digital

Computer
Special Systems
PARIS - FRANCE

DIAGNOSTIC LISTS

XY 11

Plotter Interface

DIAGNOSTIC LISTS

logic test. start 600

plotter test start 2000

SR 5 raise pen
4 lower pen
3 move right } pen
2 move left }
1 move up } drum.
0 move down }

XY 11

Plotter Interface

timing test prim DAI.

start 2000.

PROGRAM:

XY11 PLOTTER INTERFACE TEST
CSSK-11-267 22 FEB 71
G. MUNYAN

```
****** A
1 000000 000000 @
2                ;
3                ;XY11 PLOTTER INTERFACE TEST
4                ;GLENN MUNYAN
5                ;3-2-71
6                ;
7        000002          .ASECT
8        177776 PS=177776
9        177570 SR=177570
10       177560 TKS=177560
11       177562 TKB=177562
12       177564 TPS=177564
13       177566 TPB=177566
14       000000 R0=%0
15       000001 R1=%1
16       000002 R2=%2
17       000003 R3=%3
18       000004 R4=%4
19       000005 R5=%5
20       000006 SP=%6
21       000007 PC=%7
22       104000 ERR=EMT
23       000000 XX=HALT
24       000240 NOP=240
25       000015 CR=15
26       000012 LF=12
27       000002 G=2
28       000001 B=1
29       000003 GB=3
30                       .MACRO  LOOP A,B
31                       JSR     PC,@#LERCHK     ;LOOP TO "A" ON ERROR
32                       JMP     @#A             ;IF SR13=1.
33               B:      JMP     @#.+4
34                       .ENDM
35
36       000000          .=0
37 00000 000002          .+2
38 00002 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
39 00004 000002          .+2
40 00006 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
41 00010 000012          .+2
42 00012 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
43 00014 000016          .+2
44 00016 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
45 00020 000022          .+2
46 00022 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
47 00024 000026          .+2
48 00026 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
49 00030 000032          .+2
50 00032 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
51 00034 000036          .+2
52 00036 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
53 00040 000042          .+2
54 00042 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
55 00044 000046          .+2
56 00046 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
```



```
57 00050 000052          .+2
58 00052 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
59 00054 000056          .+2
60 00056 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
61 00060 000062          .+2
62 00062 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
63 00064 000066          .+2
64 00066 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
65 00070 000072          .+2
66 00072 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
67 00074 000076          .+2
68 00076 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
69 00100 000102          .+2
70 00102 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
71 00104 000106          .+2
72 00106 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
73 00110 000112          .+2
74 00112 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
75 00114 000116          .+2
76 00116 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
77 00120 000122          .+2
78 00122 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
79 00124 000126          .+2
80 00126 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
81 00130 000132          .+2
82 00132 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
83 00134 000136          .+2
84 00136 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
85 00140 000142          .+2
86 00142 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
87 00144 000146          .+2
88 00146 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
89 00150 000152          .+2
90 00152 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
91 00154 000156          .+2
92 00156 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
93 00160 000162          .+2
94 00162 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
95 00164 000166          .+2
96 00166 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
97 00170 000172          .+2
98 00172 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
99 00174 000176          .+2
100 0176 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
101 0200 000202          .+2
102 0202 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
103 0204 000206          .+2
104 0206 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
105 0210 000212          .+2
106 0212 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
107 0214 000216          .+2
108 0216 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
109 0220 000222          .+2
110 0222 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
111 0224 000226          .+2
112 0226 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
113 0230 000232          .+2
```

```
114 0232 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
115 0234 000236          .+2
116 0236 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
117 0240 000242          .+2
118 0242 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
119 0244 000246          .+2
120 0246 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
121 0250 000252          .+2
122 0252 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
123 0254 000256          .+2
124 0256 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
125 0260 000262          .+2
126 0262 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
127 0264 000266          .+2
128 0266 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
129 0270 000272          .+2
130 0272 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
131 0274 000276          .+2
132 0276 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
133 0300 000302          .+2
134 0302 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
