM8,LB1,RF4,FFA,MF1        INCP,IF(P),OPR->                   429
                         430  *       SS2MS DOES:  STATUS -> R(8), IF(P), INCP                                           430
00A5 0530040424A90038    431  SS2MS   GEN     /N(SS3M),SF1,IM8,LB1,RF4,FFA,MF1,WR1,BB3,AA8                       431
                         432  *       SS3M DOES:  SELECT AND RESET INTERRUPT FLAG, ENABLE INTERRUPTS,                     432
                         433  *                   IBR -> I, DECODE.                                                      433
                         434          EVEN                                                                               434
00A6 F500014300000000    435  SS3M    GEN     /N(INTBAS),1(X'F),GF5,IM6                                          435
00A7                     436          ORG     INTBAS+7                                                           436
                         437  IWAIT   GEN     2(SS1M),1(7),MT1,GF4,MR1,IME,           EN INT, DIS DEC,           437
00A7 7504012709F1000D    438                  CLA1,RF1,FFF,WR1,AAD                    IO STRT, P-1->P,RD         438
                         439  *       BSC2 IS PLACED HERE BECAUSE IT IS REFERENCED WITH SS1M USING                       439
                         440  *       FIELD SELECT ADDRESSING.                                                           440
00A8                     441          ORG     SS1M+8                                                                     441
00A8 0570000063AFFFB0    442  BSC2    GMSK    /N(E620MV),LB3,RF3,FFA,MKFFFB           4 -> OPR                          442
                         443  *       CBL1 IS PLACED HERE BECAUSE IT APPEARS IN A TEST WITH SS2M                          443
00A9 04A00080C3AFCFF0    444  CBL1    GMSK    /N(LONG),GF2,IM1,LB2,FFA,RF3,MKFCFF     WAIT(MEM),SAMPLE,                 444
                         445  *                                                       IR.X'0300->OPR                     445
                         446  *       STDEVN IS PLACED HERE BECAUSE IT APPEARS IN A TEST WITH SS2M                        446
00AA 0530040404000008    447  STDEVN  GEN     /N(SS3M),SF1,IM8,RF4,AAB                INCP,IF(P),B->ALU                  447
                         448  *       STATES E620(XX) ENTER 620 MODE.                                                    448
00AB 06880001A0A90002    449  E620I   GEN     /P(VINT1),IM3,LB1,FFA,MF1,WR1,AA2       OPR->R2 (X REG)                    449
00AC 0568000008670000    450  E620M   GEN     /N(E620M1),LA1,FF6,CF3,WR1,AAD     P-R0->RD                                450
00AD 05A009020100000C    451  E620M1  GEN     /P(VSS2M),SF2,GF4,IM4,RF1,AAC      RC->620(P), IF(ALU)                     451
00AE 0560000020A90002    452  E620MV  GEN     /N(E620M),LB1,FFA,MF1,WR1,AA2      OPERAND -> 620 X.                       452
```

```
                          454  *       MDX INSTRUCTION                                                    454
                          455  *                                                                          455
                          456          MORG    16       MDXXS1, MDXXL1 REFERENCED BY CONDITIONAL FSEL     456
00B0 0598000043AFC000     457  MDXXS1  GMSK    /N(MDXXS2),LB2,RF3,FFA,MKFC00          IR & 3FF->OPR      457
                          458  MDXXL1  GEN     /F(MDXXL3),2(1),FSB,LB1,RF7,FFA,MF1,   MIR->W1 & OPR,     458
00B1 658AC00027A9001E     459                 CWR1,BB1,24(W1)                        FSEL(7), INCP      459
                          460  MDXC1   GEN     /T(MDXC2,MDXC3),TF2,SF1,GF9,IM5,       OF(ALU), BASE+W1   460
00B2 75F826428090000E     461                 CFF9,23(BASE),24(W1)                                      461
00B3 05F0008080380000     462  MDXXS2  GEN     /N(MDXC3),GF2,IM1,FF3,MF1              WAIT(MEM),0->ALU,  462
                          463  *                                                      SAMPLE             463
                          464          MORG    4        MDXSKP MUST BE EVEN, MORG 16 WITH MDXNI,MDXC8     464
00B4 0520040424A80000     465  MDXSKP  GEN     /N(SS2M),SF1,IM8,RF4,LB1,FFA,MF1       IF(P),INCP,OPR->   465
00B5 062000802068000E     466  MDXC8   GEN     /N(MDXC9),GF2,LB1,FF6,MF1,24(W1)       W1 XOR OPR, SAMPLE 466
00B6 0520040420A80000     467  MDXNI   GEN     /N(SS2M),SF1,IM8,LB1,FFA,MF1           IF(P), OPR->ALU    467
00B7 0628000000A90001     468  MDX     GEN     /N(MDX1),FFA,MF1,WR1,AA1               R0->R1             468
                          469          EVEN                                                               469
00B8 558B800043AFC000     470  MDXNX   GMSK    /F(MDXP),2(1),FSE,LB2,RF3,FFA,MKFC00   IR&3FF->OPR,FS(10)  470
00B9 0590000020B9001E     471  MDXNX2  GEN     /N(MDXC1),LB1,FFB,MF1,WR1,BB1,24(W1)   MIR.W1->W1           471
                          472          EVEN                                                               472
00BA 0520040229900040     473  MDXP    GEN     /N(SS2M),SF1,IM4,LB1,LA1,RF1,FF9,BB4   P+ORSE->P,IF(ALU)   473
                          474  MDXNX1  GEN     /N(MDXNX2),IM1,RF4,FFA,MF1,WR1,        WAIT(M),WRAP->W1,  474
00BB 05C8000084A9009E     475                 C23(WRAP),24(W1)                       INCP               475
                          476          EVEN                                                               476
00BC 05F0000080000000     477  MDXXL3  GEN     /N(MDXC3),IM1                          WAIT(MEM)          477
00BD 0590000000B9009E     478  MDXXL2  GEN     /N(MDXC1),FFB,MF1,WR1,23(WRAP),24(W1)  WRAP & W1->W1      478
                          479          EVEN                                                               479
                          480  MDXC3   GEN     /S(MDXC5,MDXC6),2(1),FSB,TF3,GF9,LB1,  MIR->W1,           480
00BE 160AF24020A9001E     481                 CFFA,MF1,WR1,BB1,24(W1)                FSEL(7) IF NO ALUZ 481
00BF 05F00000809100E1     482  MDXC2   GEN     /N(MDXC3),IM1,FF9,WR1,23(W1),24(ADDR)  WAIT(M),           482
                          483  *                                                      W1+ADDR->ADDR      483
                          484          MORG    16       MDXC5, MDXC4 REF BY CND FSEL                      484
00C0 061800802390000E     485  MDXC5   GEN     /N(MDXC7),GF2,LB1,RF3,FF9,24(W1)       OPR+W1->OPR,SAMPLE 485
00C1 0600001880000001E    486  MDXC4   GEN     /N(MDXC5),IM1,AB3,BB1,24(W1)           "MIR"->BB,WAIT(M)  486
