```
WVS*VORTEX(1),MICSIM
     1        *       MICROPROGRAM SIMULATOR.                                                  0001
     2        *                                                                                0002
     3        *       ALL INPUT VALUES ARE IN HEXADECIMAL.  KEYINS ARE:                        0003
     4        *       A:  ALTER OR DISPLAY REGISTERS.  AVAILABLE REGISTERS ARE                 0004
     5        *           A=ALU, B=IBR, C=SHIFT CNTR, I=IREG#2, JN=N'TH ENTRY IN JUMP          0005
     6        *           STACK, K=DATA LOOP KEY REGISTER, M=MIL, O=DOR, P=P,                  0006
     7        *           RN=N'TH GPR, S=STATUS MUX, TN=N'TH TEST MUX ENTRY.                   0007
     8        *       BX; BEGIN SIMULATED EXECUTION AT LOCATION X.                             0008
     9        *       CX; CHANGE/DISPLAY MEMORY LOCATION X.                                    0009
    10        *       D:  DUMP CCS, DCS OR MEMORY.  VARIANTS ARE DCX,Y=DUMP CCS FROM           0010
    11        *           LOCATION X TO Y, DDA=DUMP DECODER A, DDB=DUMP DECODER B,             0011
    12        *           DMX,Y=DUMP USER MEMORY AREA FROM LOCATION X TO Y.                    0012
    13        *       E:  EDIT CCS OR DECODER.  VARIANTS ARE ECX=EDIT CCS LOCATION             0013
    14        *           X, EDAX=EDIT DECODER A LOCATION X, EDBX=EDIT DECODER B               0014
    15        *           LOCATION X.  AFTER ECX, INPUT MAY BE A HEXADECIMAL STRING OR         0015
    16        *           THE FORM +FFV,FFV,FFV,... TO CHANGE FIELDS FF TO VALUES V, OR        0016
    17        *           THE FORM =FFV,FFV,FFV,... TO CLEAR THE CCS WORD AND INPUT            0017
    18        *           ONLY THE FIELDS FF WITH VALUES V.                                    0018
    19        *       FX,Y; DUMP CCS LOCATIONS X TO Y IN FIELD-EDITED FORMAT.                  0019
    20        *       G:  DUMP ENTIRE GPR FILE.                                                0020
    21        *       HX,Y,Z,W; SET HALTS IN CURRENT PAGE AT X,Y,Z,W.                          0021
    22        *           DEFAULTS ARE X'200 (OUTSIDE OF PAGE).                                0022
    23        *       I;  INITIALIZE.                                                          0023
    24        *       L;  LOAD CCS, DECODER OR MEMORY FROM BI.  VARIANTS ARE                   0024
    25        *           LC=LOAD CCS, LDA=LOAD DECODER A, LDB=LOAD DECODER B, LM=LOAD         0025
    26        *           MEMORY.                                                              0026
    27        *       M*; SET INPUT MEDIUM.  IF *=S, INPUT = PI, IF *=R, INPUT = SI            0027
    28        *       PX; SELECT PAGE X. X=0-3.                                                0028
    29        *       QX; SELECT MEMORY TYPE.  X=3-5.  DEFAULT IS SET BY CONFIGURATION         0029
    30        *           PARAMETER "MEMTPD".                                                  0030
    31        *       R;  RETURN TO OPERATING SYSTEM.                                          0031
    32        *       S;  SINGLE STEP OR SET RUN COUNT.  VARIANTS ARE SS=SET SINGLE            0032
    33        *           STEP MODE, SRX=SET CONTINUOUS MODE, RUN X STEPS AFTER B.             0033
    34        *       T:  SET AND RESET TRACE MODE.  VARIANTS ARE TR=RESET TRACE MODE,         0034
    35        *           TSX,Y=SET TRACE MODE WITH BOUNDARIES X,Y (DEFAULT=0,X'1FF).          0035
    36        *       Z:  ZERO TABLES.                                                         0036
    37        *                                                                                0037
    38        *       ERROR MESSAGES;                                                          0038
    39        *                                                                                0039
    40        *       MS01 = INVALID INPUT COMMAND.                                            0040
    41        *       MS02 = INVALID HEXADECIMAL CHARACTER IN INPUT.                           0041
    42        *       MS05 = PAGE JUMP TO INACCESSIBLE PAGE.                                   0042
    43        *       MS06 = BCS INSTRUCTION, BUT NO PAGE 1.                                   0043
    44        *       MS07 = READ ERROR ON BI DEVICE.                                          0044
    45        *       MS08 = EOF ENCOUNTERED BEFORE LOAD COMPLETE.                             0045
    46        *       MS09 = END OF DEVICE BEFORE LOAD COMPLETE.                               0046
    47        *       MS10 = SEQUENCE ERROR ON BI.                                             0047
    48        *       MS11 = INVALID LOADER CODE.                                              0048
    49        *       MS12 = CHECKSUM ERROR DURING LOAD.                                       0049
    50        *       MS13 = UNDEFINED MEMORY OP CODE.                                         0050
    51        *       MS14 = ATTEMPT BY MICROPROGRAM TO STORE OUTSIDE USER MEMORY AREA.        0051
    52        *       MS15 = ATTEMPT TO LOAD FROM BI OUTSIDE OF USER MEMORY AREA.              0052
    53        *       MS16 = FIELD NAME ERROR IN ECX INPUT.                                    0053
    54        *       MS17 = FIELD SIZE ERROR IN ECX INPUT.                                    0054
    55                EJEC                                                                     0055
    56        VORTEX SET     1              PUT LAST FOR VORTEX                          VDM 0056
```

```
57      DIVIDE SET     1              SET DIVIDE = 1 IF HARDWARE DIVIDE IS            0057
58      *                             TO BE USED, SET TO ZERO OTHERWISE.              0058
59      JNOT   SET     1              JNOT=1 IF JUMP NOT INSTRUCTIONS ARE             0059
60      *                             AVAILABLE, AND JNOT=0 OTHERWISE.                0060
61      MEMTPD SET     4              MEMORY TYPE DEFAULT: 3=SC, 4=CORE, 5=SLOW       0061
62      *                             MEMORY TYPE CAN BE CHANGED WITH "Q" KEYIN.      0062
63      *                                                                        VDM 0063
64      *    V.D.M.  PART NO.         92L1406-001B                               VDM 0064
65      *                                                                        VDM 0065
66      *              MICSIM                                                    VDM 0066
67      *                                                                        VDM 0067
68      ********************************************************************** VDM 0068
69      *                                                                        VDM 0069
70      *      THE FOLLOWING ARE THE MACRO DEFINITIONS NECESSARY TO RUN                0070
71      *      THIS PROGRAM USING THE INTERFACE SUBPROGRAM                       VDM 0071
72      *              THEIR EFFECT IS TO CHANGE VORTEX CALLS TO V$IOC, V$IOST,  VDM 0072
73      *              AND V$EXEC TO CALLS TO THE INTERFACE SUBPROGRAM           VDM 0073
74              TITLE  MICSIM                                                    VDM 0074
75      M1      MAC                                                              VDM 0075
76              EXT    INTR                                                      VDM 0076
77              JMPM   INTR                                                      VDM 0077
78              DATA   0100000                                                   VDM 0078
79      F       FORM   1,3,4,8                                                   VDM 0079
80              F      P(1),P(2),P(3),P(4)                                       VDM 0080
81              DATA   P(5),0,0                                                  VDM 0081
82              EMAC                                                             VDM 0082
83      STAT    MAC                                                              VDM 0083
84              EXT    INTRST                                                    VDM 0084
85              JMPM   INTRST                                                    VDM 0085
86              DATA   P(1),P(2),P(3),P(4),P(5)                                  VDM 0086
87              EMAC                                                             VDM 0087
88      IOLINK  MAC                                                              VDM 0088
89              EXT    INLINK                                                    VDM 0089
90              JMPM   INLINK                                                    VDM 0090
91      F       FORM   4,6,6                                                     VDM 0091
92              F      0,014,P(1)                                                VDM 0092
93              DATA   P(2),P(3)                                                 VDM 0093
94              EMAC                                                             VDM 0094
95      *       THE FOLLOWING MACROS MAKE USE OF THE JUMP "NOT" INSTRUCTIONS            0095
96      *       IF AVAILABLE.  AVAILABILITY IS DETERMINED BY SETTING OF JNOT.           0096
97      *                                                                               0097
98              IFF    1-JNOT                                                           0098
99              GOTO   MJNZ1                                                            0099
100     MJANZ   MAC                                                                     0100
101             JAZ      *+4                                                            0101
102             JMP    P(1)                                                             0102
103             EMAC                                                                    0103
104     MJBNZ   MAC                                                                     0104
105             JBZ    *+4                                                              0105
106             JMP    P(1)                                                             0106
107             EMAC                                                                    0107
108             GOTO   MJNZ2                                                            0108
109     MJNZ1   CONT                                                                    0109
110     MJANZ   MAC                                                                     0110
111             JANZ   P(1)                                                             0111
112             EMAC                                                                    0112
113     MJBNZ   MAC                                                                     0113
```

```
114                JBNZ    P(1)                                                  0114
115                EMAC                                                          0115
116       MJNZ2    CONT                                                          0116
117       *                                                                      0117
118       SKIP     OPSY    DATA                                                  0118
119       NEXT     EQU     01006                                                 0119
120                EJEC                                                    VDM   0120
121                NAME    SMLTR                                           VDM   0121
122       X        EQU     1                                                     0122
123       B        EQU     2                                                     0123
124       SI       EQU     2               SYSTEM INPUT                          0124
125       SO       EQU     3               SYSTEM OUTPUT                         0125
126       PI       EQU     4               PROCESSOR INPUT                       0126
127       LO       EQU     5               LIST OUTPUT                           0127
128       BI       EQU     6               BINARY INPUT                          0128
129       *                                                                      0129
130       *        EXTERNAL REFERENCES                                           0130
131       *                                                                      0131
132       *        FILE CONTROL BLOCKS                                           0132
133                EXT     BIFCB,LOFCB,PIFCB,SIFCB                               0133
134       *        INTERFACE PROGRAM                                             0134
135                EXT     $DATE,$ELOC,EXIT,$JOB,$LUT,$SYST,V$LLUP               0135
136       *        SEPARATELY ASSEMBLED DATA AREAS                               0136
137                EXT     $DBUF           620 EMULATOR ROM                      0137
138                EXT     $DRM1,$DRM2     620 EMULATOR DECODERS                 0138
139                EXT     $DRM1B,$DRM1C,$DRM1D    DECODERS FOR PAGES 1, 2 & 3.  0139
140       *        VORTEX                                                        0140
141       V$DSTB   EQU     0355            DST POINTER                           0141
142       V$JCB    EQU     0412            ADDRESS OF JCP INPUT BUFFER           0142
143       V$LCNT   EQU     054             LINE COUNT FROM JCP /FORM DIRECTIVE   0143
144       V$LUT1   EQU     0400            LUT POINTER                           0144
145       *                                                                      0145
146       *        NOTE - THE FOLLOWING SECTION OF CODE WILL BE OVERLAID AND     0146
147       *        CLEARED TO ZERO AFTER INITIALIZATION, TO BE USED FOR BUFFERS. 0147
148       *                                                                      0148
149       *        THIS IS SOME INITIALIZATION NEEDED AFTER LOADING IN ORDER TO  *  VDM 0149
150       *        SET UP FCBS AND DCBS FOR THE LOGICAL UNITS SI,PI,LO,AND BI    *     0150
151       *        DEPENDING UPON WHETHER THEY ARE ON A RMD OR NON-RMD DEVICE.   *  VDM 0151
152       *                                                                      *  VDM 0152
153       *                IF ON A RMD THE GLOBAL FCB WILL BE USED               *  VDM 0153
154       *                IF ON A NON-RMD A LOCAL DCB WILL BE USED              *  VDM 0154
155       *                                                                      *  VDM 0155
156       *                                                                      0156
157       *        DETERMINE IF SO = LO.                                         0157
158       *                                                                      0158
159       SMLTR    LDXE*   $LUT            LUT BASE                              0159
160                LDA     SO,1            SO ASSIGNMENT                         0160
161                ERA     LO,1            LO ASSIGNMENT                         0161
162                ANA     =0377                                                 0162
163                STAE    SOELO                                                 0163
164                LDAE    $SYST           GET SYSTEM FLAG                       0164
165                JAN     HDRSET          DON'T CHECK FOR RMD IF MOS            0165
166                LDA     V$LCNT          GET VORTEX LINE COUNT                 0166
167                SUB     =4              ALLOW 4 LINES FOR HEADER, BOTTOM MARGIN 0167
168                STAE    LOLMAX          SET PAGE LENGTH                       0168
169                LDAI    TPVOR           ADDRESS OF 'VORTEX' FOR HEADER        0169
170                STA     TPHDR           STORE ADDRESS OF SYSTEM TITLE FOR HEADER 0170
```

```
171                CALL    RMD,SI          IS SI ON RMD?                                  0171
172                JAZ     SIRMD           YES                                    VDM     0172
173       SMPI     CALL    RMD,PI          IS PI ON RMD?                                  0173
174                JAZ     PIRMD           YES                                    VDM     0174
175       SMLO     CALL    RMD,LO          IS LO ON RMD?                                  0175
176                JAZ     LORMD           YES                                    VDM     0176
177       SMBI     CALL    RMD,BI          IS BI ON RMD?                                  0177
178                JAZ     BIRMD           YES                                    VDM     0178
179       *                                                                               0179
180       *        SETUP PART OF HEADER LINE.  DO NOT SETUP DATE NOW.  ;DATE COULD        0180
181       *        CHANGE DATE DURING EXECUTION.                                          0181
182       *                                                                               0182
183       HDRSET   LDXE*   $JOB                                                          0183
184                LDBI    HDR+12                                                         0184
185                CALL    TPST            STORE JOB NAME IN HEADER                       0185
186                LDX     TPHDR                                                          0186
187                LDBI    HDR+17                                                         0187
188                CALL    TPST            STORE SYSTEM NAME IN HEADER                    0188
189       *                                                                               0189
190       *        CONTINUE INITIALIZATION                                                0190
191       *                                                                               0191
192                CALL    TPFRM           TOP OF FORM ON LO                              0192
193                CALL    SIOUT,EXC2L,EXC2   PROGRAM ID LINE                             0193
194       *        COMPUTE PAGE LIMIT                                                     0194
195                LDAE*   VSLLUP         LAST ALLOCATED CELL                            0195
196                SUBE*   $ELOC          LAST LOADED CELL                               0196
197                ASRA    11              NUMBER OF PAGES (4*512 WORDS EACH)             0197
198                ADD     =DROM           USED FOR INDEXING IN PGOK                      0198
199                STA     PAG4                                                           0199
200                SUB     =10+DROM                                                       0200
201                JAN     HDR10                                                          0201
202                ADD     =' 10'-' 0'-10                                                 0202
203       HDR10    ADD     =' 0'+10        CONVERT TO ASCII                               0203
204                STA     PAG5+6                                                         0204
205                CALL    SIOUT,PAG5L,PAG5                                               0205
206       *        COMPUTE PAGE ADDRESSES                                                 0206
207                LDAE*   $ELOC           LAST LOADED CELL                               0207
208                IAR                                                                    0208
209                STA     PSTBL+1         ADDRESS OF PAGE 1                              0209
210                ADD     =2048           1 PAGE = 4*512 WORDS                           0210
211                STA     PSTBL+2         ADDRESS OF PAGE 2                              0211
212                ADD     =2048                                                          0212
213                STA     PSTBL+3         ADDRESS OF PAGE 3                              0213
214                JMP     EXC             BEGIN SIMULATOR EXEC                   VDM     0214
215       TPHDR    DATA    TPMOS                                                          0215
216       TPMOS    DATA    '  MOS   '                                                     0216
217                IFF     VORTEX-2                                                       0217
218       TPVOR    DATA    'VORTX II'                                                     0218
219                IFF     VORTEX-1                                                       0219
220       TPVOR    DATA    ' VORTEX '                                                     0220
221       EXC2     DATA    ' V70 MICSIM F'                                                0221
222       EXC2L    EQU     *-EXC2                                                         0222
223       PAG5     DATA    '  PAGE LIMIT *'                                               0223
224       PAG5L    EQU     *-PAG5                                                         0224
225       *                                                                               0225
226       *  NOTE - RMD CHECK ROUTINE SETS B NEGATIVE                                     0226
227       SIRMD    STBE    SIFLG           SET SI ON RMD FLAG NEGATIVE                    0227
```

```
228             LDAI    SIFCB                                           0228
229             STAE    SIIN1+4         STORE GLOBAL FCB IN READ REQUEST 0229
230             JMP     SMPI                                        VDM 0230
231     PIRMD   STBE    PIFLG           SET PI ON RMD FLAG NEGATIVE     0231
232             LDX     PIOPN+4                                         0232
233             STXE    PIIN1+4         STORE GLOBAL FCB IN READ REQUEST 0233
234             LDA     3,1             CURRENT POSITION OF FILE    VDM 0234
235             DAR                                                 VDM 0235
236             JAP     SMLO            JUMP IF ALREADY OPEN            0236
237     PIOPN   OPEN    PIFCB,PI,0,0    OPEN AND REWIND PIFCB           0237
238             JMP     SMLO                                        VDM 0238
239     LORMD   STBE    LOFLG           SET LO ON RMD FLAG NEGATIVE     0239
240             LDX     LOOPN+4                                         0240
241             STXE    LO1+4           STORE GLOBAL FCB IN WRITE REQUEST 0241
242             LDA     3,1             CURRENT POSITION OF FILE    VDM 0242
243             DAR                                                 VDM 0243
244             JAP     SMBI            JUMP IF ALREADY OPEN            0244
245     LOOPN   OPEN    LOFCB,LO,0,0    OPEN AND REWIND LOFCB           0245
246             JMP     SMBI                                        VDM 0246
247     BIRMD   STBE    BIFLG           SET BI ON RMD FLAG NEGATIVE     0247
248             LDA     =120                                            0248
249             STAE    BIBLK           SET WORD COUNT TO 120           0249
250             LDX     BIOPN+4                                         0250
251             STXE    LREAD+4         STORE GLOBAL FCB IN READ REQUEST 0251
252             LDA     3,1             CURRENT POSITION OF FILE    VDM 0252
253             DAR                                                 VDM 0253
254             JAP     HDRSET          JUMP IF ALREADY OPEN            0254
255     BIOPN   OPEN    BIFCB,BI,0,0    OPEN AND REWIND BIFCB           0255
256             JMP     HDRSET                                          0256
257     *                                                               0257
258     *       THIS ROUTINE DETERMINES IF THE LOGICAL UNIT IS ON RMD   0258
259     *       THIS ROUTINE IS USED ONLY IF VORTEX IS THE OPERATING SYSTEM 0259
260     *                                                               0260
261     *       CALLING SEQUENCE                                        0261
262     *       CALL RMD,LUN                                            0262
263     *       RETURN                  A=0 IF LUN IS ON RMD AND THE OPERATING 0263
264     *                               SYSTEM IS VORTEX.               0264
265     *                                                               0265
266     RMDO    LDB     RMD                                             0266
267             INR     RMD             UPDATE RETURN ADDRESS           0267
268             LDA     0,2             LUN PARAMETER ON CALL           0268
269             ADD     V$LUT1          ADD LUT BASE TO GET LUT ENTRY ADDRESS 0269
270             TAB                                                     0270
271             LDA     0,2             GET LUN ENTRY                   0271
272             ANA     =0377           EXTRACT RIGHT HALFWORD          0272
273             DAR                                                     0273
274             STA     *+3                                             0274
275             ASLA    1                                               0275
276             ADDI    *-*             MULTIPLY BY 3                   0276
277             ADD     V$DST,B         ADD DST BASE                    0277
278             TAB                                                     0278
279             LDA     1,2             FIRST WORD OF DST               0279
280             LSRA    8               RIGHT JUSTIFY LEFT CHARACTER OF EQUIP CODE 0280
281             SUB     ='D'                                            0281
282             DECR    2               SET B NEGATIVE = USED TO SET FLAGS 0282
283             JMP     *-*                                             0283
284     RMD     BES     0                                               0284
```

```
285                 JMP     RMDO                                                    0285
286     ********************************************************************** VDM 0286
287                 EJEC                                                        VDM 0287
288     *                                                                           0288
289     *       AREAS CO-RESIDENT WITH INITIALIZATION                               0289
290     *                                                                           0290
291     MMEM    EQU     SMLTR           MAIN MEMORY BLOCK                         0291
292     EMEM    EQU     SMLTR+0177      END OF MAIN MEMORY                        0292
293     BUFR    EQU     EMEM+1          PRINTER BUFFER                            0293
294     BUFI    EQU     BUFR+121        INPUT BUFFER                              0294
295     DROM    EQU     BUFI+36         MACHINE SIGNALS                           0295
296     *                                                                           0296
297     *       WORDS ADDRESSED RELATIVE TO (X)=DROM                                0297
298     *                                                                           0298
299     *       OUTPUT SIGNALS FROM ROM                                             0299
300     *                                                                           0300
301     CQ1     EQU     0               FIRST QUARTER OF CURRENT CCS WORD         0301
302     CQ2     EQU     CQ1+1           SECOND QUARTER                            0302
303     CQ3     EQU     CQ2+1           THIRD QUARTER                             0303
304     CQ4     EQU     CQ3+1           FOURTH QUARTER                            0304
305     XTS     EQU     CQ4+1   TS      TEST ADDRESS AND B OFFSET                 0305
306     XAF     EQU     XTS+1   AF      CONTROL STORE ADDRESS                     0306
307     XMS     EQU     XAF+1   MS      CONTROL ADDRESS MASK                      0307
308     XMT     EQU     XMS+1   MT                                                0308
309     XFS     EQU     XMT+1   FS      CONTROL FIELD SELECT GATING               0309
310     XT      EQU     XFS+1   T       TEST CONTROL                              0310
311     XS      EQU     XT+1    S       SPECIAL CONTROL                           0311
312     XG      EQU     XS+1    G       GENERAL CONTROL                           0312
313     XM      EQU     XG+1    M       FILE ADDRESS EXTRACTION MASK              0313
314     XAB     EQU     XM+1    AB      FILE ADDRESS LOAD CONTROL                 0314
315     XIMC    EQU     XAB+1   IMC     I/O AND MEMORY CONTROL                    0315
316     XLB     EQU     XIMC+1  LB      LATCH B CONTROL                           0316
317     XLA     EQU     XLB+1   LA      LATCH A CONTROL                           0317
318     XR      EQU     XLA+1   R       REGISTER CONTROL                          0318
319     XF      EQU     XR+1    F       ALU FUNCTION                              0319
320     XMD     EQU     XF+1    MD      ALU MODE CONTROL                          0320
321     XC      EQU     XMD+1   C       CARRY CONTROL                             0321
322     XW      EQU     XC+1    W       WRITE FILE CONTROL                        0322
323     XOS     EQU     XW+1    OS      OP REG SHIFT/NO-SHIFT CONTROL             0323
324     XV      EQU     XOS+1   V       OP REG GATING CONTROL                     0324
325     XY      EQU     XV+1    Y       CONTROL FOR QS (MUL/DIV OPS)              0325
326     XX      EQU     XY+1    X       OP REG LONG SHIFTING CONTROL              0326
327     XTC     EQU     XX+1    TC      OVERFLOW CONTROL                          0327
328     XB      EQU     XTC+1   B       FILE B SOURCE ADDRESS, B MUX SELECTOR     0328
329     XA      EQU     XB+1    A       FILE A SOURCE ADDRESS, FILE A&B DESTINATION 0329
330     *                                                                         0330
331     MASK    EQU     XA+1            LITERAL MASK, FIELDS MD-B COMPLEMENTED    0331
332     *                                                                         0332
333     DFA     EQU     MASK+1          DATA LOOP FILE A OUTPUT                   0333
334     LTCHA   EQU     DFA+1           DATA LOOP LATCH A INPUT                   0334
335     LTCHB   EQU     LTCHA+1         DATA LOOP LATCH B INPUT                   0335
336     XALU    EQU     LTCHB+1         ALU OUTPUT                                0336
337     XCIN    EQU     XALU+1          CARRY IN                                  0337
338     CARRY   EQU     XCIN+1          STORED CARRY                              0338
339     OVRFL   EQU     CARRY+1         OVERFLOW                                  0339
340     *                                                                         0340
341     IOD     EQU     OVRFL+1         I/O DATA REGISTER                         0341
```

```
342       STUS    EQU     IOD+1           STATUS MULTIPLEXOR                                  0342
343       *                                                                                   0343
344       *       THE NEXT 7 MUST REMAIN IN ORDER TO MAINTAIN PROPER INDEXING                 0344
345       *                                                                                   0345
346       PREG    EQU     STUS+1          PROGRAM COUNTER                                     0346
347       SREG    EQU     PREG+1          SHIFT COUNTER                                       0347
348       OREG    EQU     SREG+1          OPERAND REGISTER                                    0348
349       FIL1    EQU     OREG+1          FILLER FOR INDEXING PURPOSES                        0349
350       FIL2    EQU     FIL1+1          FILLER FOR INDEXING PURPOSES                        0350
351       KREG    EQU     FIL2+1          KEY REGISTER                                        0351
352       IOKR    EQU     KREG+1          I/O KEY REGISTER                                    0352
353       *                                                                                   0353
354       IRG1    EQU     IOKR+1          INSTRUCTION REGISTER #1                             0354
355       IRG2    EQU     IRG1+1          INSTRUCTION REGISTER #2                             0355
356       XIRG2   EQU     IRG2+1          USED FOR R-R XFER.  SEE X108.                       0356
357       *                                                                                   0357
358       MADS    EQU     XIRG2+1         MEMORY ADDRESS SOURCE                               0358
359       MOPC    EQU     MADS+1          MEMORY OPERATION CODE                               0359
360       MCCO    EQU     MOPC+1          MEMORY CONDITION CODE                               0360
361       MIL     EQU     MCCO+1          MEMORY INTERFACE LATCH                              0361
362       MBYC    EQU     MIL+1           MEMORY BYTE SELECTOR:  1=RIGHT, 0=LEFT              0362
363       *                                                                                   0363
364       INTF    EQU     MBYC+1          INTERRUPT PENDING FLAG                              0364
365       INTA    EQU     INTF+1          INTERRUPT ADDRESS FROM I/O                          0365
366       QS      EQU     INTA+1          SAVE ALU15/OR01 DURING MUL/DIV OPS                  0366
367       DS      EQU     QS+1            SIGN STORAGE FOR OREG SHIFTING                      0367
368       SUPK    EQU     DS+1            SUPERVISOR KEY                                      0368
369       IORF    EQU     SUPK+1          I/O REQUEST FLAG      >>> I/O NOT SIMULATED         0369
370       INPF    EQU     IORF+1          INTERRUPT BEING PROCESSED FLAG                      0370
371       LREG    EQU     INPF+1          MEMORY ADDRESS REGISTER                             0371
372       MULS    EQU     LREG+1          MULTIPLY SIGN BIT                                   0372
373       NSTP    EQU     MULS+1          STEP F/F              >>> STEP NOT SIMULATED        0373
374       *                                                                                   0374
375       JSPT    EQU     NSTP+1          JUMP STACK POINTER                                  0375
376       IMD     EQU     JSPT+1          FLAG FOR IMC=X'D                                    0376
377       NROM    EQU     IMD+1           CURRENT ROM ADDRESS                                 0377
378       PROM    EQU     NROM+1          PREVIOUS ROM ADDRESS                                0378
379       *                                                                                   0379
380       DRL     EQU     PROM+1          DROM BUFFER LENGTH                                  0380
381       STACK   EQU     DROM+DRL        PUSH-POP STACK                                      0381
382       *                                                                                   0382
383       *       REGISTER FILE                                                               0383
384       *                                                                                   0384
385       R0      EQU     STACK+16                                                            0385
386       R1      EQU     R0+1                                                                0386
387       R2      EQU     R1+1                                                                0387
388       R3      EQU     R2+1                                                                0388
389       R4      EQU     R3+1                                                                0389
390       R5      EQU     R4+1                                                                0390
391       R6      EQU     R5+1                                                                0391
392       R7      EQU     R6+1                                                                0392
393       R8      EQU     R7+1                                                                0393
394       R9      EQU     R8+1                                                                0394
395       RA      EQU     R9+1                                                                0395
396       RB      EQU     RA+1                                                                0396
397       RC      EQU     RB+1                                                                0397
398       RD      EQU     RC+1                                                                0398
```

```
399       RE      EQU     RD+1                                                    0399
400       RF      EQU     RE+1                                                    0400
401       *                                                                       0401
402       *       TMUX IS INDEXED BY G FIELD                                      0402
403       *       ENTRY = 1 IF CONDITION IS TRUE OR SENSE SWITCH IS ON            0403
404       *       ENTRY = 0 IF CONDITION IS FALSE OR SENSE SWITCH IS OFF          0404
405       *                                                                       0405
406       *       G=0     ALU OVERFLOW                                            0406
407       *       G=1     I/O SENSE                                               0407
408       *       G=2     SENSE SWITCH 3                                          0408
409       *       G=3     SENSE SWITCH 2                                          0409
410       *       G=4     SENSE SWITCH 1                                          0410
411       *       G=5     620/F TEST (JMP, JMPM, XEC)                             0411
412       *       G=6     EQUALS F/F                                              0412
413       *       G=7     SIGN F/F                                                0413
414       *       G=8     CARRY F/F                                               0414
415       *       G=9     ALU ZERO                                                0415
416       *       G=10    DS F/F                                                  0416
417       *       G=11    MIL SIGN                                                0417
418       *       G=12    SHIFT COUNT = -1                                        0418
419       *       G=13    GPR 0 SIGN                                              0419
420       *       G=14    DAL15 NOT EQUAL DAL14 (NORMALIZED SHIFT)                0420
421       *       G=15    QS F/F                                                  0421
422       *                                                                       0422
423       TMUX    EQU     RF+1                                                    0423
424       TMUXX   EQU     TMUX-DROM       USED TO ACCESS TMUX USING (X)=DROM      0424
425       ZSIZE   EQU     TMUX+16-MMEM    SIZE OF AREA OVER WHICH "Z" DIRECTIVE WORKS 0425
426       TRACE   EQU     TMUX+16         TRACE FLAGS                             0426
427       STEP    EQU     TRACE+4         SINGLE STEP FLAG                        0427