135 0304 000306          .+2
136 0306 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
137 0310 000312          .+2
138 0312 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
139 0314 000316          .+2
140 0316 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
141 0320 000322          .+2
142 0322 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
143 0324 000326          .+2
144 0326 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
145 0330 000332          .+2
146 0332 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
147 0334 000336          .+2
148 0336 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
149 0340 000342          .+2
150 0342 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
151 0344 000346          .+2
152 0346 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
153 0350 000352          .+2
154 0352 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
155 0354 000356          .+2
156 0356 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
157 0360 000362          .+2
158 0362 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
159 0364 000366          .+2
160 0366 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
161 0370 000372          .+2
162 0372 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
163 0374 000376          .+2
164 0376 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
165 0400 000402          .+2
166 0402 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
167 0404 000406          .+2
168 0406 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
169 0410 000412          .+2
170 0412 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
```



```
171 0414 000416          .+2
172 0416 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
173 0420 000422          .+2
174 0422 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
175 0424 000426          .+2
176 0426 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
177 0430 000432          .+2
178 0432 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
179 0434 000436          .+2
180 0436 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
181 0440 000442          .+2
182 0442 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
183 0444 000446          .+2
184 0446 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
185 0450 000452          .+2
186 0452 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
187 0454 000456          .+2
188 0456 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
189 0460 000462          .+2
190 0462 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
191 0464 000466          .+2
192 0466 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
193 0470 000472          .+2
194 0472 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
195 0474 000476          .+2
196 0476 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
197 0500 000502          .+2
198 0502 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
199 0504 000506          .+2
200 0506 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
201 0510 000512          .+2
202 0512 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
203 0514 000516          .+2
204 0516 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
205 0520 000522          .+2
206 0522 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
207 0524 000526          .+2
208 0526 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
209 0530 000532          .+2
210 0532 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
211 0534 000536          .+2
212 0536 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
213 0540 000542          .+2
214 0542 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
215 0544 000546          .+2
216 0546 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
217 0550 000552          .+2
218 0552 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
219 0554 000556          .+2
220 0556 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
221 0560 000562          .+2
222 0562 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
223 0564 000566          .+2
224 0566 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
225 0570 000572          .+2
226 0572 000000          HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
227 0574 000576          .+2
```

```
228 057E 000000         HALT                    ;TRAPPED OR INTERRUPTED TO PREVI
229
230      000030         .=30
231 0030 003144         .WORD   ERROR
232 0032 000340         .WORD   340             ;BPU AT LEVEL #7 DURING ERROR
```



```
1
2        000600         .=600
3                ;
4                ;LOGIC TEST
5                ;
6 000600 012737 TEST1:  MOV     #340,PS         ;SET BPU TO LEVEL #7
         000340
         177776
7 000606 012706         MOV     #500,SP         ;INITIALIZE STACK POINTER
         000500
8
9                ;TEST XYDB AFTER INITIATION
10 00612 000005 T1.1:   RESET                   ;RESET
11 00614 005037         CLR     GOOD
         004212
12 00620 017737         MOV     @XYDB,BAD       ;READ XYDB
         003406
         004204
13 00626 001405         BEQ     T1.1LP
14 00630 104003         ERR+GB                  ;XYDB NOT ZERO AFTER INITIATION
15 00632                LOOP    T1.1,T1.1LP
16
17               ;TEST XYCS AFTER INITIATION
18 00646 000005 T1.2:   RESET                   ;RESET
19 00650 012737         MOV     #200,GOOD       ;SET GOOD TO 200
         000200
         004212
20 00656 017737         MOV     @XYCS,BAD       ;READ XYCS
         003346
         004204
21 00664 023737         CMP     GOOD,BAD        ;COMPARE XYCS TO 200
         004212
         004204
22 00672 001405         BEQ     T1.2LP
23 00674 104003         ERR+GB                  ;XYCS NOT 200 AFTER RESET
24 00676                LOOP    T1.2,T1.2LP
25
26               ;TEST TO SEE IF THE READY FLAG FUNCTIONS PROPERLY
27                       ;FIRST TEST TO SEE IF A COMMAND CLEARS THE READY FLAG