00C2 061800802390004E     487  MDXC6   GEN     /N(MDXC7),GF2,LB1,RF3,FF9,BB4,24(W1)   ORSE+W1->OPR,SAMPL  487
                          488  MDXC7   GEN     /T(MDXC8,MDXSKP),TF3,SF3,GF9,IM6,      CND OS(ALU),       488
00C3 25A83E4300000001     489                 C24(ADDR)                              ADDR->ALU          489
                          490  MDXC9   GEN     /T(MDXNI,MDXSKP),TF3,GF7,SF1,IM6,      OS(ALU),           490
00C4 25B035C300000001     491                 C24(ADDR)                              ADDR->ALU          491
                          492  MDX1    GEN     /S(MDXXS1,MDXNX),2(1),FSE,TF3,SF2,GF9, CND OF(A),FSEL(10) 492
00C5 458BBA42A0910051     493                 CIM5,LB1,FF9,WR1,BB5,24(ADDR)          OLSE+ADDR->ADDR    493
```

```
                         495  *       SHIFT INSTRUCTIONS                                                                  495
                         496  *                                                                                           496
                         497  *       THIS IS THE SHIFT COUNT COMPUTATION.                                                497
00C6 0638000000A9000E    498  SHIFT1  GEN     /N(SHIFT2),FFA,MF1,WR1,AAE          R0 -> RE                        498
                         499  SHIFT2  GEN     /T(SHFTDF,SHFTX),TF2,SF2,GF9,IM5,  0->QS,ALUZ=>SHFTDF;                      499
00C7 56402A42A091205E    500                  CLB1,FF9,WR1,WF1,BB5,AAE NO ALUZ->OF(ALU=TAG+BASE),=>SHFTX                  500
00C8 0668000043AFFC00    501  SHFTDF  GMSK    /N(SHIFT3),LB2,RF3,FFA,MKFFC0       LOW 6 (DISP) -> OPR             501
                         502          EVEN                                                                                502
00CA 0658000080002005    503  SHFTX   GEN     /N(SHFTX1),IM1,WF1,AA5              WAIT(INDEX), 1->QS              503
00CB 0660000020A90011    504  SHFTX1  GEN     /N(SHFTX2),LB1,FFA,MF1,WR1,BB1,AA1      MIR -> R(1)                 504
00CC 0668000063BFFC01    505  SHFTX2  GMSK    /N(SHIFT3),LB3,RF3,FFB,MKFFC0,AK1 LOW 6 (R(1)) -> OPR                   505
                         506  *       SHIFT3 DOES:  -OPR -> R(1) AND SHFT, SAMPLE ALU CONDITIONALS.                       506
00CD 760BC08022670101    507  SHIFT3  GEN     /F(SHFTL),2(1),FSF,GF2,LB1,RF2,FF6,CF3,WR1,SH1,AA1                      507
                         508  *       SHIFT INSTRUCTIONS COME HERE WITH THE NEGATIVE OF THE SHIFT                          508
                         509  *       COUNT IN THE SHIFT COUNT REGISTER AND R1.  NO SHIFTING IS DONE                       509
                         510  *       IF THE SHIFT COUNT IS ZERO.                                                         510
                         511  *       SHFTL AND SHFTR MUST BE KEPT TOGETHER: THEY ARE FIELD SELECT                        511
                         512  *       ADDRESSED.  SHFTL AND SHFTR DO THE SAME THING TO LEFT AND                           512
                         513  *       RIGHT SHIFT INSTRUCTIONS RESPECTIVELY:  RB->OPR, INCP,                              513
                         514  *       FIELD SELECT (7-6) IF NOT ALUZ, IF(P) IF ALUZ.                                      514
                         515          EVEN                                                                                515
00CE 769ABE4407000000B   516  SHFTL   GEN     /S(SHFTL0,SS3MX),2(3),FSA,TF3,SF3,GF9,IM8,RF7,AAB   LEFT                516
00CF 271ABE4407000000B   517  SHFTR   GEN     /S(SHFTR0,SHFTRX),2(3),FSA,TF3,SF3,GF9,IM8,RF7,AAB  RIGHT               517
                         518          MORG    16              SHFTL0-SHFTL3 REF BY CND FSEL FROM SHFTL                    518
                         519  SHFTL0  GEN     /T(*,SHFTLX),TF3,GFC,LA2,RF5,WR1,VF1,  SHIFT RA LEFT                       519
00D0 268033001501400A    520                  CAAA                                                                        520
                         521  SHFTL1  GEN     /T(SHFT11,SHFTL0),TF2,GFF,FFC,CF3,WR1,  TEST QS,                           521
00D1 06A823C000C7000E    522                  CAAE                                    RE+RE+1 -> RE                       522
                         523  SHFTL2  GEN     /T(*,SHFTLX),TF3,GFC,LA2,RF5,WR1,SC1,  SHIFT RA&OPR LEFT                   523
00D2 269033001501DA0A    524                  CVF1,XF3,SH2,AAA                                                            524
                         525  SHFTL3  GEN     /T(SHFT31,SHFTL2),TF2,GFF,FFC,CF3,WR1,  TEST QS,                           525
00D3 16C023C000C7000E    526                  CAAE                                    RE+RE+1 -> RE                       526
                         527          EVEN                                                                                527
00D4 0728000020A9000B    528  SHFTLX  GEN     /N(SHFMSK),LB1,FFA,MF1,WR1,AAB      OPR -> RB                       528
00D5 06B000000000200A    529  SHFT11  GEN     /N(SHFT12),WF1,AAA                  RA(S)->QS                       529
00D6 56B833C000070001    530  SHFT12  GEN     /T(SHFT13,SHFDR1),TF3,GFF,CF3,WR1,AA1  R1+1->R1,TEST QS                530
                         531  SHFT13  GEN     /T(SHFT12,SHFMR1),TF3,GFC,LA2,RF5,WR1,  RA(L)->RA,INC SC,                   531
00D7 66B033001501200A    532                  CWF1,AAA                                ALUS->QS,TEST SC                    532
00D8 06C800000000200A    533  SHFT31  GEN     /N(SHFT32),WF1,AAA                  RA(S)->QS                       533
00D9 56D833C000070001    534  SHFT32  GEN     /T(SHFT33,SHFDR1),TF3,GFF,CF3,WR1,AA1  R1+1->R1,TEST QS                534
                         535          EVEN                                                                                535
00DA 06E0000000F10001    536  SHFDR1  GEN     /N(SHFMR1),FFF,WR1,AA1              R1-1->R1                        536
                         537  SHFT33  GEN     /T(SHFT31,SHFMR1),TF3,GFC,LA2,RF5,WR1,  SHFT RA&OPR LEFT,                  537
00DB 66C033001501 9A0A   538                  CXF3,SC1,SH2,AAA                  TEST SC,INC SC                        538
                         539          EVEN                                                                                539
00DC 06E8000000004001    540  SHFMR1  GEN     /N(SHFTBS),VF1,AA1                  R1(15)->DSB                     540
00DD 06F808039800000E    541  SHFTBS  GEN     /N(SHFTM),AAE,LA3,FF0,IM7,SF2       RE/2, 1 -> BYTE A               541