428       CPAG    EQU     STEP-DROM+1     CURRENT PAGE NUMBER                     0428
429       PPAG    EQU     CPAG+1          PREVIOUS PAGE NUMBER                    0429
430       PTRC    EQU     PPAG+1          TRACE FLAG OF PPAG                      0430
431       NTRC    EQU     PTRC+1          TRACE FLAG OF CPAG                      0431
432       V       EQU     NTRC+DROM+1     HEX INPUT STRING                        0432
433       MTYP    EQU     V+4-DROM        MEMORY TYPE; 3=SC, 4=CORE, 5=SLOW CORE  0433
434       TBLEN   EQU     V+5-MMEM        TOTAL TABLE LENGTH                      0434
435               BSS     TBLEN+MMEM+1-*      MAKE ROOM FOR TABLES                0435
436       *                                                                       0436
437       *       CCS                                                             0437
438       *                                                                       0438
439       DBUF    DATA    $DBUF           CURRENT CONTROL STORE PAGE              0439
440       PSTBL   DATA    $DBUF           CONTROL STORE PAGE ZERO                 0440
441               BSS     3               SPACE FOR PAGES 1-3 ADDRESSES           0441
442       *                                                                       0442
443       *       DECODERS                                                        0443
444       *                                                                       0444
445       DRM1    DATA    $DRM1           CURRENTLY SELECTED DECODER 1 (B)        0445
446       DRM2    DATA    $DRM2           CURRENTLY SELECTED DECODER 2 (A)        0446
447       DCTBL   DATA    $DRM1           PAGE ZERO DECODERS                      0447
448               DATA    $DRM1B          PAGE 1 DECODERS                         0448
449               DATA    $DRM1C          PAGE 2 DECODERS                         0449
450               DATA    $DRM1D          PAGE 3 DECODERS                         0450
451       *                                                                       0451
452       *       HALT ADDRESS TABLES                                             0452
453       *                                                                       0453
454       RHLT    DATA    RHLT0,RHLT1,RHLT2,RHLT3                                 0454
455       RHLT0   BSS     5                                                       0455
```

```
456       RHLT1  BSS     5                                              0456
457       RHLT2  BSS     5                                              0457
458       RHLT3  BSS     5                                              0458
459              EJEC                                                VDM 0459
460       *                                                          VDM 0460
461       *      SIMULATOR EXECUTIVE                                 VDM 0461
462       *                                                          VDM 0462
463              NAME    EXC                                            0463
464       EXC    TZA                                                    0464
465              LDX     =DROM          ADDRESS OF DATA BUFFER          0465
466              STA     STEP-DROM,1    SET RUN MODE                    0466
467              STAE    LOLINE         SET LO CURRENT LINE COUNT       0467
468              CALL    PGOK           INITIALIZE TO PAGE ZERO         0468
469              LDB     =TRACE                                         0469
470       *      PGOK SETS A=-1                                         0470
471              STA     0,2            SET TRACE                       0471
472              STA     1,2            ON FOR                          0472
473              STA     2,2            ALL                             0473
474              STA     3,2            PAGES                           0474
475              STA     NTRC,1         SET TRACE FLAG                  0475
476              STA     PTRC,1         FOR PAGE                        0476
477              STA     RCNT           SET CONTINUOUS RUN COUNT TO INFINITY 0477
478              INCR    1                                              0478
479              STAE    PGNO                                           0479
480       *      SET DEFAULT MEMORY TYPE.                               0480
481              LDA     =MEMTPD                                        0481
482              STA     MTYP,1         MEMORY TYPE MAY BE CHANGED WITH "Q" KEYIN. 0482
483              TZA                                                    0483
484              CALL    SAR,4,STRTRC   INITIALIZE START TRACE          0484
485              LDA     =0777                                          0485
486              CALL    SAR,4,ENDTRC   INITIALIZE END TRACE            0486
487              IAR                                                    0487
488              CALL    SAR,20,RHLTO   INITIALIZE HALT TABLES          0488
489              LDA     ='  '                                          0489
490              STAE    BUFI-1                                         0490
491              STAE    BUFR                                           0491
492              ROF                    RESET HARDWARE OVERFLOW F/F     0492
493       EXCS   TZA                                                    0493
494              CALL    SAR,ZSIZE,MMEM     CLEAR TABLES                0494
495              EJEC                                                VDM 0495
496       *                                                          VDM 0496
497       *      PROCESS OPERATOR INPUT                              VDM 0497
498       *                                                          VDM 0498
499       EXC10  LDAE    MEDIA          GET MEDIUM SELECTION.           0499
500              JAN     EXC11          SKIP SOLICITATION IF MEDIUM=PI  0500
501              INRE    SONOLO         SUPRESS LO COPY                 0501
502              CALL    ERRMSC,0       SOLICIT INPUT.                  0502
503       EXC11  JMPM    FETCH          FETCH FIRST CHARACTER FROM INPUT BUFFER 0503
504              SUB     =('Z'+1)                                       0504
505              JAP     EXC90          INVALID IF > Z                  0505
506              SUB     =('A'-'Z'-1)                                   0506
507              JAN     EXC90          INVALID IF < A                  0507
508              ADD     COMM                                           0508
509              STA     *+3                                            0509
510              LDX     =DROM          DATA BUFFER ADDRESS             0510
511              JMP*    *-*            GO TO PROCESSING DEPENDING UPON INPUT 0511
512       *                                                             0512
```

```
513     *       INPUT COMMAND PROCESSING ADDRESS TABLE                          0513
514     *                                                                       0514
515     COMM    DATA    (*+1)*                                                  0515
516             DATA    E100            A  ALTER/DISPLAY REGISTERS              0516
517             DATA    S100            B  BEGIN SIMULATED EXECUTION            0517
518             DATA    CMEM            C  CHANGE/DISPLAY MEMORY                0518
519             DATA    E150            D  DUMP CONTROL STORE, DECODER OR MEMORY 0519
520             DATA    EDRM            E  CHANGE/DISPLAY CONTROL STORE OR DECODER 0520
521             DATA    FDMP            F  DUMP CCS IN FIELD-EDITED FORMAT      0521
522             DATA    GPRS            G  DUMP GPR FILE                        0522
523             DATA    E120            H  SET CCS HALT ADDRESS                 0523
524             DATA    EXC             I  INITIALIZE                           0524
525             DATA    EXC90,EXC90     JK INVALID                              0525
526             DATA    LDRM            L  LOAD CCS, DECODER OR MEMORY          0526
527             DATA    CHME            M  SELECT INPUT MEDIUM                  0527
528             DATA    EXC90,EXC90     NO INVALID                              0528
529             DATA    PSEL            P  PAGE SELECT                          0529
530             DATA    SETM            Q  SET MEMORY TYPE                      0530
531             DATA    E170            R  RETURN TO OPERATING SYSTEM           0531
532             DATA    E130            S  SINGLE STEP OR SET RUN COUNT         0532
533             DATA    E140            T  SET TRACE BOUNDARIES                 0533
534             DATA    EXC90,EXC90     UV INVALID                              0534
535             DATA    EXC90,EXC90     WX INVALID                              0535
536             DATA    EXC90           Y  INVALID                              0536
537             DATA    EXCS            Z  ZERO TABLES                          0537
538     *                                                                   VDM 0538
539     *       INVALID INPUT                                               VDM 0539
540     *                                                                   VDM 0540
541     EXC90   CALL    ERRMSG,1        INVALID INPUT, GET NEXT INPUT           0541
542             EJEC                                                        VDM 0542
543     *                                                                   VDM 0543
544     *       THIS ROUTINE HANDLES THE PAGE SELECT COMMAND (P)            VDM 0544
545     *                                                                   VDM 0545
546     *               AVAILABLE MEMORY IS TESTED TO DETERMINE IF THERE IS ROOM VDM 0546
547     *               FOR THE PAGE AND IF NOT THEN ERROR MESSAGE IS GIVEN     0547
548     *                                                                   VDM 0548
549     PSEL    JMPM    FETCHA          GET PAGE NUMBER                     VDM 0549
550             SUB     ='0'                                                    0550
551             JAN     EXC90           INVALID INPUT                           0551
552             CALL    PGOK            CHECK IF PAGE AVAILABLE                 0552
553             JAN     EXC10           SPACE IS AVAILABLE                      0553
554             TZA                                                             0554
555             CALL    PGOK            SETUP FOR PAGE ZERO                     0555
556             CALL    ERRMSG,3        NO ROOM IN CORE, GET NEXT INPUT         0556
557             EJEC                                                        VDM 0557
558     *                                                                   VDM 0558
559     *       THIS ROUTINE CHECKS IF THERE IS SUFFICIENT CORE FOR THE     VDM 0559
560     *       DESIRED PAGE                                                VDM 0560
561     *                                                                   VDM 0561
562     *       CALLING SEQUENCE                                            VDM 0562
563     *       LDX     =DROM                                                   0563
564     *       LDA     PAGE NUMBER                                         VDM 0564
565     *       CALL    PGOK                                                VDM 0565
566     *       RETURN A REG POS=NOT ROOM, A REG NEG=ENOUGH ROOM            VDM 0566
567     *       PAGE TABLE POINTERS UPDATED                                 VDM 0567
568     *                                                                   VDM 0568
569     PGOK    ENTR                                                        VDM 0569
```

```
570                 STA     CPAG,1                                          0570
571                 ADD     =DROM                                           0571
572                 TAB                                                     0572
573                 SUBI    *-*             PAGE LIMIT                      0573
574       PAG4      BES     0                                               0574
575                 DAR                                                     0575
576                 JAP*    PGOK            EXCEED PAGE LIMIT           VDM 0576
577                 LDA     TRACE-DROM,2                                    0577
578                 STA     NTRC,1          TRACE FLAG FOR PAGE OF NEXT MICRO 0578
579                 LDA     PSTBL-DROM,2    PAGE ADDRESS TABLE              0579
580                 STA     DBUF-DROM,1     POINTER INTO TABLE              0580
581                 LDA     DCTBL-DROM,2    DCSA ADDRESS                    0581
582                 STA     DRM1-DROM,1     SET DCSA ADDR                   0582
583                 ADD     =16                                             0583
584                 STA     DRM2-DROM,1     SET DCSB ADDR                   0584
585                 DECR    01              SET A NEG                   VDM 0585
586                 JMP*    PGOK                                        VDM 0586
587                 EJEC                                                VDM 0587
588       *                                                             VDM 0588
589       *         THIS ROUTINE WILL CONTROL THE SIMULATED OPERATION    VDM 0589
590       *         OF THE COMPUTER                                     VDM 0590
591       *                                                             VDM 0591
592       *         ENTER VIA THE BEGIN COMMAND                         VDM 0592
593       *                                                             VDM 0593
594       S100      CALL    INA,'N'         GET START SIMULATION ADDRESS     0594
595                 JBZ     S101            NO ADDR GIVEN--CONTINUE FROM LAST ADDR VDM 0595
596                 STAE    NROM*DROM                                       0596
597       S101      LDX     =DROM                                           0597
598                 LDA     NROM,1                                          0598
599                 STA     PROM,1          SAVE ROM ADDRESS                0599
600                 CALL    FELD            BREAK CCS WORD INTO FIELDS      0600
601                 LDA     CPAG,1                                          0601
602                 STA     PPAG,1          SAVE PAGE NUMBER                0602
603                 LDA     NTRC,1                                          0603
604                 STA     PTRC,1          SAVE TRACE FLAG                 0604
605                 CALL    CRAS            DETERMINE NEXT CCS ADDRESS      0605
606                 CALL    LIST            LIST ADDRESSES AND DECODED FIELDS 0606
607                 CALL    S120            CHECK FOR ROM HALT ADDRESS      0607
608                 CALL    S300            DATA LOOP OPERATIONS            0608
609                 CALL    MOPA            MEMORY OPERATIONS               0609
610                 CALL    X100            DETERMINE STATUS REGISTER SETTINGS 0610
611                 CALL    LTMX            LIST TEST MUX STATUS        VDM 0611
612       *                                                             VDM 0612
613       *         CHECK FOR RETURN TO EXEC (STEP MODE OR RUN COUNT SATISFIED) VDM 0613
614                 LDAE    STEP            STEP/RUN FLAG                   0614
615                 MJANZ   EXC10           RETURN TO EXEC (STEP MODE)      0615
616                 LDA     RCNT            MICRO COUNT                 VDM 0616
617                 JAN     S101            RETURN IF NOT SET               0617
618                 JAZ     S106A           COUNT SATISFIED             VDM 0618
619                 DAR                                                 VDM 0619
620                 STA     RCNT            DECR RUN COUNT              VDM 0620
621                 JMP     S101                                            0621
622       S106A     CALL    SIOUT,S106L,S106B       OUTPUT MESSAGE          0622
623                 JMP     EXC10           RETURN TO EXEC              VDM 0623
624       S106B     DATA    '  EXECUTION LIMIT SATISFIED '              VDM 0624
625       S106L     EQU     *-S106B         LENGTH OF MSG                   0625
626       *                                                             VDM 0626
```

```
627         *   CHECK FOR ROM HALT ADDR                                          0627
628         *   YES=EXIT TO EXEC  NO=NORMAL RETURN                         VDM 0628
629         S120  ENTR                                                     VDM 0629
630               LDX     =DROM                                                0630
631               LDB     PPAG,1          CURRENTLY SELECTED PAGE              0631
632               LDBE    RHLT,2          ADDRESS OF HALT TABLE FOR SELECTED PAGE 0632
633               STB     S120A+1                                              0633
634               LDB     =4                                                   0634
635         S120A LDAE    *=*,2           GET ENTRY FROM HALT TABLE            0635
636               SUB     PROM,1          CURRENT ROM ADDRESS                  0636
637               JAZ     S123                                           VDM 0637
638               JBZ*    S120            RETURN IF ALL ENTRIES EXAMINED       0638
639               DBR                     DECREMENT POINTER                    0639
640               JMP     S120A           LOOK AT NEXT ENTRY                   0640
641         S123  CALL    SIOUT,S111L,S111        OUTPUT CCS HALT MESSAGE      0641
642               JMP     EXC10                                                0642
643         *                                                              VDM 0643
644         S111  DATA    '  CCS ADDR HALT'                                    0644
645         S111L EQU     *-S111          LENGTH OF MESSAGE                    0645
646               EJEC                                                       0646
647         *                                                                  0647
648         *     SINGLE STEP / CONTINUOUS EXECUTION CONTROL.  ENTERED VIA "S". 0648
649         *                                                                  0649
650         E130  CALL    FETCHA          GET MODE FLAG                        0650
651               SUB     ='R'                                                 0651
652               JAZ     E131            JUMP IF CONTINUOUS RUN MODE          0652
653               DAR                     TEST FOR SINGLE STEP MODE            0653
654               MJANZ   EXC90           INCORRECT INPUT                      0654
655               INCR    1               SET SINGLE STEP MODE                 0655
656         E131  STAE    STEP            SET STEP/RUN MODE FLAG               0656
657               CALL    INA,'N'         LOOK FOR RUN COUNT                   0657
658               JBZ     E133            NO COUNT GIVEN                       0658
659               DAR                                                          0659
660               DATA    01006           SKIP NEXT INSTRUCTION                0660
661         E133  DECR    1               SET UNLIMITED RUN COUNT              0661
662               STA     RCNT            STORE RUN COUNT                      0662
663               JMP     EXC10           RETURN TO EXECUTION CONTROL          0663
664         RCNT  DATA    -1              RUN COUNT LIMIT                      0664
665               EJEC                                                   VDM 0665
666         *                                                            VDM 0666
667         *     ROUTINE TO DETERMINE NEXT ROM ADDRESS                  VDM 0667
668         *                                                            VDM 0668
669         *     CALLING SEQUENCE                                       VDM 0669
670         *       LDX     =DROM         ADDRESS OF DATA BUFFER               0670
671         *       CALL    CRAS                                         VDM 0671
672         *                                                            VDM 0672
673         *       RETU    NEXT ROM ADDR IS STORED IN "NROM". ROM ADDR SELECT IS VDM 0673
674         *               A 12 BIT WORD WHERE THE UPPER 4 BITS DESIGNATE A VDM 0674
675         *               SPECIFIC 512 WORD CONTROL STORE PAGE.                0675
676         *                                                            VDM 0676
677         *       THIS ROUTINE USES THE FOLLOWING FIELDS OF THE CURRENT VDM 0677
678         *       CONTROL ROM WORD: T,S,G,TS,AF,FS,MS,MT,&IMC          VDM 0678
679         *              ALSO USED: IRG2,INTF,INTA,INPF,&TMUX          VDM 0679
680         *                                                            VDM 0680
681         CRAS  ENTR                    **                             VDM 0681
682               LDA     XT,1            GET TEST CONTROL FIELD         VDM 0682
683               LSRA    1                                                    0683
```

```
684       *       THE CONDITION T=1 IS NOT HANDLED COMPLETELY BY THE                  0684
685       *       PROCESSOR.  TREAT AS T=0.                                           0685
686               MJANZ   CR50            TEST STATUS LOGIC IF T>1.                   0686
687               LDA     XS,1                                                VDM 0687
688               ADD     CRAST           ADDRESS OF JUMP TABLE                       0688
689               STA     *+2                                                         0689
690               JMP*    *-*                                                         0690
691       CRAST   DATA    (*+1)*                                                      0691
692       *       S FIELD=    0    1    2    3                                       0692
693               DATA    CRAS0,CRA7,CRA4,CRA7                                        0693
694       CRAS0   LDA     XIMC,1                                                      0694
695               SUB     =13             IM=X'D?                                     0695
696               MJANZ   CRA0                                                        0696
697               DAR                                                         VDM 0697
698               STA     IMD,1           SET BRANCH/PUSH-POP FLAG                    0698
699               LDA     XA,1            A FIELD                             VDM 0699
700               TAB                                                         VDM 0700
701               ANA     =4              POP?                                        0701
702               JAZ     CRAS1           NO                                  VDM 0702
703               INCR    1               CHECK IF POP WITHOUT JUMP                   0703
704               ANA     XB,1            BO SET?                                     0704
705               JAZ     CR34            NO                                  VDM 0705
706               LDB     JSPT,1          STACK POINTER                               0706
707               JBZ     CRAX            STACK EMPTY - DO STANDARD PAGE BRANCH       0707
708               DBR                                                                 0708
709               STB     JSPT,1                                                      0709
710               TZA                                                         VDM 0710
711               STAE    STACK,2         CLEAR ENTRY                         VDM 0711
712               STA     IMD,1           RESET BRANCH/PUSH-POP FLAG                  0712
713               JMP     CRAX            PERFORM STANDARD PAGE BRANCH        VDM 0713
714       CRAS1   TBA                                                         VDM 0714
715               ANA     =2              PUSH?                                       0715
716               MJANZ   CRAX            YES                                         0716
717       CRA0    TZA                                                         VDM 0717
718               STA     IMD,1           RESET BRANCH/PUSH-POP FLAG                  0718
719               LDA     XG,1                                                VDM 0719
720               ANA     =4              G=X1XX?                                     0720
721               JAZ     CRA2            NO=                                 VDM 0721
722       *    T=0 & S=0 & G=X1XX         SELECT DECODER AND SAMPLE INTERRUPTS        0722
723               LDA     INTF,1          ANY INTERRUPT PENDING ?             VDM 0723
724               JAZ     CRA2A           NO                                          0724
725               LDA     INTA,1          GET INTERRUPT ADDRESS FROM I/O ,MASK WITH    VDM 0725
726               ANA     XTS,1           TS FIELD AND USE FOR NEXT ROM ADDR   VDM 0726
727       ********************************                                    VDM 0727
728               LDA     =0777           * COMPLETE INTERRUPT SIMULATION WILL NOT     0728
729               JMP     CR32            * BE DONE NOW. USE LAST ROM ADDRESS          0729
730       ******************************** ADDR AS DEFAULT.                   VDM 0730
731       *   DETERMINE IF NEXT ADDRESS COMES FROM DECODE ROM.                  VDM 0731
732       *   YES IF: NO INTERRUPT PENDING, AND G=X1XX & IMC=0110 , AND AN      VDM 0732
733       *           INTERRUPT IS NOT BEING PROCESSED(INPF=0).                 VDM 0733
734       CRA2B   LDA     INPF,1          INT BEING PROCESSED FLAG SET ?      VDM 0734
735               JAZ     CRA3                                                VDM 0735
736               JMP     CRA6Z                                                       0736
737       CRA2A   LDA     XIMC,1                                              VDM 0737
738               SUB     =3              IM=3?                                       0738
739               JAZ     CRA2C                                               VDM 0739
740               SUB     =3              IM=6?                                       0740
```

```
741             JAZ     CRA2B           YES                                     VDM 0741
742     CRA3    LDA     XIMC,1          DISABLE D-ROM                           VDM 0742
743             SUB     =12             ADDR SELECT                                 0743
744             JAZ     CRA6            IF S=0 AND IMC=1100.                    VDM 0744
745             SUB     =2                                                          0745
746             JAZ     CRA6            IMC=1110 (REQ I/O & DISABLE D-ROM)          0746
747             JMP     CR70A           GET NEXT ADDR FROM DECODE               VDM 0747
748     CRA2C   LDA     XTS,1           GET PAGE                                VDM 0748
749             CALL    PGOK            SELECT CCS AND DCS FOR PAGE             VDM 0749
750             JAP     CRPE            PAGE NOT AVAILABLE                          0750
751             JMP     CR70A           GO TO DCS FOR NEXT ADDRESS              VDM 0751
752     *                                                                       VDM 0752
753     *   FORM B-OFFSET PART OF ADDR IF AB FIELD IS NOT 1-OR-2.               VDM 0753
754     *   OTHERWISE B-OFFSET IS ZERO                                          VDM 0754
755     *   DON'T USE TS FIELD FOR B-OFFSET IF I/O REQUEST (IM=E/F)                 0755
756     CRA2    LDA     XIMC,1                                                      0756
757             SUB     =3              IM=3?                                       0757
758             JAZ     CRAX            YES-PAGE BRANCH                             0758
759             LDA     XIMC,1                                                      0759
760             SUB     =14             IM=X'E OR X'F?                              0760
761             JAP     CRA6Z           YES                                         0761
762             JMP     CRA7                                                        0762
763     *                                                                           0763
764     *   CHECK FOR PAGE ADDRESS JUMP: YES IF G FIELD NOT=X0XX                    0764
765     *   IF YES, TS FIELD SPECIFIES A PARTICULAR 512 WORD CONTROL                0765
766     *   STORE PAGE.                                                             0766
767     CRA4    LDA     XG,1                                                        0767
768             ANA     =4              G=X0XX?                                     0768
769             MJANZ   CRAX            NO-FORM PAGE JUMP ADDRESS                   0769
770     *   FORM B-OFFSET                                                       VDM 0770
771     CRA7    LDA     XAB,1           AB FIELD                                VDM 0771
772             JAZ     CRA7A           AB=0                                        0772
773             SUB     =3                                                          0773
774             JAN     CRA6Z           AB=01 OR AB=10.                             0774
775     CRA7A   LDA     XTS,1           PUT TS FIELD INTO BITS 4-1.                 0775
776             LRLA    1               BITS 0 &5-8=0.                          VDM 0776
777             DATA    01006           SKIP NEXT WORD                              0777
778     CRA6Z   TZA                     SET B-OFFSET TO ZERO                        0778
779     CRA6    STA     CRB0            STORAGE FOR B OFFSET                        0779
780     *                                                                       VDM 0780
781     *  FORM BASE ADDRESS: AF FIELD INTO BITS 8-4, BITS3-0 ARE ZERO          VDM 0781
782             LDA     XAF,1           GET AF FIELD                                0782
783             LRLA    4               POSITION FOR UPPER 5 BITS OF BASE ADDR  VDM 0783
784             STA     CRBA            STORAGE FOR BASE ADDR                   VDM 0784
785     *                                                                       VDM 0785
786     *   FORM A OFFSET; THE UPPER FOUR BITS ARE ZERO, THE LOWER 5 BITS ARE   VDM 0786
787     *                  SET BY SELECTING 5 CONTIQUOUS BITS OF INST REG2 (IRG2 VDM 0787
788     *                  VIA THE FS FIELD, AND MASKING THEM WITH 5 BITS       VDM 0788
789     *                  OF THE MT/MS FIELDS, OPEN BIT POSITIONS ARE FORCED   VDM 0789
790     *                  TO "1" (EX, FOR FS=0: SELECTED IRG2 BITS=0XXXX)      VDM 0790
791             LDA     XFS,1                                                       0791
792             SUB     =4              FOR FS<4, FORCE UNUSED BITS TO 1            0792
793             JAN     CR17            GO PERFORM SPECIAL PROCESSING           VDM 0793
794             ERA     =04340          FORM LSRA TO SELECT THE SPECIFIED 5          0794
795             STA     *+2             BITS OF IRG2.  STORE INSTRUCTION IN LINE.   0795
796             LDA     IRG2,1                                                  VDM 0796
797             DATA    *-*             RIGHT JUSTIFY SELECTED BITS                 0797
```

```
798             JMP     CR30            GO MASK SELECTED IRG2 BITS              VDM 0798
799     *                                                                       VDM 0799
800     *   SPECIAL PROCESSING FOR FS<4.                                        VDM 0800
801     CR17    LDA     XFS,1           FS=0 ?                                  VDM 0801
802             JAZ     CR23            YES-                                    VDM 0802
803             DAR                     FS=0001 ?                               VDM 0803
804             JAZ     CR21            YES-                                    VDM 0804
805             DAR                     FS=0010 ?                               VDM 0805
806             JAZ     CR19            YES-                                    VDM 0806
807     *   FS=0011                                                             VDM 0807
808             LDA     NSTP,1          STEP F/F                                    0808
809             ERA     =036            FORCE 4 MSBS TO 1                           0809
810             JMP     CR30            GO MASK BITS                            VDM 0810
811     *   FS=0010                                                             VDM 0811
812     CR19    LDA     DS,1            PUT STATUS OF DS F/F                        0812
813             LRLA    1               INTO BIT POSITION 1                         0813
814             ERA     MBYC,1          MEMORY BYTE DESIGNATOR                      0814
815             TAB                                                                 0815
816             LDA     IRG2,1          PUT BIT ONE                             VDM 0816
817             LRLA    2               IRG2 WORD                               VDM 0817
818             ANA     =8                                                          0818
819             MERG    031             A V B -> A                                  0819
820             ORA     =024            FORCE BITS 4 & 2 TO 1                       0820
821             JMP     CR30            GO MASK BITS                            VDM 0821
822     *   FS=0001                                                             VDM 0822
823     CR21    LDA     INPF,1          INCLUDE COMPLEMENT OF INTERRUPT FLAG        0823
824     *   FS=0000                                                             VDM 0824
825     *     ACCUMULATOR IS ZERO IF WE JUMP HERE.                                  0825
826     CR23    ERA     =037            FORCE 4 MSBS=1, LSB = 1 OR NOT INPF         0826
827     *   COME HERE FOR MASKING OF SELECTED IRG2 BITS.                        VDM 0827
828     *   FIELD MT MASKS MSB OF SELECTED 5 BITS, THE MS FIELD                 VDM 0828
829     *   MASKS THE OTHER 4 BITS.                                             VDM 0829
830     CR30    STA     CRAO            TEMP STORE OF SELECTED IRG2 BITS        VDM 0830
831             LDA     XMT,1           MAKE 5 BIT MASK OUT                     VDM 0831
832             LRLA    4               OF THE ONE BIT MT FIELD                 VDM 0832
833             ERA     XMS,1           AND THE 4 BIT MS FIELD.                 VDM 0833
834             ANA     CRAO            MASK IRG2 BITS.                         VDM 0834
835     *   FORM EFFECTIVE NEXT ROM ADDRESS BY INCLUSIVE ORING                  VDM 0835
836     *   BASE ADDR & A-OFFSET & B-OFFSET.                                    VDM 0836
837             ORA     CRBA            BASE ADDR & A-OFFSET                    VDM 0837
838             ORA     CRBO            & B-OFFSET                              VDM 0838
839     CR32    STA     NROM,1          THIS IS THE NEXT ROM ADDRESS                0839
840             LDA     IMD,1           BRANCH PUSH?                                0840
841             JAP*    CRAS            NO-RETURN                               VDM 0841
842             TZA                                                                 0842
843             STA     IMD,1           CLEAR BRANCH/PUSH-POP FLAG                  0843
844             LDA     MASK,1                                                      0844
845             CPA                                                                 0845
846             ANA     =017777         USE 13 BITS OF TRUE MASK FIELD              0846
847             TAB                                                             VDM 0847
848             LDA     JSPT,1                                                      0848
849             INR     JSPT,1                                                      0849
850             SUB     =16             STACK LIMIT                                 0850
851             JAP     CR36            STACK OVERFLOW - GET ENTRY ZERO             0851
852             LLRL    16              EXCHANGE A & B                              0852
853             STAE    STACK+16,2      STORE VALUE ON STACK                        0853
854             RETU*   CRAS                                                   VDM 0854
```

```
855     *                                                                    VDM 0855
856     *                                                                    VDM 0856
857     CR34   LDB     JSPT,1          JUMP STACK POINTER                       0857
858            JBZ     CR36            JUMP IF POP EMPTY STACK                  0858
859            DATA    01006           SKIP NEXT INSTRUCTION                    0859
860     CR36   INCR    2               GET FIRST STACK ENTRY                    0860
861            DBR                                                              0861
862            STB     JSPT,1          UPDATE STACK POINTER                     0862
863            LDAE    STACK,2         GET STACK ENTRY                          0863
864            LLSR    9                                                        0864
865            LSRB    7                                                        0865
866            STB     NROM,1          SET NEXT CCS ADDRESS                     0866
867            ANA     =15                                                      0867
868            CALL    PGOK            CHECK IF LEGAL PAGE                      0868
869            JAP     CRPE            ILLEGAL PAGE JUMP                        0869
870            JMP*    CRAS            RETURN                                VDM 0870
871     *                                                                    VDM 0871
872     *   COME HERE (SEE CRA4) IF G NOT=X0XX TO SELECT ROM PAGE ADDR.      VDM 0872
873     *   IF YES ;  PUT TS FIELD BITS INTO CRAZ (BITS 12-9) FOR LATER ORING VDM 0873
874     *   WITH THE 9 BIT ADDR.                                             VDM 0874
875     CRAX   LDA     XTS,1           GET TS FIELD                          VDM 0875
876            CALL    PGOK            CHECK IF PAGE AVAILABLE               VDM 0876
877            JAN     CRA6Z           GO FORM 9-BIT ADDR(512-WORD ROM)         0877
878     *                                                                    VDM 0878
879     CRPE   CALL    ERRMSG,5        PAGE JUMP TO UNAVAILABLE PAGE, GET INPUT 0879
880     *                                                                    VDM 0880
881     *    COME HERE IF T FIELD = 1X.                                         0881
