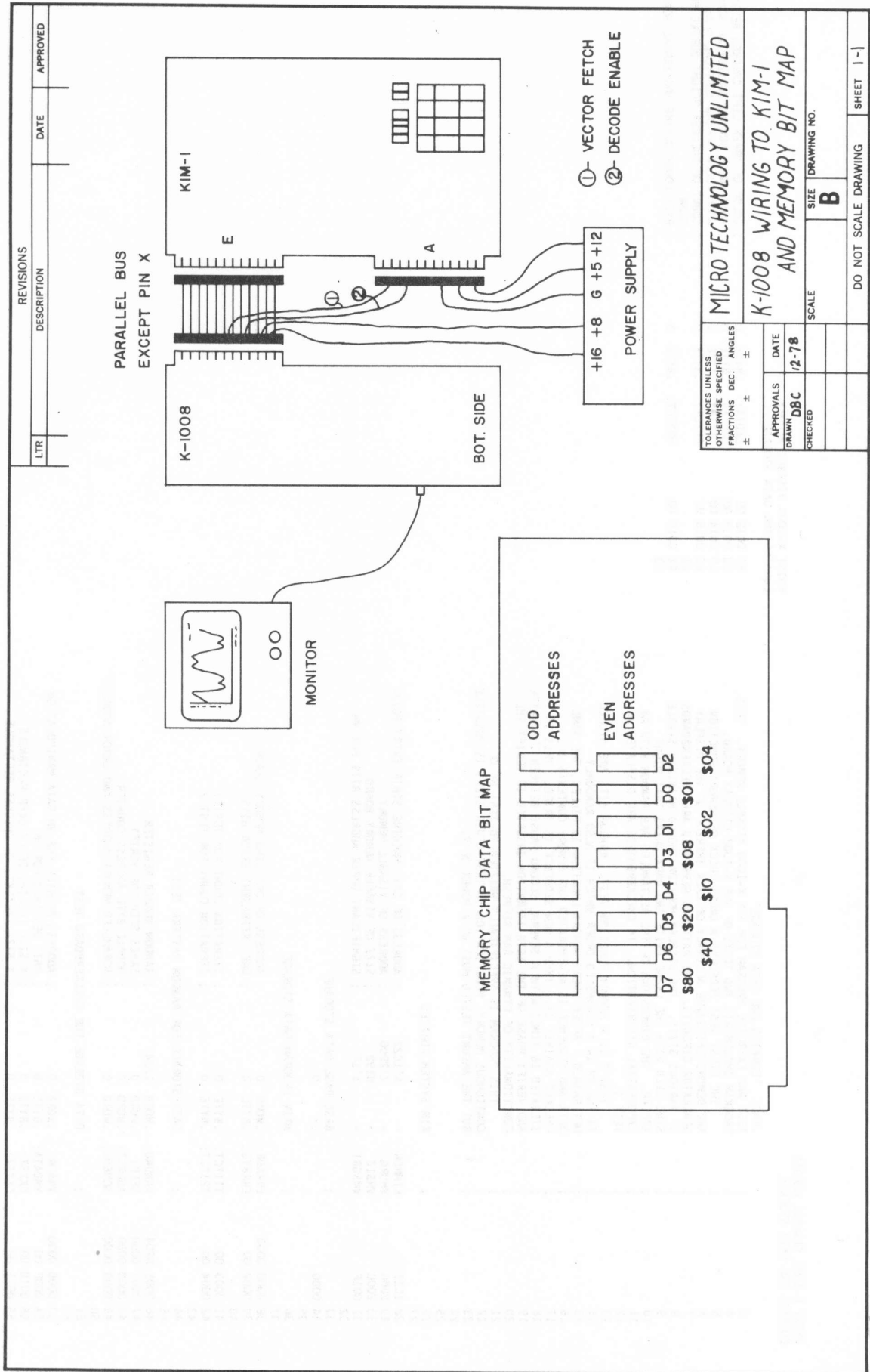




WMST K-1008 VISABLE MEMORY
EQUATES AND DATA STORAGE

3 ;
4 ;
5 ;
6 ;
7 ;
8 ;
9 ;
10 ;
11 ;
12 ;
13 ;
14 ;
15 ;
16 ;
17 ;
18 ;
19 ;
20 ;
21 ;
22 ;
23 ;
24 ;
25 ;
26 ;
27 ;
28 1C22 ; KIMMON ; ADDRESS OF SAVE MACHINE STATE ENTRY POINT
29 2000 ; VMORG ; ADDRESS OF VISABLE MEMORY
30 2000 ; VMSIZ ; SIZE OF VISABLE MEMORY BOARD
31 001F ; VMSGBT ; SIGNIFICANT UPPER ADDRESS BITS FOR VM
32 ;
33 ;
34 0000 ; BASE PAGE DATA STORAGE
35 ;
36 ;
37 ;
38 0000 0000 ; ERRADR: ; ADDRESS OF DETECTED MEMORY ERROR
39 0002 00 ; ERRBTS: ; ONES REPRESENT ERROR BITS
40 ;
41 0003 00 ; T1TCT: ; ITERATION COUNT FOR TEST 1
42 0004 00 ; T2TCT: ; ITERATION COUNT FOR TEST 2
43 ;
44 ;
45 ;
46 0005 D204 ; RANDNO: ; RANDOM NUMBER REGISTER
47 0007 0000 ; SEED: ; SAVES SEED FOR VERIFY
48 0009 0000 ; ADDRCT: ; DOUBLE BYTE ADDRESS COUNTER
49 0008 0000 ; SCHEMA: ; SCRAMBLED MEMORY ADDRESS AND ERROR ADDRES
50 ;
51 ;
52 ;
53 0000 0000 ; VMADR: ; ADDRESS POINTER FOR VM DATA MANIPULATION
54 000F 00 ; VMDATA: ; DATA DESTINED FOR VM
55 0010 00 ; CKXSZ: ; X SIZE (WIDTH) OF CHECKER RECTANGLE
56 0011 00 ; CKYSZ: ; Y SIZE (HEIGHT) OF CHECKER RECTANGLE

WMST K-1008 VISABLE MEMORY
EQUATES AND DATA STORAGE

57 0012 00 ; CKDTA: ; .BYTE 0
58 0013 00 ; CKDTAX: ; .BYTE 0