                         542  *       SS3MX IS IDENTICAL TO SS3M BUT IS PLACED HERE DUE TO                                542
                         543  *       ADDRESSING RESTRICTIONS.                                                            543
                         544          EVEN                                                                                544
00DE F500014300000000    545  SS3MX   GEN     /N(INTBAS),1(X'F),GF5,IM6                                               545
00DF 06A0000080670111    546  SHFTM   GEN     /N(SHFTLX),IM1,FF6,CF3,WR1,SH1,BB1,AA1 WAIT(M),-R1->R1                  546
                         547          MORG    16              SHFTR0-SHFTR3 REF BY CND FSEL FROM SHFTR                    547
00E0 27003F041D01040A    548  SHFTR0  GEN     /T(*,SHFTRX),TF3,SF3,GFC,IM8,LA3,RF5,WR1,SH4,AAA                        548
00E1 27083F041D01040A    549  SHFTR1  GEN     /T(*,SHFTRX),TF3,SF3,GFC,IM8,LA3,RF5,WR1,SH4,AAA                        549
                         550  SHFTR2  GEN     /T(*,SHFTRX),TF3,SF3,GFC,IM8,LA3,RF5,                                   550
00E2 27103F041D01AA0A    551                  CWR1,SC1,WF1,XF1,SH2,AAA                                                    551
```

```
                         552  SHFTR3 GEN     /T(*,SHFTRX),TF3,SF3,GFC,IM8,LA3,RF5,                                552
00E3 27183F041D01AB0A    553                CWR1,SC1,WF1,XF1,SH3,AAA                                             553
                         554  *       SHFTRX IS SIMILAR TO SS3M, AND IS PUT HERE DUE TO                           554
                         555  *       ADDRESSING RESTRICTIONS WHEN USING CONDITIONAL FIELD SELECTION.             555
                         556  *       IN ADDITION TO EVERYTHING THAT SS3M DOES, SHFTRX DOES:  OPR->RB.            556
                         557  *       SHFTRX IS ALSO USED BY THE MULTIPLY ROUTINE.                                557
                         558         EVEN                                                                       558
00E4 F500014320A9000B    559  SHFTRX GEN     /N(INTBAS),1(X'F),GF5,IM6,LB1,FFA,MF1,WR1,AAB                       559
                         560  SHFMSK GMSK    /T(SHFCOM,SHFARG),TF3,GFA,LB3,RF3,     CARRY MASK (X'800)          560
00E5 3738328063AF7FF0    561                CFFA,MKF7FF                            -> OPR                       561
                         562         EVEN                                                                       562
00E6 0530040420110008    563  SHFARG GEN     /N(SS3M),SF1,IM8,LB1,FF1,WR1,AA8       IF(P), CARRY = 1            563
00E7 0530040420790008    564  SHFCOM GEN     /N(SS3M),SF1,IM8,LB1,FF7,MF1,WR1,AA8   IF(P), CARRY = 0            564
```

```
                         566  *       THE FOLLOWING STATES CANNOT BE IDENTIFIED AS BELONGING TO ANY          566
                         567  *       SUBROUTINE OF SIGNIFICANT SIZE.                                       567
                         568  *                                                                             568
                         569  *       ADDRESSES ARE (MOSTLY) IN ALPHABETICAL ORDER.                         569
                         570  *                                                                             570
00E8 052802802091001A    571  A2      GEN     /N(SS2MS),GFA,LB1,FF9,WR1,BB1,AAA      RA+MIR -> RA,         571
                         572  *                                                       SAMPLE OFL & COND     572
                         573  ADSD2   GEN     /T(ADSD3,ASDODD),TF2,SF3,GF9,IM5,CF3,   R1+1, CND OF(ALU)      573
00E9 57582E4280060001    574                 CAA1                                                          574
                         575          EVEN                                                                  575
00EA 0758000080000000    576  ASDODD  GEN     /N(ADSD3),IM1                           WAIT(MEM)             576
                         577  ADSD3   GEN     /F(AD4),2(2),FSF,IM1,LB1,FFA,MF1,       FS(12),               577
00EB 6713C000A0A9001E    578                 CWR1,BB1,AAE                            WAIT(MEM), MIR->RE    578
                         579          MORG    4               AD4, SD4 REF BY FSEL FROM ADSD3               579
00EC 076800802091001B    580  AD4     GEN     /N(AD5),GF2,LB1,FF9,WR1,BB1,AAB         B+MIR->B,SAMPLE       580
00ED 0528028000930 0EA   581  AD5     GEN     /N(SS2MS),GFA,FF9,CF1,WR1,BBE,AAA       A+E+CRY->A,SAMPLE     581
00EE                     582          ORG     AD4+2                                                         582
                         583  SD4     GEN     /N(SD5),GF2,LB1,FF6,CF3,WR1,BB1,  EXT-MIR->EXT,SAMPLE ALU     583
00EE 077800802067001B    584                 C24(EXT)                                                      584
                         585  SD5     GEN     /N(SD6),GFA,FF6,CF1,WR1,23(W1),   ACC-W1+CRY->ACC,            585
00EF 07800280006300EA    586                 C24(ACC)                          SAMPLE ALU & OVFL           586
00F0 0B18000063AF7FF0    587  SD6     GMSK    /N(S3),LB3,RF3,FFA,MKF7FF         X'800->OPR                 587
                         588  AND2    GEN     /N(INTBAS),1(X'F),GF5,IM6,LB1,    RA.MIR->RA,IBR->I,          588
00F1 F500014320B9001A    589                 CFFB,MF1,WR1,BB1,AAA              DECODE,EN INT,              589
                         590  *                                                 SEL & RESET INT FLAG        590
00F2 0798000088660000    591  BSI2    GEN     /N(BSI3),IM1,LA1,FF6,CF3          WAIT(MEM),P-R0 -> ALU       591
00F3 0520040201060001    592  BSI3    GEN     /N(SS2M),SF1,IM4,RF1,CF3,AA1      IF(ALU), R1+1 -> P          592
                         593  DIV1    GEN     /N(DIV2),LB1,FFA,MF1,WR1,WF1,BB1, MIR->RF,MIRS->QS            593
00F4 07A8000020A9201F    594                 CAAF                                                          594
00F5 47B023C00000400A    595  DIV2    GEN     /T(DIV3,DIV4),TF2,GFF,VF1,AAA     RA(15)->DSB,TEST QS         595
00F6 07C00000006701FF    596  DIV3    GEN     /N(DIV4),FF6,CF3,WR1,SH1,BBF,AAF  -RF -> RF                   596
                         597          EVEN                                                                  597
                         598  DIV4    GEN     /T(DIV5,DIV7),TF2,GFA,FF6,CF3,    -RF -> RE,                  598