28 00712 000005 T1.3:   RESET
29 00714 005037         CLR     GOOD
         004212
30 00720 013777         MOV     GOOD,@XYDB      ;ISSUE A DUMMY COMMAND
         004212
         003304
31 00726 017737         MOV     @XYCS,BAD       ;READ XYCS
         003276
         004204
32 00734 023737         CMP     GOOD,BAD
         004212
         004204
33 00742 001405         BEQ     T1.3LP
34 00744 104003         ERR+GB                  ;XYCS REGISTER NOT EQUAL TO 0 AF
35                               ;(READY FLAG-BIT 7-SHOULD BE CLE
36 00746                LOOP    T1.3,T1.3LP
37                       ;NOW TEST TO SEE THAT THE READY FLAG GETS SET AFTER A CO
38 00762 005037         CLR     CNT
```

```
                004206
39 00766 012737         MOV     #200,GOOD
                000200
                004212
40 00774 017737 T1.3A:  MOV     @XYCS,BAD       ;READ XYCS
                003230
                004204
41 01002 023737         CMP     GOOD,BAD
                004212
                004204
42 01010 001412         BEQ     T1.4
43 01012 005237         INC     CNT
                004206
44 01016 001366         BNE     T1.3A
45 01020 104003         ERR+GB                  ;XYCS REGISTER NOT EQUAL TO 200
46                                              ;(READY FLAG-BIT 7-SHOULD BE SET
47 01022                LOOP    T1.3,T1.3AL
```



```
1
2               ;PERFORM LOAD-READ TEST ON XYDB WITH VALUE OF 77
3
4 001036 000005 T1.4:   RESET
5 001040 012737         MOV     #77,GOOD
                000077
                004212
6 001046 013777         MOV     GOOD,@XYDB      ;LOAD XYDB WITH 77
                004212
                003156
7 001054 017737         MOV     @XYDB,BAD       ;READ XYDB AFTER LOAD
                003152
                004204
8 001062 023737         CMP     GOOD,BAD        ;XYDB SHOULD=77
                004212
                004204
9 001070 001405         BEQ     T1.4LP          ;DOES IT?
10 01072 104003         ERR+GB                  ;XYDB-LOAD-READ COMPARE ERROR
11 01074                LOOP    T1.4,T1.4LP
12 01110 037727 T1.4WT: BIT     @XYCS,#200
                003114
                000200
13 01116 001774         BEQ     T1.4WT          ;WAIT FOR READY
14 01120 005037         CLR     GOOD
                004212
15 01124 017737         MOV     @XYDB,BAD       ;READ XYDB AFTER READY FLAG
                003102
                004204
16 01132 023737         CMP     GOOD,BAD        ;XYDB SHOULD=0
                004212
                004204
17 01140 001405         BEQ     T1.4AL          ;DOES IT?
18 01142 104003         ERR+GB                  ;XYDB NOT EQUAL TO 0 AFTER READY
19                                              ;(THE COMMAND SHOULD HAVE CLEARE
20 01144                LOOP    T1.4,T1.4AL
21
22              ;PERFORM LOAD-READ TEST ON XYDB WITH VALUE OF 0
23
24 01160 000005 T1.5:   RESET
25 01162 005037         CLR     GOOD
                004212
26 01166 013777         MOV     GOOD,@XYDB      ;LOAD XYDB WITH 0
                004212
                003036
27 01174 017737         MOV     @XYDB,BAD       ;READ XYDB AFTER LOAD
                003032
                004204
28 01202 023737         CMP     GOOD,BAD        ;XYDB SHOULD=0
                004212
                004204
29 01210 001405         BEQ     T1.5LP          ;DOES IT?
30 01212 104003         ERR+GB                  ;XYDB-LOAD-READ COMPARE ERROR
31 01214                LOOP    T1.5,T1.5LP
32 01230 037727 T1.5WT: BIT     @XYCS,#200
                002774
                000200
33 01236 001774         BEQ     T1.5WT          ;WAIT FOR READY
```

```
34 01240 005037          CLR     GOOD
         004212
35 01244 017737          MOV     @XYDB,BAD       ;READ XYDB AFTER READY FLAG
         002762
         004204
36 01252 023737          CMP     GOOD,BAD        ;XYDB SHOULD=0
         004212
         004204
37 01260 001405          BEQ     T1.5AL          ;DOES IT?
38 01262 104003          ERR+GB                  ;XYDB NOT EQUAL TO 0 AFTER READY
39                                               ;(THE COMMAND SHOULD HAVE CLEARE
40 01264                 LOOP    T1.5,T1.5AL
```



```
1
2                ;TEST THE INTERRUPT ENABLE BIT
3                        ;FIRST SET AND READ IT
4 001300 000005 T1.6:    RESET
5 001302 012737          MOV     #340,PS
         000340
         177776
6 001310 012737          MOV     #300,GOOD
         000300
         004212
7 001316 012777          MOV     #100,@XYCS      ;SET THE INTERRUPT ENABLE BIT
         000100
         002704
8 001324 017737          MOV     @XYCS,BAD       ;READ XYCS
         002700
         004204
9 001332 023737          CMP     GOOD,BAD
         004212
         004204
10 01340 001405          BEQ     T1.6LP
11 01342 104003          ERR+GB                  ;XYCS REGISTER LOAD-READ COMPARE
12 01344                 LOOP    T1.6,T1.6LP
13                       ;THEN CLEAR AND READ IT
14 01360 012737 T1.6A:   MOV     #200,GOOD
         000200
         004212
15 01366 005077          CLR     @XYCS           ;CLEAR THE INTERRUPT ENABLE BIT
         002636
16 01372 017737          MOV     @XYCS,BAD       ;READ XYCS
         002632
         004204
17 01400 023737          CMP     GOOD,BAD
         004212
         004204
18 01406 001405          BEQ     T1.6AL
19 01410 104003          ERR+GB                  ;XYCS REGISTER LOAD-READ COMPARE
20 01412                 LOOP    T1.6,T1.6AL
21               ;
22               ;TEST THE ABILITY OF THE INTERFACE TO CAUSE AN INTERRUPT TO THE
23               ;BY ENABLING THE INTERRUPT WITH THE READY FLAG ALREADY SET
24 01426 000005 T1.7:    RESET
25 01430 012737          MOV     #140,PS         ;SET BPU TO LEVEL #3
         000140
         177776
26 01436 005004          CLR     R4              ;SET R4 TO 0
27 01440 012777          MOV     #T1INT1,@RDYPCV ;SETUP INTERRUPT PC VECTOR