59 0014 00 ; CKDTAY: ; .BYTE 0
60 0015 00 ; CKCYCT: ; .BYTE 0
61 ;
62 0016 00 ; HBYTCT: ; .BYTE 0
63 ;

TEST AND EXERCISE PROGRAM FOR THE K-1008 VISABLE MEMORY. THIS PROGRAM IMPLEMENTS TWO TESTS OF THE K-1008 DISPLAY BOARD. THE FIRST TEST PERFORMS A GROSS CHECK OF MEMORY FUNCTION AND DEMONSTRATES THE ACCURACY OR LACK THEREOF OF THE DISPLAY GENERATOR CIRCUITS. THE PATTERNS GENERATED ARE CHECKERBOARDS OF VARIOUS SIZES. THE DISPLAY MONITOR SHOULD MAINTAIN STABLE SYNC EVEN WITH THE LARGE WHITE AND DARK AREAS THAT MIGHT APPEAR. THE DIMENSIONS OF THE RECTANGLES ARE RANDOM WITH AN EXPONENTIAL DISTRIBUTION. 16 CHECKERBOARDS ARE DISPLAYED IN TEST 1.
TEST 2 IS A MEMORY FUNCTION TEST. RANDOM BITS ARE STORED IN THE VM IN A SCRAMBLED ORDER WHICH IS ALSO RANDOMLY DETERMINED. AFTER EVERY MEMORY LOCATION IS FILLED, THE SAME DATA AND SEQUENCE IS REGENERATED AND MEMORY CONTENTS ARE CHECKED AGAINST IT. THEN A NEW SEQUENCE IS TRIED. THIS IS ITERATED 16 TIMES WITH A SEVERAL SECOND PAUSE BETWEEN THE WRITE AND VERIFY PHASE OF THE 16TH ITERATION INSERTED TO VERIFY THE FUNCTIONALITY OF DYNAMIC RAM REFRESH.
THIS PROGRAM IS SPECIFICALLY WRITTEN TO TEST 8K OF CONTIGUOUS MEMORY. MODIFICATION TO TEST OTHER SIZES IS POSSIBLE BUT THE AMOUNT TESTED MUST BE A POWER OF 2.