00F8 67C82280006701FE    599                 CWR1,SH1,BBF,AAE                  TEST DSB                    599
                         600  DIV5    GEN     /N(DIV6),GF2,FF6,CF3,WR1,SH1,BBB, -RB -> RB                   600
00F9 07D000800067 01BB   601                 CAAB                                                          601
00FA 07E000000063 01AA   602  DIV6    GEN     /N(DIV7),FF6,CF1,WR1,SH1,BBA,AAA  -RA+CARRY->RA               602
                         603          EVEN                                                                  603
00FC 07E80080009100EA    604  DIV7    GEN     /N(DIV8),GF2,FF9,WR1,BBE,AAA      RA+RE->RA, SAMPLE           604
                         605  DIV8    GMSK    /T(DIV10,DIV9),TF2,GF7,LB3,RF2,   -15 -> CNT, TEST ALUS       605
00FD 77F821C062A000F0    606                 CFFA,MKF                                                      606
                         607          EVEN                                                                  607
                         608  DIV9    GEN     /N(DIV10),SF1,GF2,LB1,FFE,MF1,    OPR V STAT -> STAT,         608
00FE 07F8048020E90008    609                 CWR1,24(STAT)                     SET OVERFLOW                609
00FF 0808000003000008    610  DIV10   GEN     /N(DIV11),RF3,AAB                 RB -> OPR                   610
                         611          EVEN        DIV11 & DIV12 MUST HAVE SAME 5 HIGH-ORDER BITS            611
0100 0810008000000000A   612  DIV12   GEN     /N(DIV13),GF2,AAA                 RA -> ALU, SAMPLE           612
                         613  DIV11   GEN     /T(*,DIV12),TF3,GFC,MR1,LA2,RF5,  DIVIDE STEP                 613
0101 08083320159192FA    614                 CFF9,WR1,SC1,XF2,SH2,BBF,AAA                                  614
0102 281821C000000000    615  DIV13   GEN     /T(DIV14,DIV15),TF2,GF7           TEST ALUS                   615
0103 0820000009100FA     616  DIV14   GEN     /N(DIV15),FF9,WR1,BBF,AAA         RA+RF -> RA                 616
                         617          EVEN                                                                  617
0104 3828328000000000    618  DIV15   GEN     /T(DIV16,DIV17),TF3,GFA           TEST NO DSB                 618
                         619  DIV16   GEN     /T(DIV18,DIV19),TF2,GFF,FFA,MF1,  RA -> RB                    619
0105 483823C000A900AB    620                 CWR1,BBA,AAB                                                  620
                         621          EVEN                                                                  621
                         622  DIV17   GEN     /T(DIV18,DIV19),TF3,GFF,FF6,CF3,  -RA -> RB                   622
```

```
0106 483833C0006701AB 623                    CWR1,SH1,BBA,AAB                                                    623
                      624  DIV18  GEN        /N(SS3M),SF1,IM8,LB1,RF4,FF6,CF3, -OPR->RA,IF(P),INCP               624
0107 053004042467010A 625                    CWR1,SH1,AAA                                                        625
                      626         EVEN                                                                           626
                      627  DIV19  GEN        /N(SS3M),SF1,IM8,LB1,RF4,FFA,MF1, OPR->RA,IF(P),INCP               627
0108 053004042A9000A 628                     CWR1,AAA                                                            628
0109 0850008060701008 629  ENTR   GMSK       /N(ENTR1),GF2,LB3,FF7,MK0100,AK8  R8&X'0100 -> ALU,SAMPLE         629
010A 6858324000000000 630  ENTR1  GEN        /T(ENTRS0,ENTRR0),TF3,GF9         TEST ALUZ                         630
010B 0530048404000000 631  ENTRS0 GEN        /N(SS3M),SF1,GF2,IM8,RF4          SET OVFL,IF(P),INCP               631
                      632         EVEN                                                                           632
010C 0530050404000000 633  ENTRR0 GEN        /N(SS3M),SF1,GF4,IM8,RF4          RESET OVFL,IF(P),INCP             633
                      634  EOR2   GEN        /N(INTBAS),1(X'F),GF5,IM6,LB1,    RA EOR MIR -> RA,                 634
010D F50001432069001A 635                    CFF6,MF1,WR1,BB1,AAA              IBR->I,DECODE,EN INT,             635
                      636  *                                                    SEL&RESET INT FLAG                636
                      637  INTEN0 GMSK       /N(INTEN1),SF1,IM5,LB3,FF9,       R1+8->ALU,OF(ALU)                 637
010E 08780402E09FFF71 638                    CMKFFF7,AK1                                                         638
010F 0880000080A90091 639  INTEN1 GEN        /N(INTEN2),IM1,FFA,MF1,WR1,BB9,AA1 WAIT(MEM), R9->R1                639
0110 0888000020B90011 640  INTEN2 GEN        /N(INTEN3),LB1,FFB,MF1,WR1,BB1,AA1 MIR.R1->R1                       640
0111 0890040301900001 641  INTEN3 GEN        /N(INTEN4),SF1,IM6,RF1,FF9,AA1    R1+R0->P,OS(ALU)                  641
0112 0848040404000000D 642 INTEN4 GEN        /N(ENTR),SF1,IM8,RF4,AAD          IF(P), INCP, RD->ALU              642
                      643  LDA2   GEN        /N(INTBAS),1(X'F),GF5,IM6,LB1,    MIR->RA,IBR->I,DECODE,            643
0113 F50001432OA9001A 644                    CFFA,MF1,WR1,BB1,AAA              EN INT, SEL&RST INT FLG           644
                      645  LDD2   GEN        /T(LDD3,LDDODD),TF2,SF3,GF9,IM5,  R1+1, CND OF(ALU)                 645
0114 38882E4280060001 646                    CCF3,AA1                                                            646
                      647         EVEN                                                                           647
0116 08B8000080000000 648  LDDODD GEN        /N(LDD3),IM1                                  WAIT(MEM)             648
0117 08C00000A0A9001A 649  LDD3   GEN        /N(LDD4),IM1,LB1,FFA,MF1,WR1,BB1,AAA  WAIT(MEM), MIR->RA            649
                      650  LDD4   GEN        /N(SS3M),SF1,IM8,RF4,LB1,FFA,     INCP,IF(P),MIR->RB                650
0118 053004042A9001B 651                     CMF1,WR1,BB1,AAB                                                    651
                      652         EVEN                                  LDSRC,LDSSC REF BY FSEL                 652
011A 6889008000300000 653  LDSRC  GEN        /F(LDSR0),2(1),FS4,GF2,FF3        RST CRY, SAMPLE, FS(0)            653
011B 6889008000360000 654  LDSSC  GEN        /F(LDSR0),2(1),FS4,GF2,FF3,CF3    SET CRY, SAMPLE, FS(0)            654
                      655         EVEN                                  LDSR0,LDSS0 REF BY FSEL                 655
