

CMR11

CMR11 HOST EXERCISER AH-F914A-MC
CZCMJAO FICHE 1 OF 1

MAY 1980
COPYRIGHT © 79.80
MADE IN USA



Table with multiple columns and rows of data, likely representing a host exerciser output or a data grid. The content is mostly illegible due to low contrast and blurring.



I D E N T I F I C A T I O N

B 1

SEQ 0001

PROGRAM CODE: AC-F912A-MC
PROGRAM NAME: CZCMJAO CMR1' HOST EXERCISER
DATE CREATED: 30-MAY-79
MODIFIED: 14-JAN-80
MAINTAINER: SPECIAL SYSTEMS KANATA
AUTHOR: DAVID G. WIENS

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSIDERED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1979, 1980 BY DIGITAL EQUIPMENT OF CANADA LIMITED.

TABLF OF CONTENTS

HEADING -----	SECTION -----
GENERAL	1.
REQUIREMENTS	2.
RESTRICTIONS	3.
CAUTIONS	4.
ASSUMPTIONS	5.
OPTIONS / CONTROLS	6.
LOADING	7.
STARTING / RESTARTING	8.
TEST DESCRIPTION	9.
START-UP	9.1
TEST SELECTOR	9.2
LOGIC TEST	9.3
PORT LOOP-BACK	9.4
HOST-REMOTE EXERCISE	9.5
MAINTENANCE ROUTINE	9.6
END OF PASS	9.7
ERRORS	10

1. GENERAL

THIS PROGRAM IS TO BE USED TO TEST THE COMPACT MICRO REMOTE (CMR-11) HOST HARDWARE AND TO BASICALLY CHECK COMMUNICATIONS BETWEEN THE HOST HARDWARE ON THE UNIBUS AND THE REMOTE HARDWARE.

THE PROGRAM IS DIVIDED INTO THREE PARTS, PLUS A FOURTH MULTI-PURPOSE 'MAINTENANCE' ROUTINE:

- HOST LOGIC TEST
- PORT LOOP-BACK TEST
- CMR-11 EXERCISE ROUTINE
- 'MAINTENANCE' ROUTINE

THE HOST LOGIC TEST IS INTENDED TO BE RUN WITH ALL REMOTE COMMUNICATION DISABLED OR DISCONNECTED AS IS THE PORT LOOP-BACK TEST. THE CMR-11 EXERCISE ROUTINE, HOWEVER, CAN BE RUN WITHOUT CHANGING ANY OF THE HARDWARE FROM ITS 'INSTALLED' CONDITION. FOR THE DETAILS OF EACH TEST, SEE THE ACCOMPANYING TEXT.

2. REQUIREMENTS

HARDWARE:

PDP-11 OR LSI-11
CMR-11 HOST MODULE ON THE UNIBUS (Q-BUS)
CONSOLE TERMINAL ON LINE.

OTHER:

THE PROCESSOR MAINDECS AND CONSOLE TERMINAL TESTS MUST HAVE BEEN RUN SUCCESSFULLY BEFORE RUNNING THIS PROGRAM.

3. RESTRICTIONS

(NONE)

4. CAUTIONS

IT IS NOT ADVISABLE TO RUN THE LOGIC TEST OR THE PORT LOOP-BACK TEST ON AN 'ON-LINE' SYSTEM WITH THE REMOTES ATTACHED AS 'ON-LINE' DEVICES MAY BE AFFECTED. THE CMR-11 EXERCISE ROUTINE, HOWEVER, MAY BE RUN UNDER THESE CONDITIONS.

COMMUNICATION PORTIONS OF THESE TESTS FROM HOST TO REMOTE AND REMOTE TO HOST MAY TAKE A CONSIDERABLE LENGTH OF TIME IF THE PORT BAUD RATES ARE SET LOW (300 BAUD).

CAUTION MUST BE EXERCISED WHEN USING THE 'MAINTENANCE' ROUTINE SO THAT FUNCTIONS ARE NOT STARTED WHICH WILL COMPROMISE ANY 'ON-LINE' SITUATIONS.

5. ASSUMPTIONS

5.1 IN THE PORT LOOP-BACK TEST, IT IS ASSUMED THAT WHEN THE OPERATOR TYPES <CR> INDICATING 'READY', HE HAS LOOPED A PORT FROM OUTPUT TO INPUT PHYSICALLY, AND HAS SELECTED THAT PORT TO BE TESTED.

5.2 IT IS ASSUMED, IN THE CMR-11 EXERCISE ROUTINE, THAT THERE ARE REMOTES ATTACHED (AT LEAST ONE) AND THAT THEY ARE 'REGULAR'. IF THIS IS NOT THE CASE, THE ROUTINE WILL BE IDLE.

6. OPTIONS / CONTROLS

6.1 SWITCHABLE OPTIONS (COMBINABLE)

SWR =	MEANS
-----	-----
100000	DON'T HALT AFTER ERRORS
040000	DON'T PRINT ERRORS
020000	LOOP ON ERRONEOUS ROUTINE
004000	INHIBIT PROGRAM ITERATIONS
002000	DON'T PRINT END OF PASS

6.2 CONTROL CHARACTERS

CNTRL-C	RESTART TEST SELECTOR
CNTRL-T	
CNTRL-S	SHOW 'SWR' OPTIONS
CNTRL-P	PROCEED
CNTRL-O	ENTER 'MAINTENANCE' DEBUG ROUTINE
CNTRL-F	SHOW CONSOLE FILL COUNT

6.3 NON-SWR PROCESSORS

ON PROCESSORS WITH NO PHYSICAL SWITCH REGISTER THE OPTIONS CAN BE EXAMINED OR CHANGED BY THE USE OF 'CNTRL-S' AND AT THE TIME OF AN ERROR HALT. THE PRINTOUT: SWR = XXXXXX : WILL SHOW THE OPTIONS AND THE OPERATOR CAN LEAVE THEM (<CR>) OR CHANGE THEM (YYYYYY<CR>), THEN PROCEED (CNTRL-P)

7. LOADING

THE PROGRAM IS SUPPLIED IN .APS FORMAT AND IS LOADED BY MEANS OF THE ABSOLUTE LOADER OR THE 'LOAD' COMMAND OF THE DIAGNOSTIC MONITOR XXDP.

8. STARTING / RESTARTING

STARTING ADDRESS :	200
RESTARTING ADDRESS :	204

9. TEST DESCRIPTION

9.1 START UP:

INTERFACE ADDRESS:	TYPE <CR> IF IT 164000	TYPE XXXXXX<CR> IF ITS XXXXXX
INTERFACE VECTOR:	TYPE <CR> IF ITS 170	TYPE XXX<CR> IF ITS XXX
INTERFACE PRIORITY:	TYPE <CR> IF ITS 4	TYPE X<CR> IF ITS X

IF THE PROCESSOR HAS NO SWITCH REGISTER, THE SWITCH OPTIONS WILL NOW BE SHOWN AND MAY BE CHANGED (AS IN #6).

9.2 TEST SELECTOR:

SELECT TEST (4 = HELP):

(HELP MESSAGE READS:)

SELECT ONE OF THE FOLLOWING:

CR OR 1 = LOGIC TEST
 2 = PORT LOOP-BACK TEST
 3 = CMR-11 EXERCISE
 4 = THIS HELP INFO
 CNTR.-C = TEST SELECT
 CNTRL-S = SWITCH OPTION SELECT
 CNTRL-F = CONSOLE FILL COUNT SELECT
 CNTRL-C = DEBUG ROUTINE WHICH ALLOWS
 LOADING SCPD AND STARTING
 FUNCTIONS AND EXAMINING THE
 RESULT.
 CNTRL-P - PROCEED FROM SW MONITOR

9.3 LOGIC TEST:

THE FOLLOWING TESTS ARE PERFORMED:

- TEST OF THE PWR-UP SEQUENCE
- TEST THE SCPD ADDRESS REG.
- TEST THE SCPD MEMORY
- TEST ALL OF THE CSR
- TEST INTERRUPT
- TEST PROGRAM INIT.
- CHECK SCPD ACCESS KEYS
- SIMPLE FUNCTIONS TEST

9.4 PORT LOOP-BACK TEST

THE PORT LOOP-BACK TEST USES THE PORT LOOP MAINTENANCE FUNCTION OF THE HOST FIRMWARE. THE TEST WILL RUN WITH ONE PORT OR WITH ALL PORTS SELECTED:

PORT # (<CR> = ALL):

THE OPERATOR MUST ENSURE THAT THE PORT(S) BEING TESTED IS(ARE) LOOPED FROM XMTR TO RCVR.

9.5 CMR-11 HOST-REMOTE EXERCISE

THIS ROUTINE STARTS A BACKGROUND SCAN AND WAITS FOR THE END OF A SCAN OF ALL REMOTES ON ALL PORTS ('THRU'). DURING THIS WAIT, IF ANY ALARMS OCCUR THE BELL ON THE CONSOLE WILL RING (4 TIMES PER ALARM).

AT THE COMPLETION OF THE SCAN, THE PROGRAM SHOWS THE PORT CONFIGURATION STATUS AS FOLLOWS:

PORT # 0 0 1 2 3 ...
PORT # 1 100 101 102 103 ...
ETC.

SHOWING EACH PORT ONLY IF THERE ARE REMOTES ON THE PORT WHICH ARE 'REGULAR'

FOLLOWING THIS PRINTOUT, THE 'DIAGNOSTIC' FUNCTION IS ATTEMPTED ON EACH REMOTE PRESENT.

DURING THE REMAINDER OF THE EXERCISE, ALL REMOTES WILL BE EXERCISED WITH THE 'DIAGNOSTIC' FUNCTION AND ALL ALARMS WILL BE INDICATED (BY BELLS) AND ANY CHANGES OF REMOTE STATUS WILL BE REPORTED:

*** REMOTE STATUS CHANGE ***

REMOTE #X
REGULAR
REMOTE STATUS NOW- IRREGULAR
YYYY

9.6 'MAINTENANCE' DEBUG ROUTINE

CAUTION SHOULD BE EXERCISED WHEN USING THIS ROUTINE SEE SECT. 4.