KIM SYSTEM EQUATES

= X'1C22 ; ADDRESS OF SAVE MACHINE STATE ENTRY POINT
= X'2000 ; ADDRESS OF VISABLE MEMORY
= 8192 ; SIZE OF VISABLE MEMORY BOARD
= X'1F ; SIGNIFICANT UPPER ADDRESS BITS FOR VM
;
; BASE PAGE DATA STORAGE
= 0

MAIN PROGRAM DATA STORAGE

WORD 0 ; ADDRESS OF DETECTED MEMORY ERROR
BYTE 0 ; ONES REPRESENT ERROR BITS
;
BYTE 0 ; ITERATION COUNT FOR TEST 1
BYTE 0 ; ITERATION COUNT FOR TEST 2
;
DATA STORAGE FOR RANDOM PATTERN TEST
;
WORD 1234 ; RANDOM NUMBER REGISTER
WORD 0 ; SAVES SEED FOR VERIFY
WORD 0 ; DOUBLE BYTE ADDRESS COUNTER
WORD 0 ; SCRAMBLED MEMORY ADDRESS AND ERROR ADDRES
;
DATA STORAGE FOR CHECKERBOARD TEST
;
WORD 0 ; ADDRESS POINTER FOR VM DATA MANIPULATION
BYTE 0 ; DATA DESTINED FOR VM
BYTE 0 ; X SIZE (WIDTH) OF CHECKER RECTANGLE
BYTE 0 ; Y SIZE (HEIGHT) OF CHECKER RECTANGLE

VMTST K-1008 VISIBLE MEMORY
MAIN TEST PROGRAM

```

64 0017      ;PAGE 'MAIN TEST PROGRAM'
65          ; START PROGRAM CODE AT 200
66 0200 A9E0 LDA #X'E0
67 0202 9A   TXS
68 0203 08   CLD
69          ;
70          ; TEST 1 16 CHECKERBOARD PATTERNS
71          ;
72 0204 A910 MAIN: LDA #16
73 0206 8503 STA T1T1CT
74 0208 208B03 JSR RNDEXP
75          ;
76 0208 8510 STA CKXSZ
77 0200 208B03 JSR RNDEXP
78 0210 8511 STA CKYSZ
79 0212 207103 JSR RAND
80 0215 A505 LDA RANDNO
81 0217 0A   ASLA
82 0218 A900 LDA #0
83 021A E900 SBC #0
84 021C 8512 STA CKOTA
85 021E 207F02 JSR CKGEN
86 0221 20D302 JSR CKVER
87 0224 D03F BNE CKERLG
88 0226 C603 DEC T1T1CT
89 0228 D0DE BNE MAIN1
90          ;
91          ; TEST 2 16 PASSES WITH RANDOM DATA, PAUSE IN 16TH PASS
92          ;
93 022A A90F MAIN10: LDA #15
94 022C 8504 STA T2T1CT
95 022E 207103 MAIN11: LDA RANDNO
96 0231 A505 LDA SEED
97 0233 8507 STA SEED+1
98 0235 A506 LDA RANDNO+1
99 0237 8508 STA SEED+1
100 0239 202903 JSR RNDGEN
101 023C A504 LDA T2T1CT
102 023E D011 BNE MAIN15
103 0240 A200 LDX #0
104 0242 A000 LDY #0
105 0244 A910 LDA #16
106 0246 18   CLC
107 0247 69FF ADC #1
108 0249 D0F8 BNE MAIN14
109 024B 88   DEY
110 024D D0F6 BNE MAIN13
111 024E CA   DEX
112 024F D0F1 BNE MAIN12
113 0251 A507 LDA SEED
114 0253 8505 STA RANDNO
115 0255 A508 LDA SEED+1
116 0257 8506 STA RANDNO+1
117 0259 204603 JSR RNDYER