011C 08F0050404000000 656  LDSR0  GEN        /N(LDSX),SF1,GF4,IM8,RF4          RESET OVFL, INCP, IF(P)           656
011D 08F0048404000000 657  LDSS0  GEN        /N(LDSX),SF1,GF2,IM8,RF4          SET OVFL, INCP, IF(P)             657
                      658  LDSX   GEN        /N(INTBAS),1(X'F),GF5,IM6,LB1,    STAT->R8,IBR->I,DECODE,           658
011E F50001432OA90038 659                    CFFA,MF1,WR1,BB3,AA8              EN INT, SEL&RST INT FLG           659
                      660         MORG       16              LXLXNI, LDXLXI REF BY CND FSEL FROM LDXL1           660
0120 0920040320900050 661  LXLXNI GEN        /N(LDXLXR),SF1,IM6,LB1,FF9,BB5    OLSE+R0 -> ALU, OS(ALU)           661
                      662  LDXLXI GEN        /N(LDXLX1),SF1,IM8,RF4,FFA,MF1,WR1,    R9->R1,INCP,IF(P)           662
0121 0910040404A90091 663                    CBB9,AA1                                                            663
0122 0918000025B90011 664  LDXLX1 GEN        /N(LDXLX2),LB1,RF5,FFB,MF1,WR1,BB1,AA1 MIR.R1->R1,INC CNT           664
0123 0900040280910001 665  LDXLX2 GEN        /N(LXLXNI),SF1,IM5,FF9,WR1,AA1    R1+R0->R1, OF(ALU)                665
                      666  LDXLXR GEN        /T(SS3M,SS2M),TF2,SF1,GFC,IM8,LB1,      TEST SHFT CNT,              666
0124 253027042A80010 667                     CRF4,FFA,MF1,BB1                        MIR->R1,INCP,IF(P)          667
                      668  LXLNX1 GEN        /S(LDXPJ,LDXPJ),2(1),FSB,TF3,GFC,LB1,   TEST SHFT CNT,              668
0125 098AF30023B90011 669                    CRF3,FFB,MF1,WR1,BB1,AA1                MIR.R1->R1,OPR              669
                      670         EVEN                                                                           670
0126 0928000080A90091 671  LDXLNX GEN        /N(LXLNX1),IM1,FFA,MF1,WR1,BB9,AA1      WAIT(MEM), R9->R1           671
                      672  LDXS1  GMSK       /T(LDXSP,LDXSX),TF2,GF9,LB2,FFA,RF3,    TAG & DISP -> OPR           672
0127 4948224043AFC000 673                    CMKFC00                                                             673
                      674         EVEN                                                                           674
0128 0950040320900050 675  LDXSX  GEN        /N(LDXSX1),SF1,IM6,LB1,FF9,BB5    OLSE+R0 -> ALU, OS(ALU)           675
0129 0980000023B80049 676  LDXSP  GEN        /N(LDXPJ),LB1,RF3,FFB,MF1,BB4,AA9 ORSE.R9->OPR(WRAP)                676
012A 053004042A80040 677   LDXSX1 GEN        /N(SS3M),SF1,IM8,LB1,RF4,FFA,MF1,BB4    ORSE->ALU(FOR OS),          677
                      678  *                                                        INCP,IF(P)                  678
                      679  LDXL1  GMSK       /S(LXLXNI,LDXLNX),2(1),FSB,TF3,GF9,     CND FSEL(7)                 679
012B 390AF24062A00010 680                    CLB3,RF2,FFA,MK1                        -2 -> CNT                   680
```

```
                          681          MORG    16                              LDXPJ, LXLNXI REF BY CND FSEL          681
0130 0520040221900000     682  LDXPJ   GEN      /N(SS2M),SF1,IM4,LB1,RF1,FF9                OPR+R0->P,IF(ALU)      682
                          683  *                                                R1+R0->ALU             683
0131 0930040285900001     684  LXLNXI  GEN      /N(LDXLNX),SF1,IM5,RF5,FF9,AA1              INC CNT, OF(ALU),      684
0132 0998040201000000     685  LMODE2  GEN      /N(LMODE3),SF1,IM4,RF1                      R0->P, IF(ALU)         685
0133 09A0000000A90019     686  LMODE3  GEN      /N(LMODE4),FFA,MF1,WR1,BB1,AA9              R(1) -> R(9)           686
0134 0848000000010108     687  LMODE4  GEN      /N(ENTR),WR1,SH1,AA8                0 -> R8                687
                          688  MBM1    GEN      /N(MBM2),SF1,IM9,LB1,RF4,FFA,MF1,   AF(P),INCP,MIR->R0     688
0135 09B00404A4A90010     689                   CWR1,BB1,AA0                                                   689
                          690  MBM2    GEN      /N(MBMX),SF1,IM8,LB1,RF4,FFA,MF1,   IF(P),INCP,MIR->W1,    690
0136 09B8040424A9201E     691                   CWR1,WF1,BB1,24(W1)                 ALU(15)->QS            691
0137 09C0000023A80010     692  MBMX    GEN      /N(MBM3),LB1,RF3,FFA,MF1,BB1        MIR->OPR               692
0138 09C8000000F1000E     693  MBM3    GEN      /N(MBM4),FFF,WR1,24(W1)             W1-1->W1               693
0139 09D0000000801000F    694  MBM4    GEN      /N(MBM5),LA1,WR1,24(W2)             P->W2                  694
013A 09D8000081F04000     695  MBM5    GEN      /N(MBM6),IM1,RF1,FFF,VF1,AA0        R0-1->P,R0(15)->DSB    695
013B 09E8000020A90000     696  MBM6    GEN      /N(MBM7),LB1,FFA,MF1,WR1,BB0,AA0    OPR->R0                696
                          697          EVEN                                                                   697
013C 0A580000A0A80000     698  MBM11   GEN      /N(MBM19),IM1,LB1,FFA,MF1           WAIT(M),OPR->ALU       698
                          699  MBM7    GEN      /T(MBM8,MBM12),TF3,SF1,GFF,IM9,     OF(P),INCP,OPR->ALU    699
013D 79F837C4A4A80000     700                   CLB1,RF4,FFA,MF1,BB0                                           700
                          701          EVEN                                                                   701
013E 0A08008080F10000     702  MBM12   GEN      /N(MBM13),GF2,IM1,FFF,WR1,AA0       WAIT(M),R0-1->R0,SAMPLE 702
013F 0A00008000F10000     703  MBM8    GEN      /N(MBM9),GF2,FFF,WR1,AA0            R0-1->R0,SAMPLE ALU    703
                          704  MBM9    GEN      /T(MBM10,MBM18),TF3,SF2,GF7,IM6,    CND(NO ALUS) OS(ALU),  704
0140 4A5039C30007000E     705                   CCF3,WR1,24(W1)                     W1+1->W1               705
0141 0A10000023A80010     706  MBM13   GEN      /N(MBM14),LB1,RF3,FFA,MF1,BB1       MIR->OPR               706
0142 0A20000062A00030     707  MBM14   GMSK     /N(MBM15),LB3,RF2,FFA,MK3           -4->SC                 707
                          708          EVEN                                                                   708
0144 2A28230005008000     709  MBM15   GEN      /T(MBM16,*),TF2,GFC,RF5,SC1         OPR(L),INC SC, TEST SC 709
                          710  MBM16   GEN      /T(MBM17,MBM18),TF3,SF2,GF7,IM6,    CND(NO ALUS) OS(ALU),  710
0145 4A4839C30007000E     711                   CCF3,WR1,24(W1)                     W1+1->W1               711
                          712          MORG     4                                                              712