THIS ROUTINE OPERATES LIKE A SIMPLE 'ODT'. HOWEVER, IT ALLOWS EXAMINING AND/OR DEPOSITING ONLY IN THE CMR-11 SCRATCHPAD MEMORY. FOLLOWING ARE THE OPERATING CHARACTERS:

- XXXX/ (SLASH) OPEN AND DISPLAY CONTENTS OF THE SCPD ADDR 'XXXX'
- <CR> (RETURN) CLOSE THE CURRENT ADDRESS.
- <LF> (LINE FEED) CLOSE THE CURRENT ADDR. AND OPEN THE NEXT SEQUENTIAL ONE.
- XXXX<CR> DEPOSIT 'XXXX' IN THE CURRENTLY OPEN ADDRESS AND CLOSE IT.
- XXXX<LF> DEPOSIT 'XXXX' IN THE CURRENTLY OPEN ADDRESS, CLOSE IT AND OPEN THE NEXT SEQUENTIAL ONE.
- G SET 'STFUN' OR 'MAIN' TO START THE CURRENTLY LOADED FUNCTION
- P PROCEED OUT OF THE 'DE-BUG' ROUTINE AND RETURN TO THE SPOT WHICH WAS LEFT WHEN CNTRL-O WAS TYPED. (ANY PROMPTS WILL BE RE-ASKED).
- (CSR ADDR)/ ALLOWS EXAMINATION OF THE HOST CSR REGISTER ONLY. NO OTHER UNIBUS ADDRESSES MAY BE EXAMINED IN THIS FASHION.

AT THE END OF EACH PASS OF ANY OF THE THREE TESTS,
AN 'END PASS' MESSAGE IS PRINTED UNLESS INHIBITED BY
SWITCH OPTION BIT 10.

THE PRINTOUT IS IN THE FOLLOWING FORM:

END PASS # X ERR. CNT. = Y

WHERE:

X = THE PASS NUMBER IN DECIMAL

Y = THE TOTAL NUMBER OF ERRORS SINCE START (DECIMAL).

IF THE NUMBER OF ERRORS BECOMES LARGER THAN 64K,
THEN '** OVERFLOW **' WILL BE PRINTED INSTEAD.

IF SWITCH OPTION BIT 11 = 0, THIS MESSAGE WILL BE
PRINTED ONLY AFTER 10 (OCTAL) PASSES THROUGH WHICHEVER
TEST IS RUNNING. OTHERWISE, IT IS PRINTED EVERY PASS.

10. ERRORS

FOR THE MOST PART, ERROR PRINTOUTS ARE SELF-EXPLANATORY.
THE ERROR NUMBERS AND ADDRESSES CAN BE USED FOR REFERENCE TO
THE PROGRAM LISTING FOR FURTHER DETAILS.

1
2
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
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```
.TITLE CZCMJAO CMR-11 HOST EXERCISER
.SBTTL SYMBOLIC DEFINITIONS
.NLIST TTM
.LIST ME
```

```
;INTERNAL DEFINITIONS:
```

```
177776 PS = 177776
177570 HWSWR = 177570
026346 SSWR = SWREG
```

```
;REGISTER DEFINITIONS:
```

```
000000 R0 = %0
000001 R1 = %1
000002 R2 = %2
000003 R3 = %3
000004 R4 = %4
000005 R5 = %5
000006 SP = %6
000007 PC = %7
```

```
;BUS REQUEST DEFINITIONS:
```

```
000340 P7 = 340
000300 P6 = 300
000240 P5 = 240
000200 P4 = 200
000140 P3 = 140
000100 P2 = 100
000040 P1 = 40
000000 P0 = 0
```

```
;BIT DEFINITIONS:
```

```
100000 B15 = 100000
040000 B14 = 40000
020000 B13 = 20000
010000 B12 = 10000
004000 B11 = 4000
002000 B10 = 2000
001000 B09 = 1000
000400 B08 = 400
000200 B07 = 200
000100 B06 = 100
000040 B05 = 40
000020 B04 = 20
000010 B03 = 10
000004 B02 = 4
000002 B01 = 2
000001 B00 = 1
002000 ISP = BEGIN
```

```
;INITIAL STACK POINTER
```

```
.SBTTL MACRO DEFINITIONS
```

57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112

000001

N = 1 ;INITIAL ERROR NUMBER
;ERROR MACROS

.MACRO ERROR P,MSG ;***** ERROR P *****
BIT #B14,@SR
BNE .+20
MOV #MSG,R0
MOV #P,ERRNUM
JSR PC,ERR
N = N+1
.ENDM

.MACRO DATERR P,MSG ;***** ERROR P *****
BIT #B14,@SR
BNE .+20
MOV #MSG,R0
MOV #P,ERRNUM
JSR PC,DERR
N = N+1
.ENDM

.MACRO HLT
JSR PC,SWHLT
.ENDM

;PRINT MACRO (MSG ADDR IN R0)

.MACRO PNTM A ;PRINT MESSAGE
MOV #A,R0 ;POINTED TO BY A
JSR PC,TYPOUT
.ENDM

;SCOPE LOOP MACRO

.MACRO SCOPE X
JSR R5,SCPRTN
X
.ENDM

;INTER-PDP-11 COMPATABLE MOVE TO PS

.MACRO MTPS SRC,?LLL
MOV SRC,-(SP)
MOV #LLL,-(SP)
RTI

LLL: .ENDM

;REGISTER SAVE AND RESTORE MACROS

.MACRO REGSAV