882     *    T=10 SET TEST MODE AND TEST SELECTED TEST CONDITION.            VDM 0882
883     *    T=11 RESET TEST MODE AND TEST SELECTED TEST CONDITION.          VDM 0883
884     *                                                                    VDM 0884
885     CR50   LDA     XT,1                                                     0885
886            SUB     =3                                                       0886
887            STA     TMOT            SET TEST-MODE-TRUE FLAG               VDM 0887
888     *                                                                    VDM 0888
889     *   SELECT TEST CONDITION SPECIFIED BY THE G FIELD                   VDM 0889
890     *                                                                    VDM 0890
891     CR54   LDA     XG,1                                                     0891
892            SUB     =5              620/F TEST?                              0892
893            JAZ     CR60            YES-GO PROCESS                        VDM 0893
894            LDB     XG,1                                                     0894
895            LDAE    TMUX,2          IS CONDITION SPECIFIED BY G FIELD TRUE?  0895
896            JAZ     CR56            NO-                                      0896
897            JMP     CR58                                                  VDM 0897
898     *   COME HERE IF TESTED CONDITION(S) TRUE.                           VDM 0898
899     *                                                                    VDM 0899
900     *   RESET OVERFLOW F/F IF :   T=NOT ZERO AND G=5 AND TEST COND.TRUE AND VDM 0900
901     *                             BIT 0 OF IRG2=1                        VDM 0901
902     *                             THIS MEANS: RESET OVFL IF DOING JOF,JOFM, VDM 0902
903     *                             XOF, OR THE "NOT" EQUIVALENTS          VDM 0903
904     *                                                                    VDM 0904
905            IFF     1=JNOT                                                   0905
906            GOTO    CR55A                                                    0906
907     CR55   INCR    1               IS OVERFLOW BIT OF                       0907
908            ANA     IRG2,1          IRG2 SET?                                0908
909            JAZ     CR58            NO=                                   VDM 0909
910            TZA                                                           VDM 0910
911            STA     OVRFL,1         RESET STORED OVERFLOW                 VDM 0911
```

```
912                 STA     TMUXX+0,1       RESET OVERFLOW IN TMUX                          0912
913                 ROF                     RESET HARDWARE OVERFLOW F/F                     0913
914                 GOTO    CR55B                                                           0914
915       CR55A     CONT                                                                    0915
916       CR55      LDX     =DROM           ADDRESS OF DATA BUFFER                          0916
917                 TZA                                                                     0917
918                 AOFA                                                                    0918
919                 STA     OVRFL,1         HARDWARE DECIDES WHETHER TO RESET OVERFLOW      0919
920                 STA     TMUXX+0,1                                                       0920
921       CR55B     CONT                                                                    0921
922       CR58      INR     TMOT            INVERT TEST MODE TRUE FLAG (-1 = TRUE)          0922
923       *                                                                         VDM 0923
924       CR56      LDA     TMOT            TEST MODE FLAG FALSE ?                  VDM 0924
925                 JAZ     CRA6            YES-SET B-OFFSET TO ZERO                        0925
926       *   FORM ROM ADDR FOR TEST CONDITION NOT MET                              VDM 0926
927       *   BITS 8-5=AF FIELD, BITS 3-1=TS FIELD BITS 2-0, BIT 0= ZERO            VDM 0927
928       *   BIT 4=AF BIT 0-INCLUSIVE OR0-WITH TS FIELD BIT 3.                     VDM 0928
929       CR57      TZB                                                                     0929
930                 LDA     XTS,1                                                           0930
931                 LLSR    3                                                               0931
932                 ORA     XAF,1                                                           0932
933                 LLSR    12                                                              0933
934                 STB     NROM,1          NEXT CCS ADDRESS                                0934
935                 LDA     CPAG,1          CURRENTLY SELECTED PAGE                         0935
936                 CALL    PGOK            CHECK IF PAGE AVAILABLE                         0936
937                 JAP     CRPE            NO--OUTPUT ERROR                        VDM 0937
938       *                                   TEST CONDITION IS MET.                VDM 0938
939                 RETU*   CRAS            EXIT                                    VDM 0939
940       *                                                                           VDM 0940
941       *                                                                           VDM 0941
942       *   620/F JMP/JMPM/XEC TESTING                                              VDM 0942
943                 IFF     1-JNOT                                                          0943
944                 GOTO    CR60A                                                           0944
945       CR60      LDA     IRG2,1          GET INST--S/B JMP,JMPM, OR XEC          VDM 0945
946                 ANA     =0777           LOOK AT 9 LSBS                                  0946
947                 JAZ     CR58            UNCONDITIONAL INST                              0947
948                 TAB                     SAVE 9 LSB'S                            VDM 0948
949                 ANA     =6                                                              0949
950                 SUB     =6              "NOT" INST?                                     0950
951                 JAZ     CR6J            YES-                                    VDM 0951
952                 TZA                     NO-                                     VDM 0952
953                 DATA    01006           SKIP NEXT WORD                                  0953
954       CR6J      INCR    01                                                      VDM 0954
955                 STA     CNOT            "NOT" INST FLAG                         VDM 0955
956                 TBA                                                             VDM 0956
957                 TZB                                                                     0957
958                 LLSR    1                                                       VDM 0958
959                 STA     CRTS                                                    VDM 0959
960                 JBZ     CR6B                                                            0960
961                 LDA     TMUXX+0,1       001: CHECK OVERFLOW                             0961
962                 ERA     CNOT                                                            0962
963                 JAZ     CR56            CONDITION NOT MET                               0963
964                 LDA     CRTS            RELOAD SHIFTED 9 BITS OF INSTRUCTION WORD       0964
965       CR6B      LDB     CNOT            "NOT" TYPE INST ?                       VDM 0965
966                 JBZ     CR6K            NO-                                     VDM 0966
967                 TZB                     YES-                                    VDM 0967
968                 LSRA    2               SKIP NEXT 2 BITS WHICH                  VDM 0968
```

```
969                 JMP     CR6D            REPRESENT "NOT" TYPES.                  VDM 0969
970         CR6K    LLSR    1                                                       VDM 0970
971                 JBZ     CR6C                                                        0971
972                 LRLB    2               POSITION BIT                                0972
973                 CALL    CRCHK,R0        002: CHECK A EQUAL/GREATER THAN ZERO        0973
974         CR6C    LLSR    1                                                       VDM 0974
975                 JBZ     CR6D                                                        0975
976                 LRLB    3               POSITION BIT                                0976
977                 CALL    CRCHK,R0        004: CHECK A NEGATIVE                       0977
978         CR6D    LLSR    1                                                       VDM 0978
979                 JBZ     CR6E                                                        0979
980                 LRLB    4               POSITION BIT                                0980
981                 CALL    CRCHK,R0        010: CHECK A=0                              0981
982         CR6E    LLSR    1                                                       VDM 0982
983                 JBZ     CR6F                                                        0983
984                 LRLB    4               POSITION BIT                                0984
985                 CALL    CRCHK,R1        020: CHECK B=0                              0985
986         CR6F    LLSR    1                                                       VDM 0986
987                 JBZ     CR6G                                                        0987
988                 LRLB    4               POSITION BIT                                0988
989                 CALL    CRCHK,R2        040: CHECK X=0                              0989
990         CR6G    LLSR    1                                                       VDM 0990
991                 JBZ     CR6H                                                        0991
992                 STA     CRTS                                                        0992
993         *   100: CHECK IF SS1 SET                                                   0993
994                 LDA     TMUXX+4,1       GET SS1                                     0994
995                 ERA     CNOT            COMPARE WITH "NOT" FLAG                     0995
996                 JAZ     CR56            CONDITION NOT MET                           0996
997                 TZB                                                                 0997
998                 LDA     CRTS            RELOAD SHIFTED 9 BITS OF INSTRUCTION WORD   0998
999         CR6H    LLSR    1                                                       VDM 0999
1000                JBZ     CR6I                                                        1000
1001                STA     CRTS                                                        1001
1002        *   200: CHECK IF SS2 SET                                                   1002
1003                LDA     TMUXX+3,1       GET SS2                                     1003
1004                ERA     CNOT            COMPARE WITH "NOT" FLAG                     1004
1005                JAZ     CR56            CONDITION NOT MET                           1005
1006                TZB                                                                 1006
1007                LDA     CRTS            RELOAD SHIFTED 9 BITS OF INSTRUCTION WORD   1007
1008        CR6I    LLSR    1                                                       VDM 1008
1009                JBZ     CR55            NO-BUT PREV COND WAS TRUE TO GET HERE   VDM 1009
1010                LDA     TMUXX+2,1       GET SS3                                     1010
1011                ERA     CNOT            COMPARE WITH "NOT" FLAG                     1011
1012                JAZ     CR56            CONDITION NOT MET                           1012
1013                JMP     CR55            ALL CONDITIONS MET                          1013
1014        *                                                                           1014
1015        *       CHECK CONDITION SPECIFIED BY BIT IN B REGISTER                      1015
1016        *       CALL    CRCHK,RN        WHERE N = 0, 1 OR 2.                        1016
1017        *                                                                           1017
1018        CRCHKA  STA     CRTS                                                        1018
1019                LDA     CRCHK-1         GET JUMP INSTRUCTION                        1019
1020                MERG    031             CREATE CONDITIONAL JUMP                     1020
1021                STA     CRCHKT                                                      1021
1022                LDBE*   CRCHK           REGISTER TO BE TESTED                       1022
1023                LDA     0,2             CONTENTS OF REGISTER                        1023
1024        CRCHKT  JMP     *+4             TEST SPECIFIED CONDITION                    1024
1025                TZA                     SET FALSE                                   1025
```

```
1026                DATA    01006           SKIP NEXT WORD                                  1026
1027                INCR    1               SET TRUE                                        1027
1028                ERA     CNOT            COMPARE WITH "NOT" FLAG                         1028
1029                JAZ     CR56            CONDITION NOT MET                               1029
1030                TZB                                                                     1030
1031                LDA     CRTS            RELOAD SHIFTED 9 BITS OF INSTRUCTION WORD       1031
1032                INR     CRCHK                                                           1032
1033                JMP     *-*             RETURN                                          1033
1034      CRCHK     BES     0                                                               1034
1035                JMP     CRCHKA                                                          1035
1036      CRTS      DATA    0               TEMPORARY STORAGE FOR TEST BITS                 1036
1037      CNOT      DATA    0               FLAG FOR "NOT" TYPES OF TESTS                   1037
1038                GOTO    CR60B                                                           1038
1039      CR60A     CONT                                                                    1039
1040      CR60      ROF                                                                     1040
1041                LDA     TMUXX+0,1                                                       1041
1042                JAZ     *+3             SIMULATED OVERFLOW OFF                          1042
1043                SOF                     SIMULATED OVERFLOW ON                           1043
1044                LDA     IRG2,1          INSTRUCTION REGISTER                            1044
1045                ANA     =0777           ISOLATE 9 BITS                                  1045
1046                ORA     =01000          FORM JUMP INSTRUCTION                           1046
1047                STA     CR6J            STORE IT IN LINE                                1047
1048                LDA     R0-DROM,1       GET SIMULATED A REGISTER                        1048
1049                LDB     R1-DROM,1       GET SIMULATED B REGISTER                        1049
1050                LDX     R2-DROM,1       GET SIMULATED X REGISTER                        1050
1051      CR6J      JIF     *-*,CR55        TEST SPECIFIED CONDITIONS                       1051
1052                LDX     =DROM           DATA BUFFER ADDRESS                             1052
1053                JMP     CR56            TEST CONDITIONS FALSE                           1053
1054      CR60B     CONT                                                                    1054
1055      *                                                                         VDM     1055
1056      TMOT      DATA    0               TEST-MODE-TRUE FLAG                       VDM     1056
1057                EJEC                                                              VDM     1057
1058      *                       -DECODE ROM ADDRESS GENERATION-                     VDM     1058
1059      *   ROUTINE TO SELECT THE NEXT CONTROL ROM ADDRESS FROM THE OUTPUT          VDM     1059
1060      *   OF THE DECODE ROM                                                       VDM     1060
1061      CR70A     LDA     DRM1-DROM,1     GET DCS SELECTION                               1061
1062                SUB     =$DRM1          COMPARE TO PAGE ZERO DCS                        1062
1063                MJANZ   CR70            CURRENT DCS NOT PAGE ZERO DCS                   1063
1064                LDA     IRG1,1          GET IBR                                         1064
1065                LSRA    8               GET 8 MSBS                                VDM   1065
1066                SUB     =0212           BCS OPCODE                                      1066
1067                JAZ     BCS                                                       VDM   1067
1068      CR70      TZA                     DECODE STORES ZERO IN SHIFT COUNTER             1068
1069                STA     SREG,1          (LOOK IN THE CIRCUIT DIAGRAMS)                  1069
1070                LDA     IRG1,1                                                          1070
1071                LSRA    8               RIGHT JUSTIFY                             VDM   1071
1072                ANA     =15             SELECT IBR BITS 11-8                            1072
1073                ADD     DRM2-DROM,1     ADDR OF DCSA                                    1073
1074                TAB                                                               VDM   1074
1075                LDA     0,2                                                       VDM   1075
1076                STA     CR7B            TEMP STORE                                VDM   1076
1077      *                                                                         VDM   1077
1078                LDA     IRG1,1          GET IBR                                         1078
1079                LSRA    12              USE BITS 15,14,13,&12                           1079
1080                ADD     DRM1-DROM,1     ADDR OF DCSB                                    1080
1081                TAB                                                                     1081
1082                LDA     0,2                                                             1082
```

```
1083                ERA     =0170000        COMPLEMENT 4 MSBS FOR SIMULATOR LOGIC                   1083
1084                STA     CR7A            TEMP STORE FOR SELECTED DRM1 WORD                       1084
1085                ANA     =0110000        GET BITS 15 & 12 OF DCSB                                1085
1086                JAZ     CR73            JUMP IF NEITHER SET                                     1086
1087                DECR    2               USE ALL OF DCSA                                         1087
1088                LDA     CR7A                                                                    1088
1089                LRLA    5               BIT "00" SET?                                           1089
1090                JAP     CR71            NO-                                                     1090
1091                LDB     =0177760        MASK OUT 4 LSBS OF DCSA                                 1091
1092      CR71      TBA                                                                             1092
1093                ANA     CR7B                                                                    1093
1094      CR73      ORA     CR7A            DCSB                                                    1094
1095                ANA     =0777           USE 9 BITS FOR ADDRESS                                  1095
1096                STA     CR7C            TEMP STORE FOR GENERATED ADDRESS                        1096
1097                LDA     CR7A            DCSB                                                    1097
1098                LSRA    1                                                                       1098
1099                ANA     CR7B            FORM B15 & A 14                                         1099
1100                ORA     CR7A            OR B14                                                  1100
1101                LRLA    1                                                                       1101
1102                JAP     CR74            EXPRESSION FALSE                                        1102
1103                LDA     IRG1,1                                                                  1103
1104                LSRA    4                                                                       1104
1105                ANA     =017                                                                    1105
1106                ORA     CR7C                                                                    1106
1107                STA     CR7C            INCLUDE IRG1 BITS 7-4 IN GENERATED ADDRESS              1107
1108      CR74      LDA     CR7A            DCSB                                                    1108
1109                LSRA    2                                                                       1109
1110                ANA     CR7B            FROM B15 & A13                                          1110
1111                ORA     CR7A            OR B13                                                  1111
1112                LRLA    2                                                                       1112
1113                JAP     CR80            JUMP IF EXPRESSION FALSE                                1113
1114                LDA     IRG1,1                                                                  1114
1115                ANA     =017                                                                    1115
1116                ORA     CR7C                                                                    1116
1117                STA     CR7C            INCLUDE IRG1 BITS 3-0 IN GENERATED ADDRESS              1117
1118      *                                                                                 VDM     1118
1119      *     IF REGISTER-REGISTER INSTRUCTION, MODIFY THE CONTROL STORE                  VDM     1119
1120      *     ADDRESS GENERATED.                                                          VDM     1120
1121      *     IF BIT "00" OF TEMP 1(CR7A) AND BIT XX5 OF TEMP 2(CR7B) ARE                 VDM     1121
1122      *     BOTH TRUE; PUT BIT 3 OF IRG2 INTO BIT 2 OF GENERATED ADDRESS.               VDM     1122
1123      *                                                                                 VDM     1123
1124      CR80      LDA     CR7A            DCSB                                                    1124
1125                LSRA    1                                                                       1125
1126                ANA     CR7B            BOTH "00" & "XX5" SET                                   1126
1127                ANA     =01000                                                                  1127
1128                JAZ     CR81            NO-                                                     1128
1129                LDA     CR7C                                                            VDM     1129
1130                LRLA    13              LEFT JUSTIFY BIT 2 OF ADDRESS                           1130
1131                LDB     IRG1,1                                                                  1131
1132                LRLB    12              LEFT JUSTIFY BIT 3 OF IRG1                              1132
1133                LLRL    1               REPLACE BIT 2 OF ADDRESS WITH BIT 3 OF IRG1             1133
1134                LRLA    2               POSITION ADDRESS PROPERLY                               1134
1135                STA     CR7C                                                            VDM     1135
1136      CR81      LDA     CR7C                                                                    1136
1137                JMP     CR32                                                                    1137
1138      *                                                                                 VDM     1138
1139      *     COME HERE FOR BCS INSTRUCTION                                               VDM     1139
```

```
1140        BCS     INCR    1               SELECT PAGE 1                                   1140
1141                CALL    PGOK            CHECK IF PAGE AVAIL AND SET POINTERS    VDM 1141
1142                JAP     BCSB                                                    VDM 1142
1143                LDA     IRG1,1          GET IRG1                                    1143
1144                ANA     =037            USE 5 LSBS OF IRG1                          1144
1145                STA     NROM,1          SET NEXT CCS ADDRESS                        1145
1146                JMP*    CRAS            RETURN                                  VDM 1146
1147        BCSB    CALL    ERRMSG,6        BCS INSTRUCTION, BUT NO PAGE 1              1147
1148        *     DATA ITEMS                                                        VDM 1148
1149        CR7A    DATA    0               TEMP STORE                              VDM 1149
1150        CR7B    DATA    0               TEMP STORE                              VDM 1150
1151        CR7C    DATA    0               TEMP STORE                              VDM 1151
1152        CR7D    DATA    0               TEMP STORE                              VDM 1152
1153        CRBA    DATA    0               STORAGE FOR BASE ADDRESS                VDM 1153
1154        CRAO    DATA    0               STORAGE FOR A OFFSET                    VDM 1154
1155        CRBO    DATA    0               STORAGE FOR B OFFSET                    VDM 1155
1156                EJEC                                                            VDM 1156
1157        *                                                                       VDM 1157
1158        *       DATA LOOP PROCESSING                                            VDM 1158
1159        *                                                                       VDM 1159
1160        *       CALLING SEQUENCE                                                VDM 1160
1161        *       LDX     =DROM           ADDRESS OF DATA BUFFER                      1161
1162        *       JMPM    S300                                                    VDM 1162
1163        *                                                                       VDM 1163
1164        *       RETURN  ROM PROCESSING WILL BE COMPLETE                         VDM 1164
1165        *                                                                       VDM 1165
1166        S300    ENTR                                                            VDM 1166
1167        *       DETERMINE STATE OF MULTIPLY SIGN (MULS) AT LAST ALU TIME.           1167
1168        *       MULS=(LA15 & LB15) V (LB15 & AL15) V (LA15 & AL15)                  1168
1169                LDA     LTCHA,1                                                     1169
1170                ANA     LTCHB,1         FORM (A & B)                            1170
1171                TAB                                                             1171
1172                LDA     LTCHA,1                                                 1172
1173                ORA     LTCHB,1         FORM (A V B)                            1173
1174                ANA     XALU,1          FORM (ALU & (A V B))                    1174
1175                MERG    031             FORM (A & B) V (ALU & (A V B))          1175
1176                TZB                                                             1176
1177                LLRL    1                                                       1177
1178                STB     MULS,1                                                  1178
1179        *       DETERMINE DFA OUTPUT.  DFA CLOCK IS SUPRESSED BY SH=XX1.        1179
1180                LDB     XA,1                                                    1180
1181                LDAE    RD,2            DFA REGISTER                            1181
1182                STA     DFA,1                                                   1182
1183                LDA     XLB,1                                                   1183
1184                SUB     =2              MASK IN USE?                            1184
1185                JAP     S304            YES=                                    1185
1186                LDA     XTC,1                                                   1186
1187                ANA     =1              SH=XX1?                                 1187
1188                JAZ     S302            NO=                                     1188
1189                TZA                                                             1189
1190                STA     DFA,1           DFA CLOCK SUPRESSED BY SH=XX1.          1190
1191        *       IF LB=0X AND V=1, DFA15 => DS                                   1191
1192        S302    LDA     XV,1                                                    1192
1193                JAZ     S304                                                    1193
1194                LDA     DFA,1                                                   1194
1195                TZB                                                         VDM 1195
1196                LLRL    1                                                   VDM 1196
```

```
1197            STB     DS,1            DS IS F/F FOR SIGN STORAGE                      VDM 1197
1198    S304    CALL    X000            DETERMINE LATCH A INPUT                         VDM 1198
1199            STA     LTCHA,1         SAVE LATCH A INPUT                              VDM 1199
1200    *                                                                               VDM 1200
1201            JMPM    X010            DETERMINE LATCH B INPUT TO ALU                  VDM 1201
1202            STA     LTCHB,1         SAVE LATCH B INPUT                              VDM 1202
1203    *                                                                               VDM 1203
1204            JMPM    X020            FORM ALU OUTPUT                                 VDM 1204
1205            STA     XALU,1          SAVE ALU OUTPUT                                 VDM 1205
1206    *                                                                               VDM 1206
1207            LDA     XLB,1                                                             1207
1208            LSRA    1               PERFORM OPERAND REGISTER SHIFTING IF              1208
1209            JAZM    DORG            LB DOES NOT SPECIFY LITERAL IN USE                1209
1210            CALL    X200            * TRANSFER ALU OUTPUT TO DESIGNATED REGS.       VDM 1210
1211    *                                                                               VDM 1211
1212    *   TRANSFER INST REG #1(IRG1) TO INST REG #2(IRG2) IF                          VDM 1212
1213    *   T=0 & S#3 & G=XXX1                                                          VDM 1213
1214            LDA     XS,1                                                              1214
1215            SUB     =3              S=3?                                              1215
1216            JAZ     S305            YES-                                              1216
1217            LDA     CQ2,1           SECOND 16 BITS OF ROM WORD                        1217
1218            ANA     =030100         SELECT T, G0                                      1218
1219            ERA     =0100           T=0 & G=XXX1?                                     1219
1220            MJANZ   S305            NO                                                1220
1221            LDA     IRG2,1          SAVE CONTENTS OF IRG2 AT LAST CLOCK.            VDM 1221
1222            STA     XIRG2,1         USED FOR R-R XFER.  SEE X108.                     1222
1223            LDA     IRG1,1          YES-MOVE IRG1                                   VDM 1223
1224            STA     IRG2,1          TO IRG2.                                        VDM 1224
1225    S305    LDA     PTRC,1          TRACE FLAG FOR CURRENTLY SELECTED PAGE            1225
1226            JAZ*    S300            JUMP IF NOT TRACE MODE                          VDM 1226
1227            JMPM    TRSETA          CHECK IF WITHIN BOUNDS                          VDM 1227
1228            JAN*    S300            NO-RETURN                                       VDM 1228
1229    *                                                                                 1229
1230    *       OUTPUT STACK LENGTH AND STACK TOP TO LO                                   1230
1231    *                                                                                 1231
1232            LDA     JSPT,1                                                            1232
1233            ADDI    STACK           ADDRESS OF CURRENT TOP OF STACK                   1233
1234            STA     *+3                                                               1234
1235            CALL    OHM,*-*,S320+12,JSPT+DROM,S320+19,0                               1235
1236            CALL    SPAC                                                              1236
1237            CALL    LOOUT,S320L,S320                                                  1237
1238    *                                                                                 1238
1239    *       OUTPUT ALU INFORMATION TO LO                                              1239
1240    *                                                                                 1240
1241            CALL    SPAC                                                              1241
1242            CALL    LOOUT,S325L,S325                                                  1242
1243            CALL    PBUF            CLEAR BUFR                                        1243
1244            CALL    OHM,LTCHA+DROM,BUFR+5,LTCHB+DROM,BUFR+10,XALU+DROM                1244
1245            DATA    BUFR+17,0                                                         1245
1246            LDX     =DROM           ADDRESS OF DATA BUFFER                            1246
1247            LDB     =(BUFR+1)       OUTPUT AREA                                       1247
1248            LDA     XCIN,1          CARRY IN                                          1248
1249            ADD     ='0'                                                              1249
1250            STA     13,2                                                              1250
1251            LDA     CARRY,1         CARRY OUT                                         1251
1252            ADD     ='0'                                                              1252
1253            STA     20,2                                                              1253
```

```
1254                CALL    LOOUT,S325L,BUFR                                    1254
1255                JMPM    REGS            DUMP CONTENTS OF REGISTERS      VDM 1255
1256        *                                                               VDM 1256
1257                CALL    LTAA            LIST V73 REGISTERS              VDM 1257
1258                JMP*    S300            RETURN                          VDM 1258
1259        *                                                               VDM 1259
1260        *                                                               VDM 1260
1261        S320    DATA    '  STACK:  CURRENT TOP=  DDDD, LENGTH= DDDD'        1261
1262        S320L   EQU     *-S320                                              1262
1263        S325    DATA    '  ALU:  INPUT A   INPUT B   CIN  OUTPUT  COUT '    1263
1264        S325L   EQU     *-S325                                              1264
1265        *                                                                   1265
1266        *       LIST V73 MACHINE REGISTERS                                  1266
1267        *                                                                   1267
1268        LTAA    ENTR                                                        1268
1269                CALL    SPAC                                                1269
1270                CALL    LOOUT,LTAZL,LTAZ                                    1270
1271                CALL    OHM,DROM+PREG,BUFR+1,DROM+SREG,BUFR+4,DROM+OREG,BUFR+7  1271
1272                DATA    DROM+KREG,BUFR+10,DROM+IOKR,BUFR+13,DROM+IRG1,BUFR+16   1272
1273                DATA    DROM+IRG2,BUFR+19,DROM+STUS,BUFR+22,DROM+IOD,BUFR+25    1273
1274                DATA    DROM+DS,BUFR+28,DROM+MULS,BUFR+31,0                 1274
1275                CALL    LOOUT,36,BUFR                                       1275
1276                RETU*   LTAA                                                1276
1277        *                                                                   1277
1278        *       LIST TEST CONDITION MUX                                     1278
1279        *                                                                   1279
1280        LTMX    ENTR                                                        1280
1281                LDB     CPAG,1                                              1281
1282                LDAE    TRACE,2         TRACE FLAG FOR PAGE ON?             1282
1283                JAZ*    LTMX            NO-EXIT                             1283
1284                CALL    TRSET#A         CHECK IF WITHIN BOUNDS              1284
1285                JAN*    LTMX            NO-EXIT                             1285
1286                CALL    SPAC                                                1286
1287                CALL    LOOUT,LTASL,LTAS                                    1287
1288                CALL    LOOUT,LTATL,LTAT                                    1288
1289                CALL    OHM,TMUX,BUFR+1,TMUX+1,BUFR+4,TMUX+2,BUFR+7         1289
1290                DATA    TMUX+3,BUFR+10,TMUX+4,BUFR+13,TMUX+5,BUFR+16        1290
1291                DATA    TMUX+6,BUFR+19,TMUX+7,BUFR+22,0                     1291