         003102
         002566
28 01446 012777          MOV     #340,@RDYPSV    ;SETUP INTERRUPT PS VECTOR
         000340
         002562
29 01454 012777          MOV     #100,@XYCS      ;ENABLE INTERRUPT
         000100
         002546
30 01462 000240          NOP
31 01464 000240          NOP
32 01466 000240          NOP
```

```
33 01470 012737          MOV     #340,PS         ;LOCK OUT INTERRUPT IF IT HASN'T
         000340
         177776
34 01476 012737          MOV     #1,GOOD         ;SET GOOD TO 1
         000001
         004212
35 01504 010437          MOV     R4,BAD          ;PUT CONTENTS OF R4 IN BAD
         004204
36 01510 023737          CMP     GOOD,BAD        ;R4 SHOULD HAVE BEEN SER TO 1
         004212
         004204
37 01516 001405          BEQ     T1.7LP          ;WAS IT?
38 01520 104003          ERR+GB                  ;INTERRUPT DIDN'T OCCUR OR MULTI
39                                               ;THE INTERRUPT SERVICE ROUTINE A
40 01522                 LOOP    T1.7,T1.7LP
```



```
1
2                ;
3                ;TEST THE ABILITY OF THE INTERFACE TO CAUSE AN INTERRUPT BY ISSU
4                ;COMMAND THEN ENABLING THE INTERRUPT.  AN INTERRUPT SHOULD OCCUR
5                ;WHEN THE READY FLAG SETS.
6                ;
7  001536 000005 T1.6:   RESET
8  001540 012737         MOV     #140,PS         ;SET BPU TO LEVEL #3
          000140
          177776
9  001546 005004         CLR     R4              ;SET R4 TO 0
10 01550 012777         MOV     #T1INT2,@RDYPCV ;SETUP INTERRUPT PC VECTOR
          003110
          002456
11 01556 012777         MOV     #340,@RDYPSV    ;SETUP INTERRUPT PS VECTOR
          000340
          002452
12 01564 005077         CLR     @XYDB           ;ISSUE A DUMMY COMMAND TO CLEAR
          002442
13 01570 012777         MOV     #100,@XYCS      ;ENABLE THE INTERRUPT
          000100
          002432
14 01576 037727 T1.8WT: BIT     @XYCS,#200
          002426
          000200
15 01604 001774         BEQ     T1.8WT          ;WAIT FOR READY
16 01606 012737         MOV     #340,PS         ;LOCKOUT INTERRUPT IF IT HASN'T
          000340
          17777b
17 01614 012737         MOV     #2,GOOD
          000002
          004212
18 01622 010437         MOV     R4,BAD
          004204
19 01626 023737         CMP     GOOD,BAD
          004212
          004204
20 01634 001405         BEQ     T1.8LP
21 01636 104003         ERR+GB                  ;INTERRUPT DIDN'T OCCUR OR MULTI
22                                               ;THE INTERRUPT SERVICE ROUTINE A
23 01640                LOOP    T1.8,T1.8LP
24 01654 004737 T1LPCK: JSR     PC,ENDCK        ;IF SR0=0 PRINT "END" AND HALT
          003120
25 01660 000137         JMP     T1.1
          000612
26
```

```
                        .=2000
                ;
                ;TIMING SCOPE LOOP-THIS TEST ALLOWS THE OPERATOR TO DYNAMICALLY
                ;TIMING DELAYS IN THE INTERFACE BY PROVIDING AN OPPORTUNITY FOR
                ;OPERATOR TO SELECT A COMMAND IN SR5-0.  THE PROGRAM WILL ISSUE
                ;THE COMMAND, WAIT FOR READY OR WAIT FOR 6 MILLI SECONDS, THEN I
                ;
002000 012737   TEST2:  MOV     #340,PS         ;SET BPU TO LEVEL #7
       000340
       177776
002006 000005           RESET                   ;RESET
002010 013737   T2A:    MOV     SR,GOOD         ;GET SWITCHES
       177570
       004212
002016 042737           BIC     #177700,GOOD    ;MASK OUT ALL EXCEPT BITS 5-0
       177700
       004212
002024 013777           MOV     GOOD,@XYDB      ;ISSUE THESE BITS AS THE COMMAND
       004212
       002200
002032 005737           TST     SR              ;TEST SWITCH REGISTER BIT 15
       177570
002036 100405           BMI     T2C             ;IF SR15=1 GO TO DELAY
002040 037727   T2B:    BIT     @XYCS,#200      ;IF SR15=0
       002164
       000200
002046 001774           BEQ     T2B             ;WAIT FOR THE READY FLAG
002050 000757           BR      T2A             ;THEN GO AND ISSUE THE COMMAND A
002052 012737   T2C:    MOV     #5000,CNT       ;SET CNT
       005000
       004206
002060 005337   T2D:    DEC     CNT             ;DELAY FOR
       004206
002064 001375           BNE     T2D             ;     6 MILLI SECONDS
002066 000750           BR      T2A             ;THEN GO AND ISSUE THE COMMAND A

                ;*********************************************************
                ;**      THE DELAY WILL BE 6 MILLI SECONDS IF PROGRAM     **
                ;**      IS RUN ON AN 11/45 AND 12 MILLI SECONDS IF       **
                ;**      RUN ON AN 11/20. TO CHANGE DELAY TO 6 MILLI-     **
                ;**      SECONDS FOR AN 11/20 CHANGE LOCATION 2054 TO     **
                ;**      2400.                                            **
                ;*********************************************************
```



```
                        .=2100
                ;
                ;PATTERN TEST - THIS TEST DRAWS PREDETERMINED PATTERNS ON THE PL
                ;THE ONLY REAL DIAGNOSTIC FEATURE OF THE TEST IS THAT IT DRAWS T
                ;HALF OF THE BOX PATTERN AT HALF SPEED TO HELP THE OPERATOR DETE
                ;IF THE DELAYS IN THE INTERFACE ARE SET TOO FAST FOR THE PLOTTER
                ;
002100 012737   TEST3:  MOV     #340,PS         ;SET BPU TO LEVEL #7
       000340
       177776
002106 012706           MOV     #600,SP         ;INITIALIZE STACK POINTER
       000600
002112 000005           RESET
002114 004737           JSR     PC,GETSPS       ;GET THE NUMBER OF STEPS FOR THI
       003722
002120 012777           MOV     #T3IN1,@RDYPCV  ;SETUP INTERRUPT PC VECTOR
       003116
       002106
002126 012777           MOV     #340,@RDYPSV    ;SETUP INTERRUPT PS VECTOR
       000340
       002102
002134 012737           MOV     #140,PS         ;SET BPU TO LEVEL #3
       000140
       177776