```
113 JSR R5,REGSAV
114 .ENDM
115
116 .MACRO REGRES
117 JSR R5,REGRES
118 .ENDM
119
120 .SBTTL TRAP CATCHERS
121 .ENABLE ABS
122 = 0
123 .WORD .+2
124 .WORD 0
125 .WORD DVATS ;TRAP BAD DEVICE ADDRESSES
126 .WORD 340
127 .REPT 126.
128 .WORD .+2
129 .WORD 0
130 .ENDR
131 .WORD .+2
132 .WORD 0
133 .WORD .+2
134 .WORD 0
135 .WORD .+2
136 .WORD 0
137 .WORD .+2
138 .WORD 0
139 .WORD .+2
140 .WORD 0
141 .WORD .+2
142 .WORD 0
143 .WORD .+2
144 .WORD 0
145 .WORD .+2
146 .WORD 0
147 .WORD .+2
148 .WORD 0
149 .WORD .+2
150 .WORD 0
151 .WORD .+2
152 .WORD 0
153 .WORD .+2
154 .WORD 0
155 .WORD .+2
156 .WORD 0
157 .WORD .+2
158 .WORD 0
159 .WORD .+2
160 .WORD 0
161 .WORD .+2
162 .WORD 0
163 .WORD .+2
164 .WORD 0
165 .WORD .+2
166 .WORD 0
167 .WORD .+2
168 .WORD 0
```

169	000124	000126	.WORD	.+2
170	000126	000000	.WORD	0
171	000130	000132	.WORD	.+2
172	000132	000000	.WORD	0
173	000134	000136	.WORD	.+2
174	000136	000000	.WORD	0
175	000140	000142	.WORD	.+2
176	000142	000000	.WORD	0
177	000144	000146	.WORD	.+2
178	000146	000000	.WORD	0
179	000150	000152	.WORD	.+2
180	000152	000000	.WORD	0
181	000154	000156	.WORD	.+2
182	000156	000000	.WORD	0
183	000160	000162	.WORD	.+2
184	000162	000000	.WORD	0
185	000164	000166	.WORD	.+2
186	000166	000000	.WORD	0
187	000170	000172	.WORD	.+2
188	000172	000000	.WORD	0
189	000174	000176	.WORD	.+2
190	000176	000000	.WORD	0
191	000200	000202	.WORD	.+
192	000202	000000	.WORD	0
193	000204	000206	.WORD	.+2
194	000206	000000	.WORD	0
195	000210	000212	.WORD	.+2
196	000212	000000	.WORD	0
197	000214	000216	.WORD	.+2
198	000216	000000	.WORD	0
199	000220	000222	.WORD	.+2
200	000222	000000	.WORD	0
201	000224	000226	.WORD	.+2
202	000226	000000	.WORD	0
203	000230	000232	.WORD	.+2
204	000232	000000	.WORD	0
205	000234	000236	.WORD	.+2
206	000236	000000	.WORD	0
207	000240	000242	.WORD	.+2
208	000242	000000	.WORD	0
209	000244	000246	.WORD	.+2
210	000246	000000	.WORD	0
211	000250	000252	.WORD	.+2
212	000252	000000	.WORD	0
213	000254	000256	.WORD	.+2
214	000256	000000	.WORD	0
215	000260	000262	.WORD	.+2
216	000262	000000	.WORD	0
217	000264	000266	.WORD	.+2
218	000266	000000	.WORD	0
219	000270	000272	.WORD	.+2
220	000272	000000	.WORD	0
221	000274	000276	.WORD	.+2
222	000276	00CJ00	.WORD	0
223	000300	000302	.WORD	.+2
224	000302	000000	.WORD	0

225	000304	000306	.WORD	.+2
226	000306	000000	.WORD	0
227	000310	000312	.WORD	.+2
228	000312	000000	.WORD	0
229	000314	000316	.WORD	.+2
230	000316	000000	.WORD	0
231	000320	000322	.WORD	.+2
232	000322	000000	.WORD	0
233	000324	000326	.WORD	.+2
234	000326	000000	.WORD	0
235	000330	000332	.WORD	.+2
236	000332	000000	.WORD	0
237	000334	000336	.WORD	.+2
238	000336	000000	.WORD	0
239	000340	000342	.WORD	.+2
240	000342	000000	.WORD	0
241	000344	000346	.WORD	.+2
242	000346	000000	.WORD	0
243	000350	000352	.WORD	.+2
244	000352	000000	.WORD	0
245	000354	000356	.WORD	.+2
246	000356	000000	.WORD	0
247	000360	000362	.WORD	.+2
248	000362	000000	.WORD	0
249	000364	000366	.WORD	.+2
250	000366	000000	.WORD	0
251	000370	000372	.WORD	.+2
252	000372	000000	.WORD	0
253	000374	000376	.WORD	.+2
254	000376	000000	.WORD	0
255	000400	000402	.WORD	.+2
256	000402	000000	.WORD	0
257	000404	000406	.WORD	.+2
258	000406	000000	.WORD	0
259	000410	000412	.WORD	.+2
260	000412	000000	.WORD	0
261	000414	000416	.WORD	.+2
262	000416	000000	.WORD	0
263	000420	000422	.WORD	.+2
264	000422	000000	.WORD	0
265	000424	000426	.WORD	.+2
266	000426	000000	.WORD	0
267	000430	000432	.WORD	.+2
268	000432	000000	.WORD	0
269	000434	000436	.WORD	.+2
270	000436	000000	.WORD	0
271	000440	000442	.WORD	.+2
272	000442	000000	.WORD	0
273	000444	000446	.WORD	.+2
274	000446	000000	.WORD	0
275	000450	000452	.WORD	.+2
276	000452	000000	.WORD	0
277	000454	000456	.WORD	.+2
278	000456	000000	.WORD	0
279	000460	000462	.WORD	.+2
280	000462	000000	.WORD	0

281	000464	000466	.WORD	.+2
282	000466	000000	.WORD	0
283	000470	000472	.WORD	.+2
284	000472	000000	.WORD	0
285	000474	000476	.WORD	.+2
286	000476	000000	.WORD	0
287	000500	000502	.WORD	.+2
288	000502	000000	.WORD	0
289	000504	000506	.WORD	.+2
290	000506	000000	.WORD	0
291	000510	000512	.WORD	.+2
292	000512	000000	.WORD	0
293	000514	000516	.WORD	.+2
294	000516	000000	.WORD	0
295	000520	000522	.WORD	.+2
296	000522	000000	.WORD	0
297	000524	000526	.WORD	.+2
298	000526	000000	.WORD	0
299	000530	000532	.WORD	.+2
300	000532	000000	.WORD	0
301	000534	000536	.WORD	.+2
302	000536	000000	.WORD	0
303	000540	000542	.WORD	.+2
304	000542	000000	.WORD	0
305	000544	000546	.WORD	.+2
306	000546	000000	.WORD	0
307	000550	000552	.WORD	.+2
308	000552	000000	.WORD	0
309	000554	000556	.WORD	.+2
310	000556	000000	.WORD	0
311	000560	000562	.WORD	.+2
312	000562	000000	.WORD	0
313	000564	000566	.WORD	.+2
314	000566	000000	.WORD	0
315	000570	000572	.WORD	.+2
316	000572	000000	.WORD	0
317	000574	000576	.WORD	.+2
318	000576	000000	.WORD	0
319	000600	000602	.WORD	.+2
320	000602	000000	.WORD	0
321	000604	000606	.WORD	.+2
322	000606	000000	.WORD	0
323	000610	000612	.WORD	.+2
324	000612	000000	.WORD	0
325	000614	000616	.WORD	.+2
326	000616	000000	.WORD	0
327	000620	000622	.WORD	.+2
328	000622	000000	.WORD	0
329	000624	000626	.WORD	.+2
330	000626	000000	.WORD	0
331	000630	000632	.WORD	.+2
332	000632	000000	.WORD	0
333	000634	000636	.WORD	.+2
334	000636	000000	.WORD	0
335	000640	000642	.WORD	.+2
336	000642	000000	.WORD	0

337	000644	000646	.WORD	.+2
338	000646	000000	.WORD	0
339	000650	000652	.WORD	.+2
340	000652	000000	.WORD	0
341	000654	000656	.WORD	.+2
342	000656	000000	.WORD	0
343	000660	000662	.WORD	.+2
344	000662	000000	.WORD	0
345	000664	000666	.WORD	.+2
346	000666	000000	.WORD	0
347	000670	000672	.WORD	.+2
348	000672	000000	.WORD	0
349	000674	000676	.WORD	.+2
350	000676	000000	.WORD	0
351	000700	000702	.WORD	.+2
352	000702	000000	.WORD	0
353	000704	000706	.WORD	.+2
354	000706	000000	.WORD	0
355	000710	000712	.WORD	.+2
356	000712	000000	.WORD	0
357	000714	000716	.WORD	.+2
358	000716	000000	.WORD	0
359	000720	000722	.WORD	.+2
360	000722	000000	.WORD	0
361	000724	000726	.WORD	.+2
362	000726	000000	.WORD	0
363	000730	000732	.WORD	.+2
364	000732	000000	.WORD	0
365	000734	000736	.WORD	.+2
366	000736	000000	.WORD	0
367	000740	000742	.WORD	.+2
368	000742	000000	.WORD	0
369	000744	000746	.WORD	.+2
370	000746	000000	.WORD	0
371	000750	000752	.WORD	.+2
372	000752	000000	.WORD	0
373	000754	000756	.WORD	.+2
374	000756	000000	.WORD	0
375	000760	000762	.WORD	.+2
376	000762	000000	.WORD	0
377	000764	000766	.WORD	.+2
378	000766	000000	.WORD	0
379	000770	000772	.WORD	.+2
380	000772	000000	.WORD	0
381	000774	000776	.WORD	.+2
382	000776	000000	.WORD	0

383				
384			.SMTL	SYMBOLIC LISTING
385				
386		000200		
387	000200	000167	JMP	= 200
388	000204	012706	MOV	BEGIN #ISF,SP
389	000210	000005	RESET	
390	000212	000167	JMP	RESTR
391				
392		002000	.	- 2000


```

393
394 002000 012706 002000 BEGIN: MOV #ISP,SP ;INITIALIZE THE STACK
395 002004 000005 RESET ;CLEAR THE WORLD
396 002006 MTPS #P7 ;STOP INTERRUPTS
397 002006 012746 000340 MOV #P7,-(SP)
398 002012 012746 002020 MOV #64$,-(SP)
399 002016 000002 RTI
400 002020 64$:
401 002020 005067 024254 CLR DEVER
402 002024 105067 024220 CLRB LSIFLG ;PREPARE TO CHECK FOR LSI-11
403 002030 012737 015440 000004 MOV #LSTST,@#4
404 002036 012737 000340 000006 MOV #P7,@#6
405 002044 005737 177776 TST @#177776 ;CHECK FOR PSW BSET FLAG
406 002050 005067 024274 CLR SWRFLG
407 002054 012737 015450 000004 MOV #SRTST,@#4 ;SET UP TO TRAP IF NO SWR.
408 002062 012737 000340 000006 MOV #P7,@#6
409 002070 012767 177570 024246 MOV #HWSWR,SR ;SET UP TO CHECK IF ANY HD SWR
410 002076 005777 024242 TST @SR ;SEE IF IT'S THERE
411 002102 PNTM TSTHDR
412 002102 012700 017770 MOV #TSTHDR,RO ;PRINT MESSAGE
413 002106 004767 014320 JSR PC,TYPOUT ;POINTED TO BY TSTHDR
414 002112 PROMT: PNTM FRAD ;ASK FOR 1ST UNIBUS ADDR OF CMR-11
415 002112 012700 020344 MOV #FRAD,RO ;PRINT MESSAGE
416 002116 004767 014310 JSR PC,TYPOUT ;POINTED TO BY FRAD
417 002122 016767 024240 015216 MOV DEVADR,KBBUF ;LOAD DEFAULT ADDRESS
418 002130 012767 002112 024160 MOV #PROMT,PCONT ;LOAD RETURN ADDRESS
419 002136 004767 014332 JSR PC,INPKB ;GET KBD INPUT
420 002142 016767 015200 024216 MOV KBBUF,DEVADR ;REPLACE CMR-11 BASE ADDR
421
422 002150 026727 015172 164000 CMP KBBUF,#164000 ;IS IT WITHIN LIMIT?
423 002156 103006 BHS PRMT1 ;YES, TEST IT
424 002160 PNTM AGAIN ;NO, ERROR & ASK AGAIN
425 002160 012700 020371 MOV #AGAIN,RO ;PRINT MESSAGE
426 002164 004767 014242 JSR PC,TYPOUT ;POINTED TO BY AGAIN
427 002170 000167 177716 JMP PROMT
428 002174 012737 015466 000004 PRMT1: MOV #DVATST,@#4
429 002202 005777 024160 TST @DEVADR ;IS IT A GOOD ADDR?
430 002206 012737 015644 000004 MOV #HDER,@#4 ;SET UP FOR FURTHER TRAPS
431 002214 005037 000006 CLR @#6 ;ALLOW CLOCK TO FUNCTION
432 002220 PRMT2: PNTM DVECT ;YES, ASK FOR DEVICE VECTOR
433 002220 012700 020424 MOV #DVECT,RO ;PRINT MESSAGE
434 002224 004767 014202 JSR PC,TYPOUT ;POINTED TO BY DVECT
435 002230 016767 024130 015110 MOV VECTOR,KBBUF ;LOAD DEFAULT VECTOR
436 002236 012767 002220 024052 MOV #PRMT2,PCONT ;LOAD RETURN ADDRESS
437 002244 004767 014224 JSR PC,INPKB ;GET KBD INPUT
438 002250 016767 015072 024106 MOV KBBUF,VECTOR ;REPLACE CMR-11 VECTOR
439 002256 026727 015064 000776 CMP KBBUF,#776 ;IS IT WITHIN LIMITS?
440 002264 101406 BLOS PRMT3 ;YES, GET PRIORITY
441 002266 PNTM AGAIN ;NO, ERROR AND ASK AGAIN
442 002266 012700 020371 MOV #AGAIN,RO ;PRINT MESSAGE
443 002272 004767 014134 JSR PC,TYPOUT ;POINTED TO BY AGAIN
444 002276 000167 177716 JMP PRMT2
445 002302 PRMT3: PNTM DVPRIO ;ASK FOR CMR-11 PRIORITY
446 002302 012700 020450 MOV #DVPRIO,RO ;PRINT MESSAGE
447 002306 004767 014120 JSR PC,TYPOUT ;POINTED TO BY DVPRIO
448 002312 016767 024052 015026 MOV #KPRIO,KBBUF ;LOAD DEFAULT PRIORITY

```