1292                CALL    LOOUT,24,BUFR                                       1292
1293                CALL    SPAC                                                1293
1294                CALL    LOOUT,LTAPL,LTAP                                    1294
1295                CALL    OHM,TMUX+8,BUFR+1,TMUX+9,BUFR+4,TMUX+10,BUFR+7      1295
1296                DATA    TMUX+11,BUFR+10,TMUX+12,BUFR+13,TMUX+13,BUFR+16     1296
1297                DATA    TMUX+14,BUFR+19,TMUX+15,BUFR+22,0                   1297
1298                CALL    LOOUT,24,BUFR                                       1298
1299                RETU*   LTMX                                                1299
1300        LTAZ    DATA    '  P      SC     OPR   KREG  IOKR  IBR   I      STAT'    VDM 1300
1301                DATA    '  IOR    SHFT   MULS'                                1301
1302        LTAZL   EQU     *-LTAZ          MESSAGE LENGTH                      1302
1303        LTAS    DATA    '                TEST CONDITION STATES'             VDM 1303
1304        LTASL   EQU     *-LTAS          MESSAGE LENGTH                      1304
1305        LTAT    DATA    '  OVFL  SENS  SSW3  SSW2  SSW1  EMUL  ALUO  ALUS'    VDM 1305
1306        LTATL   EQU     *-LTAT          MESSAGE LENGTH                      1306
1307        LTAP    DATA    '  ALUC  ALUZ  SHFT  MIRS  SFTC  ROAO  NORM  QUOS'    VDM 1307
1308        LTAPL   EQU     *-LTAP          MESSAGE LENGTH                      1308
1309                EJEC                                                        1309
1310        *   DETERMINE IF SPECIAL ALU MODE: LA=0&LB=0/1&SH=1XX .             VDM 1310
```

```
1311        *   IF NOT: INCR RETURN BY 2.                                  VDM 1311
1312        SAL1    LDA     CQ3,1           THIRD 16 BITS OF ROM WORD          1312
1313                ANA     =054000         SELECT LA & LB1                    1313
1314                ORA     XTC,1                                              1314
1315                ANA     =054004         SELECT LA, LB1, TC2                1315
1316                ERA     =4              LA=0 & LB=0X & TC=1XX?             1316
1317                JAZ*    SALU            YES, SPECIAL ALU MODE              1317
1318                INR     SALU                                               1318
1319                INR     SALU                                               1319
1320                JMP     *-*                                                1320
1321        SALU    BES     0                                                  1321
1322                JMP     SAL1                                               1322
1323                EJEC                                                   VDM 1323
1324        *                                                              VDM 1324
1325        *       DETERMINE LATCH A INPUT TO ALU                         VDM 1325
1326        *                                                              VDM 1326
1327        *       CALLING SEQUENCE                                           1327
1328        *       LDX     =DROM           ADDRESS OF DATA BUFFER             1328
1329        *       JMPM    X000                                           VDM 1329
1330        *                                                              VDM 1330
1331        *       RETURN  LATCH A INPUT IN A REGISTER                    VDM 1331
1332        *                                                              VDM 1332
1333        X000    ENTR                                                   VDM 1333
1334                LDB     DFA,1           GET FILE A DATA                    1334
1335        *                                                              VDM 1335
1336                LDA     XLA,1           LATCH A CONTROL                    1336
1337                ADD     X000T           PROCESSING ROUTINES JUMP TABLE     1337
1338                STA     *+2                                                1338
1339                JMP*    *-*             GO TO PROCESSING DEPENDING ON LA   1339
1340        X000T   DATA    (*+1)*                                             1340
1341        *       LA FIELD = 0    1    2    3                                1341
1342                DATA    XA30,X004,X002,XA49                                1342
1343        *                                                              VDM 1343
1344        *   IF LB=0X AND TC=001; FORCE LATCH A TO ZERO                     1344
1345        *   IF LB=0X AND TC=010; FORCE LATCH A TO ALL 1'S                  1345
1346        *   IF LB=0X AND TC=100; SET LATCH A=0 IF IRG2 BITS 7&6 # 11.  IF  1346
1347        *   IRG2 BITS 7+6=11(DECR),SET LATCH A=-1                      VDM 1347
1348        *   OTHERWISE, SELECT CONTENTS OF REGISTER DESIGNATED BY A FIELD VDM 1348
1349        XA30    LDA     XLB,1                                              1349
1350                SUB     =2              LB=0X                              1350
1351                JAP     XA32            NO=                                1351
1352                LDA     XTC,1                                          VDM 1352
1353                LRLA    13              TC=1XX?                            1353
1354                JAN     XS33            YES=                               1354
1355                LRLA    1               TC=X1X?                            1355
1356                JAN     XS36            YES=                               1356
1357                JAZ     XA32            JUMP IF TC=0                       1357
1358                JMP     XS34                                               1358
1359        XS33    LDA     IRG2,1                                         VDM 1359
1360                ANA     =0300           IF IRG2 BITS 7&6=11 SET LATCH A = -1.  1360
1361                SUB     =0300           IF IRG2 BITS 7&6 =00, 01 OR 10     1361
1362                JAZ     XS36            SET LATCH A TO ZERO.           VDM 1362
1363        XS34    TZB                     SET LATCH A=0                  VDM 1363
1364                DATA    01006           SKIP NEXT INSTRUCTION              1364
1365        *                                                              VDM 1365
1366        XS36    DECR    02              SET LATCH A = -1                   1366
1367        XA32    TBA                                                        1367
```

```
1368            RETU*   X000            SET LATCH A=-1                                  VDM 1368
1369    *                                                                               VDM 1369
1370    *   FORM RIGHT SHIFTED LATCH A INPUT                                            VDM 1370
1371    *                                                                               VDM 1371
1372    XA49    TBA                                                                         1372
1373            ANA     =1                                                                  1373
1374            STA     MBYC,1          BYTE DESIGNATOR FOR MEMORY OPERATIONS       VDM 1374
1375            LDA     XTC,1           GET TC FIELD                                        1375
1376            ADD     =XA40           COMPUTE PROCESSING ADDRESS                          1376
1377            STA     *+3                                                                 1377
1378            TBA                     FA IN BOTH A & B NOW                                1378
1379            XEC     *=*             GET LATCH A WORD BIT 15 INTO A LSB                  1379
1380            LRLB    1               REMOVE HSB FROM B REG                               1380
1381            LLRL    15              USE HSB FROM A LSB, RESULT TO A                     1381
1382            RETU*   X000                                                                1382
1383    *   TC=000: FA01 INTO LA00 AND MUL SIGN INTO LA15                                   1383
1384    XA40    LDA     MULS,1          MUL SIGN FF                                         1384
1385    *   TC=001: STORE BIT 00 OF FILE A INTO BIT 15 OF LATCH A WORD                      1385
1386    XA41    NOP                                                                         1386
1387    *   TC=010: FA01 INTO LA00 AND FA15 INTO LA15                                       1387
1388    XA42    LRLA    1               FA15 -> A LSB                                       1388
1389    *   TC=011: STORE BIT 00 OF OP REG INTO BIT 15 OF LATCH A WORD                      1389
1390    XA43    LDA     OREG,1          OPERAND REGISTER                                    1390
1391    *   TC=1XX: STORE ZERO INTO BIT 15 OF LATCH A WORD                                  1391
1392    XA44    TZA                                                                         1392
1393            TZA                                                                         1393
1394            TZA                                                                         1394
1395            TZA                                                                         1395
1396    *                                                                               VDM 1396
1397    *   COME HERE IF LA=10: FORM LEFT SHIFTED LATCH A INPUT                         VDM 1397
1398    *                                                                               VDM 1398
1399    X002    TBA                                                                         1399
1400            ANA     =1                                                                  1400
1401            STA     MBYC            BYTE DESIGNATOR FOR MEMORY OPERATIONS               1401
1402            LDA     XTC,1           GET TC FIELD                                VDM 1402
1403            ADD     XA10T           COMPUTE PROCESSING ADDRESS                          1403
1404            STA     *+4                                                                 1404
1405            TBA                                                                         1405
1406            LDB     OREG,1                                                              1406
1407            JMP*    *=*             GO TO PROCESSING DEPENDING ON TC                    1407
1408    XA10T   DATA    (*+1)*                                                              1408
1409    *       TC FIELD=  0    1    2    3    4    5    6    7                            1409
1410            DATA    XA10,XA11,XA12,XA17,XA14,XA15,XA16,XA17                             1410
1411    *   TC=000: FA14 INTO LA15 AND "ZERO" INTO LA00                                 VDM 1411
1412    XA10    ANA     =077777         ZERO => FA15                                        1412
1413    *                                                                               VDM 1413
1414    *   TC=001: FA14 INTO LA15 AND FA15 INTO LA00                                   VDM 1414
1415    XA11    LRLA    1                                                                   1415
1416            DATA    01006           SKIP NEXT WORD                                      1416
1417    *                                                                               VDM 1417
1418    *   TC=010: FA14 INTO LA15 AND OR15 INTO LA00                                   VDM 1418
1419    XA12    LLRL    1                                                                   1419
1420            DATA    01006           SKIP NEXT WORD                                      1420
1421    *                                                                                   1421
1422    *   TC=100: FA15 INTO LA15 AND ZERO INTO LA00                                       1422
1423    XA14    ASLA    1                                                                   1423
1424            RETU*   X000                                                                1424
```

```
1425      *  TC=101; FA15 INTO LA15 AND FA15 INTO LA00                    1425
1426      XA15   TAB                                                      1426
1427      *                                                               1427
1428      *  TC=110; FA15 INTO LA15 AND OR15 INTO LA00                    1428
1429      XA16   LSRB    1                                                1429
1430             LASL    1                                                1430
1431             DATA    01006          SKIP NEXT WORD                    1431
1432      *                                                               1432
1433      *  TC=X11; NOT USED, FLAG AS ERROR                              1433
1434      XA17   DECR    1                                                1434
1435             DATA    01006          SKIP NEXT WORD                    1435
1436      *                                                        VDM    1436
1437      *      PROGRAM COUNTER IS INPUT                          VDM    1437
1438      *                                                        VDM    1438
1439      X004   LDA     PREG,1         GET PROGRAM COUNTER       VDM    1439
1440             JMP*    X000                                     VDM    1440
1441             EJEC                                             VDM    1441
1442      *                                                        VDM    1442
1443      *      DETERMINE LATCH B INPUT TO ALU                    VDM    1443
1444      *                                                        VDM    1444
1445      *      CALLING SEQUENCE                                  VDM    1445
1446      *      LDX     =DROM          ADDRESS OF DATA BUFFER            1446
1447      *      JMPM    X010                                     VDM    1447
1448      *                                                        VDM    1448
1449      *      RETURN  LATCH B INPUT TO ALU IN A REGISTER        VDM    1449
1450      *                                                        VDM    1450
1451      X010   ENTR                                             VDM    1451
1452             LDA     XLB,1          LATCH B CONTROL                   1452
1453             ADD     X010T          JUMP TABLE BASE                   1453
1454             STA     *+3                                              1454
1455             LDA     MASK,1         IN CASE LB>1                      1455
1456             JMP*    *-*                                              1456
1457      X010T  DATA    (*+1)*                                           1457
1458      *      LATCH B =   0    1     2     3                           1458
1459             DATA    X010A,X012,X010C,(X010)*                         1459
1460      X010A  LDB     XB,1           FILE B SOURCE ADDRESS             1460
1461             LDAE    R0,2           B FILE REGISTER CONTENTS          1461
1462             DATA    01006          SKIP NEXT WORD                    1462
1463      *      MASKED I REGISTER IS B SOURCE                            1463
1464      X010C  ANA     IRG2,1         AND IRG2 WITH MASK COMPLEMENT     1464
1465             JMP*    X010                                             1465
1466      *                                                        VDM    1466
1467      *    COME HERE IF LB FIELD=01: SELECT B MUX FOR INPUT TO LATCH B VDM    1467
1468      *      INPUTS ARE OR REG, MAIN MEM, DATA REG, I/O DATA REG, STATUS VDM    1468
1469      *      REG, AND 4 CONFIGURATIONS OF BYTES FROM THE OP REG.         VDM    1469
1470      *                                                                  VDM    1470
1471      *                                                                  VDM    1471
1472      X012   LDA     XB,1           FILE B SOURCE ADDR                VDM    1472
1473             ANA     =7             3 LSB OF ADDR                     1473
1474             ADD     X012T          JUMP TABLE BASE                   1474
1475             STA     *+3                                              1475
1476             LDA     OREG,1         NECESSARY IF B EQ ZERO OR B GT 3  1476
1477             JMP*    *-*                                              1477
1478      X012T  DATA    (*+1)*         INDIRECT ADDRESS OF JUMP TABLE    1478
1479      *      B FIELD= X000   X001  X010  X011 X100 X101 X110  X111     1479
1480             DATA    (X010)*,X012B,X012C,X012D,X013,X014,X015,X012E    1480
1481      *                                                                1481
```

```
1482        *       B=X001  MEMORY INPUT LATCH                              1482
1483        *                                                               1483
1484        X012B   LDA     MIL,1           MEMORY INPUT LATCH              1484
1485                DATA    01006           SKIP NEXT WORD                  1485
1486        *                                                               1486
1487        *       B=X010  I/O DATA LATCH                                  1487
1488        *                                                               1488
1489        X012C   LDA     IOD,1           I/O DATA LATCH                  1489
1490                DATA    01006           SKIP NEXT WORD                  1490
1491        *                                                               1491
1492        *       B=X011  STATUS MUX                                      1492
1493        *                                                               1493
1494        X012D   LDA     STUS,1          STATUS MUX                      1494
1495                JMP*    X010                                            1495
1496        *                                                       VDM     1496
1497        *       B=X111  OP1 RIGHT BYTE TO LEFT BYTE             VDM     1497
1498        *                                                       VDM     1498
1499        X012E   ANA     =0377           MASK OUT LEFT BYTE              1499
1500                LRLA    8               LEFT JUSTIFY BYTE       VDM     1500
1501                JMP*    X010            RETURN                  VDM     1501
1502        *                                                       VDM     1502
1503        *       B=X100  OP1 RIGHT BYTE/SIGN EXTENDED            VDM     1503
1504        *                                                       VDM     1504
1505        X013    LRLA    8               LEFT JUSTIFY BYTE               1505
1506        *                                                               1506
1507        *       B=X101  OP1 LEFT BYTE/SIGN EXTENDED                     1507
1508        *                                                               1508
1509        X014    ASRA    8               EXTEND SIGN                     1509
1510                DATA    01006           SKIP NEXT WORD                  1510
1511        *                                                       VDM     1511
1512        *       B=X110  OP1 RIGHT BYTE                          VDM     1512
1513        *                                                       VDM     1513
1514        X015    ANA     =0377           MASK OUT LEFT BYTE              1514
1515                JMP*    X010            RETURN                  VDM     1515
1516                EJEC                                            VDM     1516
1517        *                                                       VDM     1517
1518        *       FORM ALU OUTPUT                                 VDM     1518
1519        *                                                       VDM     1519
1520        *       CALLING SEQUENCE                                VDM     1520
1521        *       LDX     =DROM           ADDRESS OF DATA BUFFER          1521
1522        *       JMPM    X020                                    VDM     1522
1523        *                                                       VDM     1523
1524        *       RETURN  OUTPUT IN A REGISTER                    VDM     1524
1525        *                                                       VDM     1525
1526        X020    ENTR                                            VDM     1526
1527        *   ARITHMETIC MODE: (M=0-AND-LB=0X) OR (LB=1X-AND-F=XX0X)      1527
1528        *   LOGICAL MODE: (M=1-AND-LB=0X) OR (LB=1X-AND-F=XX1X)         1528
1529        *                                                       VDM     1529
1530                ROF                     DON'T USE UNPREDICTABLE OVERFLOW1530
1531                LDA     XLB,1           GET LATCH B CONTROL             1531
1532                SUB     =2              LB=1X (MASK FIELD IN USE?)      1532
1533                JAN     XAR1            NO-M FIELD CONTROLS MODE        1533
1534                LDA     XF,1            GET F FIELD                     1534
1535                ANA     =2              BIT 1 OF F FIELD CONTROLS MODE  1535
1536                LSRA    1               POSITION BIT 1 LIKE M FIELD WILL BE1536
1537                DATA    01006           SKIP NEXT WORD                  1537
1538        *   LB<2, M FIELD CONTROLS ALU MODE                             1538
```

```
1539      XAR1   LDA     XMD,1          GET ALU MODE FIELD                          1539
1540             DAR                    IS MODE LOGICAL?                            1540
1541             JAZ     X023           YES                                         1541
1542      *                                                                   VDM 1542
1543      *    DETERMINE CARRY IN FOR ARITHMETIC MODE OR SPECIAL ALU MODE     VDM 1543
1544      *                                                                   VDM 1544
1545             LDA     XLB,1          GET LB FIELD                                1545
1546             SUB     =2                                                         1546
1547             JAP     AR12           LB=3 OR 4                           VDM 1547
1548             CALL    SALU           DETERMINE IF SPECIAL ALU MODE       VDM 1548
1549             JMP     AR20           YES=                                VDM 1549
1550             LDB     XC,1           CARRY CONTROL                       VDM 1550
1551             TZA                                                        VDM 1551
1552             JBZ     X026           JUMP IF ZERO CARRY IN               VDM 1552
1553             LDA     CARRY,1        STORED CARRY                        VDM 1553
1554             DBR                                                        VDM 1554
1555             JBZ     X026           JUMP IF USE STORED CARRY            VDM 1555
1556             ERA     =1             USE COMPLEMENTED CARRY                      1556
1557             DBR                                                        VDM 1557
1558             JBZ     X026                                               VDM 1558
1559             INCR    01             XCIN=1                              VDM 1559
1560      X026   STA     XCIN,1         CARRY IN FOR OPERATION              VDM 1560
1561             JMP     AR13                                               VDM 1561
1562      AR12   LDA     XF,1                                               VDM 1562
1563             ANA     =2                                                         1563
1564             JAZ     X026                                               VDM 1564
1565             LSRA    1                                                  VDM 1565
1566             JMP     X026                                               VDM 1566
1567      *                                                                  VDM 1567
1568      *    IF IRG2 BITS 7&6=01: SET CARRY-IN                             VDM 1568
1569      *    IF IRG2 BITS 7&6=00,10,11: RESET CARRY-IN                     VDM 1569
1570      AR20   LDA     IRG2,1         NO RETURN POINT FROM SALU           VDM 1570
1571             LSRA    6                                                          1571
1572             ANA     =3                                                         1572
1573             DAR                                                        VDM 1573
1574             JAZ     X026-1         GO SET CARRY-IN                     VDM 1574
1575             TZA                                                        VDM 1575
1576             JMP     X026           GO RESET CARRY-IN                   VDM 1576
1577      *                                                                  VDM 1577
1578      *      DO ARITHMETIC OPERATION                                     VDM 1578
1579      *                                                                  VDM 1579
1580      AR13   LDA     =(X022-X025)   DISTANCE FROM LOGICAL TO                    1580
1581      *                             ARITHMETIC FUNCTION TABLE                   1581
1582      *                                                                         1582
1583      *      DO LOGICAL OPERATION                                               1583
1584      *                                                                         1584
1585      X023   ADD     X025           ADDRESS OF LOGICAL FUNCTION TABLE           1585
1586             ADD     XF,1           ADD ALU FUNCTION INDEX                      1586
1587             STA     X021           SAVE FOR JUMP                       VDM 1587
1588             LDA     LTCHA,1        FREQUENTLY NEEDED                           1588
1589             JMP*    *-*            DO FUNCTION REQUESTED                       1589
1590      X021   BES     0              SAVED JUMP ADDRESS                  VDM 1590
1591      *                                                                  VDM 1591
1592      X022   DATA    (*+1)*         ARITHMETIC FUNCTIONS                        1592
1593             DATA    X063           0000 ADD A AND CIN                          1593
1594             DATA    X064           0001 OR A AND B, ADD CIN            VDM 1594
1595             DATA    X066           0010 COMPLEMENT B, OR WITH A,       VDM 1595
```

```
1596        *                                   ADD CIN                                  VDM 1596
1597                DATA    X068        0011 GENERATE ALL ONES, ADD CIN                  VDM 1597
1598                DATA    X070        0100 COMPLEMENT B, AND WITH A,                   VDM 1598
1599        *                                   ADD A, ADD CIN                           VDM 1599
1600                DATA    X072        0101 OR A AND B,                                 VDM 1600
1601        *                                   COMPLEMENT B, AND WITH A,                VDM 1601
1602        *                                   ADD RESULT, ADD CIN                      VDM 1602
1603                DATA    X074        0110 ADD A AND CIN, SUBTRACT B                   VDM 1603
1604        *                                   SUBTRACT 1                               VDM 1604
1605                DATA    X076        0111 COMPLEMENT B, AND WITH A,                   VDM 1605
1606        *                                   SUBTRACT 1, ADD CIN                      VDM 1606
1607                DATA    X078        1000 AND A AND B, ADD A, ADD CIN                 VDM 1607
1608                DATA    X080        1001 ADD A AND B, ADD CIN                        VDM 1608
1609                DATA    X082        1010 COMPLEMENT B, OR WITH A                     VDM 1609
1610        *                                   AND A AND B,                             VDM 1610
1611        *                                   ADD RESULT, ADD CIN                      VDM 1611
1612                DATA    X084        1011 AND A AND B, SUBTRACT 1, ADD CIN            VDM 1612
1613                DATA    X086        1100 ADD A AND A, ADD CIN                        VDM 1613
1614                DATA    X088        1101 OR A AND B, ADD A, ADD CIN                  VDM 1614
1615                DATA    X090        1110 COMPLEMENT B, OR WITH A,                    VDM 1615
1616        *                                   ADD A, ADD CIN                           VDM 1616
1617                DATA    X092        1111 ADD A, ADD CIN, SUBTRACT 1                  VDM 1617
1618        *                                                                            VDM 1618
1619        X025    DATA    (*+1)*      LOGICAL FUNCTIONS                                    1619
1620                DATA    X030        0000 COMPLEMENT A                                VDM 1620
1621                DATA    X032        0001 OR A AND B, COMPLEMENT                      VDM 1621
1622                DATA    X034        0010 COMPLEMENT A, AND WITH B                    VDM 1622
1623                DATA    X036        0011 GENERATE ALL ZEROS                          VDM 1623
1624                DATA    X038        0100 AND A AND B, COMPLEMENT                     VDM 1624
1625                DATA    X040        0101 COMPLEMENT B                                VDM 1625
1626                DATA    X042        0110 EXCLUSIVE OR A AND B                        VDM 1626
1627                DATA    X044        0111 COMPLEMENT B, AND WITH A                    VDM 1627
1628                DATA    X046        1000 COMPLEMENT A, OR WITH B                     VDM 1628
1629                DATA    X048        1001 EXCLUSIVE OR A AND B, COMPLEMENT            VDM 1629
1630                DATA    X050        1010 B                                           VDM 1630
1631                DATA    X052        1011 AND A AND B                                 VDM 1631
1632                DATA    X054        1100 GENERATE ALL ONES                           VDM 1632
1633                DATA    X056        1101 COMPLEMENT B, OR WITH A                     VDM 1633
1634                DATA    X058        1110 OR A AND B                                  VDM 1634
1635                DATA    (X020)*     1111 A                                               1635
1636                EJEC                                                                     1636
1637        *       NOTE - IN THE FOLLOWING ROUTINES A REFERS TO THE LATCH A             VDM 1637
1638        *       INPUT TO THE ALU AND B REFERS TO THE LATCH B INPUT                   VDM 1638
1639        *                                                                            VDM 1639
1640        *                                                                            VDM 1640
1641        *       NOTE - LATCH A CONTENTS ARE IN THE ACCUMULATOR WHEN THESE                1641
1642        *       ROUTINES ARE ENTERED                                                     1642
1643        *                                                                            VDM 1643
1644        *       COMPLEMENT A                (0000)                                   VDM 1644
1645        *                                                                            VDM 1645
1646        X030    CPA                         COMPLEMENT                                   1646
1647                JMP*    X020                RETURN                                   VDM 1647
1648        *                                                                            VDM 1648
1649        *       OR A AND B, COMPLEMENT  (0001)                                       VDM 1649
1650        *                                                                            VDM 1650
1651        X032    ORA     LTCHB,1             OR WITH B                                    1651
1652                JMP     X030                GO COMPLEMENT ACCUMULATOR                    1652
```

```
1653        *
1654        *       COMPLEMENT A, AND WITH B  (0010)
1655        *
1656   X034 CPA                       COMPLEMENT A
1657        ANA     LTCHB,1           AND WITH B
1658        JMP*    X020              RETURN
1659        *
1660        *       GENERATE ALL ZEROS     (0011)
1661        *
1662   X036 TZA
1663        JMP*    X020              RETURN
1664        *
1665        *       AND A AND B, COMPLEMENT  (0100)
1666        *
1667   X038 ANA     LTCHB,1           AND WITH B
1668        JMP     X030              GO COMPLEMENT ACCUMULATOR
1669        *
1670        *       COMPLEMENT B           (0101)
1671        *
1672   X040 LDA     LTCHB,1           B
1673        JMP     X030              GO COMPLEMENT ACCUMULATOR
1674        *
1675        *       EXCLUSIVE OR A AND B   (0110)
1676        *
1677   X042 ERA     LTCHB,1           EXCLUSIVE OR WITH B
1678        JMP*    X020              RETURN
1679        *
1680        *       COMPLEMENT B, AND WITH A  (0111)
1681        *
1682   X044 CPA                       COMPLEMENT A
1683        ORA     LTCHB,1           OR WITH B
1684        JMP     X030              COMPLEMENT RESULT (DE MORGAN'S THEOREM)
1685        *
1686        *       COMPLEMENT A, OR WITH B  (1000)
1687        *
1688   X046 CPA                       COMPLEMENT A
1689        ORA     LTCHB,1           OR WITH B
1690        JMP*    X020              RETURN
1691        *
1692        *       EXCLUSIVE OR A AND B, COMPLEMENT  (1001)
1693        *
1694   X048 ERA     LTCHB,1           EXCLUSIVE OR WITH B
1695        JMP     X030              GO COMPLEMENT ACCUMULATOR
1696        *
1697        *       B                      (1010)
1698        *
1699   X050 LDA     LTCHB,1           B
1700        JMP*    X020              RETURN
1701        *
1702        *       AND A AND B            (1011)
1703        *
1704   X052 EQU     X034+1
1705        *
1706        *       GENERATE ALL ONES      (1100)
1707        *
1708   X054 DECR    01                ALL ONES
1709        JMP*    X020              RETURN
```

```
VDM 1653
VDM 1654
VDM 1655
    1656
VDM 1657
VDM 1658
VDM 1659
VDM 1660
VDM 1661
VDM 1662
VDM 1663
VDM 1664
VDM 1665
VDM 1666
    1667
    1668
VDM 1669
VDM 1670
VDM 1671
VDM 1672
    1673
VDM 1674
VDM 1675
VDM 1676
    1677
VDM 1678
VDM 1679
VDM 1680
VDM 1681
    1682
    1683
    1684
VDM 1685
VDM 1686
VDM 1687
    1688
VDM 1689
VDM 1690
VDM 1691
VDM 1692
VDM 1693
    1694
    1695
VDM 1696
VDM 1697
VDM 1698
VDM 1699
VDM 1700
VDM 1701
VDM 1702
VDM 1703
    1704
VDM 1705
VDM 1706
VDM 1707
VDM 1708
VDM 1709
```

```
1710      *                                                                    VDM 1710
1711      *       COMPLEMENT B, OR WITH A  (1101)                              VDM 1711
1712      *                                                                    VDM 1712
1713      X056  CPA                     COMPLEMENT A                             1713
1714            ANA     LTCHB,1         AND WITH B                               1714
1715            JMP     X030            COMPLEMENT RESULT (DE MORGAN'S THEOREM)  1715
1716      *                                                                    VDM 1716
1717      *       OR A AND B            (1110)                                 VDM 1717
1718      *                                                                    VDM 1718
1719      X058  ORA     LTCHB,1         OR WITH B                                1719
1720            JMP*    X020            RETURN                                 VDM 1720
1721            EJEC                                                           VDM 1721
1722      *       NOTE - THE CONTENTS OF LATCH A ARE IN THE ACCUMULATOR WHEN        1722
1723      *       THESE ROUTINES ARE ENTERED                                       1723
1724      *                                                                    VDM 1724
1725      *       0000    ADD A AND CIN                                        VDM 1725
1726      *                                                                    VDM 1726
1727      X063  ADD     XCIN,1          ADD CIN                                VDM 1727
1728            JMPM    X094            SAVE OVERFLOW                          VDM 1728
1729            LDA     XCIN,1          CIN                                    VDM 1729
1730            JAZ     X095            NO CARRY OUT IF NO CARRY IN            VDM 1730
1731            LDA     X099            ALU RESULT                             VDM 1731
1732            JAZ     X097            SET CARRY ON ZERO RESULT               VDM 1732
1733            JMP     X095            NO CARRY OUT                           VDM 1733
1734      *                                                                    VDM 1734
1735      *       0001    OR A AND B, ADD CIN                                  VDM 1735
1736      *                                                                    VDM 1736
1737      X064  ORA     LTCHB,1         OR WITH B                                1737
1738            JMP     X063            ADD CIN                                VDM 1738
1739      *                                                                    VDM 1739
1740      *       0010    COMPLEMENT B, OR WITH A, ADD CIN                     VDM 1740
1741      *                                                                    VDM 1741
1742      X066  LDA     LTCHB,1         B                                      VDM 1742
1743            COMPL   011             COMPLEMENT                             VDM 1743
1744            ORA     LTCHA,1         OR WITH A                              VDM 1744
1745            JMP     X063            ADD CIN                                VDM 1745
1746      *                                                                    VDM 1746
1747      *       0011    GENERATE ALL ONES, ADD CIN                           VDM 1747
1748      *                                                                    VDM 1748
1749      X068  DECR    01              ALL ONES                               VDM 1749
1750            JMP     X063            ADD CIN                                VDM 1750
1751      *                                                                    VDM 1751
1752      *       0100    COMPLEMENT B, AND WITH A, ADD A, ADD CIN             VDM 1752
1753      *                                                                    VDM 1753
1754      X070  LDA     LTCHB,1         B                                      VDM 1754
1755            COMPL   011             COMPLEMENT                             VDM 1755
1756            ANA     LTCHA,1         AND WITH A                             VDM 1756
1757            STA     XOAB            SECOND RESULT                          VDM 1757
1758            ADD     LTCHA,1         ADD A                                  VDM 1758
1759            ADD     XCIN,1          ADD CIN                                VDM 1759
1760            JMPM    X094            SAVE OVERFLOW                          VDM 1760