                          713  MBM18   GEN      /P(VSS3M),SF2,GF4,IM4,RF1,CF3,      IF(ALU),W2+1->P        713
0148 016809020106000F     714                   C24(W2)                                                        714
                          715  MBM17   GEN      /T(MBM7,MBM11),TF3,GFA,LB1,         OPR->ALU,TEST DSB      715
0149 69E8328020A80000     716                   CFFA,MF1                                                       716
                          717  MBM10   GEN      /T(MBM7,MBM11),TF3,GFA,LB1,RF3,     MIR->OPR,TEST DSB      717
014A 69E8328023A80010     718                   CFFA,MF1,BB1                                                   718
014B 0A28008000F10000     719  MBM19   GEN      /N(MBM16),GF2,FFF,WR1,AA0           R0-1->R0,SAMPLE        719
014C 0A6800000039800E     720  MUL2    GEN      /N(MUL3),FF3,MF1,WR1,SC1,AAE                0->E,LEFT CIRC OPR     720
014D 0A700000E2A000E0     721  MUL3    GMSK     /N(MUL4),IM1,LB3,RF2,FFA,MK000E             -14->CNTR, WAIT(M)     721
014E 0A78000020A9001F     722  MUL4    GEN      /N(MUL5),LB1,FFA,MF1,WR1,BB1,AAF            MIR->F                 722
014F 0A8000200039A00A     723  MUL5    GEN      /N(MUL6),MR1,FF3,MF1,WR1,SC1,WF1,AAA        0->A,CRC RT OPR,       723
                          724  *                                                            OPR(1)->BAD(0)         724
                          725  MUL6    GEN      /N(MUL7),MR1,RF5,FF9,WR1,SC1,WF1,XF1,       A+(E OR F)->A,         725
0150 0A9000200591A80A     726                   CAAA      RT OPR, OPR(1)->BAD(0), INC CNTR                     726
                          727          EVEN                                                                   727
                          728  MUL7    GEN      /T(MUL8,*),TF2,GFC,MR1,LA3,RF5,FF9,         A+(E OR F) -> A,       728
0152 1A9823201D91A80A     729                   CWR1,SC1,WF1,XF1,AAA      MUL SIGN-> DLA(15), RT A, RT OPR,    729
                          730  *                                           DFA(0)->OPR(15), OPR(1)->BAD(0),     730
                          731  *                                           INC CNTR, TEST CNTR = -1             731
                          732  MUL8    GEN      /T(MUL10,MUL9),TF3,GFA,LA3,WR1,SC1,         RT A, MUL SGN->A15     732
0153 2AA832801801A80A     733                   CWF1,XF1,AAA    RT OPR, DFA(0)->OPR(15), DSB=>MUL9                 733
                          734          EVEN                                                                   734
0154 0AA80000006700FA     735  MUL9    GEN      /N(MUL10),FF6,CF3,WR1,BBF,AAA               A-F->A                 735
                          736  MUL10   GEN      /N(SHFTRX),SF1,IM8,LA3,RF4,WR1,SC1,          RT A, A15->A15,        736
0155 072004041C01AA0A     737                   CWF1,XF1,SH2,AAA    RT OPR, DFA(0)->OPR(15), INCP, IF(P)       737
                          738  OR2     GEN      /N(INTBAS),1(X'F),GF5,IM6,LB1,      RA OR MIR ->RA, IBR->I,  738
```

```
0156 F50001432011001A 739                    CFF1,WR1,BB1,AAA                    DECODE, EN INT,                739
                      740  *                                                     SEL & RST INT FLG              740
0157 0AC0000020A9000E 741  PSCAN2 GEN         /N(PSCAN3),LB1,FFA,MF1,WR1,AAE      OPR->RE                        741
0158 0AC8000063A00770 742  PSCAN3 GMSK        /N(PSCAN4),LB3,RF3,FFA,MK0077       -120->OPR                      742
0159 0AD00000E2A000F0 743  PSCAN4 GMSK        /N(PSCAN5),IM1,LB3,RF2,FFA,MK000F   -16->SC, WAIT(M)               743
                      744  PSCAN5 GEN         /N(PSCAN6),LB1,FFA,MF1,WR1,WF1,     MIR->RF, ALU(15)->QS           744
015A 0AE0000020A9201F 745                    CBB1,AAF                                                           745
                      746  *      PSCANX AND PSCAN6 MUST HAVE THE SAME 5 HIGH-ORDER BITS                        746
015B 0168090404000000 747  PSCANX GEN         /P(VSS3M),SF2,GF4,IM8,RF4           INCP, IF(P)                    747
                      748         EVEN                                                                           748
                      749  PSCAN6 GEN         /N(PSCAN7),GF2,LB1,RF3,FF9,CF3,     OPR+1->OPR, SAMPLE             749
015C 0AE8008023960100 750                    CSH1                                                               750
                      751  PSCAN7 GEN         /T(PSCAN9,PSCAN8),TF3,SF3,GFF,      OS(ALU) IF QS, R2->ALU         751
015D 7AF83FC300000002 752                    CIM6,AA2                                                           752
                      753         EVEN                                                                           753
015E 0AF8000080000001 754  PSCAN8 GEN         /N(PSCAN9),IM1,AA1                  WAIT(M),R1->ALU                754
                      755  PSCAN9 GEN         /T(PSCANB,PSCANA),TF3,GFC,RF5,      R2+1->R2, INC SC               755
015F 0B08330005070002 756                    CCF3,WR1,AA2                                                       756
                      757         EVEN                                                                           757
0160 0AC804028007000E 758  PSCANA GEN         /N(PSCAN4),SF1,IM5,CF3,WR1,AAE      RE+1->RE, OF(ALU)              758
                      759  PSCANB GEN         /T(PSCANX,PSCAN6),TF2,GF9,LA2,      RF(L)->RF, ALU(15)->QS         759
0161 6AD822401001200F 760                    CWR1,WF1,AAF                                                       760
                      761  S2     GEN         /N(S3),GFA,LB1,FF6,CF3,WR1,BB1,     ACC-MIR->ACC,                  761
0162 0B1802802067001A 762                    C24(ACC)                            SAMPLE ALU & OVFL              762
                      763  S3     GEN         /N(S4),SF1,IM8,LB1,RF4,FFA,MF1,     INCP,IF(P),SREG->STAT          763
0163 0B2004042 4A90038 764                   CWR1,BB3,24(STAT)                                                  764
                      765  S4     GEN         /N(INTBAS),1(X'F),GF5,IM6,LB1,      STAT XOR OPR -> STAT,          765
0164 F500014320690008 766                    CFF6,MF1,WR1,BB0,24(STAT)           SEL&RST CINTF, IBR->I,         766
                      767  *                                                     ENABLE INTS, DECODE            767
0165 0B30000080000000A 768 STD2   GEN         /N(STD3),IM1,AAA                        WAIT(MEM), RA->ALU         768