```

449 002320 012767 002302 0c 70      MOV      #PRMT3,PCONT      ;LOAD RETURN ADDRESS
450 002326 004767 014142      JSR      PC,INPKB        ;GET KBD INPUT
451 002332 026727 015010 000007  CMP      KBBUF,#7        ;IS IT WITHIN LIMITS?
452 002340 003406      BLE      PRMT4          ;LOW ENOUGH, OKAY
453 002342      PNTM      AGAIN        ;NO, ERROR AND ASK AGAIN
454 002342 012700 020371      MOV      #AGAIN,RO      ;PRINT MESSAGE
455 002346 004767 014060      JSR      PC,TYPOUT      ;POINTED TO BY AGAIN
456 002352 000167 177724      JMP      PRMT3
457 002356 026727 014764 000004  PRMT4:  CMP      KBBUF,#4        ;HIGH ENOUGH?
458 002364 002006      BGE      PRMT5          ;YES, OKAY
459 002366      PNTM      AGAIN        ;NO, ERROR & ASK AGAIN
460 002366 012700 020371      MOV      #AGAIN,RO      ;PRINT MESSAGE
461 002372 004767 014034      JSR      PC,TYPOUT      ;POINTED TO BY AGAIN
462 002376 000167 177700      JMP      PRMT3
463 002402 006367 014740      PRMT5:  ASL      KBBUF
464 002406 006367 014734      ASL      KBBUF
465 002412 006367 014730      ASL      KBBUF
466 002416 006367 014724      ASL      KBBUF
467 002422 006367 014720      ASL      KBBUF
468 002426 016767 014714 023670  MOV      KBBUF,PRIO     ;SHIFT IT INTO PLACE
469 002434 004767 013140      JSR      PC,DEVGEN      ;LOAD NEW PRIORITY
470                                     ;GENERATE CMR-11 ADDRESSES
471                                     ;CLEAR OUT ALL TABLES FOR START-UP
472
473 002440 005067 022540      CLR      EXTBL1
474 002444 005067 022534      CLR      EXTBL2
475 002450 005067 023130      CLR      EXTBL3
476 002454 005067 023324      CLR      EXTBL4
477 002460 012737 015644 000004  RESTRT: MOV      #HDR,#4        ;RESET TRAPS
478
479 002466 012737 002532 000100  MOV      #CLKINT,#100   ;SET CLOCK VECTOR
480 002474 012737 000340 000102  MOV      #340,#102
481 002502      MTPS      #P7          ;RAISE C.P. PRIORITY
482 002502 012746 000340      MOV      #P7,-(SP)
483 002506 012746 002514      MOV      #64$,-(SP)
484 002512 000002      RTI
485 002514      64$:
486
487 002514 005767 023630      TST      SWRFLG         ;NON-SWR-PROCESSOR?
488 002520 001402      BEQ      1$
489 002522 004767 013446      JSR      PC,SWDMP
490 002526 000167 000002 1$:      JMP      BCONT         ;YES, DUMP SWITCH LOCATION
491
492
493                                     ;LINE CLOCK SERVICE ROUTINE.
494                                     ; SIMPLY RETURN. CLOCK NOT USED.
495
496 002532 000002      CLKINT: RTI           ;RETURN FROM CLOCK INTERRUPT
497
498                                     .SBTTL SUPERVISOR DISPATCH
499
500
501 002534 000005      BCONT:  RESET
502 002536 105067 023513      CLR      SAFE
503 002542 105067 023503      CLR      D$FLG
504 002546      PNTM      TSTSEL
                                     ;ALWAYS DO RESET UPON ARRIVAL
                                     ;CLEAR 'FILL CNT SAFETY FLAG'
                                     ;CLEAR 'DEBUG' FLAG
                                     ;PRINT 'SELECT TEST (<CR>=HELP)''

```

```
505 002546 012700 020704      MOV      #TSTSEL,RO      ;PRINT MESSAGE
506 002552 004767 013654      JSR      PC,TYPOUT      ;POINTED TO BY TSTSEL
507 002556 012767 000001 014562  MOV      #1,KBBUF      ;DEFAULT LOGIC TEST
508 002564 012767 002534 023524  MOV      #BCONT,PCONT  ;LOAD RETURN ADDRESS
509 002572 004767 013676      JSR      PC,INPKB      ;GET KBD INPUT
510 002576 005067 023514      CLR      PCONT
511 002602 026727 014540 000001  CMP      KBBUF,#1      ;CHECK IT FOR '1'
512 002610 001422      BEQ      LOGTST        ;IF CLR, DO LOGIC TESTS
513 002612 026727 014530 000002  CMP      KBBUF,#2      ;CHECK IT FOR '2'
514 002620 001002      BNE      BCNT2
515 002622 000167 006016      JMP      PORTLP        ;IF '2' DO PORT LOOP TEST
516 002626 026727 014514 000003  BCNT2:  CMP      KBBUF,#3  ;CHECK IT FOR '3'
517 002634 001002      BNE      TSTRSK        ;IF NOT, GIVE HELP
518 002636 000167 007324      JMP      CMREXC        ;IF '3' DO EXERCISE.
519
520 002642      TSTRSK: PNTM      HELPM      ;PRINT HELPFUL INFO
521 002642 012700 020733      MOV      #HELM,RO      ;PRINT MESSAGE
522 002646 004767 013560      JSR      PC,TYPOUT      ;POINTED TO BY HELPM
523 002652 000167 177656      JMP      BCNT          ;AND RE-ASK
524
525
526      .SBT'L  HOST LOGIC TEST
527
528
529      ;CHECK POWER UP OF THE HOST MICRO FIRMWARE.
530
531 002656 005067 023476      LOGTST: CLR      VERFLG      ;CLEAR VERSION PRINT FLAG
532 002662 012767 000010 023414  MOV      #10,ITER      ;SET ITERATION COUNT
533 002670 005067 023416      CLR      PASCNT        ;CLEAR PASS COUNT
534 002674 105067 023354      CLR      OFLOF        ;CLEAR 'OVERFLOW' FLAG
535 002700 005067 023376      CLR      ERRCNT        ;CLEAR ERROR COUNT
536 002704      PNTM      LGKTSM      ;PRINT 'LOGIC TEST'
537 002704 012700 021543      MOV      #LGKTSM,RO    ;PRINT MESSAGE
538 002710 004767 013516      JSR      PC,TYPOUT      ;POINTED TO BY LGKTSM
539 002714 000005      STO:    RESET        ;INIT
540 002716 012767 000000 023350  MOV      #0,CNTR        ;SET UP TIMER
541 002724 116767 177774 177772  MOVB:  MOVB      MOVB,MOVB ;MOVB GIVE THE MICRO TIME
542 002732 116767 177766 177764      MOVB      MOVB,MOVB    ;MOVB TO GET GOING.
543 002740 116767 177760 177756      MOVB      MOVB,MOVB
544 002746 116767 177752 177750      MOVB      MOVB,MOVB
545 002754 116767 177744 177742      MOVB      MOVB,MOVB
546 002762 116767 177736 177734      MOVB      MOVB,MOVB
547 002770 032777 002000 023300  BIT      #B10,@CSR      ;IS 'PURUP' SET?
548 002776 001016      BNE      ST1          ;IF YES, OKAY
549 003000      ERROR      \N,NOGO    ;ERROR:HOST U.P. FAILED TO START.
550      ;***** ERROR 1 *****
551 003000 032777 040000 023336  BIT      #B14,@SR
552 003006 001007      BNE      .+20
553 003010 012700 022552      MOV      #NOGO,RO
554 003014 012767 000001 013042  MOV      #1,ERRNUM
555 003022 004767 012670      JSR      PC,ERR
556      =      N+1
557 003026      SCOPE      STO
558 003026 004567 012474      JSR      RS,SCRPTN
559 003032 002714      STO
560 003034 012767 102000 013024  ST1:    MOV      #102000,GOOD ;NOW WATCH FOR INTR
```

```
561 003042 017767 023230 013020 MOV @CSR,BAD
562 003050 042767 075777 013012 BIC #75777,BAD
563 003056 026767 013004 013004 CMP GOOD,BAD
564 003064 001421 BEQ ST2 ;INTR & PWRUP SET; OKAY
565 003066 005367 023202 DEC CNTR ;WAIT A BIT
566 003072 001360 BNE -T1
567 003074 DATERR \N,NOTUP ;ERROR:PWR UP SEQ NOT COMPLETED(BAD CSR)
568 ;***** ERROR 2 *****
569 003074 032777 040000 023242 BIT #B14,@SR
570 003102 001007 BNE .+20
571 003104 012700 022607 MOV #NOTUP,RO
572 003110 012767 000002 012746 MOV #2,ERRNUM
573 003116 004767 012660 JSR PC,DERR
574 000003 = N+1
575 003122 SCOPE ST0
576 003122 004567 012400 JSR R5,SCPRTN
577 003126 002714 ST0
578 003130 017767 023142 012732 ST2: MOV @CSR,BAD ;NOW CHK CSR FOR 0'S
579 003136 042767 102000 012724 BIC #102000,BAD ;EXCEPT PWRUP & INTR
580 003144 005067 012716 CLR GOOD
581 003150 026767 012712 012712 CMP GOOD,BAD
582 003156 001416 BEQ ST3 ;CSR = 0'S, O.K.
583 003160 DATERR \N,CSRRST ;ERROR:CSR NOT RESET
584 ;***** ERROR 3 *****
585 003160 032777 040000 023156 BIT #B14,@SR
586 003166 001007 BNE .+20
587 003170 012700 022660 MOV #CSRRST,RO
588 003174 012767 000003 012662 MOV #3,ERRNUM
589 003202 004767 012574 JSR PC,DERR
590 000004 = N+1
591 003206 SCOPE ST0
592 003206 004567 012314 JSR R5,SCPRTN
593 003212 002714 ST0
594 003214 012767 000020 023050 ST3: MOV #20,CLK ;COUNTER FOR VERSION ITEMS
595 003222 012777 001000 023110 MOV #1000,@SPAR ;ADDR RESULT SCPD
596 003230 012701 026204 MOV #VERSP,R1 ;POINT AT MEM SPACE
597 003234 117721 023102 ST4: MOV @SPDR,(R1)+ ;STORE VERSION INFO
598 003240 005367 023026 DEC CLK ;DONE?
599 003244 002403 BLT ST6
600 003246 005277 023066 INC @SPAR ;NO, KEEP GOING
601 003252 000770 BR ST4
602
603
604 ;SPAR REGISTER TEST:
605 ;DO A SLIDING BIT (0) TEST ON THE SCPD ADDRESS REGISTER
606 ;WHILE IGNORING SPAR BIT 08 (UNUSED).
607
608 003254 005077 023060 ST6: CLR @SPAR ;TEST CLEAR OF SPAR
609 003260 005067 012602 CLR GOOD
610 003264 017767 023050 012576 MOV @SPAR,BAD ;SPAR CLEAR?
611 003272 001416 BEQ ST60 ;YES, OK
612 003274 DATERR \N,SPRCLR ;ERROR:CAN'T CLEAR SPAR
613 ;***** ERROR 4 *****
614 003274 032777 040000 023042 BIT #B14,@SR
615 003302 001007 BNE .+20
616 003304 012700 022705 MOV #SPRCLR,RO
```