002142 012777           MOV     #100,@XYCS      ;ENABLE INTERRUPT
       000100
       002060
002150 000240           NOP
002152 000240           NOP
002154 000240           NOP                     ;WAIT TO ALLOW FIRST INTERRUPT T
002156 012777   T3BRD:  MOV     #40,@XYDB       ;PEN UP
       000040
       002046
002164 000001           WAIT
002166 012737           MOV     #62,COUNT
       000062
       004210
002174 012777   T3BRD1: MOV     #10,@XYDB       ;PEN RIGHT (50 STEPS FOR BORDER)
       000010
       002030
002202 000001           WAIT
002204 005337           DEC     COUNT
       004210
002210 001371           BNE     T3BRD1
002212 000137           JMP     T3BOX
       002216
```

```
1
2                       ;DRAW A BOX
3 002216 033727 T3BOX:  BIT     SR,#10          ;TEST SR BIT 3
         177570
         000010
4 002224 001473         BEQ     T3X             ;IF IT IS A 1, DRAW A "BOX"
5 002226 012777         MOV     #20,@XYDB       ;PEN DOWN
         000020
         001776
6 002234 000001         WAIT
7 002236 005077         CLR     @XYDB           ;DUMMY COMMAND
         001770
8 002242 000001         WAIT
9 002244 013737         MOV     STEPS,COUNT
         004220
         004210
10 02252 012777 T3B1:   MOV     #10,@XYDB       ;PEN RIGHT
         000010
         001752
11 02260 000001         WAIT
12 02262 005077         CLR     @XYDB           ;DUMMY COMMAND
         001744
13 02266 000001         WAIT
14 02270 005337         DEC     COUNT
         004210
15 02274 001366         BNE     T3B1
16 02276 013737         MOV     STEPS,COUNT
         004220
         004210
17 02304 012777 T3B2:   MOV     #1,@XYDB        ;DRUM DOWN
         000001
         001720
18 02312 000001         WAIT
19 02314 005077         CLR     @XYDB           ;DUMMY COMMAND
         001712
20 02320 000001         WAIT
21 02322 005337         DEC     COUNT
         004210
22 02326 001366         BNE     T3B2
23 02330 013737         MOV     STEPS,COUNT
         004220
         004210
24 02336 012777 T3B3:   MOV     #4,@XYDB        ;PEN LEFT
         000004
         001666
25 02344 000001         WAIT
26 02346 005337         DEC     COUNT
         004210
27 02352 001371         BNE     T3B3
28 02354 013737         MOV     STEPS,COUNT
         004220
         004210
29 02362 012777 T3B4:   MOV     #2,@XYDB        ;DRUM UP
         000002
         001642
30 02370 000001         WAIT
31 02372 005337         DEC     COUNT
```



```
         004210
32 02376 001371         BNE     T3B4
33 02400 012777         MOV     #40,@XYDB       ;PEN UP
         000040
         001624
34 02406 000001         WAIT
35 02410 000137         JMP     T3X             ;GO TO NEXT PATTERN
         002414
```

```
1
2                ;DRAW AN "X"
3  002414 033727 T3X:    BIT     SR,#4           ;TEST SR BIT 2
          177570
          000004
4  002422 001472         BEQ     T3PLUS          ;IF IT IS A 1 DRAW A "X"
5  002424 012777         MOV     #20,@XYDB       ;PEN DOWN
          000020
          001600
6  002432 000001         WAIT
7  002434 013737         MOV     STEPS,COUNT
          004220
          004210
8  002442 012777 T3X1:   MOV     #11,@XYDB       ;DRUM DOWN & PEN RIGHT
          000011
          001562
9  002450 000001         WAIT
10 02452  005337         DEC     COUNT
          004210
11 02456  001371         BNE     T3X1
12 02460  012777         MOV     #40,@XYDB       ;PEN UP
          000040
          001544
13 02466  000001         WAIT
14 02470  013737         MOV     STEPS,COUNT
          004220
          004210
15 02476  012777 T3X2:   MOV     #2,@XYDB        ;DRUM UP
          000002
          001526
16 02504  000001         WAIT
17 02506  005337         DEC     COUNT
          004210
18 02512  001371         BNE     T3X2
19 02514  012777         MOV     #20,@XYDB       ;PEN DOWN
          000020
          001510
20 02522  000001         WAIT
21 02524  013737         MOV     STEPS,COUNT
          004220
          004210
22 02532  012777 T3X3:   MOV     #5,@XYDB                ;DRUM DOWN & PEN LEFT
          000005
          001472
23 02540  000001         WAIT
24 02542  005337         DEC     COUNT
          004210
25 02546  001371         BNE     T3X3
26 02550  012777         MOV     #40,@XYDB       ;PEN UP
          000040
          001454
27 02556  000001         WAIT
28 02560  013737         MOV     STEPS,COUNT
          004220
          004210
29 02566  012777 T3X4:   MOV     #2,@XYDB        ;DRUM UP
          000002
```



```
          001436
30 02574  000001         WAIT
31 02576  005337         DEC     COUNT
          004210
32 02602  001371         BNE     T3X4
33 02604  000137         JMP     T3PLUS
          002610
34
```

```
1
2                       ;DRAW A "+"
3  002610 033727 T3PLUS: BIT     SP,#2           ;TEST SP BIT 1
          177570
          000002
4  002616 001504         BEQ     T3LPCK          ;IF 1T IS A 1, DRAW A "+"
5  002620 013737         MOV     HALF,COUNT
          004214
          004210
6  002626 012777 T3P1:   MOV     #10,@XYDB       ;PEN RIGHT
          000010
          001376
7  002634 000001         WAIT
8  002636 005337         DEC     COUNT
          004210
9  002642 001371         BNE     T3P1
10 002644 012777         MOV     #20,@XYDB       ;PEN DOWN
          000020
          001360
11 002652 000001         WAIT
12 002654 013737         MOV     STEPS,COUNT
          004220
          004210
13 002662 012777 T3P2:   MOV     #1,@XYDB        ;DRUM DOWN
          000001
          001342
14 002670 000001         WAIT
15 002672 005337         DEC     COUNT
          004210
16 002676 001371         BNE     T3P2
17 002700 012777         MOV     #40,@XYDB       ;PEN UP
          000040
          001324
18 002706 000001         WAIT
19 002710 013737         MOV     HALF,COUNT
          004214
          004210
20 002716 012777 T3P3:   MOV     #12,@XYDB       ;DRUM UP + PEN RIGHT
          000012
          001306