1761            LDA     LTCHA,1         A                                      VDM 1761
1762            STA     XOAA            FIRST RESULT                           VDM 1762
1763      *                                                                    VDM 1763
1764      X071  LDA     XCIN,1          CIN                                    VDM 1764
1765            JAZ     XOAO            DO CARRY CALCULATION                   VDM 1765
1766            LDA     X099            ALU RESULT                             VDM 1766
```

```
1767                JAZ     X097            SET CARRY ON ZERO RESULT                        VDM 1767
1768                JMP     X0A0            DO CARRY CALCULATION                            VDM 1768
1769      *                                                                                 VDM 1769
1770      *         0101    ADD (OR A AND B) WITH (COMPLEMENT B, AND WITH A).               VDM 1770
1771      *                 ADD CIN                                                         VDM 1771
1772      *                                                                                 VDM 1772
1773      X072      ORA     LTCHB,1         OR WITH B                                           1773
1774                STA     X0AA            FIRST RESULT                                    VDM 1774
1775                LDA     LTCHB,1         B                                               VDM 1775
1776                COMPL   011             COMPLEMENT                                      VDM 1776
1777                ANA     LTCHA,1         AND WITH A                                      VDM 1777
1778                STA     X0AB            SECOND RESULT                                   VDM 1778
1779                ADD     X0AA            ADD FIRST RESULT                                VDM 1779
1780                ADD     XCIN,1          ADD CIN                                         VDM 1780
1781                JMPM    X094            SAVE OVERFLOW                                   VDM 1781
1782                JMP     X071            DO CARRY CALCULATION                            VDM 1782
1783      *                                                                                 VDM 1783
1784      *         0110    ADD A AND CIN, SUBTRACT B, SUBTRACT 1                           VDM 1784
1785      *                                                                                 VDM 1785
1786      X074      SUB     LTCHB,1         SUBTRACT B                                          1786
1787                DAR                     SUBTRACT 1                                      VDM 1787
1788                ADD     XCIN,1          ADD CIN                                         VDM 1788
1789                JMPM    X094            SAVE OVERFLOW                                   VDM 1789
1790                LDA     XCIN,1          CIN                                             VDM 1790
1791                JAZ     X075            JUMP IF NO CARRY IN                             VDM 1791
1792      *                                                                                 VDM 1792
1793                LDA     LTCHA,1         A                                               VDM 1793
1794                ERA     LTCHB,1         B                                               VDM 1794
1795                JAP     *+7             JUMP IF SIGNS THE SAME                          VDM 1795
1796                LDA     LTCHA,1         A                                               VDM 1796
1797                JAN     X097            SET CARRY IF A NEGATIVE                         VDM 1797
1798                JMP     X096            NO CARRY                                        VDM 1798
1799      *                                                                                 VDM 1799
1800                LDA     LTCHA,1         A                                               VDM 1800
1801                SUB     LTCHB,1         LESS B                                          VDM 1801
1802                JAP     X097            SET CARRY IF A GREATER                          VDM 1802
1803                JMP     X096            NO CARRY                                        VDM 1803
1804      *                                                                                 VDM 1804
1805      X075      LDA     X099            RESULT                                          VDM 1805
1806                CPA                                                                     VDM 1806
1807                JAZ     X096            NO CARRY ON -1 RESULT                           VDM 1807
1808                JMP     X097            SET CARRY                                       VDM 1808
1809      *                                                                                 VDM 1809
1810      *         0111    COMPLEMENT B, AND WITH A, SUBTRACT 1, ADD CIN                   VDM 1810
1811      *                                                                                 VDM 1811
1812      X076      LDA     LTCHB,1         B                                               VDM 1812
1813                COMPL   011             COMPLEMENT                                      VDM 1813
1814                ANA     LTCHA,1         AND WITH A                                      VDM 1814
1815                DAR                     SUBTRACT 1                                      VDM 1815
1816                ADD     XCIN,1          ADD CIN                                         VDM 1816
1817                JMPM    X094            SAVE OVERFLOW                                   VDM 1817
1818                LDA     XCIN,1          CIN                                             VDM 1818
1819                JAZ     X075            JUMP IF NO CARRY IN                             VDM 1819
1820                JMP     X096            NO CARRY OUT                                    VDM 1820
1821      *                                                                                 VDM 1821
1822      *         1000    AND A AND B, ADD A, ADD CIN                                     VDM 1822
1823      *                                                                                 VDM 1823
```

```
1824      X078  STA     XOAA          STORE FIRST RESULT                    1824
1825            ANA     LTCHB,1       AND WITH B                      VDM 1825
1826            STA     XOAB          SECOND RESULT                   VDM 1826
1827            ADD     LTCHA,1       ADD A                           VDM 1827
1828            ADD     XCIN,1        ADD CIN                         VDM 1828
1829            JMPM    X094          SAVE OVERFLOW                   VDM 1829
1830            JMP     X071          DO CARRY CALCULATION            VDM 1830
1831      *                                                             VDM 1831
1832      *     1001    ADD A AND B, ADD CIN                            VDM 1832
1833      *                                                             VDM 1833
1834      X080  STA     XOAA          STORE FIRST RESULT                    1834
1835            ADD     LTCHB,1       ADD B                           VDM 1835
1836            ADD     XCIN,1        ADD CIN                         VDM 1836
1837            JMPM    X094          SAVE OVERFLOW                   VDM 1837
1838            LDA     LTCHB,1       B                               VDM 1838
1839            STA     XOAB          SECOND RESULT                   VDM 1839
1840            JMP     X071          DO CARRY CALCULATION            VDM 1840
1841      *                                                             VDM 1841
1842      *     1010    ADD (COMPLEMENT B, OR WITH A) WITH (AND A AND B),VDM 1842
1843      *             ADD CIN                                         VDM 1843
1844      *                                                             VDM 1844
1845      X082  LDA     LTCHB,1       B                               VDM 1845
1846            COMPL   011           COMPLEMENT                      VDM 1846
1847            ORA     LTCHA,1       OR WITH A                       VDM 1847
1848            STA     XOAA          FIRST RESULT                    VDM 1848
1849            LDA     LTCHA,1       A                               VDM 1849
1850            ANA     LTCHB,1       AND WITH B                      VDM 1850
1851            STA     XOAB          SECOND RESULT                   VDM 1851
1852            ADD     XOAA          ADD FIRST RESULT                VDM 1852
1853            ADD     XCIN,1        ADD CIN                         VDM 1853
1854            JMPM    X094          SAVE OVERFLOW                   VDM 1854
1855            JMP     X071          DO CARRY CALCULATION            VDM 1855
1856      *                                                             VDM 1856
1857      *     1011    AND A AND B, SUBTRACT 1, ADD CIN                VDM 1857
1858      *                                                             VDM 1858
1859      X084  ANA     LTCHB,1       AND WITH B                            1859
1860            DAR                   SUBTRACT 1                      VDM 1860
1861            ADD     XCIN,1        ADD CIN                         VDM 1861
1862            JMPM    X094          SAVE OVERFLOW                   VDM 1862
1863            LDA     XCIN,1        CIN                             VDM 1863
1864            MJANZ   X096          NO CARRY OUT                          1864
1865            LDA     X099          ALU RESULT                      VDM 1865
1866            CPA                   COMPLEMENT                      VDM 1866
1867            JAZ     X096          NO CARRY OUT ON -1 RESULT       VDM 1867
1868            JMP     X097          SET CARRY OUT                   VDM 1868
1869      *                                                             VDM 1869
1870      *     1100    ADD A AND A, ADD CIN                            VDM 1870
1871      *                                                             VDM 1871
1872      X086  ADD     LTCHA,1       ADD A                                 1872
1873            ADD     XCIN,1        ADD CIN                         VDM 1873
1874            JMPM    X094          SAVE OVERFLOW                   VDM 1874
1875            LDA     LTCHA,1       A                               VDM 1875
1876            JAN     X097          SET CARRY IF A NEGATIVE         VDM 1876
1877            JMP     X096          NO CARRY OUT                    VDM 1877
1878      *                                                             VDM 1878
1879      *     1101    OR A AND B, ADD A, ADD CIN                      VDM 1879
1880      *                                                                 1880
```

```
1881        X088    ORA     LTCHB,1         OR WITH B                                   1881
1882                ORA     LTCHB,1         OR WITH B                           VDM     1882
1883                STA     X0AB            SECOND RESULT                       VDM     1883
1884                ADD     LTCHA,1         ADD A                               VDM     1884
1885        *                                                                   VDM     1885
1886        X089    ADD     XCIN,1          ADD CIN                             VDM     1886
1887                JMPM    X094            SAVE OVERFLOW                       VDM     1887
1888                JMP     X071-2          DO CARRY CALCULATION                VDM     1888
1889        *                                                                   VDM     1889
1890        *       1110    COMPLEMENT B, OR WITH A, ADD A, ADD CIN             VDM     1890
1891        *                                                                   VDM     1891
1892        X090    LDA     LTCHB,1         B                                   VDM     1892
1893                COMPL   011             COMPLEMENT                          VDM     1893
1894                ORA     LTCHA,1         OR WITH A                           VDM     1894
1895                STA     X0AB            SECOND RESULT                       VDM     1895
1896                ADD     LTCHA,1         ADD A                               VDM     1896
1897                JMP     X089            DO CARRY CALCULATION                VDM     1897
1898        *                                                                   VDM     1898
1899        *       1111    ADD A AND CIN, SUBTRACT 1                           VDM     1899
1900        *                                                                   VDM     1900
1901        X092    DAR                     SUBTRACT 1                                  1901
1902                ADD     XCIN,1          ADD CIN                             VDM     1902
1903                JMPM    X094            SAVE OVERFLOW                       VDM     1903
1904                LDA     XCIN,1          CIN                                 VDM     1904
1905                MJANZ   X096            NO CARRY OUT                                1905
1906                LDA     LTCHA,1         A                                   VDM     1906
1907                JAZ     X096            NO CARRY OUT IF A ZERO              VDM     1907
1908                JMP     X097            SET CARRY OUT                       VDM     1908
1909                EJEC                                                        VDM     1909
1910        *                                                                   VDM     1910
1911        *       SAVE OVERFLOW FROM ARITHMETIC OPERATION                     VDM     1911
1912        *                                                                   VDM     1912
1913        X094    ENTR                                                        VDM     1913
1914                STA     X099            SAVE RESULT                         VDM     1914
1915                TZA                                                                 1915
1916                AOFA                                                                1916
1917                ORA     OVRFL,1         SET STORED OVERFLOW                         1917
1918                STA     OVRFL,1         DON'T RESET STORED OVERFLOW                 1918
1919                JMP*    X094            RETURN                              VDM     1919
1920        *                                                                   VDM     1920
1921        *       SUBROUTINE TO CHECK IF SAMPLING (T=00 & S=X0 & G=xX1X)              1921
1922        *               RETURN INCR BY TWO IF NOT SAMPLING                  VDM     1922
1923        *                                                                   VDM     1923
1924        X09B    LDA     CQ2,1           SECOND 16 BITS OF ROM WORD                  1924
1925                ANA     =032200         SELECT T, S0, G1                            1925
1926                ERA     =0200           T=00 & S=X0 & G=XX1X                        1926
1927                JAZ*    X09A            YES-X09A (SAMPLING)                         1927
1928                INR     X09A                                                        1928
1929                INR     X09A                                                        1929
1930                JMP     *-*                                                         1930
1931        X09A    BES     0                                                           1931
1932                JMP     X09B                                                        1932
1933        *                                                                   VDM     1933
1934        *       CARRY CALCULATION ROUTINE                                   VDM     1934
1935        *                                                                   VDM     1935
1936        X0A0    JMPM    X09A            CHECK IF SAMPLING                           1936
1937                JMP     *+4             YES-                                        1937
```

```
1938                JMP     X096        NO=                                          1938
1939                LDA     X0AA        FIRST RESULT                                 1939
1940                ANA     X0AB        SECOND RESULT                          VDM   1940
1941                JAN     X097        SET CARRY IF BOTH NEGATIVE             VDM   1941
1942                LDA     X0AA        FIRST RESULT                           VDM   1942
1943                JAN     X0A1        JUMP IF NEGATIVE                       VDM   1943
1944                LDA     X0AB        SECOND RESULT                          VDM   1944
1945                JAP     X095        NO CARRY IF RESULT POSITIVE            VDM   1945
1946                ADD     X0AA        PLUS FIRST RESULT                      VDM   1946
1947                JMP     X098        SET CARRY IF FIRST RESULT LARGER             1947
1948          *                                                                  VDM   1948
1949          *                                                                  VDM   1949
1950    X0A1  LDA     X0AB        SECOND RESULT                                  VDM   1950
1951                JAN     X095        NO CARRY IF RESULT POSITIVE            VDM   1951
1952                ADD     X0AA        PLUS FIRST RESULT                      VDM   1952
1953    X098  JAP     X097        SET CARRY IF FIRST RESULT NOT LARGER               1953
1954    X095  TZA                 SET CARRY = 0                                      1954
1955                DATA    01006       SKIP NEXT WORD                               1955
1956    X097  INCR    1           SET CARRY = 1                                      1956
1957                STA     CARRY*,1    SAVE CARRY                                   1957
1958    X096  LDA     X099        ALU RESULT                                         1958
1959                RETU*   X020                                                     1959
1960          *                                                                  VDM   1960
1961    X099  DATA    0           TEMP STORE                                         1961
1962    X0AA  DATA    0           FIRST RESULT                           VDM   1962
1963    X0AB  DATA    0           SECOND RESULT                          VDM   1963
1964                EJEC                                                       VDM   1964
1965          *                                                                  VDM   1965
1966          *   FORM MACHINE STATUS REGISTER SETTING(STUS)                     VDM   1966
1967          *   TEST CONDITION SETTINGS FOR TEST MUX(TMUX)                     VDM   1967
1968          *   AND MISC SETTINGS: SUPERVISOR KEY, I/O KEY REG, INTERRUPT FLAG,VDM   1968
1969          *                      MUL SIGN                                    VDM   1969
1970          *                                                                  VDM   1970
1971          *   THE MACHINE STATUS REGISTER(STUS) IS AN INPUT TO THE LATCH B MUX VDM   1971
1972          *                                                                  VDM   1972
1973          *   THE TEST CONDITIONS(SEE TMUX TABLE UP FRONT) ARE SELECTED BY THE VDM   1973
1974          *   G FIELD(WHEN T FIELD IS NOT=00) IN DETERMINING THE NEXT ROM    VDM   1974
1975          *   ADDRESS,THIS IS DONE IN COMPONENT CRAS.                        VDM   1975
1976          *                                                                  VDM   1976
1977          *   A PSEUDO 16-BIT STATUS REGISTER(STUS) IS USED IN THE PROGRAM. IN VDM   1977
1978          *   THE HARDWARE THERE IS NO "REGISTER, ITS JUST GATING OF F/F'S,ETC.VDM   1978
1979          *   INTO THE BMUX.                                                 VDM   1979
1980          *   SIMILARLY, A TEST CONDITION TABLE(TMUX) IS USED HERE-IN.        VDM   1980
1981          *                                                                  VDM   1981
1982          *                                                                  VDM   1982
1983          *   MACHINE STATUS "REGISTER" (STUS) BITS                                1983
1984          *      0   NOT USED                                                VDM   1984
1985          *      1   NOT USED                                                VDM   1985
1986          *      2   ALU ZERO                                                     1986
1987          *      3   SHIFT COUNTER BIT 0                                     VDM   1987
1988          *      4   SHIFT COUNTER BIT 1                                     VDM   1988
1989          *      5   SHIFT COUNTER BIT 2                                     VDM   1989
1990          *      6   SHIFT COUNTER BIT 3                                     VDM   1990
1991          *      7   SHIFT COUNTER BIT 4                                     VDM   1991
1992          *      8   ALU OVERFLOW                                            VDM   1992
1993          *      9   ALU EQUALS                                              VDM   1993
1994          *      10  ALU SIGN                                                VDM   1994
```

```
1995      *       11  ALU CARRY                                                    VDM 1995
1996      *       12  KEY REG BIT 0,DK12    THE DATA LOOP KEY REGISTER IS            1996
1997      *       13  KEY REG BIT 1,DK13    A 4-BIT REGISTER THAT                    1997
1998      *       14  KEY REG BIT 2,DK14    STORES THE STATE OF                      1998
1999      *       15  KEY REG BIT 3,DK15    ALU-OUT BITS 12-15                       1999
2000      *                                                                        VDM 2000
2001      *                                                                        VDM 2001
2002      *   TEST MUX (TMUX) TABLE                                                   2002
2003      *       0   ALU OVERFLOW                                                 VDM 2003
2004      *       1   I/O SENSE                                                    VDM 2004
2005      *       2   SSW3                                                         VDM 2005
2006      *       3   SSW2                                                         VDM 2006
2007      *       4   SSW1                                                         VDM 2007
2008      *       5   620/F TEST (FOR JMP, JMPM, $XEC GROUPS)                      VDM 2008
2009      *       6   ALU EQUALS                                                   VDM 2009
2010      *       7   ALU SIGN                                                     VDM 2010
2011      *       8   ALU CARRY                                                    VDM 2011
2012      *       9   ALU ZERO                                                     VDM 2012
2013      *       10  DS  FLIP/FLOP  (SIGN STORAGE FOR OREG SHIFTING                 2013
2014      *       11  MIL15  SIGN BIT OF MEMORY LATCH                            VDM 2014
2015      *       12  SHIFT COUNT = -1                                              2015
2016      *       13  A15-SIGN OF A REG FOR MUL, OPERATIONS                      VDM 2016
2017      *       14  DAL15 NOT EQUAL TO DAL14  (ALU OUTPUT BITS 14 & 15)           2017
2018      *       15  QS FLIP/FLOP   (SAVES ALU15/OR01 FOR MUL/DIV OPS)             2018
2019      *                                                                        VDM 2019
2020      ***************************************************************        VDM 2020
2021      *  BIT 11 (MIL15) IS ESTABLISHED IN MEM, OPERATIONS  COMPONENT.  *       VDM 2021
2022      *                                                                        VDM 2022
2023      *                                                                        VDM 2023
2024      *    CALLING SEQUENCE                                                    VDM 2024
2025      *      LDX     =DROM          ADDRESS OF DATA BUFFER                      2025
2026      *      CALL    X100                                                    VDM 2026
2027      *                                                                      VDM 2027
2028      *                                                                      VDM 2028
2029      X100  ENTR                                                            VDM 2029
2030      *                                                                      VDM 2030
2031      *                                                                      VDM 2031
2032      *   THIS SEQUENCE CONTROLS TEST CONDITION BITS 1-5,10,AND 12-15        VDM 2032
2033      *   THE ALU RELATED BITS; 0 & 6-9 FOLLOW.                              VDM 2033
2034      *                                                                      VDM 2034
2035      *                                                                      VDM 2035
2036      *   TEST BIT 1; I/O SENSE                                              VDM 2036
2037      *      I/O SENSE NOT SIMULATED                                           2037
2038      *                                                                      VDM 2038
2039      *                                                                      VDM 2039
2040      *   TEST BITS 2,3,4; SENSE SWITCHES 3,2,1                              VDM 2040
2041      *                                                                      VDM 2041
2042            INCR    1              A=1                                        2042
2043            TZB                    B=0                                     VDM 2043
2044            STB     4+TMUXX,1      SS1=0                                      2044
2045            STB     3+TMUXX,1      SS2=0                                      2045
2046            STB     2+TMUXX,1      SS3=0                                      2046
2047            XS1     SETSW1                                                    2047
2048            XS2     SETSW2                                                    2048
2049            XS3     SETSW3                                                    2049
2050      *   TEST BIT 13; STORES SIGN BIT(DAL15) OF ALU OUTPUT WHENEVER         VDM 2050
2051      *              FILE REG.0(A REG) IS WRITTEN INTO.                      VDM 2051
```

```
2052                LDA     XW,1            DON'T UPDATE A15 F/F                    2052
2053                JAZ     XS30            IF NOT WRITING INTO RO                  2053
2054                LDA     XLB,1           LB IS SPECIFYING                VDM 2054
2055                SUB     =2              MASK FIELD                              2055
2056                JAP     XS30            DO NOT SET A15 F/F.             VDM 2056
2057                LDA     XA,1                                                    2057
2058                MJANZ   XS30                                                    2058
2059                LDB     XALU,1                                                  2059
2060                LLRL    1                                                       2060
2061                STA     13+TMUXX,1                                              2061
2062          *   TEST BIT 10; STATUS OF DS F/F  1=SET  0=RESET               VDM 2062
2063          XS30  LDA     DS,1                                            VDM 2063
2064                STA     10+TMUXX,1                                              2064
2065          *   TEST BIT 12; SHIFT COUNT=-1                                 VDM 2065
2066                ROF                                                             2066
2067                TZB                                                             2067
2068                LDA     SREG,1                                                  2068
2069                ADD     =1              WAS SREG =-1? (DON'T USE IAR INSTRUCTION)2069
2070                AOFB                    SREG WAS = -1 IF OVERFLOW               2070
2071                STB     12+TMUXX,1                                              2071
2072          *   TEST BIT 14; DAL15 NOT EQUAL TO DAL14.                      VDM 2072
2073                LDA     XALU,1          ALU OUTPUT                              2073
2074                LRLA    1               POSITION BIT 14 TO 15                   2074
2075                ERA     XALU,1          COMPARE BITS                    VDM 2075
2076                LSRA    15              POSITION RESULT                         2076
2077                STA     14+TMUXX,1      SET NORMALIZE FLAG                      2077
2078          *   TEST BIT 15; STATUS OF QS F/F  1=SET  0=RESET               VDM 2078
2079                LDA     QS,1                                                    2079
2080                STA     15+TMUXX,1                                              2080
2081          *                                                               VDM 2081
2082          *                                                               VDM 2082
2083          *   THE FOLLOWING STATUS REGISTER AND TEST CONDITIONS ARE       VDM 2083
2084          *   CONDITIONALLY CONTROLLED BY FIELDS S,T,&G.                  VDM 2084
2085          *                                                               VDM 2085
2086          *                                                               VDM 2086
2087          *   SET/RESET "EQUALS","STORED-CARRY","ZEROS","OVERFLOW" & "ALU SIGN"IF VDM 2087
2088          *    (S=0&T=0&G=XX1X)-OR-(S=1&T=0&GX11X)                        VDM 2088
2089          *    OTHERWISE,SKIP TO X104                                     VDM 2089
2090          *                                                               VDM 2090
2091                JMPM    X09A            CHECK IF SAMPLING ALU CONDITIONALS      2091
2092                JMP     XS65            YES                                     2092
2093                LDA     CQ2,1           SECOND 16 BITS OF ROM WORD              2093
2094                ANA     =036600         SELECT T, S, G1, G2                     2094
2095                ERA     =02600          T=00 & S=01 & G =X11X                   2095
2096                JAZ     X103            YES, GO SAMPLE OVERFLOW                 2096
2097                LDA     CQ2,1           SECOND 16 BITS OF ROM WORD              2097
2098                ANA     =037000         SELECT T, S, G3                         2098
2099                ERA     =01000          T=00 & S=00 & G=1XXX                    2099
2100                JAZ     X103            GO SAMPLE OVERFLOW                      2100
2101                JMP     X104            NOT SAMPLING ANYTHING                   2101
2102          *   SET "EQUALS" F/F IF ALU RESULT=ALL ONES                     VDM 2102
2103          XS65  INCR    1                                                       2103
2104                LDB     XALU,1          IS ALU OUTPUT                           2104
2105                IBR                     EQUAL TO ALL ONES ?                     2105
2106                JBZ     XS70            YES                                     2106
2107                TZA                                                             2107
2108          XS70  STA     6+TMUXX,1       "EQUALS" IS POSITION 6 IN TMUX          2108
```

```
2109        *   SET STORED-CARRY F/F IF CARRY OUT WAS GENERATED                  VDM 2109
2110            LDA     CARRY,1                                                     2110
2111            STA     8+TMUXX,1       "CARRY" IS POSITION 8 IN TMUX               2111
2112        *   SET "ALU OUT=ZERO" F/F IF ALU RESULT IS ZERO                     VDM 2112
2113            INCR    1                                                           2113
2114            LDB     XALU,1          IS ALU OUTPUT EQUAL TO ZERO ?               2114
2115            JBZ     XS78            YES                                         2115
2116        *   RESET ALU OUT=ZERO BIT IF ALU RESULT IS NOT ZERO                 VDM 2116
2117            TZA                                                                 2117
2118        XS78    STA     9+TMUXX,1       "ZEROS" IS POSITION 9 IN TMUX           2118
2119        *                                                                    VDM 2119
2120        *   SET ALU SIGN BIT IF ALU RESULT WAS NEGATIVE                      VDM 2120
2121            TZA                                                                 2121
2122            LDB     XALU,1                                                      2122
2123            LLRL    1                                                           2123
2124            STA     7+TMUXX,1       ALU SIGN BIT IS POSITION 7 IN TMUX          2124
2125            LDA     XG,1                                                        2125
2126            ANA     =8              G FIELD = 1XXX?                             2126
2127            JAZ     XS90                                                    VDM 2127
2128        *  SAMPLE STORED OVERFLOW IF: 1. SPECIAL ALU MODE(LA=0&LB=0X&SH(TC)=1XX     2128
2129        *                                AND BIT 6 OF IRG2=1.               VDM 2129
2130        *                             2. S=0&T=0&G=1XXX.                    VDM 2130
2131        *                             3. S=1&T=NOT 0&G=X11X.                VDM 2131
2132        X103    CALL    SALU            SPECIAL ALU MODE ?                  VDM 2132
2133            JMP     X108            YES-                                    VDM 2133
2134        X109    LDA     OVRFL,1         OVERFLOW ON?                        VDM 2134
2135            JMP     XS89                                                        2135
2136        *                                                                    VDM 2136
2137        X108    LDA     XIRG2,1         IRG2 AT LAST CLOCK                      2137
2138            ANA     =0100           IRG2 BIT 6 = 1 ?                            2138
2139            JAZ     XS90            NO-DON'T SAMPLE OVFL                        2139
2140            JMP     X109            YES-SAMPLE OVFL.                    VDM 2140
2141        *                                                                    VDM 2141
2142        *   SET/RESET  "OVERFLOW" BIT IN STUS &TMUX IF S=01&T=00&G=X01X/X10X  VDM 2142
2143        *                                                                    VDM 2143
2144        X104    LDA     CQ2,1           SECOND 16 BITS OF ROM WORD              2144
2145            ANA     =036600         SELECT T, S, G1, G2                         2145
2146            ERA     =032100         T=00 & S=01 & G=X01X                        2146
2147            JAZ     X107            YES- GO SET OVERFLOW BIT                    2147
2148            ERA     =0300           T=00 & S=01 & G=X10X                        2148
2149            MJANZ   XS90                                                        2149
2150            DATA    01006           SKIP NEXT WORD                              2150
2151        X107    INCR    1                                                       2151
2152        XS89    STA     TMUXX,1         "OVERFLOW" IS POSITION ZERO IN TMUX     2152
2153            STA     OVRFL,1         SET STORED OVERFLOW                         2153
2154        *   IF S=0, SET/RESET INTERRUPT FLAG, SUPERVISOR KEY, AND I/O        VDM 2154
2155        *           REQUEST FLAG--AS SPECIFIED BY FIELD IMC                  VDM 2155
2156        XS90    LDA     XS,1            S=0 ?                               VDM 2156
2157            MJANZ   X140            NO                                          2157
2158            LDA     XIMC,1                                                  VDM 2158
2159            SUB     =12             IMC>1011                                    2159
2160            JAP     X140            YES                                         2160
2161            ADD     =7              IMC<0101                                    2161
2162            JAN     X140            YES                                         2162
2163            ADD     XS90T                                                       2163
2164            STA     *+3                                                         2164
2165            TZA                                                                 2165
```

```
2166                JMP*    *-*                                                 2166
2167        *       IM FIELD =      5   6   7   8   9   A   B                   2167
2168        XS90T   DATA    (*+1)*,XS91,XS91,XS92,X130,X140,X134,X136            2168
2169        *   IMC=0111: SET INTERRUPT FLAG                                VDM 2169
2170        XS92    INCR    01                                              VDM 2170
2171        *   IMC=0101/0110: RESET INTERRUPT FLAG (A=0 IF WE JUMPED HERE)     2171
2172        XS91    STA     INPF,1          SET VALUE OF INT BEING PROCESSED FLAG 2172
2173                JMP     X140                                                2173
2174        *   IMC=1011: SET SUPERVISOR KEY                                VDM 2174
2175        X136    INCR    1               SET A=1                             2175
2176        *   IMC=1010: RESET SUPERVISOR KEY (A=0 IF WE JUMPED HERE)          2176
2177        X134    STA     SUPK,1          SET VALUE OF SUPERVISOR KEY         2177
2178                JMP     X140                                                2178
2179        *                                                               VDM 2179
2180        *   IMC=1000: LOAD I/O KEY REGISTER WITH 4 MSB'S OF                 2180
2181        *             ALU OUTPUT: RIGHT JUSTIFIED                       VDM 2181
2182        X130    LDA     XALU,1                                              2182
2183                LSRA    12              RIGHT JUSTIFY                       2183
2184                STA     IOKR,1                                              2184
2185        *                                                                   2185
2186        *       RECONSTRUCT THE STATUS "REGISTER" (STUS).                   2186
2187        *                                                                   2187
2188        X140    TZB                                                         2188
2189                LDA     SUPK,1          SUPERVISOR KEY                      2189
2190                LLSR    1                                                   2190