```

617 003310 012767 000004 012546      MOV      #4,ERRNUM
618 003316 004767 012460      JSR      PC,DERR
619          000005      N        =      N+1
620 003322          SCOPE  ST6
621 003322 004567 012200      JSR      R5,SCPRTN
622 003326 003254          ST6
623 003330 012767 177777 022756 ST60:  MOV      #-1,PAT          ;SET UP FOR SLIDING 0 TEST
624 003336 012767 000001 022744      MOV      #B00,MASK      ;SET BIT MASK
625 003344 016767 022744 012514 ST61:  MOV      PAT,GOOD
626 003352 042767 176000 012506      BIC      #176000,GOOD
627 003360 016777 012502 022752      MOV      GOOD,@SPAR      ;LOAD PATTERN
628 003366 017767 022746 012474      MOV      @SPAR,BAD        ;AND READ IT BACK
629 003374 042767 000400 012466      BIC      #B08,BAD
630 003402 042767 000400 012456      BIC      #B08,GOOD        ;IGNORE BIT 08
631 003410 026767 012452 012452      CMP      GOOD,BAD        ;SPAR DATA O.K. ?
632 003416 001416          BEQ      ST62            ;IF YES NEXT PATTERN
633 003420          DATERR  \N,SPRDAT      ;ERROR:BAD BITS IN SPAR
634          ;***** ERROR 5 *****
635 003420 032777 040000 022716      BIT      #B14,@SR
636 003426 001007          BNE      .+20
637 003430 012700 022732      MOV      #SPRDAT,R0
638 003434 012767 000005 012422      MOV      #5,ERRNUM
639 003442 004767 012334      JSR      PC,DERR
640          000006      N        =      N+1
641 003446          SCOPE  ST61
642 003446 004567 012054      JSR      R5,SCPRTN
643 003452 003344          ST61
644 003454 032767 001000 022632 ST62:  BIT      #B09,PAT          ;DONE WHOLE REG?
645 003462 001414          BEQ      ST7            ;IF YES, NEXT RTN
646 003464 012767 177777 022622      MOV      #-1,PAT          ;RE-LOAD ALL 1'S
647 003472 046767 022612 022614      BIC      MASK,PAT        ;NO, PREPARE FOR NEXT BIT
648 003500 006367 022604      ASL      MASK            ;ROTATE MASK
649 003504 004767 012432      JSR      PC,MONIT
650 003510 000167 177630      JMP      ST61            ;AND CONTINUE
651
652          ;SCRATCH PAD MEMORY TEST
653          ; (1) UP COUNT IN REQ SCPD WHILE FILLING RES SCPD WITH 377'S
654          ; (2) UP COUNT IN RES SCPD WHILE FILLING REQ SCPD WITH 377'S
655
656 003514 005077 022620      ST7:    CLR      @SPAR          ;ADDRESS 0
657 003520 005067 022636          CLR      WORD
658 003524 016777 022632 022610 ST70:  MOV      WORD,@SPDR      ;LOAD REQUEST SCRATCH PAD
659 003532 052777 001000 022600      BIS      #B09,@SPAR
660 003540 012777 000377 022574      MOV      #377,@SPDR
661 003546 042777 001000 022564      BIC      #B09,@SPAR
662 003554 027727 022560 000377      CMP      @SPAR,#377
663 003562 103005          CMPI    ST71            ;DONE.
664 003564 005277 022550          BHI     ST71
665 003570 105267 022566          INC     @SPAR
666 003574 000753          INCB   WORD
667 003576 005077 022536          BR     ST70
668 003602 005067 022554      ST71:  CLR      @SPAR          ;NOW...CHECK IT
669 003606 016767 022550 012252 ST80:  CLR      WORD
670 003614 017767 022522 012246      MOV      WORD,GOOD
671 003622 026767 012240 012240      MOV      @SPDR,BAD
672 003630 001432          CMP      GOOD,BAD
          BEQ      ST81            ;O.K. CONTINUE
  
```

```

673 003632          DATERR  \N,ISATE          ;ERROR:REQ SCPD ADR TEST ERROR
674                                     ;***** ERROR 6 *****
675 003632 032777 040000 022504          BIT      #B14,@SR
676 003640 001007          BNE      .+20
677 003642 012700 022761          MOV      #ISATE,RO
678 003646 012767 000006 012210          MOV      #6,ERRNUM
679 003654 004767 012122          JSR      PC,DERR
680                                     =      N+1
681 003660 032777 040000 022456          BIT      #B14,@SR          ;PRINT ALLOWED?
682 003666 001010          BNE      ST80S          ;NO, SKIP IT
683 003670          PNTM      SCPAD
684 003670 012700 021560          MOV      #SCPAD,RO          ;PRINT MESSAGE
685 003674 004767 012532          JSR      PC,TYP0UT          ;POINTED TO BY SCPAD
686 003700 017700 022434          MOV      @SPAR,RO          ;PRINT SCPD ADDRESS OF ERROR
687 003704 004767 013464          JSR      PC,OCTPNT
688 003710          ST80S: SCOPE      ST80
689 003710 004567 011612          JSR      R5,SCPRTN
690 003714 003606          ST80
691 003716 027727 022416 000377  ST81:  CMP      @SPAR,#377
692 003724 103005          BHS      ST82
693 003726 005277 022406          INC      @SPAR
694 003732 105267 022424          INCB     WORD
695 003736 000723          BR       ST80
696 003740 012777 001000 022372  ST82:  MOV      #1000,@SPAR
697 003746 012767 000377 012112          MOV      #377,GOOD
698 003754 017767 022362 012106  ST90:  MOV      @SPDR,BAD
699 003762 026767 012100 012100          CMP      GOOD,BAD
700 003770 001432          BEQ      ST91
701 003772          DATERR  \N,OSDTE          ;ERROR:RESULT SCPD DATA TEST ERROR
702                                     ;***** ERROR 7 *****
703 003772 032777 040000 022344          BIT      #B14,@SR
704 004000 001007          BNE      .+20
705 004002 012700 023012          MOV      #OSDTE,RO
706 004006 012767 000007 012050          MOV      #7,ERRNUM
707 004014 004767 011762          JSR      PC,DERR
708                                     =      N+1
709 004020 032777 040000 022316          BIT      #B14,@SR          ;PRINT ALLOWED?
710 004026 001010          BNE      ST90S          ;NO, SKIP IT
711 004030          PNTM      SCPAD
712 004030 012700 021560          MOV      #SCPAD,RO          ;PRINT MESSAGE
713 004034 004767 012372          JSR      PC,TYP0UT          ;POINTED TO BY SCPAD
714 004040 017700 022274          MOV      @SPAR,RO          ;PRINT SCPD ADDRESS OF ERROR
715 004044 004767 013324          JSR      PC,OCTPNT
716 004050          ST90S: SCOPE      ST90
717 004050 004567 011452          JSR      R5,SCPRTN
718 004054 003754          ST90
719 004056 027727 022256 001377  ST91:  CMP      @SPAR,#1377
720 004064 001403          BEQ      ST10
721 004066 005277 022246          INC      @SPAR
722 004072 000730          BR       ST90
723
724 004074 012777 001000 022236  ST10:  MOV      #B09,@SPAR          ;DO IT AGAIN WITH A SWITCH
725 004102 005067 022254          CLR      WORD
726 004106 016777 022250 022226  ST100: MOV      WORD,@SPDR          ;COUNT DOWN UNTIL HALF-WAY
727 004114 042777 001000 022216          BIC      #B09,@SPAR
728 004122 012777 000377 022212          MOV      #377,@SPDR          ;AND LOAD REQ SCPD WITH 1'S

```

729	004130	052777	001000	022202		BIS	#B09,@SPAR		
730	004136	027727	022176	001377		CMP	@SPAR,#1377		:DONE?
731	004144	103005				BHIS	ST101		:YES
732	004146	005277	022166			INC	@SPAR		:NO, KEEP GOING DOWN
733	004152	105267	022204			INCB	WORD		
734	004156	000753				BR	ST100		
735	004160	012777	001000	022152	ST101:	MOV	#B09,@SPAR		:NOW, ...CHECK IT
736	004166	005067	022170			CLR	WORD		
737	004172	016767	022164	011666	ST110:	MOV	WORD,GOOD		
738	004200	017767	022136	011662		MOV	@SPDR,BAD		
739	004206	026767	011654	011654		CMP	GOOD,BAD		
740	004214	001432				BEQ	ST111		:OUT ADDR OK?
741	004216					DATERR	\N,OSATE		:ERROR:RESULT SCPD ADR TEST ERROR
742									:***** ERROR 10 *****
743	004216	032777	040000	022120		BIT	#B14,@SR		
744	004224	001007				BNE	.+20		
745	004226	012700	023047			MOV	#OSATE,RO		
746	004232	012767	000010	011624		MOV	#10,ERRNUM		
747	004240	004767	011536			JSR	PC,DERR		
748		000011			N	=	N+1		
749	004244	032777	040000	022072		BIT	#B14,@SR		:PRINT ALLOWED?
750	004252	001010				BNE	ST110S		:NO, SKIP IT
751	004254					PNTM	SCPAD		
752	004254	012700	021560			MOV	#SCPAD,RO		:PRINT MESSAGE
753	004260	004767	012146			JSR	PC,TYP0UT		:POINTED TO BY SCPAD
754	004264	017700	022050			MOV	@SPAR,RO		:PRINT SCPD ADR OF ERROR
755	004270	004767	013100			JSR	PC,0CTPNT		
756	004274				ST110S:	SCOPE	ST110		
757	004274	004567	011226			JSR	R5,SCPRTN		
758	004300	004172				ST110			
759	004302	027727	022032	001377	ST111:	CMP	@SPAR,#1377		
760	004310	103005				BHIS	ST112		
761	004312	005277	022022			INC	@SPAR		
762	004316	105267	022040			INCB	WORD		
763	004322	000723				BR	ST110		
764	004324	012767	000377	011534	ST112:	MOV	#377,GOOD		:NOW CHECK REQ SCPD
765	004332	005077	022002			CLR	@SPAR		
766	004336	017767	022000	011524	ST120:	MOV	@SPDR,BAD		
767	004344	026767	011516	011516		CMP	GOOD,BAD		:ALL 1'S?
768	004352	001432				BEQ	ST121		:YES, CHECK THE REST
769	004354					DATERR	\N,ISDTE		:ERROR:REQ SCPD DATA TEST ERR
770									:***** ERROR 11 *****
771	004354	032777	040000	021762		BIT	#B14,@SR		
772	004362	001007				BNE	.+20		
773	004364	012700	023103			MOV	#1:DTE,RO		
774	004370	012767	000011	011466		MOV	#11,ERRNUM		
775	004376	004767	011400			JSR	PC,DERR		
776		000012			N	=	N+1		
777	004402	032777	040000	021734		BIT	#B14,@SR		:PRINT ALLOWED?
778	004410	001010				BNE	ST120S		:NO, SKIP IT
779	004412					PNTM	SCPAD		
780	004412	012700	021560			MOV	#SCPAD,RO		:PRINT MESSAGE
781	004416	004767	012010			JSR	PC,TYP0UT		:POINTED TO BY SCPAD
782	004422	017700	021712			MOV	@SPAR,RO		:PRINT SCPD ADR OF ERROR
783	004426	004767	012742			JSR	PC,0CTPNT		
784	004432				ST120S:	SCOPE	ST120		