```

VMTST K-1008 VISIBLE MEMORY
MAIN TEST PROGRAM

```

118 025C D014 BNE RNERLG
119 025E C604 DEC T2T1CT
120 0260 10CC BPL MAIN11
121 0262 4C0402 JMP MAIN
122          ;
123          ; ERROR LOG ROUTINES
124          ;
125          ; CKERLG:
126 0265 8502 STA ERRBTS
127 0267 A50D LDA YMAADR
128 0269 850D STA ERRADR
129 026B 850E LDA YMAADR+1
130 026D 850E STA ERRADR+1
131 026F 4C221C JMP KIMMON
132          ;
133 0272 8502 STA ERRBTS
134 0274 A50B LDA SCHEMA
135 0276 850C STA ERRADR
136 0278 A50C LDA SCHEMA+1
137 027A 850C STA ERRADR+1
138 027C 4C221C JMP KIMMON
139          ;

```

; GO TO ERROR LOG IF ERROR
; DECREMENT AND CHECK ITERATION COUNT
; LOOP UNTIL 16 ITERATIONS DONE
; REPEAT THE ENTIRE TEST WITH DIFFERENT
; DATA

; STORE ERROR BITS
; STORE ERROR ADDRESS

; GO TO KIM MONITOR

; STORE ERROR BITS
; STORE ERROR ADDRESS

; GO TO KIM MONITOR

WMST K-1008 VISABLE MEMORY
CHECKERBOARD PATTERN GENERATE

AND VERIFY ROUTINES

```
140 ; PAGE 'CHECKERBOARD PATTERN GENERATE AND VERIFY ROUTINES'  
141 ; CHECKERBOARD PATTERN GENERATOR  
142 ; STARTS AT UPPER LEFT CORNER OF SCREEN AND GENERATES A CHECKER-  
143 ; BOARD PATTERN  
144 ; ENTER WITH CKXSZ SET TO CHECKER SQUARE WIDTH AND CKYSZ SET TO  
145 ; CHECKER SQUARE HEIGHT AND CKDTA SET TO 0 FOR A BLACK UPPER LEFT  
146 ; SQUARE OR SET TO X'FF FOR A WHITE UPPER LEFT SQUARE.  
147 ; USES ALL REGISTERS, PRESERVES CKXSZ, CKYSZ, CKDTA  
148  
149 LDA #VMORG/256 ; INITIALIZE ADDRESS POINTER TO BEGINNING  
150 STA VMADR+1 ; OF VM  
151 LDA #0  
152 STA VMADR  
153 LDA #8  
154 STA CKDTA ; INITIALIZE BIT COUNT  
155 STA CKDTA ; COPY CKDTA TO VERTICAL WORK LOCATION  
156  
157 ; START A ROW OF CHECKER BLOCKS  
158  
159 CKGNV: LDA CKYSZ ; SET Y SIZE IN CKYCT  
160 STA CKYCT  
161  
162 ; START A HORIZONTAL SCAN  
163  
164 CKGNH: LDA CKDTA ; COPY VERTICAL CKDTA TO HORIZONTAL WORK  
165 STA CKDTAX ; LOCATION  
166 LDA #40 ; INITIALIZE COUNT OF BYTES GENERATED IN  
167 STA HBYTCT ; A HORIZONTAL SCAN  
168  
169 CKGNH1: LDX CKXSZ ; SET X SIZE IN INDEX X  
170 LDA CKDTAX ; GENERATE A DOT = TO CURRENT VALUE OF  
171 CKGNH2: ROLA CKDTAX ; CKDTAX  
172 ROL ; COUNT DOTS GENERATED  
173 DEY ; SKIP AHEAD IF NOT 8 YET  
174 LDA CKGNH4 ; STORE A COMPLETED BYTE IN VM  
175 VMDATA (VMADR),Y  
176 STA VMADR ; INCREMENT VM ADDRESS  
177 CKGNH3: BNE VMADR+1 ; TEST IF ENTIRE VM FILLED  
178 INC VMADR+1 ; JUMP OUT IF SO  
179 LDA CKGNF ; RESTORE 8 BIT COUNT  
180 CMP #VMORG+8192/256 ; TEST IF FINISHED WITH A HORIZONTAL SCAN  
181 BEQ CKGNF ; JUMP IF SO  
182 LDA #8 ; DECREMENT SQUARE WIDTH COUNT  
183 CKGNH3: DEC HBYTCT ; AT SQUARE BOUNDARY, FLIP COLOR  
184 BEQ CKGNV1 ; GO GENERATE NEXT DOT  
185 CKGNH4: BNE CKDTAX ; GO GENERATE NEXT DOT  
186 LDA CKDTAX ; FINISH VERTICAL SCAN  
187 STA CKDTA  
188 EOR #X'FF  
189 STA CKDTAX  
190 JMP CKGNV  
191  
192  
193
```