21 002724 000001         WAIT
22 002726 005337         DEC     COUNT
          004210
23 002732 001371         BNE     T3P3
24 002734 012777         MOV     #20,@XYDB       ;PEN DOWN
          000020
          001270
25 002742 000001         WAIT
26 002744 013737         MOV     STEPS,COUNT
          004220
          004210
27 002752 012777 T3P4:   MOV     #4,@XYDB        ;PEN LEFT
          000004
          001252
28 002760 000001         WAIT
29 002762 005337         DEC     COUNT
          004210
```



```
30 002766 001371         BNE     T3P4
31 002770 012777         MOV     #40,@XYDB       ;PEN UP
          000040
          001234
32 002776 000001         WAIT
33 003000 013737         MOV     HALF,COUNT
          004214
          004210
34 003006 012777 T3P5:   MOV     #2,@XYDB        ;DRUM UP
          000002
          001216
35 003014 000001         WAIT
36 003016 005337         DEC     COUNT
          004210
37 003022 001371         BNE     T3P5
38 003024 000137         JMP     T3LPCK
          003030
39
40 003030 004737 T3LPCK: JSR     PC,ENDCK        ;IF SR0=0, PRINT "END" AND HALT
          003120
41 003034 012777         MOV     #40,@XYDB       ;PEN UP
          000040
          001170
42 003042 000001         WAIT
43 003044 013737         MOV     STEPS,COUNT
          004220
          004210
44 003052 062737         ADD     #62,COUNT
          000062
          004210
45 003060 012777 T3LP1:  MOV     #1,@XYDB        ;DRUM DOWN TO STARTING POSITION
          000001
          001144
46 003066 000001         WAIT
47 003070 005337         DEC     COUNT
          004210
48 003074 001371         BNE     T3LP1
49 003076 000137         JMP     T3BOX
          002216
```

```
1
2 003102 062704 T1INT1: ADD     #1,P4
         000001
3 003106 000002         RTI
4 003110 062704 T1INT2: ADD     #2,R4
         000002
5 003114 000002         RTI
6 003116 000002 I3INT:  RTI
7
8 003120 032737 ENDCK:  BIT     #1,SR
         000001
         177570
9 003126 001005         BNE     ENLXIT
10 03130 012700         MOV     #ENDMSG,R0
         004176
11 03134 004737         JSR     PC,TYPMSG       ;PRINT "END"
         003372
12 03140 000000         XX                      ;HALT AT END OF TEST
13 03142 000207 ENDXIT: RTS     PC
```



```
1
2 003144 032737 ERROR:  BIT     #40000,SR       ;IF SR14=1 (DELETE PRINTOUT)
         040000
         177570
3 003152 001044         BNE     E4              ;EXIT NOW
4 003154 012700         MOV     #PCMSG,R0       ;PUT MESSAGE POINTER IN R0
         004116
5 003160 004737         JSR     PC,TYPMSG       ;PRINT "ERROR AT ADDRESS"
         003372
6 003164 011600         MOV     (SP),R0         ;GET SAVED PC
7 003166 162700         SUB     #2,R0           ;PUT ADDRESS OF ERROR CALL
         000002
8 003172 010002         MOV     R0,R2           ; IN R2
9 003174 012701         MOV     #4,R1           ;SET COUNT TO 4
         000004
10 03200 004737         JSR     PC,TYPOCT       ;PRINT PC
         003474
11 03204 012700         MOV     #PSMSG,R0       ;PUT MESSAGE POINTER IN R0
         004142
12 03210 004737         JSR     PC,TYPMSG       ;PRINT "PS="
         003372
13 03214 016600         MOV     2(SP),R0        ;GET SAVED STATUS
         000002
14 03220 012701         MOV     #3,R1           ;SET COUNT TO 3
         000003
15 03224 004737         JSR     PC,TYPOCT       ;PRINT STATUS
         003474
16 03230 032712 E1:     BIT     #2,(R2)         ;GOOD?
         000002
17 03234 001402         BEQ     E2              ;NO
18 03236 004737         JSR     PC,GDTYP        ;YES
         003300
19 03242 032712 E2:     BIT     #1,(R2)         ;BAD?
         000001
20 03246 001402         BEQ     E3              ;NO
21 03250 004737         JSR     PC,BDTYP        ;YES
         003326
22 03254 012700 E3:     MOV     #CR,R0          ;PUT CR IN R0
         000015
23 03260 004737         JSR     PC,TTO          ;PRINT CR+LF
         003434
24 03264 032737 E4:     BIT     #100000,SR      ;IF SR15=1
         100000
         177570
25 03272 001001         BNE     E5              ;BYPASS THE ERROR HALT
26 03274 000000 EHLT:   XX                      ;ERROR HALT
27 03276 000002 E5:     RTI                     ;EXIT
28 03300 012700 GDTYP:  MOV     #GDMSG,P0       ;PUT MESSAGE POINTER IN R0
         004150
29 03304 004737         JSR     PC,TYPMSG       ;PRINT "GOOD="
         003372
30 03310 013700         MOV     GOOD,R0
         004212
31 03314 012701         MOV     #3,P1           ;SET COUNT TO 3
         000003
32 03320 004737         JSR     PC,TYPOCT       ;PRINT CONTENTS OF GOOD
         003474
```

```
33 03324 000207          RTS     PC
34 03326 012700 BDTYP:   MOV     #BDMSG,R0      ;PUT MESSAGE POINTER IN R0
         004157
35 03332 004737          JSR     PC,TYPMSG      ;PRINT "BAD="
         003372
36 03336 013700          MOV     BAD,R0
         004204
37 03342 012701          MOV     #3,R1          ;SET COUNT TO 3
         000003
38 03346 004737          JSR     PC,TYPOCT      ;PRINT CONTENTS OF BAD
         003474
39 03352 000207          RTS     PC
```



```
1
2 003354 032737 LERCHK: BIT     #20000,SR      ;IF SR13=1
         020000
         177570
3 003362 001002         BNE     .+6            ;LOOP ON THE ERROR
4 003364 062716         ADD     #4,(SP)        ;IF 0, BYPASS LOOP JUMP
         000004
5 003370 000207         RTS     PC
6                ;MESSAGE PRINTOUT R0=ADDRESS OF BEGINNING OF MESSAGE
7 003372 010046 TYPMSG: MOV     R0,-(SP)       ;PUT MESSAGE POINTER ON STACK
8 003374 117600 TM1:    MOVB    @(SP),R0       ;GET FIRST CHARACTER
         000000
9 003400 022700         CMP     #100,R0        ;IS IT (@) THE TERMINATOR?
         000100
10 03404 001411         BEQ     TM3            ;YES - EXIT
11 03406 022700         CMP     #46,R0         ;IS IT CRLF FLAG?
         000046
12 03412 001002         BNE     TM2            ;NO