```

785 004432 004567 011070      JSR      R5,SCRPTN
786 004436 004336              ST120
787 004440 027727 021674 000377 ST121:  CMP      @SPAR,#377      ;DONE REQ SCPD?
788 004446 001403              BEQ      ST13          ;YES, CHECK CLR SC. PAD
789 004450 005277 021664      INC      @SPAR
790 004454 000730              BR       ST120        ;NO, KEEP CHECKING
791
792                          ;CHECK CLEARING ALL SCRATCHPAD LOCATIONS
793                          ; FILL THE ENTIRE SCPD WITH A CONSTANT (NORMALLY '0')
794                          ; AND THE CHECK IT.
795
796 004456 005077 021656      ST13:   CLR      @SPAR      ;START AT 0
797 004462 012777 000000 021652 ST130:  MOV      #0,@SPDR    ;THIS LOC (#0) CAN BE CHANGED
798 004470 027727 021644 000377      CMP      @SPAR,#377    ; TO ANYTHING AND THIS RTN
799 004476 001403              BEQ      ST131A       ; CAN BE USED FOR ANY DATA
800 004500 005277 021634      INC      @SPAR        ; FILL-UP & TEST
801 004504 000766              BR       ST130
802 004506 012777 001000 021624 ST131A: MOV      #B09,@SPAR ;ADDR RES. SCPD
803 004514 012777 000000 021620 ST131B: MOV      #0,@SPDR  ;THIS '#0' MUST ALSO BE CHANGED
804 004522 027727 021612 001377      CMP      @SPAR,#1377
805 004530 001403              BEQ      ST131        ;FILL UP WHOLE RES. SCPD
806 004532 005277 021602      INC      @SPAR
807 004536 000766              BR       ST131B
808 004540 005077 021574      ST131:  CLR      @SPAR  ;NOW CHECK IT
809 004544 012767 000000 011314      MOV      #0,GOOD     ;THIS LOC (#0) MUST BE CHANGED
810 004552 017767 021564 011310 ST132:  MOV      @SPDR,BAD ; TO CORRESPOND TO ST130
811 004560 026767 011302 011302      CMP      GOOD,BAD    ; IN ORDER TO CHECK THE
812 004566 001432              BEQ      ST133       ; CORRECT DATA FILL-UP
813 004570              DATERR  \N,RQSPCL ;ERROR:REQ. SCPD NOT ALL 0'S
814                          ;***** ERROR 12 *****
815 004570 032777 040000 021546      BIT      #B14,@SR
816 004576 001007              BNE     .+20
817 004600 012700 023135      MOV      #RQSPCL,RO
818 004604 012767 000012 011252      MOV      #12,ERRNUM
819 004612 004767 011164      JSR      PC,DERR
820              000013      =      N+1
821 004616 032777 040000 021520      BIT      #B14,@SR    ;PRINT ALLOWED?
822 004624 001010              BNE     ST132S       ;NO, SKIP IT
823 004626              PNTM     SCPAD
824 004626 012700 021560      MOV      #SCPAD,RO   ;PRINT MESSAGE
825 004632 004767 011574      JSR      PC,TYP0UT  ;POINTED TO BY SCPAD
826 004636 017700 021476      MOV      @SPAR,RO   ;PRINT SCPD ADR OF ERROR
827 004642 004767 012526      JSR      PC,OC1PNT
828 004646              ST132S: SCOPE  ST13
829 004646 004567 010654      JSR      R5,SCRPTN
830 004652 004456              ST13
831 004654 027727 021460 000377 ST133:  CMP      @SPAR,#377    ;DONE CHECKING REQ. SCPD?
832 004662 001404              BEQ      ST134       ;YES, CHECK RES. SCPD
833 004664 005277 021450      INC      @SPAR        ;NO, FINISH UP
834 004670 000167 177656      JMP      ST132
835 004674 012777 001000 021436 ST134:  MOV      #1000,@SPAR ;NOW CHECK RES. SCPD
836 004702 012767 000000 011156      MOV      #0,GOOD     ;THIS '#0' ALSO MUST BE CHANGED
837 004710 017767 021426 011152 ST135:  MOV      @SPDR,BAD
838 004716 026767 011144 011144      CMP      GOOD,BAD    ;RES SCPD ALL 0'S?
839 004724 001432              BEQ      ST136
840 004726              DATERR  \N,RSSPCL ;ERROR:RES. SCPD NOT ALL 0'S
    
```



```

841                                     ;***** ERROR 13 *****
842 004726 032777 040000 021410      BIT    #B14,@SR
843 004734 001007                    BNE    .+20
844 004736 012700 023165              MOV    #RSSPCL,RO
845 004742 012767 000013 011114      MOV    #13,ERRNUM
846 004750 004767 011026              JSR    PC,DERR
847                                     =    N+1
848 004754 032777 040000 021362      BIT    #B14,@SR          ;PRINT ALLOWED?
849 004762 001010                    BNE    ST135S           ;NO, SKIP IT
850 004764                                PNTM   SCPAD
851 004764 012700 021560              MOV    #SCPAD,RO        ;PRINT MESSAGE
852 004770 004767 011435              JSR    PC,TYP0UT        ;POINTED TO BY SCPAD
853 004774 017700 021340              MOV    @SPAR,RO        ;PRINT SCPD ADDR OF ERROR
854 005000 004767 012370              JSR    PC,OC1PNT
855 005004                                ST135S: SCOPE  ST13
856 005004 004567 010576              JSR    R5,SCPRTN
857 005010 004456                                ST13
858 005012 027727 021322 001377  ST136: CMP    @SPAR,#1377      ;DONE RES. SCPD?
859 005020 001403                                BEQ    ST14             ;YES, NEXT ROUTINE
860 005022 005277 021312              INC    @SPAR
861 005026 000730              BR     ST135
862
863                                     ;TEST 'CLR TOP BYTE'
864                                     ; SETTING 'CLRTPB' SHOULD RESULT IN ALL BITS OF THE
865                                     ; HIGH BYTE OF THE CSR BEING CLEARED. (LOW BYTE ALREADY = 0)
866
867 005030 012777 000020 021240  ST14:  MOV    #B04,@CSR          ;SET 'CLRTPB'
868 005036 005067 011024                    CLR    GOOD             ;ALL CSR SHOULD BE 0
869 005042 017767 021230 011020      MOV    @CSR,BAD
870 005050 026767 011012 011012      CMP    GOOD,BAD        ;ALL CLEAR?
871 005056 001416                                BEQ    ST15
872 005060                                DATERR \N,CLTBER      ;ERROR:CLR TPB FAILED
873                                     ;***** ERROR 14 *****
874 005060 032777 040000 021256      BIT    #B14,@SR
875 005066 001007                    BNE    .+20
876 005070 012700 023215              MOV    #CLTBER,RO
877 005074 012767 000014 010762      MOV    #14,ERRNUM
878 005102 004767 010674              JSR    PC,DERR
879                                     =    N+1
880 005106                                SCOPE  ST14
881 005106 004567 010414              JSR    R5,SCPRTN
882 005112 005030                                ST14
883
884                                     ;NOW H A N G AND CHECK CSR
885                                     ; 'HANG' BY USING THE CMR-11 'DIE' FUNCTION
886                                     ; THEN CHECK THAT THE CSR WAS RIGHT
887                                     ; FOR THE 'DIE' FUNC. THEN CHECK ALL R/W BITS OF THE CSR
888
889 005114 005077 021220 021214  ST15:  CLR    @SPAR          ;ADDR REQ. SCPD
890 005120 012777 000213                    MOV    #213,@SPDR      ;LOAD MAINT FUNC SUBF #3
891 005126 005277 021206                    INC    @SPAR
892 005132 012777 000001 021202      MOV    #1,@SPDR        ;DATA LEN. = 1
893 005140 005277 021174                    INC    @SPAR
894 005144 012777 000001 021170      MOV    #1,@SPDR        ;ONE REQUEST
895 005152 012777 000200 021116      MOV    #200,@CSR       ;SET 'MAIN'SHD CAUSE U.P. HANG
896 005160 012767 007777 021106      MOV    #7777,CNTR     ;WAIT FOR THE MICRO
  
```