WMST K-1008 VISABLE MEMORY
CHECKERBOARD PATTERN GENERATE

AND VERIFY ROUTINES

```
194  
195 CKGNV1: DEC CKYCT ; DECREMENT SQUARE HEIGHT  
196 BNE CKGNH ; GO GENERATE NEXT LINE  
197 LDA CKDTAY ; AT SQUARE BOUNDARY, FLIP COLOR  
198 EOR #X'FF  
199 STA CKDTAY  
200 JMP CKGNV ; GO GENERATE NEXT LINE  
201  
202 CKGNF: RTS ; RETURN  
203  
204 ; CHECKERBOARD PATTERN VERIFY  
205 ; STARTS AT UPPER LEFT CORNER OF SCREEN AND VERIFIES A CHECKER-  
206 ; BOARD PATTERN.  
207 ; ENTER WITH CKXSZ SET TO CHECKER SQUARE WIDTH AND CKYSZ SET TO  
208 ; CHECKER SQUARE HEIGHT AND CKDTA SET TO 0 FOR A BLACK UPPER LEFT  
209 ; SQUARE OR SET TO X'FF FOR A WHITE UPPER LEFT SQUARE.  
210 ; USES ALL REGISTERS, PRESERVES CKXSZ, CKYSZ, CKDTA  
211 ; UPON FINDING A MISMATCH BETWEEN THE GENERATED PATTERN AND THE  
212 ; CONTENTS OF THE VM, RETURNS WITH THE ERROR BIT SET IN A AND  
213 ; THE ADDRESS OF THE ERROR IN VMADR AND VMADR+1  
214  
215 CKYER: LDA #VMORG/256 ; INITIALIZE ADDRESS POINTER TO BEGINNING  
216 STA VMADR+1 ; OF VM  
217 LDA #0  
218 STA VMADR  
219 LDA #8  
220 STA CKDTA ; INITIALIZE BIT COUNT  
221 STA CKDTA ; COPY CKDTA TO VERTICAL WORK LOCATION  
222  
223 ; START A ROW OF CHECKER BLOCKS  
224  
225 CKYVR: LDA CKYSZ ; SET Y SIZE IN CKYCT  
226 STA CKYCT  
227  
228 ; START A HORIZONTAL SCAN  
229  
230 CKYRH: LDA CKDTAY ; COPY VERTICAL CKDTA TO HORIZONTAL WORK  
231 STA CKDTAX ; LOCATION  
232 LDA #40 ; INITIALIZE COUNT OF BYTES GENERATED IN  
233 STA HBYTCT ; A HORIZONTAL SCAN  
234 CKYRH1: LDX CKXSZ ; SET X SIZE IN INDEX X  
235 CKYRH2: LDA CKDTAX ; GENERATE A DOT = TO CURRENT VALUE OF  
236 ROLA CKDTAX ; CKDTAX  
237 ROL ; COUNT DOTS GENERATED  
238 DEY ; SKIP AHEAD IF NOT 8 YET  
239 LDA CKYRH4 ; COMPARE THE COMPLETED BYTE WITH VM  
240 VMDATA (VMADR),Y  
241 EOR CKYERF ; CONTENTS  
242 INC VMADR ; JUMP IF ERROR  
243 CKYRH3: BNE VMADR+1 ; INCREMENT VM ADDRESS  
244 INC VMADR+1  
245 LDA VMADR+1 ; TEST IF ENTIRE VM FILLED  
246 CKYERF ; VMORG+8192/256  
247 BEQ CKYERF ; JUMP OUT IF SO  
248
```

```

VMTST K-1008 VISABLE MEMORY
CHECKERBOARD PATTERN GENERATE AND VERIFY ROUTINES

249 0309 A008      CKVRH3: LDY #8      ; RESTORE 8 BIT COUNT
250 0308 C616      DEC CKVRV1      ; TEST IF FINISHED WITH A HORIZONTAL SCAN
251 030D F00C      BEQ CKVRV1      ; JUMP IF SO
252 030F CA        DEX CKVRH4:      ; DECREMENT SQUARE WIDTH COUNT
253 0310 D00D      LDA CKVRH2:      ; GO GENERATE NEXT DOT
254 0312 A513      LDA CKOTAX      ; AT SQUARE BOUNDARY, FLIP COLOR
255 0314 49FF      EOR #X'FF      ;
256 0316 8513      STA CKOTAX      ;
257 0318 4CED02     JMP CKVRH1      ; GO GENERATE NEXT DOT
258
259
260
261 0318 C615      ; FINISH VERTICAL SCAN
262 031D D0C6      CKVRV1: DEC CKYCT ; DECREMENT SQUARE HEIGHT
263 031F A514      LDA CKOTAY      ; GO GENERATE NEXT LINE
264 0321 49FF      EOR #X'FF      ; AT SQUARE BOUNDARY, FLIP COLOR
265 0323 8514      STA CKOTAY      ;
266 0325 4CE102     JMP CKVRV      ; GO GENERATE NEXT LINE
267
268 0328 60        CKVERF: RTS      ; RETURN
269