2191                LDA     9+TMUXX,1       ALU ZERO                            2191
2192                LLSR    1                                                   2192
2193                LDA     SREG,1          SHIFT COUNTER LSB 5 BITS            2193
2194                LLSR    5                                                   2194
2195                LDA     TMUXX,1         OVERFLOW                            2195
2196                LLSR    1                                                   2196
2197                LDA     6+TMUXX,1       ALU EQUALS ALL ONES                 2197
2198                LLSR    1                                                   2198
2199                LDA     7+TMUXX,1       ALU SIGN                            2199
2200                LLSR    1                                                   2200
2201                LDA     8+TMUXX,1       ALU CARRY                           2201
2202                LLSR    1                                                   2202
2203                LDA     KREG,1          DATA LOOP KEY REGISTER              2203
2204                LLSR    4                                                   2204
2205                STB     STUS,1          STATUS MUX                          2205
2206                RETU*   X100            RETURN                          VDM 2206
2207        *                                                                   2207
2208        SETSW1  STA     4+TMUXX,1       SET SS1 ON                          2208
2209        SETSW2  STA     3+TMUXX,1       SET SS2 ON                          2209
2210        SETSW3  STA     2+TMUXX,1       SET SS3 ON                          2210
2211                EJEC                                                    VDM 2211
2212        *                                                               VDM 2212
2213        *                                                               VDM 2213
2214        *   THIS ROUTINE PERFORMS THE OPERAND REGISTER SHIFTING OPERATIONS   2214
2215        *     THE FOLLOWING CONTROL FIELDS ARE USED:OS(SC),Y(W),V,X,&LB  VDM 2215
2216        *     ALSO USED: DS,QS,OREG,FILE BITS FA00&FA15, & ALU-OUT BIT 15. VDM 2216
2217        *                                                               VDM 2217
2218        *     CALLING SEQUENCE                                          VDM 2218
2219        *      LDX    =DROM           ADDRESS OF DATA BUFFER                2219
2220        *      CALL   DORG                                              VDM 2220
2221        *                                                               VDM 2221
2222        *      RETU   OPERAND REGISTER SHIFTING COMPLETE                VDM 2222
```

```
2223        *                                                                   VDM 2223
2224        DORG    ENTR                                                        VDM 2224
2225                LDA     XOS,1           SHIFT OP REG. ?                     VDM 2225
2226                MJANZ   D001                                                    2226
2227                INCR    1               RESET INTERRUPT FLAG                    2227
2228                ANA     XX,1            IF SC=0 & XX=X1.                        2228
2229                JAZ     *+4             JUMP SIGNAL NOT SIMULATED               2229
2230                TZA                                                             2230
2231                STA     INPF,1          INTERRUPT BEING PROCESSED FLAG          2231
2232                LDA     XY,1            IS Y(HARDWARE W FIELD)= 0  ?        VDM 2232
2233                JAZ*    DORG            YES=NO ACTION, EXIT.                VDM 2233
2234                LDA     XALU,1          STORE SIGN                          VDM 2234
2235                ANA     =0100000        BIT OF ALU OUTPUT                       2235
2236                LRLA    1               IN                                  VDM 2236
2237                STA     QS,1            QS F/F.                             VDM 2237
2238                JMP*    DORG            EXIT                                VDM 2238
2239        D001    LDA     OREG,1                                                  2239
2240                STA     D0T1            TEMP SAVE                           VDM 2240
2241        *   Y(W) FIELD SPECIFIES LEFT/RIGHT SHIFTING                        VDM 2241
2242                LDA     XY,1                                                    2242
2243                LRLA    2                                                       2243
2244                ORA     XX,1                                                    2244
2245                ADD     D020T           COMPUTE PROCESSING ADDRESS              2245
2246                STA     D020J+1         DEPENDING ON X & Y(W).                  2246
2247                LDA     DFA,1           FILE A INPUT                            2247
2248                LDB     OREG,1                                                  2248
2249        D020J   JMP*    *-*             GO TO PROCESSING                        2249
2250        D020T   DATA    (*+1)*                                                  2250
2251        *  X,Y FIELDS = 0,0  0,1  0,2  0,3  1,0  1,1  1,2  1,3                  2251
2252                DATA    D020,D021,D022,D023,D024,D025,D026,D027                 2252
2253        *  Y=0: LEFT SHIFTS                                                     2253
2254        *                                                                       2254
2255        *  X=2; COMPLEMENT OF ALU BIT 15 -> OR00                                2255
2256        D022    LDA     XALU,1                                                  2256
2257                CPA                                                             2257
2258                DATA    01006           SKIP NEXT WORD                          2258
2259        *  X=0: OR15 -> OR00                                                    2259
2260        D020    LDA     OREG,1                                                  2260
2261                DATA    01006           SKIP NEXT WORD                          2261
2262        *  X=3; ZERO -> OR00                                                    2262
2263        D023    TZA                                                             2263
2264        *  X=1: FA15 -> OR00                                                    2264
2265        D021    LLRL    1               DO SHIFTING                             2265
2266                JMP     D030                                                    2266
2267        *  Y=1; RIGHT SHIFTS                                                    2267
2268        *                                                                       2268
2269        *  X=0; OR00 -> OR15                                                    2269
2270        D024    LDA     OREG,1                                                  2270
2271                DATA    01006           SKIP NEXT WORD                          2271
2272        *  X=3: DSB -> OR15                                                     2272
2273        D027    LDA     DS,1                                                    2273
2274        *  X=1: FA00 -> OR15                                                    2274
2275        D025    LLSR    1               DO SHIFTING                             2275
2276                DATA    01006           SKIP NEXT WORD                          2276
2277        *  X=2: OR15 -> OR15                                                    2277
2278        D026    ASRB    1               DO SHIFTING                             2278
2279        *  STORE OREG AND RETURN                                                2279
```

```
2280      D030    STB     OREG,1                                                  2280
2281              JMP*    DORG                                            VDM 2281
2282      *                                                               VDM 2282
2283      DOT1    DATA    0               TEMP STORE FOR PREVIOUS OREG CONTENTS   VDM 2283
2284              EJEC                                                    VDM 2284
2285      *                                                               VDM 2285
2286      *   STORE ALU OUTPUT AS SPECIFIED BY R-FIELD AND W FIELD        VDM 2286
2287      *                                                               VDM 2287
2288      *   CALLING SEQUENCE                                            VDM 2288
2289      *     LDX     =DROM           ADDRESS OF DATA BUFFER                2289
2290      *     JMPM    X200                                                  2290
2291      *     RETURN                                                        2291
2292      *                                                               VDM 2292
2293      *                                                               VDM 2293
2294      X200    ENTR                    **                              VDM 2294
2295              LDA     XLB,1           DON'T OUTPUT TO                 VDM 2295
2296              LSRA    1               FILE REGISTERS                  VDM 2296
2297              MJANZ   X202            IF MSB OF LB FIELD = 1                2297
2298              LDA     XW,1            WRITE TO FILE REGISTER ?        VDM 2298
2299              JAZ     X202            NO-                             VDM 2299
2300              LDA     XALU,1          STORE ALU OUTPUT                VDM 2300
2301              LDB     XA,1            REGISTER FILE ADDRESS                 2301
2302              STAE    R0,2            STORE ALU OUTPUT IN REGISTER FILE     2302
2303      *                                                               VDM 2303
2304      X202    LDA     XR,1            R-FIELD                         VDM 2304
2305              ADD     X200T                                                 2305
2306              STA     *+3                                                   2306
2307              LDA     XALU,1                                                2307
2308              JMP*    **                                                    2308
2309      X200T   DATA    (*+1)*          INDIRECT ADDRESS OF JUMP TABLE        2309
2310      *       R FIELD =    0      1       2       3       4       5    6    7 2310
2311              DATA    (X200)*,X200R1,X200R2,X200R3,X200R4,X200R5,X206,X207  2311
2312      X207    STA     OREG,1          R = 7, LOAD OPERAND & INCREMENT P     2312
2313      X200R4  INR     PREG,1          R = 4, INCREMENT P REGISTER           2313
2314              SKIP    NEXT                                                  2314
2315      X200R1  STA     PREG,1          R = 1, LOAD P REGISTER                2315
2316              DATA    01006           SKIP NEXT WORD                        2316
2317      X200R3  STA     OREG,1          R = 3, LOAD OPERAND REGISTER          2317
2318              DATA    01006           SKIP NEXT WORD                        2318
2319      X200R5  INR     SREG,1          R = 5, COUNT SHIFT REGISTER           2319
2320              JMP*    X200                                                  2320
2321      *       R=2     SHIFT COUNTER IS 8 BITS.  MUST SET UPPER BITS FOR     2321
2322      X200R2  ORA     =0177400        PROPER SIMULATION                     2322
2323              STA     SREG,1                                                2323
2324              JMP*    X200                                                  2324
2325      *       R=6     LOAD PROCESSOR KEY FROM 4 MSB OF ALU                  2325
2326      X206    LSRA    12                                                    2326
2327              STA     KREG,1                                                2327
2328              JMP*    X200                                                  2328
2329              EJEC                                                    VDM 2329
2330      *                                                               VDM 2330
2331      *                                                               VDM 2331
2332      ****** MEMORY OPERATIONS COMPONENT ******                        VDM 2332
2333      *                                                               VDM 2333
2334      *                                                               VDM 2334
2335      *     * OPERATION CODES *                                       VDM 2335
2336      *       MOPC=0: TRANSFER ALU OUTPUT TO MIL AND IREG#1.          VDM 2336
```

```
2337      *        MOPC=1: READ FROM MEMORY TO MIL AND IREG#1.                    VDM 2337
2338      *        MOPC=2: READ FROM MEMORY TO MIL.                               VDM 2338
2339      *        MOPC=3: WRITE 16-BIT ALU OUTPUT TO MEMORY                      VDM 2339
2340      *        MOPC=4: WRITE ALU OUTPUT AS A BYTE TO MEMORY                   VDM 2340
2341      *                (SPECIFIED BY BYTE POINTER)                            VDM 2341
2342      *                                                                       VDM 2342
2343      *                                                                       VDM 2343
2344      *    * ADDRESS SOURCES *                                                VDM 2344
2345      *        MADS=0: ADDRESS IS ALU OUTPUT                                  VDM 2345
2346      *        MADS=1: ADDRESS IS PROGRAM COUNTER                             VDM 2346
2347      *        MADS=2: ADDRESS IS MEMORY INTERFACE LATCH(MIL)                 VDM 2347
2348      *                                                                       VDM 2348
2349      *                                                                       VDM 2349
2350      *  * MEMORY CONDITION CODES *                                           VDM 2350
2351      *        MCCO=0: IDLE                                                   VDM 2351
2352      *        MCCO=1: IDLE WITH REQUEST PENDING                                  2352
2353      *        MCCO=2-N: ACTIVE (N=MEMORY TYPE)                                   2353
2354      *                                                                       VDM 2354
2355      *                                                                       VDM 2355
2356      *   CALLING SEQUENCE                                                    VDM 2356
2357      *      CALL    MOPA                                                     VDM 2357
2358      *                                                                       VDM 2358
2359      *                                                                       VDM 2359
2360      MOPA  ENTR                                                            VDM 2360
2361            LDX     =DROM          ADDRESS OF ROM FIELDS & DATA                2361
2362      *                                                                       VDM 2362
2363      *   DETERMINE MEMORY CONDITION                                          VDM 2363
2364            LDA     MCCO,1         MCCO IS THE COND. CODE FOR MEM.OPERATIONS    2364
2365            MJANZ   M032           0=IDLE,1=ACTIVE NOT DONE,2=ACTIVE&DONE       2365
2366      *   MEMORY IS IDLE                                                      VDM 2366
2367      M004  CALL    MREQ           ANY MEMORY REQUEST?                          2367
2368            STA     MCCO,1         CLEAR CONDITION CODE IF NO REQUEST           2368
2369            JAZ     M071           RETURN VIA LISTING ROUTINE IF NO REQUEST     2369
2370            CALL    M010           DETERMINE MEMORY OPERATION                   2370
2371      *                                                                       VDM 2371
2372      *   DETERMINE MEMORY ADDRESS SOURCE                                     VDM 2372
2373            LDA     XS,1           FIELD S=0?                                   2373
2374            JAZ     M026           YES-NOT VALID ADDR SOURCE CONDITION      VDM 2374
2375            LDB     XIMC,1         GET FIELD IMC                                2375
2376            LSRB    2              GET TWO HIGH-ORDER BITS OF IMC               2376
2377            JBZ     M026           INVALID ADDRESS SOURCE                       2377
2378            LDAE*   MADST-1,2      GET LOCATION OF MEMORY ADDRESS SOURCE        2378
2379            STA     LREG,1                                                      2379
2380            DATA    01006          SKIP NEXT WORD                               2380
2381      *   SET MADS TO INDICATE INVALID MEMORY ADDRESS SOURCE                      2381
2382      *   THIS WILL PRODUCE THE LETTER X ON PRINT OUT.                        VDM 2382
2383      M026  LDB     =('X'-'0'+1)                                                2383
2384            STB     MADS,1                                                      2384
2385            INCR    01             SET MEMORY CONDITION CODE                    2385
2386            STA     MCCO,1         TO: ACTIVE BUT NOT DONE                  VDM 2386
2387            TZA                    CLEAR COMPLETE MEMORY CYCLE FLAG             2387
2388            JMP     M071           RETURN VIA LISTING ROUTINE                   2388
2389      *     MEMORY ADDRESS SOURCE TABLE                                         2389
2390      MADST DATA    DROM+XALU      MEMORY ADDRESS SOURCE IS ALU OUTPUT          2390
2391            DATA    DROM+PREG      MEMORY ADDRESS SOURCE IS P REGISTER          2391
2392            DATA    DROM+MIL       MEMORY ADDRESS SOURCE IS MIL                 2392
2393      *                                                                       VDM 2393
```

```
2394        *   COME HERE IF MCCO > 0 (MEMORY ACTIVE)                                  2394
2395        M032  DAR                    MCCO=1                              VDM 2395
2396              MJANZ   M036           NO, CHECK FOR COMPLETION                   2396
2397        *   MCCO=1                                                         VDM 2397
2398              LDA     MOPC,1         MEM OP IS XFER ALU-OUT TO MIL/IRG1         2398
2399              JAZ     M036           YES                                        2399
2400              CALL    MOOR           OVERRIDE ?                          VDM 2400
2401              JAN     M038           YES-                                VDM 2401
2402              TZA                                                               2402
2403              STA     MPLE           CLEAR MPLE FLAG                            2403
2404              LDA     MOPC,1         CHECK IF STORE REQ                         2404
2405              SUB     =3                                                       2405
2406              JAN     M036           NO                                         2406
2407              LDA     LREG,1                                                   2407
2408              SUB     PREG,1         IF LREG=PREG SET MPLE                      2408
2409              MJANZ   M036           NO                                         2409
2410              INCR    1                                                        2410
2411              STA     MPLE                                                     2411
2412              JMP     M036                                                     2412
2413        *   MCCO = 1 WITH OVERRIDE                                             2413
2414        M038  CALL    M010           DETERMINE MEMORY OPERATION                 2414
2415        *       MAINTAIN OLD ADDRESS SOURCE                                    2415
2416        *   MEMORY REQUESTED OR ACTIVE                                         2416
2417        M036  INR     MCCO,1         UPDATE MEMORY CONDITION CODE               2417
2418        *                                                                 VDM 2418
2419        *   ENTER HERE IF MEMORY IS REQUESTED OR ACTIVE                        2419
2420        *                                                                      2420
2421        *   MEMORY CYCLE COMPLETE?                                             2421
2422              TZA                                                              2422
2423              STA     MLIP           CLEAR COMPLETE MEMORY FLAG                 2423
2424              LDA     MCCO,1                                                   2424
2425              SUB     MTYP,1                                                   2425
2426              JAP     M050           IF MEMORY DONE                             2426
2427        *   WAIT FOR MEMORY DONE?                                              2427
2428              LDA     XS,1                                                     2428
2429              MJANZ   M041           S>0                                        2429
2430              LDA     XIMC,1                                                   2430
2431              DAR                                                              2431
2432              JAZ     M045           IF IMC=1                                   2432
2433        *   PENDING MEMORY REQUEST                                             2433
2434        M041  CALL    MREQ           ANY NEW MEMORY REQUESTS                    2434
2435              JAZ     M070           NO, RETURN VIA LISTING ROUTINE             2435
2436        M045  LDA     MTYP,1         MEMORY COMPLETION COUNT                    2436
2437              STA     MCCO,1         SET CONDITION CODE TO COMPLETE             2437
2438              INCR    1              SET COMPLETE MEMORY FLAG                   2438
2439              JMP     M071           RETURN VIA LISTING ROUTINE                 2439
2440        M050  CALL    MOPB           GET MEMORY ADDRESS                  VDM 2440
2441              LDA     MOPC,1                                             VDM 2441
2442              SUB     =5                                                       2442
2443              JAP     M051           UNDEFINED OP CODE IF > 4                   2443
2444              ADD     M05T                                                     2444
2445              STA     *+2                                                      2445
2446              JMP*    *-*            GO TO PROCESSING DEPENDING UPON MOPC.      2446
2447        M05T  DATA    (**6)*         NOTE - 5 SUBTRACTED ABOVE                  2447
2448        *          MOPC=  0    1    2    3    4                               2448
2449              DATA    M052,M054,M056,M058,M060                                 2449
2450        M051  CALL    ERRMSC,13      UNDEFINED MEMORY OP CODE.                  2450
```

```
2451                    JMP     M004                                        2451
2452        *                                                           VDM 2452
2453        *   MOPC=0: XFER ALU OUTPUT TO MIL & IRG1.                  VDM 2453
2454        M052    LDA     XALU,1                                      VDM 2454
2455                DATA    01006           SKIP NEXT WORD                  2455
2456        *   MOPC=1;READ FROM MEMORY TO MIL AND IRG1                 VDM 2456
2457        M054    LDA     0,2             B REG CONTAINS MEM ADDR     VDM 2457
2458                STA     IRG1,1                                      VDM 2458
2459                DATA    01006           SKIP NEXT INSTRUCTION           2459
2460        *   MOPC=2;READ FROM MEMORY TO MIL.                         VDM 2460
2461        M056    LDA     0,2             B REG CONTAINS MEM ADDR     VDM 2461
2462                STA     MIL,1                                       VDM 2462
2463                JMP     M004                                            2463
2464        *   WRITE 16-BIT ALU OUTPUT TO MEMORY                       VDM 2464
2465        M058    CALL    PMEM            CHECK FOR PROTECTED MEMORY      2465
2466                LDA     XALU,1                                      VDM 2466
2467                JMP     M063                                            2467
2468        *       WRITE ALU OUTPUT AS A BYTE TO MEMORY                VDM 2468
2469        *                                                           VDM 2469
2470        M060    CALL    PMEM            CHECK PROTECTED MEMORY      VDM 2470
2471                LDA     MBYC,1          GET BYTE DESIGNATOR         VDM 2471
2472                MJANZ   M061            1=RIGHT BYTE, ZERO=LEFT BYTE    2472
2473                LDA     0,2             B REG CONTAINS MEM ADDR VIA MOPB VDM 2473
2474                ANA     =0377           SET RIGHT BYTE FROM MEMORY      2474
2475                STA     M06A                                        VDM 2475
2476                LDA     XALU,1          STORE LEFT BYTE OF          VDM 2476
2477                ANA     =0177400        ALU OUTPUT WITH RIGHT BYTE FROM MEMORY  2477
2478                JMP     M062                                        VDM 2478
2479        *   RIGHT BYTE                                              VDM 2479
2480        M061    LDA     0,2                                         VDM 2480
2481                ANA     =0177400        GET LEFT BYTE FROM MEMORY       2481
2482                STA     M06A                                        VDM 2482
2483                LDA     XALU,1                                      VDM 2483
2484                ANA     =0377           GET RIGHT BYTE OF ALU OUTPUT    2484
2485        M062    ERA     M06A                                            2485
2486        M063    STA     0,2             B REG HAS MEMORY ADDRESS        2486
2487        *   IF MPLE SET, WRITE OUTPUT INTO IRG1 TO PRESERVE PIPELINE    2487
2488                LDB     MPLE                                            2488
2489                JBZ     M004            GO CHECK FOR MEMORY REQUEST     2489
2490                STA     IRG1,1                                          2490
2491                TZA                                                     2491
2492                STA     MPLE            CLEAR MPLE                      2492
2493                JMP     M004                                            2493
2494        M06A    DATA    0               TEMP STORE                  VDM 2494
2495        *                                                               2495
2496        *       DETERMINE MEMORY OPERATION TO BE PERFORMED.             2496
2497        *       RETURN WITH OPERATION CODE STORED IN MOPC.              2497
2498        *                                                               2498
2499        M010    ENTR                                                    2499
2500                LDA     XS,1            S=0?                            2500
2501                JAZ     M012            YES-                            2501
2502        M011    LDA     XIMC,1          GET IMC FIELD                   2502
2503                ANA     =3              SELECT 2 LSBS OF IMC            2503
2504                IAR                     MOPC = 2 LSBS OF IMC + 1        2504
2505        M013    STA     MOPC,1                                          2505
2506                RETU*   M010                                            2506
2507        *   S=0                                                         2507
```

```
2508      MO12   LDA     XIMC,1                                          2508
2509             SUB     =4              IMC = 0100?                     2509
2510             JAZ     MO13            YES=                            2510
2511             JMP     MO11                                            2511
2512      *                                                       VDM 2512
2513      *                                                       VDM 2513
2514      *      SUBROUTINE TO CHECK IF MEMORY REF IS OUTSIDE DELARED MAIN MEMORY VDM 2514
2515      *                                                       VDM 2515
2516      *      CALLING SEQUENCE                                 VDM 2516
2517      *      LDB     MEMORY ADDRESS                           VDM 2517
2518      *      CALL    PMEM                                     VDM 2518
2519      *                                                       VDM 2519
2520      *      RETURN--IF ERROR,ERROR MESSAGE AND BYPASS READ/WRITE VDM 2520
2521      *                                                       VDM 2521
2522      PMEM   ENTR                                             VDM 2522
2523             TBA                     MEMORY ADDRESS          VDM 2523
2524             SUB     =MMEM           START OF MAIN MEMORY            2524
2525             JAN     PM1             BELOW MAIN MEMORY       VDM 2525
2526             TBA                                             VDM 2526
2527             SUB     =(EMEM+1)       TOP OF USERS MEMORY AREA + 1    2527
2528             JAN*    PMEM            MEMORY ADDR OK          VDM 2528
2529      PM1    CALL    ERRMSC,14       STORE OUT OF ALLOWED RANGE      2529
2530             LDX     =DROM           (ERRMSC CLOBBERED X)            2530
2531             LDBI    MO6A            DUMMY ADDRESS                   2531
2532             JMP*    PMEM            CONTINUE                        2532
2533      *                                                       VDM 2533
2534      *    GET MEM ADDR STORED IN L REGISTER                  VDM 2534
2535      *    RETURN WITH ADDR IN B REG.                         VDM 2535
2536      MOPB   ENTR                                             VDM 2536
2537             LDB     LREG,1                                          2537
2538             LDA     MADS,1          GET ADDR SOURCE CODE    VDM 2538
2539             SUB     =4              VALID MADS VALUE?               2539
2540             JAN*    MOPB            YES (NOTE MADS=MADS-1 STILL)    2540
2541      *    SET ADDR TO 177777 AS DEFAULT IF MADS NOT 0,1,OR 2.VDM 2541
2542             DECR    02              B=-1                    VDM 2542
2543             RETU*   MOPB                                    VDM 2543
2544      *                                                       VDM 2544
2545      *                                                       VDM 2545
2546      *    SUBROUTINE TO CHECK FOR MEMORY REQUEST             VDM 2546
2547      *     CALLING SEQUENCE                                  VDM 2547
2548      *      LDX     =DROM           ADDRESS OF DATA BUFFER          2548
2549      *      CALL    MREQ                                    VDM 2549
2550      *                                                      VDM 2550
2551      *      RETURN; A REG=0=>NO REQUEST                            2551
2552      *                                                      VDM 2552
2553      *                                                      VDM 2553
2554      MREQ   ENTR                                            VDM 2554
2555             CALL    MOOR            OVERRIDE?                       2555
2556             JAN     MREN            YES-THEN NO MEM REQUEST VDM 2556
2557             LDB     XG,1            GET G FIELD                     2557
2558             LDBE    TMUX,2          CONTENTS OF TEST CONDITION TABLE 2558
2559             LDA     XS,1                                    VDM 2559
2560             ADD     MREQT                                           2560
2561             STA     *+2                                             2561
2562             JMP*    *-*                                             2562
2563      MREQT  DATA    (*+1)*                                          2563
2564      *      S FIELD =  0    1    2    3                             2564
```

```
2565                DATA    MRE0,MREY,MRE2,MRE3                                  2565
2566          * S=3                                                              2566
2567          MRE3  JBZ     MREN          TEST FALSE => NO MEMORY REQUEST        2567
2568          MREY  INCR    1             SET A REG = +1                         2568
2569                DATA    01006         SKIP NEXT INSTRUCTION                  2569
2570          MREN  TZA                   A=0 => NO REQUEST                      2570
2571                RETU*   MREQ                                                 2571
2572          * S=2                                                              2572
2573          MRE2  LDA     XT,1          T=0?                                   2573
2574                JAZ     MREY          YES, MEMORY REQUEST                    2574
2575                JBZ     MREY          TEST FALSE => MEMORY REQUEST           2575
2576                JMP     MREN          TEST TRUE => NO MEMORY REQUEST         2576
2577          * S=0                                                              2577
2578          MRE0  LDA     XIMC,1                                               2578
2579                SUB     =4            IMC = 0100?                            2579
2580                JAZ     MREY          YES, MEMORY REQUEST                    2580
2581                JMP     MREN          NO, NO MEMORY REQUEST                  2581
2582          *                                                          VDM 2582
2583          *                                                          VDM 2583
2584          *   SUBROUTINE TO CHECK FOR OVERRIDE                       VDM 2584
2585          *    CALLING SEQUENCE                                      VDM 2585
2586          *      LDX     =DROM       ADDRESS OF DATA BUFFER                 2586
2587          *      CALL    MOOR                                               2587
2588          *                                                          VDM 2588
2589          *    RETURN; A REG=0=NO OVERRIDE, A=-1=OVERRIDE            VDM 2589
2590          *                                                          VDM 2590
2591          *   OVERRIDE CONDITIONS: IM=00XX &[S=1 OR (S=2 & T=0) OR (S=2 &    2591
2592          *                          CONDITION SPECIFIED BY G IS FALSE)      2592
2593          *                          OR=(S=3 & TEST CONDITION SPECIFIED VDM 2593
2594          *                          BY G IS TRUE)]                    VDM 2594
2595          *                                                            VDM 2595
2596          MOOR  ENTR                                                   VDM 2596
2597                LDA     XIMC,1        IMC FIELD                        VDM 2597
2598                LSRA    2             EQUALS 00XX?                           2598
2599                JAZ     MOO1          YES=CONTINUE                     VDM 2599
2600          MOO2  TZA                                                    VDM 2600
2601                RETU*   MOOR                                           VDM 2601
2602          MOO1  LDA     XS,1                                           VDM 2602
2603                ADD     MOORT         ADDRESS OF JUMP TABLE                  2603
2604                LDB     XG,1          GET CONTENTS OF G FIELD                2604
2605                LDBE    TMUX,2        GET CONTENTS OF TEST CONDITIONS TABLE  2605
2606                STA     *+2           PREPARE FOR JUMP                       2606
2607                JMP*    *-*           GO TO PROCESSING DEPENDING ON S FIELD  2607
2608          MOORT DATA    (*+1)*                                               2608
2609          *     S FIELD =  0    1    2    3                                  2609
2610                DATA    MOO2,MOO3,MOO4,MOO5                                  2610
2611          MOO4  LDA     XT,1          T=0?                                   2611
2612                JAZ     MOO3          YES=OVERRIDE                           2612
2613                JBZ     MOO3          OVERRIDE IF T>0 & TEST FALSE           2613
2614                TZB                   FORCE NO-OVERRIDE JUMP                 2614
2615          MOO5  JBZ     MOO2          NO OVERRIDE IF TEST FALSE              2615
2616          *                                                          VDM 2616
2617          MOO3  DECR    01            A=-1                         VDM 2617
2618                RETU*   MOOR          RETURN                       VDM 2618
2619          *                                                          VDM 2619
2620          *                                                          VDM 2620
2621          *   LIST MEMORY OPERATIONS(IF TRACE FLAG IS ON)            VDM 2621
```

```
2622      *                                                                    VDM 2622
2623      *                                                                    VDM 2623
2624      M071  STA   MLIP          SET/RESET COMPLETE CYCLE FLAG                 2624
2625      M070  LDA   PTRC,1        TRACE FLAG FOR CURRENTLY SELECTED PAGE        2625
2626            JAZ   M072                                                     VDM 2626
2627            JMPM  TRSETA        CHECK IF WITHIN BOUNDS                     VDM 2627
2628            JAN   M072          NO=DO NOT TRACE                            VDM 2628
2629            CALL  SPAC          SPACE LO                                   VDM 2629
2630            CALL  LOOUT,MCCHL,MCCH                                            2630
2631            LDX   =DROM         ("CALL SPAC" CLOBBERED X)                     2631
2632            LDA   MCCO,1                                                      2632
2633            ADD   ='0'          CONVERT TO ASCII                              2633
2634            LDB   =(BUFR+1)     OUTPUT BUFFER AREA                            2634
2635            STA   1,2                                                         2635
2636            LDA   MOPC,1                                                      2636
2637            ADD   ='0'                                                        2637
2638            STA   4,2                                                         2638
2639            LDA   MADS,1                                                      2639
2640            ADD   ='0'-1                                                      2640
2641            STA   7,2                                                         2641
2642            LDA   MBYC,1                                                      2642
2643            ADD   ='0'                                                        2643
2644            STA   10,2                                                        2644
2645            CALL  MOPB          GET MEMORY ADDRESS                            2645