```

897 005166 005777 021104 1$: TST @CSR ;MICRO DONE?
898 005172 100403 BMI 2$ ;NO, WAIT A BIT
899 005174 005367 021074 DEC CNTR
900 005200 001372 BNE 1$
901 005202 012767 137600 010656 2$: MOV #137600,GOOD
902 005210 017767 021062 010652 MOV @CSR,BAD ;READ CSR REGISTER
903 005216 026767 010644 010644 CMP GOOD,BAD ;DID IT HANG CORRECTLY?
904 005224 001432 BEQ ST16 ;YES, GOOD
905 005226 012767 000000 021040 MOV #0,CNTR
906 005234 DATERR \N,DIERR ;ERROR:CSR NOT RIGHT FOR 'DIE'
907 ;***** ERROR 15 *****
908 005234 032777 040000 021102 BIT #B14,@SR
909 005242 001007 BNE .+20
910 005244 012700 023257 MOV #DIERR,RO
911 005250 012767 000015 010606 MOV #15,ERRNUM
912 005256 004767 010520 JSR PC,DERR
913 000016 N = N+1
914 005262 012777 000002 021006 MOV #2,@CSR ;PROGRAM INIT
915 005270 005367 021000 ST150: DEC CNTR ;WAIT FOR INIT
916 005274 001375 BNE ST150
917 005276 012777 000020 020772 MOV #B04,@CSR ;CLR TOP BYTE
918 005304 SCOPE ST15
919 005304 004567 010216 JSR R5,SCPRTN
920 005310 005114 ST15
921
922 ;CHECK CSR WITH MICRO HUNG
923
924 005312 012767 137755 010546 ST16: MOV #137755,GOOD ;CHECK R/W BITS OF CSR
925 005320 012777 000355 020750 MOV #355,@CSR
926 005326 017767 020744 010534 MOV @CSR,BAD
927 005334 026767 010526 010526 CMP GOOD,BAD ;DO THEY SET?
928 005342 001416 BEQ ST160 ;YES, TRY CLEARING THEM
929 005344 DATERR \N,CSRWE ;ERROR:BAD BITS IN CSR WHILE HUNG
930 ;***** ERROR 16 *****
931 005344 032777 040000 020772 BIT #B14,@SR
932 005352 001007 BNE .+20
933 005354 012700 023312 MOV #CSRWE,RO
934 005360 012767 000016 010476 MOV #16,ERRNUM
935 005366 004767 010410 JSR PC,DERR
936 000017 N = N+1
937 005372 SCOPE ST16
938 005372 004567 010130 JSR R5,SCPRTN
939 005376 005312 ST16
940 005400 012767 137400 010460 ST160: MOV #137400,GOOD ;NOW, SEE IF THEY CLEAR
941 005406 012777 000000 020662 MOV #0,@CSR
942 005414 017767 020656 010446 MOV @CSR,BAD
943 005422 026767 010440 010440 CMP GOOD,BAD ;DID THEY CLR?
944 005430 001416 BEQ ST161 ;YES, NEXT
945 005432 DATERR \N,CSRCE ;ERROR:CAN'T CLR LOW CSR WHILE HUNG
946 ;***** ERROR 17 *****
947 005432 032777 040000 020704 BIT #B14,@SR
948 005440 001007 BNE .+20
949 005442 012700 023344 MOV #CSRCE,RO
950 005446 012767 000017 010410 MOV #17,ERRNUM
951 005454 004767 010322 JSR PC,DERR
952 000020 N = N+1
  
```

```

953 005460          SCOPE ST160
954 005460 004567 010042 JSR  R5,SCPRTN
955 005464 005400          ST160
956 005466 012777 000200 020602 ST161: MOV  #200,@CSR           ;CHECK ACCESS KEYS TO SCPD
957 005474 012777 000200 020636 MOV  #200,@SPAR        ;WITH MAIN SET
958 005502 017767 020634 020652 MOV  @SPDR,WORD       ;READ AN REQ SCPD WORD
959 005510 005167 020646          COM  WORD              ;COMPLEMENT IT
960 005514 042767 177400 020640 BIC  #177400,WORD     ;FORGET TOP BYTE
961 005522 016777 020634 020612 MOV  WORD,@SPDR       ;WRITE COMPL. BACK
962 005530 027767 020606 020624 CMP  @SPDR,WORD       ;READ IT BACK AGAIN
963 005536 001016          BNE  ST162            ;IT SHOULDN'T BE THE SAME
964 005540          ERROR  \N,SPAVME      ;ERROR:ACCESSED REQ SCPD WITH 'MAIN' SET
965                                     ;***** ERROR 20 *****
966 005540 032777 040000 020576 BIT  #B14,@SR
967 005546 001007          BNE  .+20
968 005550 012700 023403          MOV  #SPAVME,RO
969 005554 012767 000020 010302 MOV  #20,ERRNUM
970 005562 004767 010130          JSR  PC,ERR
971                                     =  N+1
972 005566          SCOPE ST161
973 005566 004567 007734          JSR  R5,SCPRTN
974 005572 005466          ST161
975 005574 012777 000001 020474 ST162: MOV  #1,@CSR           ;TRY AGAIN WITH 'ST FUN' SET
976 005602 012777 000200 020530 MOV  #200,@SPAR
977 005610 017767 020526 020544 MOV  @SPDR,WORD       ;READ A REQ SCPD WORD
978 005616 005167 020540          COM  WORD              ;COMPLEMENT IT
979 005622 042767 177400 020532 BIC  #177400,WORD     ;FORGET TOP BYTE
980 005630 016777 020526 020504 MOV  WORD,@SPDR       ;WRITE IT BACK
981 005636 026777 020520 020476 CMP  WORD,@SPDR       ;READ IT BACK AGAIN
982 005644 001016          BNE  ST17              ;IT SHOULDN'T BE THE SAME
983 005646          ERROR  \N,SPAVFE      ;ERROR:ACCESSED REQ SCPD WITH 'STFUN' SET
984                                     ;***** ERROR 21 *****
985 005646 032777 040000 020470 BIT  #B14,@SR
986 005654 001007          BNE  .+20
987 005656 012700 023447          MOV  #SPAVFE,RO
988 005662 012767 000021 010174 MOV  #21,ERRNUM
989 005670 004767 010022          JSR  PC,ERR
990                                     =  N+1
991 005674          SCOPE ST162
992 005674 004567 007626          JSR  R5,SCPRTN
993 005700 005574          ST162
994
995 ;TEST INTERRUPT WITH MICRO HUNG
996
997 005702 012777 006014 020454 ST17: MOV  #ERINT,@VECTOR      ;SET UP VECTOR FOR ERROR
998 005710          MTPS  #0              ;ALLOW INTERRUPTS
999 005710 012746 000000          MOV  #0,-(SP)
1000 005714 012746 005722          MOV  #64$,-(SP)
1001 005720 000002          RTI
1002
1003 005722 000240          64$:  NOP
1004 005724 000240          NOP
1005 005726 012777 006056 020430 ST170: MOV  #GDINT,@VECTOR
1006 005734 012777 000100 020334 MOV  #100,@CSR
1007 005742 000240          NOP
1008 005744 000240          NOP
  
```

```
1009 005746          ERROR  \N,NOINTR          ;ERROR:HOST CAN'T INTERRUPT UNIBUS
1010                                     ;***** ERROR 22 *****
1011 005746 032777 040000 020370      BIT    #B14,@SR
1012 005754 001007          BNE    .+20
1013 005756 012700 023514          MOV    #NOINTR,R0
1014 005762 012767 000022 010074      MOV    #22,ERRNUM
1015 005770 004767 007722          JSR    PC,ERR
1016 000023          =    N+1
1017 005774 042777 000100 020274      BIC    #100,@CSR          ;CLEAR INTR ENAB
1018 006002          SCOPE ST170
1019 006002 004567 007520          JSR    R5,SCRPTN
1020 006006 005726          ST170
1021 006010 000167 000054      ST170R: JMP    INTCNT          ;CONTINUE
1022 006014 022626      ERINT:  CMP    (SP)+,(SP)+      ;POP-POP STACK
1023 006016          ERROR  \N,ERINTR          ;ERROR:ERRONEOUS HOST INTERRUPT
1024                                     ;***** ERROR 23 *****
1025 006016 032777 040000 020320      BIT    #B14,@SR
1026 006024 001007          BNE    .+20
1027 006026 012700 023552          MOV    #ERINTR,^
1028 006032 012767 000023 010024      MOV    #23,ERRNUM
1029 006040 004767 007652          JSR    PC,ERR
1030 000024          =    N+1
1031 006044          SCOPE ST17
1032 006044 004567 007456          JSR    R5,SCRPTN
1033 006050 005702          ST17
1034 006052 000167 177650          JMP    ST170          ;CONTINUE
1035 006056 012716 006010      GDINT:  MOV    #ST170R,(SP)      ;CHANGE RETURN ADDRESS
1036 006062 005077 020210          CLR    @CSR          ;CLEAR INTR ENAB
1037 006066 000002          RTI          ;CONTINUE
1038 006070 012777 006134 020266      INTCNT: MOV    #ERINT1,@VECTOR      ;SET UP FOR 2ND ERR INTR
1039 006076 042777 000100 020172      BIC    #100,@CSR          ;CLEAR INTER ENAB
1040 006104          MTPS  PRIO          ;SET C.P AT DEVICE PRIORITY
1041 006104 016746 020214          MO.   PRIO,-(SP)
1042 006110 012746 006116          MOV    #648,-(SP)
1043 006114 000002          RTI
1044 006116          648:
1045 006116 012777 000100 020152      MOV    #100,@CSR          ;SET INTR ENAB
1046 006124 000240          NOP          ;NO INTERRUPT SHOULD OCCUR
1047 006126 000240          NOP          ;THIS TIME.
1048 006130 000167 000046      INTCNR: JMP    ST18          ;O.K. CONTINUE
1049 006134 022626      ERINT:  CMP    (SP)+,(SP)+      ;POP-POP STACK
1050 006136 005077 020134          CLR    @CSR          ;CLEAR INTR ENAB
1051 006142          ERROR  \N,WPODI          ;ERROR:WRONG PRIORITY;DEVICE IS HIGHER
1052                                     ;***** ERROR 24 *****
1053 006142 032777 040000 020174      BIT    #B14,@SR
1054 006150 001007          BNE    .+20
1055 006152 012700 023605          MOV    #WPODI,R0
1056 006156 012767 000024 007700      MOV    #24,ERRNUM
1057 006164 004767 007526          JSR    PC,ERR
1058 000024          =    N+1
1059 006170          SCOPE INTCNT
1060 006170 000167 007332          JSR    R5,SCRPTN
1061 006174 006070          INTCNT
1062 006176 000167 000000          JMP    ST18
1063
1064          ;NOW CHECK PROGRAM IN:1.
```