13 03414 012700         MOV     #CR,R0         ;YES - PUT CR IN R0
         000015
14 03420 004737 TM2:    JSR     PC,TTO         ;GO PRINT CONTENTS OF R0
         003434
15 03424 005216         INC     (SP)           ;INCREMENT POINTER TO NEXT CHARA
16 03426 000762         BR      TM1            ;GO GET NEXT CHARACTER
17 03430 005726 TM3:    TST     (SP)+          ;POP MESSAGE POINTER OFF THE STA
18 03432 000207         RTS     PC             ;EXIT
19               ;PRINT THE CHARACTER IN R0
20 03434 032737 TTO:    BIT     #40000,SR      ;IF SR14=1, DELETE PRINTOUT
         040000
         177570
21 03442 001010         BNE     TTO.2
22 03444 010037         MOV     R0,TPB         ;PRINT CHARACTER
         177566
23 03450 105737 TTO.1:  TSTB    TPS
         177564
24 03454 100375         BPL     TTO.1          ;WAIT FOR CHARACTER TO BE DONE
25 03456 022700         CMP     #CR,R0         ;WAS CHARACTER A CR?
         000015
26 03462 001401         BEQ     TTO.3          ;YES - PRINT A LINE FEED
27 03464 000207 TTO.2:  RTS     PC             ;NO - EXIT
28 03466 012700 TTO.3:  MOV     #LF,R0         ;PUT LF IN R0
         000012
29 03472 000760         BR      TTO            ;GO PRINT THE LF
30               ;OCTAL PRINTOUT. R0=VALUE R1=NUMBER OF DIGITS TO BE PRINTED
31 03474 010137 TYPOCT: MOV     R1,CNT         ;SET "CNT" FOR STACKING
         004206
32 03500 010037 TO.1:   MOV     R0,TEMP        ;PUT VALUE IN "TEMP"
         004222
33 03504 042737         BIC     #177770,TEMP   ;MASK OUT ALL BUT BITS 2,1+0
         177770
         004222
34 03512 062737         ADD     #60,TEMP       ;MAKE ASCII NUMBER
         000060
         004222
35 03520 013746         MOV     TEMP,-(SP)     ;STORE NUMBER ON STACK
         004222
36 03524 006000         ROR     R0
```

```
37 03526 006000          ROR     R0
38 03530 006000          ROR     R0
39 03532 005337          DEC     CNT             ;IF COUNT IS NOT ZERO
         004206
40 03536 003360          BGT     TO.1            ;GO STACK ANOTHER NUMBER
41 03540 022701          CMP     #6,R1
         000006
42 03544 001002          BNE     TO.2            ;IF 6 NUMBERS WERE SELECTED
43 03546 042716          BIC     #6,(SP)         ;CLEAR BITS 1+0 OF LAST NUMBER S
         000006
44 03552 010137 TO.2:    MOV     R1,CNT          ;SET "CNT" FOR PRINTING
         004206
45 03556 012600 TO.3:    MOV     (SP)+,R0        ;GET A NUMBER FROM THE STACK
46 03560 004737          JSR     PC,TTO          ;PRINT THE NUMBER
         003434
47 03564 005337          DEC     CNT             ;IF COUNT IS NOT ZERO
         004206
48 03570 003372          BGT     TO.3            ;GO PRINT ANOTHER NUMBER
49 03572 000207          RTS     PC
```



```
1
2
3                        ;GET A VALUE (0101-9999) FROM THE KEYBOARD, TERMINATE WITH CR
4                        ;R1 CONTAINS THE VALUE AT COMPLETION OF THIS ROUTINE
5 003574 012737 GETNUM:  MOV     #4,INPCNT       ;INPUT UP TO 4 DECIMAL DIGITS
         000004
         004216
6 003602 005001          CLR     R1              ;SET R1 TO 0
7 003604 004737 GET.1:   JSR     PC,TTIN         ;READ A CHAR INTO R0
         003674
8 003610 020027          CMP     R0,#215         ;IS IT A CR?
         000215
9 003614 001424          BEQ     GDEXIT          ;YES-EXIT
10 03616 004737          JSR     PC,TTO          ;NO-ECHO THE CHAR
         003434
11 03622 020027          CMP     R0,#271         ;IS CHAR GREATER THAN 9?
         000271
12 03626 101021          BHI     BDEXIT          ;YES-EXIT
13 03630 020027          CMP     R0,#257         ;IS CHAR LESS THAN 0?
         000257
14 03634 101416          BLOS    BDEXIT          ;YES-EXIT
15 03636 005337          DEC     INPCNT          ;HAVE MORE THAN 4 CHARS BEEN TYP
         004216
16 03642 100413          BMI     BDEXIT          ;YES-EXIT
17 03644 042700          BIC     #177760,R0      ;MASK 17 APPLIED TO R0
         177760
18 03650 000241          CLC
19 03652 006101          ROL     R1
20 03654 006101          ROL     R1
21 03656 006101          ROL     R1
22 03660 006101          ROL     R1              ;SHIFT R1 4 PLACES
23 03662 060001          ADD     R0,R1           ;ADD NEW NUMBER TO R1
24 03664 000747          BR      GET.1           ;GO GET ANOTHER CHAR
25 03666 062716 GDEXIT:  ADD     #4,(SP)         ;GOOD EXIT
         000004
26 03672 000207 BDEXIT:  RTS     PC              ;BAD EXIT
27
28 03674 005237 TTIN:    INC     TKS             ;FETCH A CHAR
         177560
29 03700 032737 TTI.1:   BIT     #200,TKS                ;IS DONE FLAG SET?
         000200
         177560
30 03706 001774          BEQ     TTI.1           ;NO-WAIT FOR IT TO SET
31 03710 013700          MOV     TKB,R0          ;READ A CHAR INTO R0
         177562
32 03714 042700          BIC     #177400,R0      ;MASK 377 APPLIED TO CHAR IN R0
         177400
33 03720 000207          RTS     PC
34
```

```
1
2                       ;GET THE NUMBER OF STEPS TO BE USED
3  003722 012700 GETSPS: MOV     #CR,R0
          000015
4  003726 004737         JSR     PC,TTO          ;PRINT CR+LF
          003434
5  003732 012700         MOV     #STPMSG,R0
          004165
6  003736 004737         JSR     PC,TYPMSG       ;PRINT "STEPS="
          003372
7  003742 004737         JSR     PC,GETNUM       ;GET THE NUMBER OF STEPS IN DECI
          003574
8  003746 000137         JMP     GETSPS          ;ERROR-TRY AGAIN
          003722
9  003752 004737         JSR     PC,GETOCT       ;CONVERT THE DECIMAL NUMBER TO A
          004012
10 03756 020327         CMP     R3,#144         ;IS CHAR GREATER THAN 100?
          000144
11 03762 101757         BLOS    GETSPS          ;NO-TRY A LARGER VALUE
12 03764 010337         MOV     R3,TOTAL        ;STORE TOTAL NUMBER OF STEPS
          004224
13 03770 162703         SUB     #144,R3         ;SUBPACT 100 FROM THE TOTAL
          000144
14 03774 010337         MOV     R3,STEPS        ;AND STORE THIS VALUE IN STEPS