```

```

VMTST K-1008 VISABLE MEMORY
RANDOM PATTERN GENERATE AND VERIFY ROUTINES

270
271
272 0329 A900      ;PAGE 'RANDOM PATTERN GENERATE AND VERIFY ROUTINES'
273 0328 8509      RNDGEN: STA #0    ; RANDOM PATTERN STORED IN SCRAMBLED ORDER GENERATE ROUTINE
274 032D A920      LDA ADDRCT      ; INITIALIZE ADDRESS COUNTER
275 032F 850A      STA #VMSIZ/256  ; TO 8192
276 0331 207103    STORPH: JSR RAND ; GENERATE A RANDOM NUMBER
277 0334 205F03    JSR MADR      ; FORM A SCRAMBLED MEMORY ADDRESS
278 0337 A505      LDX #0         ; STORE A RANDOM BYTE
279 0339 A200      LDA (SCHEMA,X) ; INDIRECTLY THROUGH SCRAMBLED MEMORY
280 033B 8108      DEC ADDRCT     ; ADDRESS AT SCHEMA
281 033D C609      BNE STORPH     ; DECREMENT ADDRESS COUNTER
282 033F D0F0      DEC ADDRCT+1   ; AND LOOP IF NOT ZERO
283 0341 C60A      BNE STORPH     ;
284 0343 D0EC      RTS           ; RETURN WHEN DONE
285 0345 60
286
287
288
289
290 0346 A920      ; RANDOM PATTERN STORED IN SCRAMBLED ORDER VERIFY ROUTINE
291 0348 850A      RNDVER: LDA #VMSIZ/256 ; INITIALIZE ADDRESS COUNTER
292 034A 207103    STA ADDRCT+1
293 034D 205F03    VERFPH: JSR RAND ; GENERATE A RANDOM NUMBER
294 0350 A108      LDA MADR      ; FORM SCRAMBLED MEMORY ADDRESS
295 0352 4505      EOR RANDNO    ; GET DATA FROM MEMORY INDIRECTLY
296 0354 D008      BNE VERRET    ; THROUGH SCHEMA
297 0356 C609      DEC ADDRCT     ; GO RETURN ON UNEQUAL COMPARE
298 0358 D0F0      BNE VERFPH    ; DECREMENT ADDRESS COUNTER
299 035A C60A      DEC ADDRCT+1  ; AND LOOP IF NOT ZERO
300 035C D0EC      BNE VERFPH    ;
301 035E 60        VERRET: RTS    ; RETURN
302
303
304
305 035F A507      ; SCRAMBLED MEMORY ADDRESS FORMATION ROUTINE
306 0361 4509      ; USES ADDRCT AND SEED TO FORM A SCRAMBLED ADDRESS IN SCHEMA
307 0363 8508      MADR: LDA SEED  ; GET LOWER BYTE OF RANDOM NUMBER
308 0365 A508      EOR ADDRCT     ; EXCLUSIVE-OR WITH LOWER ADDRESS
309 0367 450A      LDA SEED+1    ; LOWER BYTE OF RESULT
310 0369 291F      AND ADDRCT+1  ; GET UPPER BYTE OF RANDOM NUMBER
311 036B 18        AND #VMSGBT   ; EXCLUSIVE-OR WITH UPPER ADDRESS
312 036C 6920      CLC           ; SAVE SIGNIFICANT BITS OF RESULT
313 036E 850C      ADC #VMDRG/256 ; ADD IN FIRST PAGE NUMBER OF BOARD
314 0370 60        STA SCHEMA+1  ; BEING TESTED
315
316
317
318
319
320
321
322
323 0371 A008      RAND: LDY #8    ; SET COUNTER FOR 8 RANDOM BITS