2646            STB   MLIS+8        STORE MEMORY ADDRESS IN CALLING SEQUENCE      2646
2647      MLIS  CALL  OHM,MIL+DROM,BUFR+13,IRG1+DROM,BUFR+16,MLIS+8,BUFR+19       2647
2648            DATA  *-*,BUFR+22,0                                               2648
2649            CALL  LOOUT,24,BUFR                                               2649
2650            LDA   MLIP          COMPLETE MEM. CYCLE FLAG                  VDM 2650
2651            JAZ   M074                                                        2651
2652            CALL  SPAC                                                        2652
2653            CALL  LOOUT,MFCYL,MFCY                                            2653
2654      *                                                                       2654
2655      *     CHECK IF MEMORY CYCLE NEEDS TO BE COMPLETED                       2655
2656      *                                                                       2656
2657      M072  LDA   MLIP          MEMORY CYCLE COMPLETION FLAG                  2657
2658            JAZ   M074          CYCLE IS COMPLETE                             2658
2659            TZA                                                               2659
2660            STA   MLIP          CLEAR THE FLAG                                2660
2661            JMP   MOPA+1        GO AROUND AGAIN                               2661
2662      *                                                                       2662
2663      *     SET (MIL15 & NOT SREG02) INTO 11+TMUX                             2663
2664      *     LOOK AT CIRCUIT DIAGRAMS IF YOU DON'T BELIEVE IT.                 2664
2665      *                                                                       2665
2666      M074  LDX   =DROM         ADDRESS OF DATA BUFFER                        2666
2667            LDA   SREG,1                                                      2667
2668            CPA                                                               2668
2669            LRLA  13                                                          2669
2670            ANA   MIL,1         COMPUTE MIL15 & NOT SREG02                    2670
2671            TZB                                                               2671
2672            LLRL  1                                                           2672
2673            STB   11+TMUXX,1                                                  2673
2674            RETU. MOPA                                                        2674
2675      *                                                                   VDM 2675
2676      MCCH  DATA  '  MCCO  MOPC  MADS  MBYC  MIR   IBR   MMAD  CONT'        2676
2677      MCCHL EQU   *-MCCH                                                    2677
2678      MFCY  DATA  '  FINISH CYCLE: '                                        2678
```

```
2679      MFCYL  EQU     *-MFCY                                          2679
2680      *                                                       VDM 2680
2681      MLIP   DATA    0             COMPLETE MEMORY CYCLE     VDM 2681
2682      MPLE   DATA    0             PIPELINE FLAG                   2682
2683             EJEC                                            VDM 2683
2684      *                                                       VDM 2684
2685      *      ALTER/DISPLAY MICRO REGISTERS                    VDM 2685
2686      *                                                       VDM 2686
2687      E100   JMPM    PBUF          SETUP PRINTER BUFFER            2687
2688             JMPM    FETCHA        INPUT REGISTER NUMBER     VDM 2688
2689      E100A  STA     E105          SAVE INPUT                      2689
2690             SUB     =('T'+1)                                      2690
2691             JAP     EXC90         ERROR IF > T                    2691
2692             SUB     =('A'-'T'-1)                                  2692
2693             JAN     EXC90         ERROR IF < A                    2693
2694             TAB                                                   2694
2695             LDAE    E100T,2                                       2695
2696             JAZ     EXC90         ERROR IF NOT DEFINED            2696
2697             STA     JRFL          TURN OFF JUMP OR FILE FLAG      2697
2698             JAP     EAFR          A IS ADDRESS OF REGISTER TO BE DISPLAYED 2698
2699      *                                                              2699
2700      *      ALTER/DISPLAY JUMP STACK OR REGISTER FILE               2700
2701      *                                                              2701
2702             JMPM    FETCHA        GET REG NUMBER 0-F        VDM 2702
2703             JMPM    CONV          CONVERT CHAR TO HEX       VDM 2703
2704             JAN     INH8          INVALID HEX CHARACTER INPUT     2704
2705             STA     E104          SAVE REGISTER VALUE             2705
2706      E102   LDB     JRFL          GET JUMP/REGISTER FLAG/INDEX    2706
2707             LDA     E104          GET REGISTER VALUE              2707
2708             ADDE    E106+3,2      ADD APPROPRIATE TABLE BASE      2708
2709      EAFR   TAX                   REG LOCATION                    2709
2710             LDB     0,1                                           2710
2711      *    OUTPUT CONTENTS OF B REG TO TTY PRINTER AS 4 HEX CHARS   VDM 2711
2712      *                                                       VDM 2712
2713             STX     E108          SAVE REG. LOC.            VDM 2713
2714             LDX     =(BUFR+1)     ASCII OUTPUT BUFFER             2714
2715             CALL    OH            CONVERT TO ASCII          VDM 2715
2716             CALL    SIOUT,3,BUFR  OUTPUT CONTENTS                 2716
2717             CALL    INA,'R'       INPUT 1-4 HEX DIGITS            2717
2718             STX     E109          SAVE LAST CHAR INPUT      VDM 2718
2719             JBZ     EAC1          JUMP IF NO CHANGE VALUE INPUT VDM 2719
2720             LDX     E108          REG.LOC.                  VDM 2720
2721             STA     0,1                                     VDM 2721
2722             LDA     E105          REGISTER                  VDM 2722
2723             SUB     ='T'                                          2723
2724             MJANZ   EAC1                                          2724
2725      *                                                             2725
2726      *      USE ONLY THE LOW-ORDER BIT OF TMUX.  NOTE - CHANGING TMUX MAY 2726
2727      *      RESULT IN TMUX AND STUS HAVING INCOMPATIBLE VALUES      2727
2728             LDA     0,1           GET STORED VALUE                2728
2729             ANA     =1                                            2729
2730             STA     0,1           STORE ONLY LSB                  2730
2731      *                                                       VDM 2731
2732      EAC1   LDA     E109          LAST CHAR INPUT           VDM 2732
2733             SUB     =',                                           2733
2734             MJANZ   EXC10         NO-RETURN TO EXEC               2734
2735             LDA     JRFL                                          2735
```

```
2736                JAP     EAC2            JUMP IF NOT JUMP STACK OR FILE REGISTER          2736
2737        *       OUTPUT NEXT FILE REGISTER OR JUMP STACK CELL                              2737
2738                LDA     E104            INCREMENT POINTER                                2738
2739                IAR                                                                      2739
2740                ANA     =017            KEEP 4 BITS                                      2740
2741                STA     E104                                                             2741
2742                LLSR    4               POSITION VALUE                                   2742
2743                JMPM    OH2             CONVERT CHAR TO ASCII                    VDM 2743
2744                LRLA    8                                                        VDM 2744
2745                ORA     =0207           BEL                                              2745
2746                STAE    BUFR+1                                                           2746
2747                CALL    SIOUT,2,BUFR                                                     2747
2748                JMP     E102            GO TYPE CONTENTS                         VDM 2748
2749        *                                                                        VDM 2749
2750        EAC2    CALL    LOOUT,ABELL,ABEL   OUTPUT 'A-BELL'                               2750
2751                CALL    FETCH           GET FIRST INPUT CHARACTER                        2751
2752                JMP     E100A                                                            2752
2753        *                                                                                2753
2754        ABEL    DATA    '  A '                                                           2754
2755        ABELL   EQU     *-ABEL                                                           2755
2756        *                                                                                2756
2757        E100T   DATA    DROM+XALU       A - ALU                                          2757
2758                DATA    DROM+IRG1       B - INSTRUCTION REGISTER #1                      2758
2759                DATA    DROM+SREG       C - SHIFT COUNTER                                2759
2760                DATA    0,0,0,0,0       D-H UNDEFINED                                    2760
2761                DATA    DROM+IRG2       I - INSTRUCTION REGISTER #2                      2761
2762                DATA    -1              J - JUMP STACK                                   2762
2763                DATA    DROM+KREG       K - DATA LOOP KEY REGISTER                       2763
2764                DATA    0               L - UNDEFINED                                    2764
2765                DATA    DROM+MIL        M - MEMORY INPUT LATCH                           2765
2766                DATA    0               N - UNDEFINED                                    2766
2767                DATA    DROM+OREG       O - OPERAND REGISTER                             2767
2768                DATA    DROM+PREG       P - PROGRAM COUNTER                              2768
2769                DATA    0               Q - UNDEFINED                                    2769
2770                DATA    -2              R - REGISTER FILE                                2770
2771                DATA    DROM+STUS       S - STATUS MUX                                   2771
2772                DATA    -3              T - TEST MUX                                     2772
2773        *                                                                                2773
2774        E104    DATA    0               REGISTER NUMBER FOR AJ, AR, AT COMMANDS          2774
2775        E105    DATA    0               TEMP STORE FOR INITIAL INPUT             VDM 2775
2776        E106    DATA    TMUX,R0,STACK   ADDRESSES                                        2776
2777        E108    DATA    0               TEMP STORAGE                             VDM 2777
2778        E109    DATA    0               LAST INPUT CHAR                          VDM 2778
2779        JRFL    DATA    0               JUMP/REGISTER INDEX IF NOT < 0.                  2779
2780                EJEC                                                             VDM 2780
2781        *                                                                        VDM 2781
2782        *       CHANGE/DISPLAY MEMORY (IN HEXADECIMAL)                           VDM 2782
2783        *                                                                        VDM 2783
2784        CMEM    JMPM    PBUF            INITIALIZE BUFFER                                2784
2785                CALL    INA,'N'                                                          2785
2786                STA     CME3            SAVE ADDR                                VDM 2786
2787        CME1    LDX     =(BUFR+1)                                                        2787
2788                LDB     CME3                                                     VDM 2788
2789                LDB     0,2                                                      VDM 2789
2790                JMPM    OH              CONVERT TO ASCII                         VDM 2790
2791                CALL    SIOUT,3,BUFR      OUTPUT CONTENTS                        VDM 2791
2792                CALL    INA,'R'         INPUT NEW VALUE                                  2792
```

```
2793            JBZ     CME2            JUMP IF NO CHANGE INPUT                 VDM 2793
2794            LDB     CME3                                                    VDM 2794
2795            STA     0,2             STORE CHANGE VALUE                      VDM 2795
2796    CME2    TXA                                                             VDM 2796
2797            SUB     ='.'                                                        2797
2798            MJANZ   EXC10           NO-RETURN TO EXEC                           2798
2799            JMPM    PBUF            INITIALIZE BUFFER                       VDM 2799
2800            INR     CME3            INCR MEMORY ADDR                        VDM 2800
2801            LDB     CME3            ADDR                                    VDM 2801
2802            LDX     =(BUFR+1)                                                   2802
2803            JMPM    OH              CONVERT TO ASCII                        VDM 2803
2804            CALL    SIOUT,3,BUFR      OUTPUT ADDR                           VDM 2804
2805            LDA     CME3                                                    VDM 2805
2806            JMP     CME1            GO OUTPUT CONTENTS                      VDM 2806
2807    CME3    DATA    0               MEMORY ADDRESS                          VDM 2807
2808            EJEC                                                            VDM 2808
2809    *                                                                       VDM 2809
2810    *       LOAD CONTROL STORE, CENTRAL OR DECODE                           VDM 2810
2811    *       ENTERED VIA THE L COMMAND                                       VDM 2811
2812    *       NOTE - X = DROM WHEN ENTERED                                        2812
2813    *                                                                       VDM 2813
2814    LDRM    JMPM    FETCHA          GET WHICH TYPE CONTROL STORE                2814
2815            DECR    2                                                           2815
2816            STB     LCCS            FLAG FOR MAIN MEMORY LOAD                   2816
2817            LDB     DBUF-DROM,1     ADDRESS OF CURRENT CCS PAGE                 2817
2818            SUB     ='C'                                                        2818
2819            JAZ     LDRC            LOADING CCS                                 2819
2820            DAR                     D ?                                     VDM 2820
2821            JAZ     LDDS            LOADING DECODE                          VDM 2821
2822            SUB     =('M'-'D')      LOADING MAIN MEMORY?                        2822
2823            JAZ     LDMM                                                        2823
2824            JMP     EXC90           ERROR-ILLEGAL DESIGNATOR                VDM 2824
2825    LDDS    JMPM    FETCHA          GET WHICH DECODE                        VDM 2825
2826            SUB     ='B'                                                        2826
2827            JAZ     LDDSB                                                       2827
2828            IAR                     A?                                          2828
2829            MJANZ   EXC90           INVALID DESIGNATOR                          2829
2830            LDB     DRM2-DROM,1     ADDRESS OF CURRENT DCSA PAGE                2830
2831            DATA    01006           SKIP NEXT INSTRUCTION                       2831
2832    LDDSB   LDB     DRM1-DROM,1     ADDRESS OF CURRENT DCSB PAGE                2832
2833            INCR    1               SET FOR DCS LOAD                            2833
2834    *       START LOAD                                                          2834
2835    LDRC    STA     LCCS                                                        2835
2836            STB     LDPT            SAVE POINTER                                2836
2837            STB     LDADR           DEFAULT ORG TO 0                        VDM 2837
2838    LDMM    LDA     BIFLG                                                       2838
2839            JAP     LMOS            JUMP IF SYSTEM IS MOS OR BI NOT ON RMD.     2839
2840            IOLINK  BI,BUFR+1,120                                               2840
2841    LMOS    TZA                                                             VDM 2841
2842            STA     RCN             ZERO RECORD NUMBER                      VDM 2842
2843    *                                                                       VDM 2843
2844    *       THE READ FCB/DCB AND WORD COUNT ARE SET AT INITIALIZATION       VDM 2844
2845    *                                                                       VDM 2845
2846    LREAD   READ    BIDCB,BI,0,1    READ A CARD                                 2846
2847    LR1     STAT    LREAD,ERR,EOF,BEOD,LR1                                  VDM 2847
2848    LREAD1  TZA                                                             VDM 2848
2849            STA     WDCT            ZERO WORD COUNT                         VDM 2849
```

```
2850                LDX     BIBUF           ADDR OF BI INPUT BUFFER                 VDM 2850
2851                LDA     0,1             GET WORD ZERO                           VDM 2851
2852                TAB                                                             VDM 2852
2853                ANA     =010000         BIT 12                                      2853
2854                ERA     =010000                                                     2854
2855                STA     FREC            SET FIRST RECORD FLAG                       2855
2856                TBA                                                             VDM 2856
2857                ANA     =04000          BIT 11                                      2857
2858                ERA     =04000                                                      2858
2859                STA     LREC            SET LAST RECORD FLAG                        2859
2860                TBA                                                             VDM 2860
2861                ERA     RCN                                                         2861
2862                ANA     =0377           LSB 8 BITS                                  2862
2863                MJANZ   SEQER           SEQUENCE ERROR                              2863
2864                TBA                                                                 2864
2865                JAPM    CKSUM           CHECK CHECKSUM IF BIT 15 OFF                2865
2866                LDA     BIBUF           BINARY INPUT BUFFER ADDRESS                 2866
2867                LDB     FREC            FIRST RECORD FLAG                           2867
2868                JBZ     LDBP            MOVE PAST HEADER BLOCK                      2868
2869                ADD     =9                                                          2869
2870    LDBP        IAR                                                                 2870
2871                IAR                                                                 2871
2872                STA     BUFPTR          SET BUFFER POINTER                      VDM 2872
2873    LDWD        LDA     BUFPTR                                                  VDM 2873
2874                SUB     BIBUF                                                   VDM 2874
2875                SUB     =60                                                         2875
2876                JAP     LDNR            RECORD COMPLETE ?                       VDM 2876
2877                LDX     BUFPTR                                                  VDM 2877
2878                LDA     LCCS                                                        2878
2879                JAN     LDMEM           JUMP IF LOAD MAIN MEMORY                    2879
2880                LDA     0,1             GET WORD                                VDM 2880
2881                TAB                                                             VDM 2881
2882                LSRA    13              GET CODE                                VDM 2882
2883                JAZ     LDINC           CODE=ZERO MEANS IGNORE                      2883
2884                DAR                                                             VDM 2884
2885                JAZ     LDORG           SET ORG ADDRESS                         VDM 2885
2886                DAR                                                             VDM 2886
2887                JAZ     LDST                                                    VDM 2887
2888                JMP     LDCDE           LOADER CODE ERROR                       VDM 2888
2889    LDST        INR     BUFPTR                                                      2889
2890                TBA                                                             VDM 2890
2891                ANA     =017777         GET 13 LSBS (INST COUNT)                    2891
2892                IAR                     PLUS ONE AS COUNT IS ACTUAL MINUS ONE   VDM 2892
2893    LDOR1       ASLA    2               TIMES 4 (FOUR 16 BIT WORDS PER MICRO)   VDM 2893
2894                STA     NMIC+2          SET COUNT FOR MOVE                      VDM 2894
2895                LDA     BUFPTR                                                  VDM 2895
2896                STA     NMIC+3          SET FROM ADDRESS FOR MOVE               VDM 2896
2897                ADD     NMIC+2                                                      2897
2898                STA     BUFPTR          UPDATE BUFFER POINTER                       2898
2899                LDA     LDADR                                                   VDM 2899
2900                STA     NMIC+4          SET TO ADDRESS FOR MOVE                 VDM 2900
2901                ADD     NMIC+2                                                      2901
2902                STA     LDADR           UPDATE LOAD ADDRESS POINTER                 2902
2903    NMIC        CALL    MOVW,4,0,0                                              VDM 2903
2904                JMP     LDWD                                                    VDM 2904
2905    *                                                                           VDM 2905
2906    LDORG       TBA                                                             VDM 2906
```

```
2907                ANA     =017777         GET ORG ADDR                                    2907
2908                LDB     LCCS                                                    VDM 2908
2909                XBZ     LDOR1           IF LOADING CCS                          VDM 2909
2910                ADD     LDPT            SET LOAD ADDR                           VDM 2910
2911                STA     LDADR                                                   VDM 2911
2912        LDINC   INR     BUFPTR                                                      2912
2913                JMP     LDWD                                                    VDM 2913
2914        *  LOAD MAIN MEMORY                                                         2914
2915        LDMEM   LDA     0,1             GET WORD                                    2915
2916                LSRA    13              FETCH CODE                                  2916
2917                ADD     CODE            SERVICE ROUTINE ADDRESS                     2917
2918                STA     *+2                                                         2918
2919                JMP*    *-*             GO DO SERVICE                               2919
2920        CODE    DATA    (*+1)*                                                      2920
2921        *                SUB   ERR   ERR   ERR   DATA                               2921
2922                DATA    LDSCD,LDCDE,LDCDE,LDCDE,LDLDR                               2922
2923        LDSCD   LDA     0,1                                                         2923
2924                LSRA    9                                                           2924
2925                ANA     =017            FETCH SUBCODE                               2925
2926                SUB     =8                                                          2926
2927                JAP     LDCDE           ERROR                                       2927
2928                ADD     SCOD            SERVICE ROUTINE ADDRESS                     2928
2929                STA     *+2                                                         2929
2930                JMP*    *-*             GO DO SERVICE                               2930
2931        SCOD    DATA    (*+9)*          8 SUBTRACTED ABOVE                          2931
2932        *               IGNORE ORG   ERR   ERR   ERR   ERR   ERR  DONE              2932
2933                DATA    LDINC,LDMOR,LDCDE,LDCDE,LDCDE,LDCDE,LDCDE,LCOM              2933
2934        LDMOR   LDA     0,1             SET NEW ORIGIN                              2934
2935                LSRA    4               FETCH POINTER                               2935
2936                ANA     =037                                                        2936
2937                MJANZ   LDCDE           ERROR IF NOT ABSOLUTE                       2937
2938                INR     BUFPTR                                                      2938
2939                LDA     1,1             LOAD ADDRESS                                2939
2940                STA     LDADR                                                       2940
2941                SUB     =MMEM                                                       2941
2942                JAN     LDSIZ           IF BELOW MAIN MEMORY                        2942
2943                SUBI    EMEM-MMEM+1                                                 2943
2944                JAP     LDSIZ           IF ABOVE MAIN MEMORY                        2944
2945                JMP     LDINC                                                       2945
2946        LDLDR   INR     BUFPTR          LOAD DATA WORDS                             2946
2947                LDA     0,1                                                         2947
2948                ANA     =017777         ISOLATE WORD                                2948
2949                IAR                                                                 2949
2950                STA     LDMV+2          SET MOVE COUNT                              2950
2951                LDA     BUFPTR                                                      2951
2952                STA     LDMV+3          SET FROM ADDRESS                            2952
2953                ADD     LDMV+2                                                      2953
2954                STA     BUFPTR          UPDATE POINTER                              2954
2955                LDA     LDADR                                                       2955
2956                STA     LDMV+4          SET TO ADDRESS                              2956
2957                ADD     LDMV+2                                                      2957
2958                STA     LDADR           UPDATE LOAD ADDRESS                         2958
2959                SUB     =(EMEM+1)                                                   2959
2960                JAP     LDSIZ           JUMP IF MEMORY CAPACITY EXCEEDED            2960
2961        LDMV    CALL    MOVW,0,0,0      MOVE DATA WORDS TO MEMORY                   2961
2962                JMP     LDWD                                                        2962
2963        LDNR    INR     RCN             ADVANCE RECORD NUMBER                       2963
```

```
2964                LDA     BIFLG           BI ON RMD FLAG                                  2964
2965                JAP     LDRNS           BI NOT RMD OR SYSTEM IS MOS                     2965
2966                LDA     BUFPTR                                                  VDM     2966
2967                SUB     BIDCB+1         BUFFER ADDR                             VDM     2967
2968                SUB     =120            RECORD LENGTH ON RMD                            2968
2969                JAP     LDRNR           JUMP IF 120 WORDS READ                  VDM     2969
2970                LDA     BIBUF                                                   VDM     2970
2971                ADD     =60                                                             2971
2972                STA     BIBUF           SET FOR NEXT 60 WORD BLOCK              VDM     2972
2973                LDA     LREC            LAST RECORD FLAG                        VDM     2973
2974                JAZ     LREAD1          JUMP IF NOT LAST RECORD                 VDM     2974
2975                JMP     LCOM            LOAD COMPLETE                           VDM     2975
2976    LDRNR       LDA     BIDCB+1         BUFFER ADDR                             VDM     2976
2977                STA     BIBUF           RESET BUFFER ADDRESS                    VDM     2977
2978                STA     BUFPTR                                                  VDM     2978
2979    LDRNS       LDA     LREC            LAST RECORD FLAG                                2979
2980                JAZ     LREAD           JUMP IF NOT SET                         VDM     2980
2981    LCOM        CALL    SIOUT,LDCML,LDCM                                                2981
2982                JMP     EXC10           RETURN                                  VDM     2982
2983    *                                                                           VDM     2983
2984    ERR         CALL    ERRMSG,7        READ ERROR                                      2984
2985    EOF         CALL    ERRMSG,8        EOF ENCOUNTERED                                 2985
2986    BEOD        CALL    ERRMSG,9        END OF DEVICE                                   2986
2987    SEQER       CALL    ERRMSG,10       SEQUENCE ERROR                                  2987
2988    LDCDE       CALL    ERRMSG,11       LOADER CODE ERROR                               2988
2989    LDSIZ       CALL    ERRMSG,15                                                       2989
2990    BIFLG       DATA    0               BI ON RMD FLAG, POS=NO, NEG=YES         VDM     2990
2991    *                                                                           VDM     2991
2992    *       THIS ROUTINE PERFORMS A STANDARD CHECKSUM ON A RECORD               VDM     2992
2993    *                                                                           VDM     2993
2994    *       CALLING SEQUENCE                                                    VDM     2994
2995            CALL    CKSUM                                                       VDM     2995
2996    *       RETURN IF CHECKSUM VALID, GO TO ERROR IF NOT                                2996
2997    *                                                                           VDM     2997
2998    CKSUM   ENTR                                                                VDM     2998
2999            LDA     BIBUF           BINARY INPUT BUFFER ADDRESS                     2999
3000            ADD     BIBLK           RECORD LENGTH                                   3000
3001            STA     CKSU2                                                   VDM     3001
3002            TZB                                                             VDM     3002
3003    CKSU1   TBA                                                             VDM     3003
3004            ERA     0,1                                                     VDM     3004
3005            TAB                     SAVE ACCUM VALUE                        VDM     3005
3006            INCR    045             A=X=X+1                                 VDM     3006
3007            SUBI    0                                                       VDM     3007
3008    CKSU2   BES     0                                                       VDM     3008
3009            JAN     CKSU1           JUMP IF NOT AT END OF RECORD            VDM     3009
3010            TBA                                                             VDM     3010
3011            JAZ*    CKSUM           RETURN IF CHECKSUM OK                   VDM     3011
3012            CALL    ERRMSG,12       CHECKSUM ERROR                                  3012
3013    *       DATA BLOCKS AND TEMPORARY STORAGE                               VDM     3013
3014    *                                                                       VDM     3014
3015    BIDCB   DCB     60,BUFR+1,0     LOCAL DCB FOR BI                        VDM     3015
3016    BIBLK   DATA    60              RECORD LENGTH, SET DURING INITIALIZATION        3016
3017    BIBUF   DATA    BUFR+1          POINTER TO BUFFER                       VDM     3017
3018    BUFPTR  DATA    BUFR+1          BUFFER POINTER                          VDM     3018
3019    FREC    DATA    0               FIRST RECORD FLAG                       VDM     3019
3020    LCCS    DATA    0               0=LOAD CCS, POS=LOAD DCS, NEG=LOAD MEM          3020
```

```
3021    LDADR   DATA    0               LOAD ADDR                                VDM 3021
3022    LDPT    DATA    0               BASE ADDR                                VDM 3022
3023    LREC    DATA    0               LAST RECORD FLAG                         VDM 3023
3024    RCN     DATA    0               CURRENT EXPECTED RECORD NUMBER           VDM 3024
3025    WDCT    DATA    0               WORD COUNT                               VDM 3025
3026    LDCM    DATA    '  LOAD COMPLETE '                                       VDM 3026
3027    LDCML   EQU     *-LDCM                                                      3027
3028            EJEC                                                            3028
3029    *                                                                       3029
3030    *   DISPLAY/CHANGE DECODERS (A OR B) OR CCS.                            3030
3031    *                                                                       3031
3032    *   ENTERED FROM EXEC VIA "E" DIRECTIVE                                 3032
3033    *      NOTE - X = DROM WHEN ENTERED                                     3033
3034    *                                                                       3034
3035    EDRM    JMPM    FETCHA          GET WHICH CONTROL STORE                 3035
3036            SUB     ='C'                                                    3036
3037            JAZ     E110            CENTRAL-GO TO CENTRAL ROUTINE           3037
3038    *                                                                       3038
3039    *       CHANGE DECODERS.                                                3039
3040    *                                                                       3040
3041            DAR                     D ?                                     3041
3042            MJANZ   EXC90           INPUT ERROR                             3042
3043            JMPM    FETCHA          INPUT A OR B DCS                        3043
3044            LRLA    8                                                       3044
3045            STA     ABDCS                                                   3045
3046            LRLA    8                                                       3046
3047            SUB     ='A'                                                    3047
3048            LDB     =DRM2           ADDRESS OF ADDRESS OF DCSA              3048
3049            JAZ     ED20            YES                                     3049
3050            DAR                     B ?                                     3050
3051            DBR                     ADDRESS OF ADDRESS OF DCSB              3051
3052            MJANZ   EXC90                                                   3052
3053    ED20    LDA     0,2             ADDRESS OF DCS                          3053
3054            STA     E119            TEMP STORE                              3054
3055    *                                                                       3055
3056    *    INPUT HEX 0-F FOR DECODE ROM ADDR                                  3056
3057            JMPM    FETCHA          INPUT CONTROL STORE ADDRESS             3057
3058            CALL    CONV            CONVERT TO HEX                          3058
3059            JAN     EXC90           INPUT ERROR                             3059
3060            STA     ED41            TEMP STORE                              3060
3061    ED31    ADD     E119            ADD ADDRESS OF DCS                      3061
3062            TAX                                                             3062
3063            LDB     0,1             GET CONTENTS OF DCS                     3063
3064    *                                                                       3064
3065    *    OUTPUT CONTENTS OF B REG TO TTY PRINTER AS 4 HEX CHAR'S            3065
3066            STX     ED42            SAVE REG ADDR                           3066
3067            LDX     =(BUFR+1)       OUTPUT BUFFER ADDR                      3067
3068            CALL    OH              CONVERT TO ASCII                        3068
3069            CALL    SIOUT,3,BUFR      OUTPUT                                3069
3070    *                                                                       3070
3071    *                                                                       3071
3072            CALL    INA,'R'         INPUT 1-4 HEX DIGITS                    3072
3073            JXZ     EXC10                                                   3073
3074            STX     E118            SAVE LAST CHAR INPUT                    3074
3075            JBZ     ED51                                                    3075
3076            LDX     ED42            ADDR OF REG                             3076
3077            STA     0,1             STORE NEW VALUE                         3077
```

```