0166 0B38008060BFFFE1 769  STD3   GMSK        /N(STD4),GF2,LB3,FFB,MKFFFE,AK1         R1.1, SAMPLE               769
                      770  STD4   GEN         /T(STDEVN,SS2M),TF2,SF3,GF9,IM6,        R1+1,                      770
0167 25502E4300060001 771                    CCF3,AA1                                CND OS(ALU) (ALUZ)         771
0168 0530040404000000A 772 STO2   GEN         /N(SS3M),SF1,IM8,RF4,AAA                INCP,IF(P),A->ALU          772
0169 0850000023A80050 773  STS1   GEN         /N(STS2),LB1,RF3,FFA,MF1,BB5            OLSE->OPR                  773
                      774  *                                                          (OVFL->LO-ORD BIT)         774
016A 0B5800001800A005 775  STS2   GEN         /N(STS3),LA3,SC1,WF1,AA5                OVFL->OPR SIGN,            775
                      776  *                                                          1->BYTA                    776
016B 0B6000000000B000 777  STS3   GEN         /N(STS4),SC1,WF1,XF2                    OVFL->OPR SIGN,            777
                      778  *                                                          CARRY->OPR BIT 1           778
016C 0B6800000000B000 779  STS4   GEN         /N(STS5),SC1,WF1,XF2                    OVFL->OPR SIGN,            779
                      780  *                                                          CARRY->OPR BIT 0           780
016D 0B7000000018108 781   STS5   GEN         /N(STS6),SC1,WR1,SH1,AA8                OVFL->OPR BIT 0,           781
                      782  *                                                          CARRY->OPR BIT 1,          782
                      783  *                                                          0->R8 (STATUS)             783
                      784  STS6   GEN         /F(STSS1),2(1),FSE,LB1,FFA,MF1,         OPR->RE, FSEL(10)          784
016E 0B8B800020A9000E 785                    CWR1,AAE                                                           785
                      786         EVEN                                                                           786
                      787  STSS1  GETTAG      SHORT                                                              787
0170 04C8008043AFCFF0
                      788  STSL1  GETTGL      LONG                                                               788
0171 04A0088447AFCFF0
                      789  STS8   GMSK        /N(SS3M),SF1,IM8,RF4,LB3,FFB,           INCP, IF(P),               789
0172 0530040464BFFFCE 790                    CMKFFFC,AKE                             RE.3->ALU                  790
0173 0BA0000020A90011 791  STXL1  GEN         /N(STXL2),LB1,FFA,MF1,WR1,BB1,AA1       MIR -> R1                  791
0174 0BA8000000B90091 792  STXL2  GEN         /N(STXL3),FFB,MF1,WR1,BB9,AA1           R9.R1->R1                  792
                      793  *      STXL3 DOES: TEST ALUZ: T=> STXLP; F=> OF(OLSE+R0),FSEL(7)->STXLX                  793
0175 2C0AFA42A0900050 794  STXL3  GEN         /S(STXLX,STXLP),2(1),FSB,TF3,SF2,GF9,IM5,LB1,FF9,BB5               794
```

```
                         795          MORG    4                              STXLR, STXLI, STXLPI REF BY FSEL      795
0178 0C18040300900001    796  STXLR   GEN     /N(STXLR1),SF1,IM6,FF9,AA1                    OS(ALU), R1+R0        796
0179 0BD8000080A90091    797  STXLI   GEN     /N(STXLI1),IM1,FFA,MF1,WR1,BB9,AA1            WAIT(MEM), R9->R1     797
017A 0BC8040280910001    798  STXLPI  GEN     /N(STXLI),SF1,IM5,FF9,WR1,AA1                 OF(ALU), R1+R0->R1    798
017B 0BC0000020B90011    799  STXLI1  GEN     /N(STXLR),LB1,FFB,MF1,WR1,BB1,AA1             MIR.R1->R1            799
                         800          MORG    16                             STXLX, STXLX1 REF BY CND FSEL         800
0180 0C10000080000000    801  STXLX   GEN     /N(STXLX2),IM1                                WAIT(MEM=INDEX)       801
0181 0C10040280910001    802  STXLX1  GEN     /N(STXLX2),SF1,IM5,FF9,WR1,AA1                OF(ALU), R1+R0->R1    802
0182 4B8AC00023A80010    803  STXLX2  GEN     /F(STXLR),2(1),FSB,LB1,RF3,FFA,MF1,BB1        MIR->OPR, FSEL(7)     803
                         804  *       NOTE: DON'T START MEMORY AT STXLR1.  LET MPLE SETTLE.                       804
0183 0520000020A80000    805  STXLR1  GEN     /N(SS2M),LB1,FFA,MF1                          OPR->ALU              805
                         806  *       STXLP MUST HAVE THE SAME HIGH-ORDER 5 BITS AS STXLX                         806
                         807          EVEN                                                                        807
0184 4B92800000B660000   808  STXLP   GEN     /F(STXLR),2(2),FSA,LA1,RF3,FF6,CF3            P-R0->OPR, FSEL(7)    808
0185 0C30000043AFC000    809  STXS1   GMSK    /N(STXS2),LB2,FFA,RF3,MKFC00                  TAG & DISP -> OPR     809
                         810  *       STXS2 DOES:  R0+OLSE -> ALU, OF(ALU) IF NOT ALUZ                            810
0186 4C402A42A0900050    811  STXS2   GEN     /T(STXS3,STXS3),TF2,SF2,GF9,IM5,LB1,FF9,BB5                         811
                         812          EVEN                                                                        812
0188 0C48000008670001    813  STXS3   GEN     /N(STXS4),LA1,FF6,CF3,WR1,AA1                 P-R0->R1              813
0189 0C50000023900041    814  STXS4   GEN     /N(STXS5),LB1,RF3,FF9,BB4,AA1                 R1+ORSE->OPR          814
018A 0C58000023B80009    815  STXS5   GEN     /N(STXS6),LB1,RF3,FFB,MF1,AA9                 OPR.R9->OPR           815
                         816  STXS6   GEN     /T(STXSP,STXSX),TF2,SF1,GF9,IM6,LB1,          OPR+R0->OPR,          816
018B 7C60264323900000    817                  CRF3,FF9                                      OS(ALU)               817
                         818  *       NOTE: DON'T START MEMORY AT STXSP OR STXSX.  LET MPLE SETTLE.                818
018C 0520000003000001    819  STXSP   GEN     /N(SS2M),RF3,AA1                              R1->OPR               819
                         820          EVEN                                                                        820
018E 0520000023A80010    821  STXSX   GEN     /N(SS2M),LB1,RF3,FFA,MF1,BB1                  MIR->OPR              821
018F 0C80040281000001    822  XLATE1  GEN     /N(XLATE2),SF1,IM5,RF1,AA1                 OF(ALU),R1->P            822
                         823  XLATE2  GEN     /N(XLATE3),SF1,IM9,RF4,FF3,MF1,            OF(P),INCP,0->R1         823