```

WMST K-1008 VISABLE MEMORY
RANDOM PATTERN GENERATE AND VERIFY ROUTINES

```

324 0373 A505      RAND1:  LDA  RANDNO
325 0375 4A        LSR   RANDNO
326 0376 4505      EOR   RANDNO
327 0378 4A        LSR   RANDNO
328 0379 4A        LSR   RANDNO
329 037A 4505      EOR   RANDNO
330 037C 4A        LSR   RANDNO
331 037D 4506      EOR   RANDNO+1
332 037F 4A        LSR   RANDNO+1
333 0380 4A        LSR   RANDNO+1
334 0381 4A        LSR   RANDNO+1
335 0382 4A        LSR   RANDNO+1
336 0383 2606      ROL   RANDNO+1
337 0385 2605      ROL   RANDNO
338 0387 88        DEY
339 0388 00E9      BNE  RAND1
340 038A 60        RTS
341
342      ;
343      ; EXPONENTIALLY DISTRIBUTED RANDOM NUMBER SUBROUTINE
344      ; RULES OF USE SAME AS RAND, 8 BIT RESULT RETURNED IN A
345      ; AN EXPONENTIAL DISTRIBUTION MEANS THAT THE PROBABILITY OF A
346      ; RESULT BETWEEN 10 AND 20 IS THE SAME AS THE PROBABILITY OF A
347      ; RESULT BETWEEN 100 AND 200.
348      ; NOTE THAT THE PROBABILITY OF A ZERO RESULT IS ZERO.
349 038B 207103     RNDXP:  JSR  RAND
350 038E 207103     JSR  RAND
351 0391 A505      LDA  RANDNO
352 0393 2907      AND  #7
353 0395 A8        TAY
354 0396 C8        INY
355 0397 A506      LDA  RANDNO+1
356 0399 88        DEY
357 039A F004      BEQ  RNDXP2
358 039C 4A        LSR   RNDXP2
359 039D 4C9903     JMP  RNDXP1
360 03A0 0900      ORA  #0
361 03A2 F0E7      BEQ  RNDXP
362 03A4 60        RTS
363
364 0000      .END
NO ERROR LINES

```

; EXCLUSIVE-OR BITS 3, 12, 14, AND 15
; OF SEED

; RESULT IS IN BIT 3 OF A
; SHIFT INTO CARRY

; SHIFT RANDNO LEFT ONE BRINGING IN CARRY
; TEST IF 8 NEW RANDOM BITS COMPUTED
; LOOP FOR MORE IF NOT
; RETURN

; EXPONENTIALLY DISTRIBUTED RANDOM NUMBER SUBROUTINE
; RULES OF USE SAME AS RAND, 8 BIT RESULT RETURNED IN A
; AN EXPONENTIAL DISTRIBUTION MEANS THAT THE PROBABILITY OF A
; RESULT BETWEEN 10 AND 20 IS THE SAME AS THE PROBABILITY OF A
; RESULT BETWEEN 100 AND 200.
; NOTE THAT THE PROBABILITY OF A ZERO RESULT IS ZERO.

; GET TWO NEW RANDOM BYTES
; CONVERT ONE OF THE BYTES TO A RANDOM
; VALUE BETWEEN 0 AND 7 AND PUT IN Y AS A
; SHIFT COUNT
; GET THE OTHER RANDOM NUMBER AND SHIFT IT
; RIGHT ACCORDING TO Y

; TEST FOR A ZERO RESULT
; PROHIBIT ZERO RESULTS
; RETURN