          004220
15 04000 000241         CLC
16 04002 006003         ROR     R3
17 04004 010337         MOV     R3,HALF         ;STORE COUNT=TO HALF OF STEPS
          004214
18 04010 000207         RTS     PC
19
20                      ;CONVERT THE DECIMAL NUMBER IN R1 TO AN OCTAL VALUE AND PUT IT I
21 04012 010103 GETOCT: MOV     R1,R3
22 04014 042703         BIC     #177760,R3
          177760
23 04020 010102         MOV     R1,R2
24 04022 012737         MOV     #12,VALUE
          000012
          004226
25 04030 004737         JSR     PC,OCTSUB
          004062
26 04034 012737         MOV     #144,VALUE
          000144
          004226
27 04042 004737         JSR     PC,OCTSUB
          004062
28 04046 012737         MOV     #1750,VALUE
          001750
          004226
29 04054 004737         JSR     PC,OCTSUB
          004062
30 04060 000207         RTS     PC
31 04062 010200 OCTSUB: MOV     R2,R0
32 04064 006000         ROR     R0
33 04066 006000         ROR     R0
34 04070 006000         ROR     R0
35 04072 006000         ROR     R0
```



```
36 04074 010002         MOV     R0,R2
37 04076 042700         BIC     #177760,R0
          177760
38 04102 001404         BEQ     OS.2
39 04104 063703 OS.1:   ADD     VALUE,R3
          004226
40 04110 005300         DEC     R0
41 04112 001374         BNE     OS.1
42 04114 000207 OS.2:   RTS     PC
```

```
1
2 004116    046 PCMSG:  .ASCII  /&&ERROR AT ADDRESS @/
  004117    046
  004120    105
  004121    122
  004122    122
  004123    117
  004124    122
  004125    040
  004126    101
  004127    124
  004130    040
  004131    101
  004132    104
  004133    104
  004134    122
  004135    105
  004136    123
  004137    123
  004140    040
  004141    100
3
4 004142    040 PSMSG:  .ASCII  /  PS=@/
  004143    040
  004144    120
  004145    123
  004146    075
  004147    100
5
6 004150    046 GDMSG:  .ASCII  /&GOOD=@/
  004151    107
  004152    117
  004153    117
  004154    104
  004155    075
  004156    100
7
8 004157    046 BDMSG:  .ASCII  /&BAD=@/
  004160    102
  004161    101
  004162    104
  004163    075
  004164    100
9
10 04165    046 STPMSG: .ASCII  /&&STEPS=@/
   04166    046
   04167    123
   04170    124
   04171    105
   04172    120
   04173    123
   04174    075
   04175    100
11
12 04176    046 ENDMSG: .ASCII  /&&END@/
   04177    046
   04200    105
```



```
   04201    116
   04202    104
   04203    100
13
14                      .EVEN
15
16 04204 000000 BAD:    XX
17 04206 000000 CNT:    XX
18 04210 000000 COUNT:  XX
19 04212 000000 GOOD:   XX
20 04214 000000 HALF:   XX
21 04216 000000 INPCNT: XX
22 04220 000000 STEPS:  XX
23 04222 000000 TEMP:   XX
24 04224 000000 TOTAL:  XX
25 04226 000000 VALUE:  XX
26
27 04230 172554 XYCS:   172554          ;CONTROL STATUS REGISTER VECTOR
28 04232 172556 XYDB:   172556          ;DATA BUFFER REGISTER VECTOR
29 04234 000120 RDYPCV: 000120          ;READY PC VECTOR
30 04236 000122 RDYPSV: 000122          ;READY PS VECTOR
31
32       000001'        .END
```

```
.MAIN.  MACRO V06-03 15-APR-75 00:00 PAGE 16-2
SYMBOL TABLE

B      = 000001          BAD     004204          BDEXIT  003672
BDMSG   004157           BDTYP   003326          CNT     004206
COUNT   004210           CR     = 000015         EHLT    003274
ENDCK   003120           ENDMSG  004176          ENDXIT  003142
ERR    = 104000          ERRLP   003144          E1      003230
E2      003242           E3      003254          E4      003264
E5      003276           G      = 000002         GB     = 000003
GDEXIT  003666           GDMSG   004150          GDTYP   003300
GETNUM  003574           GETOCT  004012          GETSPS  003722
GET.1   003604           GOOD    004212          HALF    004214
INPCNT  004216           LERCHK  003354          LF     = 000012
NOP    = 000240          OCTSUB  004062          OS.1    004104
OS.2    004114           PCMSG   004116          PS     = 177776
PSMSG   004142           RDYPCV  004234          RDYPSV  004236
SR     = 177570          STEPS   004220          STPMSG  004165
TEMP    004222           TEST1   000600          TEST2   002000
TEST3   002100           TKB    = 177562         TKS    = 177560
TM1     003374           TM2     003420          TM3     003430
TOTAL   004224           TO.1    003500          TO.2    003552
TO.3    003556           TPB    = 177566         TPS    = 177564
TTIN    003674           TTI.1   003700          TTO     003434
TTO.1   003450           TTO.2   003464          TTO.3   003466
TYPMSG  003372           TYPOCT  003474          T1INT1  003102
T1INT2  003110           T1LPCK  001654          T1.1    000612
T1.1LP  000642           T1.2    000646          T1.2LP  000706
T1.3    000712           T1.3A   000774          T1.3AL  001032
T1.3LP  000756           T1.4    001036          T1.4AL  001154
T1.4LP  001104           T1.4WT  001110          T1.5    001160
T1.5AL  001274           T1.5LP  001224          T1.5WT  001230
T1.6    001300           T1.6A   001360          T1.6AL  001422
T1.6LP  001354           T1.7    001426          T1.7LP  001532
T1.8    001536           T1.8LP  001650          T1.8WT  001576
T2A     002010           T2B     002040          T2C     002052
T2D     002060           T3BOX   002216          T3BRL   002156
T3BRD1  002174           T3B1    002252          T3B2    002304
T3B3    002336           T3P4    002362          T3INT   003116
T3LPCK  003030           T3LP1   003060          T3PLUS  002610
T3P1    002626           T3P2    002662          T3P3    002716
T3P4    002752           T3P5    003006          T3X     002414
T3X1    002442           T3X2    002476          T3X3    002532
T3X4    002566           VALUE   004226          XX     = 000000
XYCS    004230           XYDB    004232
. ABS.  004240      000
ERRORS DETECTED:  1
FREE CORE:  4306. WORDS
/EN:ABS,/EN:AMA<XY11.SRC
```

Computer
Special Systems
PARIS - FRANCE