0190 0C88040484390001    824                  CWR1,AA1                                                            824
0191 0CA8000022600110    825  XLATE3  GEN     /N(XLATE4),LB1,RF2,FF6,BB1,SH1             -MIR-1->SC               825
                         826          MORG    4                                                                   826
                         827  XLATE5  GEN     /P(VSS3M),SF2,GF4,IM4,RF1,CF3,             IF(ALU),W2+1->P          827
0194 016809020106000F    828                  C24(W2)                                                             828
                         829  XLATE4  GEN     /T(XLATE6,XLATE5),TF3,GFC,IM1,             WAIT(M),INC SC,R1+1->R1  829
0195 2CB0330085070001    830                  CRF5,CF3,WR1,AA1                                                    830
0196 0CB8008020660010    831  XLATE6  GEN     /N(XLATE7),GF2,LB1,FF6,CF3,BB1             R0-MIR,SAMPLE            831
                         832  XLATE7  GEN     /T(XLATE4,XLATE5),TF3,SF2,GF9,             CND(NO ALUZ) OF(P),INCP  832
0197 2CA83A4484000000    833                  CIM9,RF4                                                            833
0198 0570000020790001    834  XIO2    GEN     /N(E620MV),LB1,FF7,MF1,WR1,24(ADDR)    MAKE ADDRESS EVEN            834
```

```
                          836  *                                                                                  836
                          837  *       STEP KEY DETECT.  ENTER 620 HALT LOOP                                      837
                          838  *                                                                                  838
0199 0CD0000000067000D    839  STEP1   GEN      /N(STEP2),FF6,CF3,WR1,BB0,AAD          RD-R0->RD                839
019A 0CD8000001000000C    840  STEP2   GEN      /N(STEP3),RF1,AAC                      RC->P                    840
019B 0CE0000063AFFFA0     841  STEP3   GMSK     /N(STEP4),RF3,LB3,FFA,MKFFFA           5->OPERAND               841
019C 0FF80001A0A90002     842  STEP4   GEN      /P(X'1FF),IM3,LB1,FFA,MF1,WR1,AA2      OPERAND->R2              842
01FF                      843           ORG      X'1FF          STEP INTERRUPTS TO HERE                        843
01FF 0CC8000009F1000D     844  STEP    GEN      /N(STEP1),LA1,RF1,FFF,WR1,AAD          P-1 ->P,RD               844
```

846 END 846

SYMBOLS

0030 A1	00E8 A2	000A ACC	0031 AD1	00EC AD4			
00ED AD5	0001 ADDR	0090 ADDREX	0063 ADL	0062 ADS			
00E9 ADSD2	00EB ADSD3	0061 AL	003C AND1	00F1 AND2			
0079 ANDL	0078 ANDS	0060 AS	00EA ASDODD	0000 BASE			
0029 BSC1	00A8 BSC2	0053 BSCL	0052 BSCS	0028 BSI1			
00F2 BSI2	00F3 BSI3	0051 BSIL	0050 BSIS	0081 CBL			
00A9 CBL1	0080 CBS	008A CTEST1	0088 CTEST2	0086 CTEST3			
0084 CTEST4	0082 CTEST5	008C CTESTF	0083 CTST0A	008E CTST0B			
008D CTST1A	008B CTST2A	0089 CTST3A	008F CTST3B	0087 CTST4A			
0085 CTST5A	0035 D1	00F4 DIV1	00FF DIV10	0101 DIV11			
0100 DIV12	0102 DIV13	0103 DIV14	0104 DIV15	0105 DIV16			
0106 DIV17	0107 DIV18	0108 DIV19	00F5 DIV2	00F6 DIV3			
00F8 DIV4	00F9 DIV5	00FA DIV6	00FC DIV7	00FD DIV8			
00FE DIV9	0040 DJUMP	006B DL	006A DS	00AB E620I			
00AC E620M	00AD E620M1	00AE E620MV	0109 ENTR	010A ENTR1			
010C ENTRR0	010B ENTRS0	003E EOR1	010D EOR2	007D EORL			
007C EORS	000B EXT	0400 FFP	0005 GETREG	00A1 INT1			
00A2 INT2	00A3 INT3	00A0 INTBAS	0004 INTEN	010E INTEN0			
010F INTEN1	0110 INTEN2	0111 INTEN3	0112 INTEN4	00A7 IWAIT			
0038 LDA1	0113 LDA2	0071 LDAL	0070 LDAS	0039 LDD1			
0114 LDD2	0117 LDD3	0118 LDD4	0073 LDDL	0116 LDDODD			
0072 LDDS	0049 LDSL	011A LDSRC	011C LDSRO	0048 LDSS			
011B LDSSC	011D LDSSO	011E LDSX	0059 LDXL	012B LDXL1			
0126 LDXLNX	0122 LDXLX1	0123 LDXLX2	0121 LDXLXI	0124 LDXLXR			
0130 LDXPJ	0058 LDXS	0127 LDXS1	0129 LDXSP	0128 LDXSX			
012A LDXSX1	0092 LGNXI1	0093 LGNXI2	0001 LMODE1	0132 LMODE2			
0133 LMODE3	0134 LMODE4	0091 LNGNXI	0094 LONG	0095 LONG1			
0096 LONGX	0097 LONGX1	0098 LONGX2	0125 LXLNX1	0131 LXLNXI			
0120 LXLXNI	0034 M1	000C MBM	0135 MBM1	014A MBM10			
013C MBM11	013E MBM12	0141 MBM13	0142 MBM14	0144 MBM15			
0145 MBM16	0149 MBM17	0148 MBM18	014B MBM19	0136 MBM2			
0138 MBM3	0139 MBM4	013A MBM5	013B MBM6	013D MBM7			
013F MBM8	0140 MBM9	0137 MBMX	00B7 MDX	00C5 MDX1			
00B2 MDXC1	00BF MDXC2	00BE MDXC3	00C1 MDXC4	00C0 MDXC5			
00C2 MDXC6	00C3 MDXC7	00B5 MDXC8	00C4 MDXC9	005D MDXL			
00B6 MDXNI	00B8 MDXNX	00BB MDXNX1	00B9 MDXNX2	00BA MDXP			
005C MDXS	00B4 MDXSKP	00B1 MDXXL1	00BD MDXXL2	00BC MDXXL3			
00B0 MDXXS1	00B3 MDXXS2	0069 ML	0068 MS	0155 MUL10			
014C MUL2	014D MUL3	014E MUL4	014F MUL5	0150 MUL6			
0152 MUL7	0153 MUL8	0154 MUL9	0020 OP00	0041 OP00L			
0040 OP00S	003D OR1	0156 OR2	007B ORL	007A ORS			
000D P1130	000C P620	0200 PAGE1	0006 PSCAN1	0157 PSCAN2			
0158 PSCAN3	0159 PSCAN4	015A PSCAN5	015C PSCAN6	015D PSCAN7			
015E PSCAN8	015F PSCAN9	0160 PSCANA	0161 PSCANB	015B PSCANX			
0003 RETRNA	0002 RETURN	0032 S1	0162 S2	0163 S3			
0164 S4	0033 SD1	00EE SD4	00EF SD5	00F0 SD6			
0067 SDL	0066 SDS	00E6 SHFARG	00E7 SHFCOM	00DA SHFDR1			
00DC SHFMR1	00E5 SHFMSK	00D5 SHFT11	00D6 SHFT12	00D7 SHFT13			
00D8 SHFT31	00D9 SHFT32	00DB SHFT33	00DD SHFTBS	00C8 SHFTDF			
00CE SHFTL	00D0 SHFTL0	00D1 SHFTL1	00D2 SHFTL2	00D3 SHFTL3			
00D4 SHFTLX	00DF SHFTM	00CF SHFTR	00E0 SHFTR0	00E1 SHFTR1			
00E2 SHFTR2	00E3 SHFTR3	00E4 SHFTRX	00CA SHFTX	00CB SHFTX1			
00CC SHFTX2	0044 SHIFT	00C6 SHIFT1	00C7 SHIFT2	00CD SHIFT3			
0099 SHORT	009A SHORT1	009B SHORTP	009C SHORTX	009E SHRTP1			
009F SHRTR1	009D SHRTX1	0065 SL	0064 SS	00A0 SS1M			

```
00A4 SS2M   00A5 SS2MS  0020 SS2MX  00A6 SS3M   00DE SS3MX
0008 STAT   003B STD1   0165 STD2   0166 STD3   0167 STD4
00AA STDEVN 0077 STDL   0076 STDS   01FF STEP   0199 STEP1
019A STEP2  019B STEP3  019C STEP4  003A STO1   0168 STO2
0075 STOL   0074 STOS   0169 STS1   016A STS2   016B STS3
016C STS4   016D STS5   016E STS6   0025 STS7   0172 STS8
004B STSL   0171 STSL1  004A STSS   0170 STSS1  005B STXL
0173 STXL1  0174 STXL2  0175 STXL3  0179 STXLI  017B STXLI1
0184 STXLP  017A STXLPI 0178 STXLR  0183 STXLR1 0180 STXLX
0181 STXLX1 0182 STXLX2 005A STXS   0185 STXS1  0186 STXS2
0188 STXS3  0189 STXS4  018A STXS5  018B STXS6  018C STXSP
018E STXSX  00D1 VINT1  00B4 VSS2M  002D VSS3M  000E W1
000F W2     004C WAIT67 0054 WAITAB 005F WAITFL 005E WAITFS
0009 WRAP   006C WT1617 007F WT1FL  007E WT1FS  0021 XIO1
0198 XIO2   0043 XIOL   0042 XIOS   000D XLATE  018F XLATE1
0190 XLATE2 0191 XLATE3 0195 XLATE4 0194 XLATE5 0196 XLATE6
0197 XLATE7
  0 ERRORS ASSEMBLY COMPLETE
 /WEOF,BO
 /FINI
```