3078      ED51   LDA     E118                                                    3078
3079             SUB     ='0'                                                    3079
3080             MJANZ   EXC10           NO-RETURN TO EXEC                       3080
3081      *    SEQUENCE TO NEXT ROM LOCATION                                     3081
3082             INR     ED41            ROM LOC 0-16                            3082
3083             LDA     ED41                                                    3083
3084             SUB     =16                                                     3084
3085             JAP     EXC10           RETURN TO EXEC IF RANGE NOT 0-16        3085
3086             LDB     ED41            NEXT DCS ADDRESS                        3086
3087             LRLB    12                                                      3087
3088             CALL    OH2             CONVERT TO ASCII                        3088
3089             ORA     ABDCS           A OR B DCS                              3089
3090             STAE    BUFR+1                                                  3090
3091             CALL    SIOUT,2,BUFR      OUTPUT NEXT LOCATION                  3091
3092             LDA     ED41            NEXT DCS ADDR                           3092
3093             JMP     ED31                                                    3093
3094      *                                                                       3094
3095      ABDCS  DATA    0               A OR B DCS FLAG                         3095
3096      ED41   DATA    0               SAVE HEX INPUT(0-F)                     3096
3097      ED42   DATA    0               TEMP STORE                              3097
3098      *                                                                VDM 3098
3099      *      CHANGE/DISPLAY ROM DATA WORD                              VDM 3099
3100      *                                                                VDM 3100
3101      E110   CALL    INA,'N'         GET DUMP ADDRESS                        3101
3102             STA     E119            SAVE ADDR                         VDM 3102
3103             SUB     =01000          ROM BUFFER LIMIT                        3103
3104             JAP     EXC90           INVALID ADDRESS                   VDM 3104
3105      *                                                                VDM 3105
3106      E112   LDA     E119            ROM ADDR                          VDM 3106
3107             LRLA    2               TIMES FOUR                        VDM 3107
3108             ADDE    DBUF            PLUS CCS BUFFER ADDRESS                 3108
3109             STA     E113            SAVE ADDR                         VDM 3109
3110      *                                                                VDM 3110
3111             JMPM    PBUF            SETUP PRINTER BUFFER              VDM 3111
3112             LDX     =(BUFR+1)       ASCII BUFFER ADDR                       3112
3113             LDB     E113            ROM ADDR                          VDM 3113
3114             LDB     0,2             FIRST 16-BIT DATA BLOCK           VDM 3114
3115             JMPM    OH              CONVERT HEX TO ASCII              VDM 3115
3116      *                                                                VDM 3116
3117             LDB     E113            ROM ADDR                          VDM 3117
3118             LDB     1,2             SECOND 16-BIT DATA BLOCK          VDM 3118
3119             JMPM    OH              CONVERT DATA TO ASCII             VDM 3119
3120      *                                                                VDM 3120
3121             LDB     E113            ROM ADDR                          VDM 3121
3122             LDB     2,2             THIRD 16-BIT DATA BLOCK           VDM 3122
3123             JMPM    OH              CONVERT DATA TO ASCII             VDM 3123
3124      *                                                                VDM 3124
3125             LDB     E113            ROM ADDR                          VDM 3125
3126             LDB     3,2             FOURTH 16-BIT DATA BLOCK          VDM 3126
3127             JMPM    OH              CONVERT DATA TO ASCII             VDM 3127
3128      *                                                                VDM 3128
3129             CALL    SIOUT,11,BUFR     OUTPUT TO SO                    VDM 3129
3130      *                                                                VDM 3130
3131             JMPM    INH             INPUT CHANGE VALUE - IF ANY       VDM 3131
3132             JXZ     EXC10                                             VDM 3132
3133             STX     E118            LAST INPUT CHAR                   VDM 3133
3134             JBZ     E114            JUMP IF NO CHANGE VALUE INPUT     VDM 3134
```

```
3135                JMPM    MOVW        MOVE DATA TO ROM BUFFER          VDM 3135
3136                DATA    4           WORD COUNT                       VDM 3136
3137                DATA    V           FROM ADDR                        VDM 3137
3138        E113    DATA    0           TO ADDR                          VDM 3138
3139        *                                                            VDM 3139
3140        E114    LDA     E118        LAST INPUT CHAR                  VDM 3140
3141                SUB     =',                                             3141
3142                MJANZ   EXC10       NO-RETURN TO EXEC                   3142
3143        E114A   INR     E119        STEP ROM RELATIVE ADDRESS           3143
3144                LDA     E119                                         VDM 3144
3145                SUB     =01000      ROM BUFFER LIMIT                    3145
3146                JAZ     EXC10       RETURN - AT END OF BUFFER        VDM 3146
3147                JMPM    PBUF        INITIALIZE LO BUFFER                3147
3148                LDX     =(BUFR+1)   ASCII BUFFER ADDR                   3148
3149                LDB     E119        ROM ADDR                         VDM 3149
3150                JMPM    OH          CONVERT TO ASCII                 VDM 3150
3151                CALL    SIOUT,4,BUFR  OUTPUT TO SO                   VDM 3151
3152                JMP     E112        OUTPUT CONTENTS OF WORD          VDM 3152
3153        *                                                            VDM 3153
3154        E118    DATA    0           LAST INPUT CHAR                  VDM 3154
3155        E119    DATA    0           TEMP STORED ROM ADDR             VDM 3155
3156                EJEC                                                 VDM 3156
3157        *                                                            VDM 3157
3158        *       ROM ADDRESS HALT                                     VDM 3158
3159        *                                                            VDM 3159
3160        E120    LDBE    CPAG+DROM   CURRENTLY SELECTED PAGE             3160
3161                LDAE    RHLT,2      PAGE HALT TABLE                     3161
3162                STA     E123        STUFF INTO SUBTRACTION           VDM 3162
3163                STA     E121        RESET HALT BUFFER POINTER        VDM 3163
3164        E124    CALL    INA,'N'     GET HALT ADDRESS                    3164
3165                LDB     E121                                         VDM 3165
3166                STA     0,2         SET HALT ADDR                    VDM 3166
3167                TXA                                                  VDM 3167
3168                SUB     =',                                             3168
3169                MJANZ   EXC10       NO-RETURN TO EXEC                   3169
3170                INR     E121        ADVANCE POINTER                     3170
3171                LDA     E121                                            3171
3172                SUBI    RHLT0                                        VDM 3172
3173        E123    BES     0           OVERLAYED BY ABOVE STORE         VDM 3173
3174                SUB     =5          FIVE ALREADY INPUT?                 3174
3175                JAN     E124        NO-LOOK FOR MORE                 VDM 3175
3176                JMP     EXC10       YES-RETURN                       VDM 3176
3177        E121    DATA    RHLT                                         VDM 3177
3178                EJEC                                                 VDM 3178
3179        *                                                            VDM 3179
3180        *       TRACE                                                VDM 3180
3181        *                                                            VDM 3181
3182        E140    JMPM    FETCHA      GET SET/RESET INPUT                 3182
3183                LDB     CPAG,1      CURRENTLY SELECTED PAGE             3183
3184                SUB     ='R'                                            3184
3185                JAZ     E143        JUMP IF TRACE RESET REQUEST         3185
3186                DAR                                                     3186
3187                MJANZ   EXC90       JUMP IF NOT TRACE SET REQUEST       3187
3188                CALL    E155        GET ADDRESS LIMITS                  3188
3189                LDX     =DROM       GET DATA BUFFER ADDRESS             3189
3190                LDB     CPAG,1      GET CURRENTLY SELECTED PAGE         3190
3191                LDA     START                                           3191
```

```
3192                STAE    STRTRC,2        SET START TRACE ADDRESS                          3192
3193                LDA     END                                                              3193
3194                STAE    ENDTRC,2        SET END TRACE ADDRESS                            3194
3195                INCR    1                                                                3195
3196        E143    STAE    TRACE,2         SET OR RESET TRACE FLAG FOR PAGE                 3196
3197                STA     NTRC,1          TRACE FLAG FOR PAGE OF NEXT MICRO                3197
3198                JMP     EXC10           RETURN FOR NEXT INPUT                            3198
3199        *                                                                          VDM   3199
3200        STRTRC  DATA    0,0,0,0         TRACE START                              VDM   3200
3201        ENDTRC  DATA    0,0,0,0         TRACE END                                VDM   3201
3202                EJEC                                                             VDM   3202
3203        *                                                                        VDM   3203
3204        *       THIS ROUTINE WILL FETCH BEGIN AND END ADDRESSES                  VDM   3204
3205        *       DEFAULT IF EITHER IS MISSING IS AS FOLLOWS:                            3205
3206        *               START TO 0                                               VDM   3206
3207        *               END TO 512 (1FF)                                         VDM   3207
3208        *       CALLING SEQUENCE                                                 VDM   3208
3209        *       CALL    E155                                                     VDM   3209
3210        *       RETURN  START ADDR IN START, END ADDR IN END                     VDM   3210
3211        *                                                                        VDM   3211
3212        E155    ENTR                                                             VDM   3212
3213                CALL    INA,'N'         GET START ADDRESS                              3213
3214                STA     START           SAVE ADDRESS                             VDM   3214
3215                JAN     EXC90           NEGATIVE ADDRESS                               3215
3216                SUB     =01000                                                         3216
3217                JAP     EXC90           OUTSIDE OF PAGE                                3217
3218                TXA                     LAST CHAR                                VDM   3218
3219                SUB     ='','                                                          3219
3220                MJBNZ   E155A           BOUNDS GIVEN                                   3220
3221                STB     START           DEFAULT START ADDRESS                          3221
3222        E155A   MJANZ   E145            LAST CHARACTER NOT A COMMA                     3222
3223                CALL    INA,'N'         GET END TRACE ADDRESS                          3223
3224                JBZ     E145                                                     VDM   3224
3225                DATA    01006           SKIP NEXT INSTRUCTION                          3225
3226        E145    LDA     =0777                                                          3226
3227                STA     END             DEFAULT END TO 0777 (1FF)                VDM   3227
3228                JAN     EXC90           NEGATIVE ADDRESS                               3228
3229                SUB     =01000                                                         3229
3230                JAP     EXC90           OUTSIDE OF PAGE                                3230
3231                JMP*    E155                                                     VDM   3231
3232                EJEC                                                                   3232
3233        *                                                                              3233
3234        *       DUMP CCS CONTENTS IN FIELD EDITED FORM.                                3234
3235        *       ENTERED FROM EXEC VIA "F" DIRECTIVE.                                   3235
3236        *                                                                              3236
3237        FDMP    CALL    E155            GET ADDRESS BOUNDS                             3237
3238        FD1     LDX     =DROM           ADDRESS OF DATA BUFFER                         3238
3239                TZA                                                                    3239
3240                STA     XAB,1           MAKE SURE FELD USES CCS AS FIELD SOURCES       3240
3241                STA     XM,1                                                           3241
3242                STA     XTC,1                                                          3242
3243                LDA     START                                                          3243
3244                CALL    FELD            BREAK CCS WORD INTO FIELDS                     3244
3245                CALL    LSTF            GO LIST FIELDS                                 3245
3246                INR     START                                                          3246
3247                LDA     END                                                            3247
3248                SUB     START           DONE YET?                                      3248
```

```
3249                JAN     EXC10           YES- RETURN TO EXEC                          3249
3250                JMP     FD1             GO DUMP ANOTHER CCS WORD                     3250
3251                EJEC                                                          VDM 3251
3252        *                                                                     VDM 3252
3253        *       DUMP CCS, DECODERS OR MEMORY                                         3253
3254        *       ENTERED FROM EXEC VIA "DC", "DDA", "DDB" OR "DM" DIRECTIVES          3254
3255        *                                                                     VDM 3255
3256        E150    CALL    FETCHA          GET SECOND CHARACTER OF INPUT                3256
3257                SUB     =C'C'           DUMP CCS?                                    3257
3258                JAZ     E151            YES                                          3258
3259                STA     E15C            FLAG FOR DUMPING 8 WORDS PER LINE.           3259
3260                DAR                     DUMP DECODER?                                3260
3261                JAZ     DDCS            YES-                                         3261
3262                SUB     =('M'-'D')      DUMP MEMORY?                                 3262
3263                MJANZ   EXC90           NO- INVALID INPUT                            3263
3264                CALL    INA,'N'         GET START ADDRESS                            3264
3265                STA     START                                                        3265
3266                TXA                                                                  3266
3267                SUB     =',             IS FINAL CHARACTER OF FIELD A COMMA?          3267
3268                MJBNZ   E150A           JUMP IF NO DIGITS INPUT                      3268
3269                LDX     =MMEM           DEFAULT STARTING VALUE TO MMEM               3269
3270                STX     START                                                        3270
3271        E150A   MJANZ   E150C           JUMP IF FINAL CHARACTER IS NOT A COMMA        3271
3272                CALL    INA,'N'         GET END ADDRESS                              3272
3273                JBZ     E150C           JUMP IF NO DIGITS                            3273
3274                SKIP    NEXT                                                         3274
3275        E150C   LDA     =EMEM           DEFAULT END ADDRESS TO EMEM                  3275
3276                STA     END                                                          3276
3277                SUB     =EMEM+1                                                      3277
3278                JAP     EXC90           ABOVE USER AREA                              3278
3279                LDA     START                                                        3279
3280                STA     E157            ADDRESS FOR DUMP ROUTINE                     3280
3281                SUB     =MMEM                                                        3281
3282                JAN     EXC90           BELOW USER AREA                              3282
3283                LDA     END                                                          3283
3284                SUB     START                                                        3284
3285                JAN     EXC90           END ADDRESS BEFORE START ADDRESS             3285
3286                JMP     E152                                                         3286
3287        DDCS    CALL    FETCHA          GET THIRD CHARACTER OF INPUT                 3287
3288                LDB     =DRM2           ADDRESS OF ADDRESS OF DECODER A              3288
3289                SUB     =C'A'           DECODER A?                                   3289
3290                JAZ     DDCS1                                                        3290
3291                DBR                     ADDRESS OF ADDRESS OF DECODER B              3291
3292                DAR                     DECODER B?                                   3292
3293                MJANZ   EXC90           NO- INVALID INPUT                            3293
3294        DDCS1   STA     START                                                        3294
3295                LDA     =017                                                         3295
3296                STA     END             DUMP ENTIRE DECODER                          3296
3297                LDA     0,2             GET DECODER ADDRESS                          3297
3298                STA     E157            SET START DUMP ADDRESS                       3298
3299                STA     E15C            FLAG FOR DUMPING DCS                         3299
3300                JMP     E152                                                         3300
3301        *       GET ADDRESS BOUNDARIES                                               3301
3302        E151    STA     E15C            FLAG FOR DUMPING CCS OR MEMORY               3302
3303                CALL    E155                                                         3303
3304        E152    CALL    SPAC            SPACE LO                                     3304
3305        *                                                                     VDM 3305
```

```
3306      *       DUMP DATA TO LO                                          VDM 3306
3307      *                                                                VDM 3307
3308      E153    JMPM    PBUF            SETUP PRINTER BUFFER             VDM 3308
3309              LDX     =(BUFR+1)       OUTPUT BUFFER ADDR                   3309
3310      *                                                                VDM 3310
3311              LDB     START                                                3311
3312              JMPM    OH              CONVERT TO ASCII IN BUFR         VDM 3312
3313              LDA     ='* '                                                3313
3314              STA     0,1             STORE '* ' AFTER ADDRESS.            3314
3315              IXR                     SPACES AFTER ADDRESS IN BUFFER   VDM 3315
3316              LDA     E15C            WHAT TO DUMP?                        3316
3317              MJANZ   D16B            MEMORY OR DCS                        3317
3318              LDA     START                                                3318
3319              LRLA    2               TIMES FOUR                       VDM 3319
3320              ADDE    DBUF            PLUS CCS BUFFER ADDRESS              3320
3321              STA     E157            SAVE ROM BUFFER ADDR             VDM 3321
3322              TAB                                                      VDM 3322
3323              LDB     0,2             FIRST 16 BITS OF ROM WORD        VDM 3323
3324              JMPM    OH              CONVERT TO ASCII                 VDM 3324
3325      *                                                                VDM 3325
3326              LDB     E157            ROM ADDR                         VDM 3326
3327              LDB     1,2             SECOND 16 BITS OF ROM WORD       VDM 3327
3328              JMPM    OH              CONVERT TO ASCII                 VDM 3328
3329              IXR                     SPACE AFTER FIRST 8 DIGITS       VDM 3329
3330      *                                                                VDM 3330
3331              LDB     E157            ROM ADDR                         VDM 3331
3332              LDB     2,2             THIRD 16 BITS OF ROM WORD        VDM 3332
3333              JMPM    OH              CONVERT TO ASCII                 VDM 3333
3334      *                                                                VDM 3334
3335              LDB     E157            ROM ADDR                         VDM 3335
3336              LDB     3,2             FOURTH 16 BITS OF ROM WORD       VDM 3336
3337              JMPM    OH              CONVERT TO ASCII                 VDM 3337
3338              IXR                     SPACE AFTER HEX WORD IN BUFFER   VDM 3338
3339      *                                                                VDM 3339
3340              LDB     E157            ROM ADDR                         VDM 3340
3341              LDB     0,2             FIRST 16 BITS OF ROM WORD        VDM 3341
3342              JMPM    E160            CONVERT TO ASCII                 VDM 3342
3343              IXR                     SPACE                            VDM 3343
3344      *                                                                VDM 3344
3345              LDB     E157            ROM ADDR                         VDM 3345
3346              LDB     1,2             SECOND 16 BITS OF ROM WORD       VDM 3346
3347              CALL    E160            CONVERT TO ASCII                     3347
3348      *                                                                VDM 3348
3349              CALL    LOOUT,31,BUFR   OUTPUT LINE                          3349
3350              JMPM    PBUF            SETUP LO BUFFER                      3350
3351              LDXI    BUFR+14         BUFFER ADDR FOR SECOND LINE          3351
3352              LDB     E157            ROM ADDR                         VDM 3352
3353              LDB     2,2             THIRD 16 BITS OF ROM WORD        VDM 3353
3354              JMPM    E160            CONVERT TO ASCII                 VDM 3354
3355              IXR                     SPACE                            VDM 3355
3356      *                                                                VDM 3356
3357              LDB     E157            ROM ADDR                         VDM 3357
3358              LDB     3,2             FOURTH 16 BITS OF ROM WORD       VDM 3358
3359              JMPM    E160            CONVERT TO ASCII                 VDM 3359
3360              CALL    LOOUT,31,BUFR     OUTPUT LINE                    VDM 3360
3361              JMPM    SPAC                                                 3361
3362              INR     START           STEP START DUMP ADDR             VDM 3362
```

```
3363        E154    LDA     END
3364                SUB     START           DONE YET?
3365                JAN     EXC10           JUMP IF DONE
3366                JMP     E153            OUTPUT NEXT LINE OF PAGE
3367        *
3368        *       DUMP 16 BIT WORDS, 8 PER LINE IN HEXADECIMAL
3369        D16B    LDBE*   E157
3370                CALL    OH              CONVERT TO ASCII
3371                IXR                     SPACE BETWEEN WORDS
3372                INR     E157
3373                INR     START
3374                LDA     END
3375                SUB     START           DONE WITH DUMP?
3376                JAN     D16BW           YES
3377                LDA     START
3378                ANA     =7              LINE FULL?
3379                MJANZ   D16B            NO
3380        D16BW   CALL    LOOUT,27,BUFR
3381                JMP     E154
3382        *
3383        START   DATA    0               START TRACE ADDRESS
3384        END     DATA    0               END TRACE ADDRESS
3385        E157    DATA    0               ROM WORD LOCATION
3386        E15C    DATA    0               WHAT KIND OF DUMP?
3387        E15D    DATA    0               MAXIMUM LINE COUNT PER PAGE
3388                EJEC
3389        *
3390        *       CONVERT 16 BITS IN B TO ASCII CODED BINARY IN BUFR
3391        *
3392        *       X CONTAINS ADDR OF STORAGE LOCATION IN BUFR
3393        *
3394        E160    ENTR
3395                TZA
3396                LLRL    1               SHIFT FIRST BIT INTO A
3397                IBR                     SET TRAILING BIT IN B
3398                JMP     *+6             START CONVERSION
3399        *
3400        E162    TZA
3401                LLRL    1               SHIFT NEXT BIT INTO A
3402                JBZ*    E160            RETURN IF COMPLETE
3403                LRLA    7               LEFT JUSTIFY CHAR - ALMOST
3404                LLRL    1               SHIFT NEXT BIT INTO A
3405                ADD     ='00'           CONVERT TO ASCII
3406                STA     0,1             STORE CHARS IN BUFFER
3407                IXR                     STEP BUFFER INDEX
3408                JMP     E162            CONVERT NEXT TWO BITS
3409                EJEC
3410        *
3411        *       SET MEMORY TYPE.
3412        *       ENTERED VIA "Q" KEYIN.
3413        *       QN      N=3: SC, N=4: CORE, N=5: SLOW CORE.
3414        *
3415        SETM    CALL    FETCHA          GET NEXT CHARACTER OF INPUT
3416                SUB     ='0'
3417                TAB
3418                SUB     =3
3419                JAN     EXC90           ERROR IF LESS THAN 3
```

```
3420            SUB     =3                                                  3420
3421            JAP     EXC9D           ERROR IF GREATER THAN 5             3421
3422            STB     MTYP,1          SET MEMORY TYPE                     3422
3423            JMP     EXC1D           RETURN TO EXEC.                     3423
3424            EJEC                                                    VDM 3424
3425    *                                                               VDM 3425
3426    *       RETURN TO OPERATING SYSTEM                              VDM 3426
3427    *                                                               VDM 3427
3428    E17D    CALL    FETCHC          GET NEXT CHARACTER OF INPUT.        3428
3429            JAZ     E171            EXIT IF NO MORE.                    3429
3430            SUB     =' '                                                3430
3431            MJANZ   EXC9D           NO, MUST BE AN ERROR            VDM 3431
3432    E171    LDAE    LOFLG           LO ON RMD FLAG                  VDM 3432
3433            JAP     E172            JUMP IF NO                      VDM 3433
3434            CLOSE   LOFCB,LO,0,1    CLOSE AND UPDATE                    3434
3435    E172    CALL    EXIT            RETURN TO OPERATING SYSTEM      VDM 3435
3436            JMP     EXC             JUST IN CASE                    VDM 3436
3437            EJEC                                                    VDM 3437
3438    *       THIS ROUTINE HANDLES THE SELECT INPUT MEDIA COMMAND (M)  VDM 3438
3439    *                                                               VDM 3439
3440    *       'MR' SELECTS SI TO BE THE INPUT MEDIA                   VDM 3440
3441    *       'MS' SELECTS PI TO BE THE INPUT MEDIA                   VDM 3441
3442    *                                                               VDM 3442
3443    *       CALLING SEQUENCE                                        VDM 3443
3444    *                                                               VDM 3444
3445    *       JMP     CHME                                            VDM 3445
3446    *       RETURN  DESIRED INPUT MEDIA SELECTED                    VDM 3446
3447    *                                                               VDM 3447
3448    CHME    JMPM    FETCHA          GET SELECTOR INDICATOR          VDM 3448
3449            SUB     ='R'                                                3449
3450            JAZ     CHMR                                            VDM 3450
3451            DAR                     S ?                             VDM 3451
3452            MJANZ   EXC9D           JUMP IF INVALID INPUT               3452
3453            DECR    1               SET TO PI INPUT                     3453
3454            DATA    01006           SKIP NEXT WORD                      3454
3455    CHMR    TZA                     SET TO SI INPUT                     3455
3456            STA     MEDIA                                               3456
3457            JMP     EXC1D                                           VDM 3457
3458    MEDIA   DATA    0                                               VDM 3458
3459            EJEC                                                        3459
3460    *                                                               VDM 3460
3461    *       CONVERT ASCII CHAR TO HEX DIGIT                         VDM 3461
3462    *                                                               VDM 3462
3463    *       CALLING SEQUENCE                                        VDM 3463
3464    *       LDA     ASCII CHAR TO BE CONVERTED                      VDM 3464
3465    *       JMPM    CONV                                            VDM 3465
3466    *                                                               VDM 3466
3467    *       RETURN  A CONTAINS HEX DIGIT IF LEGAL                   VDM 3467
3468    *               A SET NEGATIVE IF NOT                           VDM 3468
3469    *                                                               VDM 3469
3470    CONV    ENTR                                                    VDM 3470
3471            SUB     ='0'                                                3471
3472            JAN*    CONV            RETURN IF ILLEGAL                   3472
3473            SUB     =10                                                 3473
3474            JAN     CONV1           JUMP IF 0-9                         3474
3475            SUB     =7              COMPARE TO ASCII A                  3475
3476            JAN*    CONV            RETURN IF ILLEGAL                   3476
```

```
3477                SUB     =6              COMPARE TO ASCII G                      3477
3478                XAP     CONV2           INSURE NEGATIVE RESULT IF ILLEGAL       3478
3479                ADD     =6              A->0, B->1, ETC                         3479
3480      CONV1     ADD     =10                                                     3480
3481                JMP*    CONV                                                    3481
3482      CONV2     LDA     =0100000                                                3482
3483                EJEC                                                    VDM 3483
3484      *                                                                 VDM 3484
3485      *         THIS ROUTINE WILL BREAK A CCS WORD INTO ITS SEPARATE FIELDS      3485
3486      *                                                                 VDM 3486
3487      *         CALLING SEQUENCE                                        VDM 3487
3488      *         LDX     =DROM           ADDRESS OF DATA BUFFER                  3488
3489      *         LDA     CCS ADDRESS                                             3489
3490      *         JMPM    FELD                                            VDM 3490
3491      *                                                                 VDM 3491
3492      *         RETURN                                                  VDM 3492
3493      *         THE FIELDS WILL BE STORED IN XTS, XAF, XMS, ETC.        VDM 3493
3494      *         XA, XB, XF MAY NOT COME FROM CCS - E.G. REGISTER SELECTION       3494
3495      *                                                                 VDM 3495
3496      FELD      ENTR                                                    VDM 3496
3497                LRLA    2               TIMES FOUR                      VDM 3497
3498                ADD     DBUF-DROM,1     BASE OF CCS PAGE                        3498
3499                TAB                                                             3499
3500                LDA     0,2             FIRST 16 BITS OF ROM WORD               3500
3501                STA     CQ1,1                                                   3501
3502                LDA     1,2             SECOND 16 BITS OF ROM WORD              3502
3503                STA     CQ2,1                                                   3503
3504                LDA     2,2             THIRD 16 BITS OF ROM WORD               3504
3505                STA     CQ3,1                                                   3505
3506                LDA     3,2             FOURTH 16 BITS OF ROM WORD              3506
3507                STA     CQ4,1                                                   3507
3508                TAB                                                             3508
3509                LLRL    010                                             VDM 3509
3510                CALL    CDAB            DETERMINE A&B FIELDS            VDM 3510
3511      *                                                                 VDM 3511
3512                LDB     CQ1,1           FIRST 16 BITS                           3512
3513      *                                                                 VDM 3513
3514                TZA                                                     VDM 3514
3515                LLRL    4                                               VDM 3515
3516                STA     XTS,1           TS FIELD                        VDM 3516
3517      *                                                                 VDM 3517
3518                TZA                                                     VDM 3518
3519                LLRL    5                                               VDM 3519
3520                STA     XAF,1           AF FIELD                        VDM 3520
3521      *                                                                 VDM 3521
3522                TZA                                                     VDM 3522
3523                LLRL    4                                               VDM 3523
3524                STA     XMS,1           MS FIELD                        VDM 3524
3525      *                                                                 VDM 3525
3526                TZA                                                     VDM 3526
3527                LLRL    1                                               VDM 3527
3528                STA     XMT,1           MT FIELD                        VDM 3528
3529                TZA                                                     VDM 3529
3530                LLRL    2                                               VDM 3530
3531      *                                                                 VDM 3531
3532                LDB     CQ2,1           SECOND 16 BITS                          3532
3533      *                                                                 VDM 3533
```

```
 4674                DATA    '08'        EOF ENCOUNTERED BEFORE LOAD COMPLETE              4674
 4675                DATA    '09'        END OF DEVICE BEFORE LOAD COMPLETE                4675
 4676                DATA    '10'        SEQUENCE ERROR ON BI                              4676
 4677                DATA    '11'        INVALID LOADER CODE                               4677
 4678                DATA    '12'        CHECKSUM ERROR DURING LOAD                        4678
 4679                DATA    '13'        UNDEFINED MEMORY OP CODE                          4679
 4680                DATA    '14'        ATTEMPT TO WRITE OUTSIDE DEFINED MEMORY           4680
 4681                DATA    '15'        ATTEMPT TO LOAD OUTSIDE DEFINED MEMORY            4681
 4682                DATA    '16'        FIELD NAME ERROR                                  4682
 4683                DATA    '17'        FIELD RANGE ERROR                                 4683
 4684      MSGTXT    DATA    '  MSDD'    ERROR MESSAGE SKELETON                            4684
 4685      MSGL      EQU     *-MSGTXT                                                      4685
 4686                END     SMLTR                                                 VDM 4686

CPU:1.176   CTP:.105   SUPS:20.751

@HDG,P                                  SOURCE DECK: 29!

     MICXRF

@PRT,S          PFPFILE.MICXRF
FURPUR 27-S  08/16/78  10:11:48
```