```

1065
1066
1067 006202 012777 000002 020066 ST18: MOV #2,@CSR ;DO HOST "INIT"
1068 006210 016767 177774 177772 MOV: MOV MOV,MOV ;WAIT A LITTLE
1069 006216 016767 177766 177764 MOV MOV,MOV
1070 006224 016767 177760 177756 MOV MOV,MOV
1071 006232 016767 177752 177750 MOV MOV,MOV
1072 006240 016757 177744 177742 MOV MOV,MOV
1073 006246 016767 177736 177734 MOV MOV,MOV
1074 006254 012767 002000 007604 MOV #2000,GOOD ;ONLY PWR-UP SHOULD BE SET
1075 006262 017767 020010 007600 MOV @CSR,BAD ;READ CSR
1076 006270 026767 007572 007572 CMP GOOD,BAD ;CSR CORRECT AFTER INIT?
1077 006276 001416 BEQ ST180 ;YES, CONTINUE
1078 006300 DATERR \N,HITER ;ERROR:PROGRAM INIT FAILED
1079 ;***** ERROR 25 *****
1080 006300 032777 040000 020036 BIT #B14,@SR
1081 006306 001007 BNE .+20
1082 006310 012700 023646 MOV #HITER,RO
1083 006314 012767 000025 007542 MOV #25,ERRNUM
1084 006322 004767 007454 JSR PC,DERR
1085 000026 N = N+1
1086 006326 SCOPE ST18
1087 006326 004567 007174 JSR R5,SCRPTN
1088 006332 006202 ST18
1089 006334 032777 100000 017734 ST180: BIT #B15,@CSR ;WAIT FOR PWR-UP COMPL
1090 006342 001774 BEQ ST180
1091 006344 012777 001000 017766 MOV #1000,@SPAR ;NOW READY TO READ VERSION
1092 006352 012701 026204 MOV #VERS, R1 ;POINT AT ORIGINAL VERSION
1093 006356 012702 026226 MOV #VERSUN, R2 ;AND SAVE THIS VERSION
1094 006362 117712 017754 ST181: MOV B @SPDR, (R2)
1095 006366 001016 BNE ST182
1096 006370 ERROR \N,VISF ;ERROR:MICRO DIDN'T STORE VERSION STUFF
1097 ;***** ERROR 26 *****
1098 006370 032777 040000 017746 BIT #B14,@SR
1099 006376 001007 BNE .+20
1100 006400 012700 023706 MOV #VISF,RO
1101 006404 012767 000026 007452 MOV #26,ERRNUM
1102 006412 004767 007300 JSR PC,ERR
1103 000027 N = N+1
1104 006416 SCOPE ST180
1105 006416 004567 007104 JSR R5,SCRPTN
1106 006422 006334 ST180
1107 006424 111167 007436 ST182: MOV B (R1),GOOD ;NOW COMPARE VERSION WITH ORIG.
1108 006430 111267 007434 MOV B (R2),BAD
1109 006434 042767 177400 007424 BIC #177400,GOOD ;CLR TOP BYTE
1110 006442 042767 177400 007420 BIC #177400,BAD
1111 006450 026767 007412 007412 CMP GOOD,BAD ;SAME AS FIRST TIME?
1112 006456 001416 BEQ ST183
1113 006460 DATERR \N,VIDER ;ERROR:2ND VERSION NOT SAME AS 1ST
1114 ;***** ERROR 27 *****
1115 006460 032777 040000 017656 BIT #B14,@SR
1116 006466 001007 BNE .+20
1117 006470 012700 023746 MOV #VIDER,RO
1118 006474 012767 000027 007362 MOV #27,ERRNUM
1119 006502 004767 007274 JSR PC,DERR
1120 000030 N = N+1

```

```
1121 006506 SCOPE ST182
1122 006506 004567 007014 JSR R5,SCPRTN
1123 006512 006424 ST182
1124 006514 122122 ST183: CMPB (R1)+,(R2)+ ;UPDATE POINTERS
1225 006516 027727 017616 001020 CMP @SPAR,#1020 ;DONE?
1126 006524 001403 BEQ ST19
1127 006526 005277 017606 INC @SPAR ;NEXT WORD
1128 006532 000713 BR ST181
1129
1130
1131 ;CHECK ACCESS TO RESULT SCPD WITH INTERRUPT CLEAR
1132
1133
1134 006534 012777 000020 017534 ST19: MOV #20,@CSR ;CLEAR TOP BYTE OF CSR
1135 006542 012777 001200 017570 MOV #1200,@SPAR ;ADDRESS RESULT SCPD
1136 006550 017700 017566 017604 MOV @SPDR,WORD ;READ A WORD FROM IT
1137 006556 005160 017600 COM WORD ;COMPLEMENT IT
1138 006562 042767 177400 017572 BIC #177400,WORD ;IGNORE TOP 8 BITS
1139 006570 016777 017566 017544 MOV WORD,@SPDR ;AND WRITE IT BACK
1140 006576 026777 017560 017536 CMP WORD,@SPDR ;NOW READ IT BACK
1141 006604 001016 BNE ST20 ;IT SHOULDN'T BE THE SAME
1142 006606 ERROR \N,SPAVIC ;ERROR:RES. SCR. PAD ACCESSED WITH INTR CLR
1143 ;***** ERROR 30 *****
1144 006606 032777 040000 017530 BIT #B14,@SR
1145 006614 001007 BNE .+20
1146 006616 012700 024015 MOV #SPAVIC,R0
1147 006622 012767 000030 007234 MOV #30,ERRNUM
1148 006630 004767 007062 JSR PC,ERR
1149 000031 N = N+1
1150 006634 SCOPE ST19
1151 006634 004567 006666 SR R5,SCPRTN
1152 006640 006534 ST19
1153
1154
1155
1156 ;NOW CHECK SOME SIMPLE FUNCTIONALITY
1157 ; ENSURE THAT STARTING A 'PAINT' FUNCTION GIVES
1158 ; THE CORRECT ERROR STATUS.
1159 ; ENSURE THAT STARTING THE 'MAINT' FUNCTION WITH 'MAIN'
1160 ; WORKS PROPERLY AND GET PORT BAUD RATES.
1161
1162 006642 012767 000000 017424 ST20: MOV #0,CNTR
1163 006650 012777 000020 017420 MOV #20,@CSR ;CLEAR TOP BYTE
1164 006656 005077 017456 CLR @SPAR ;ADDRESS REQ SCPD
1165 006662 012777 000210 017452 MOV #210,@SPDR ;LOAD 'MAIN' FUNCTION
1166 006670 005277 017444 INC @SPAR ; TO EXAMINE HOST
1167 006674 012777 000004 017440 MOV #4,@SPDR ; MEMORY LOCATIONS
1168 006702 005277 017432 INC @SPAR
1169 006706 012777 000001 017426 MOV #1,@SPDR ;ONE REQUEST
1170 006714 005277 017420 INC @SPAR
1171 006720 012777 000002 017414 MOV #2,@SPDR ;# OF BYTES TO READ = 2
1172 006726 005277 017406 INC @SPAR
1173 006732 012777 000377 017402 MOV #377,@SPDR ;START ADDR = DFFF (H)
1174 006740 005277 017374 INC @SPAR
1175 006744 012777 000337 017370 MOV #337,@SPDR
1176 006752 012777 000001 017316 ST20: MOV #1,@CSR ;SET 'ST FUN' ON 'MAINT' FUN
```

```

1177 006760 012767 140000 007100      MOV      #B14+B15,GOOD      ;CHECK FOR FUNCOM & INTR
1178 006766 017767 017304 007074 ST201:  MOV      @CSR,BAD
1179 006774 026767 007066 007066      CMP      GOOD,BAD          ;'FUNCOM' AND 'INTR' SET?
1180 007002 001421                BEQ      ST202
1181 007004 005367 017264                DEC      CNTR                ;IF NOT, WAIT A BIT
1182 007010 001366                BNE     ST201
1183 007012                DATERR  \N,STESTF          ;ERROR:STATUS WRONG WITH 'STFUN' ON 'MAIN'
1184                                ;***** ERROR 31 *****
1185 007012 032777 040000 017324      BIT      #B14,@SR
1186 007020 001007                BNE     .+20
1187 007022 012700 024060                MOV     #STESTF,RO
1188 007026 012767 000031 007030      MOV     #31,ERRNUM
1189 007034 004767 006742                JSR    PC,DERR
1190                =          N+1
1191 007040                SCOPE  ST20
1192 007040 004567 006462                JSR    R5,SCPRTN
1193 007044 006642                ST20
1194 007046 012777 001000 017264 ST202:  MOV     #1000,@SPAR          ;ADDRESS RESULT SCPD
1195 007054 012767 000210 007004      MOV     #210,GOOD          ;CHECK ON FUNCTION
1196 007062 017767 017254 007000      MOV     @SPDR,BAD
1197 007070 026767 006772 006772      CMP     GOOD,BAD          ;FUNCTION CORRECT?
1198 007076 001416                BEQ     ST203
1199 007100                DATERR  \N,RSPEF          ;ERROR:RESULT FUNC. CODE WRONG ON MAIN
1200                                ;***** ERROR 32 *****
1201 007100 032777 040000 017236      BIT      #B14,@SR
1202 007106 001007                BNE     .+20
1203 007110 012700 024111                MOV     #RSPEF,RO
1204 007114 012767 000032 006742      MOV     #32,ERRNUM
1205 007122 004767 006654                JSR    PC,DERR
1206                =          N+1
1207 007126                SCOPE  ST20
1208 007126 004567 006374                JSR    R5,SCPRTN
1209 007132 006642                ST20
1210 007134 012777 001002 017176 ST203:  MOV     #1002,@SPAR          ;NOW ADDRESS 'USOF'
1211 007142 012767 000377 006716      MOV     #377,GOOD          ;SHOULD SHOW 'TOTAL FAILURE' CODE
1212 007150 017767 017166 006712      MOV     @SPDR,BAD
1213 007156 026767 006704 006704      CMP     GOOD,BAD          ;USOF O.K.?
1214 007164 001416                BEQ     ST204
1215 007166                DATERR  \N,USOF E1       ;ERROR:'STFUN' ON 'MAIN' GAVE BAD USOF
1216                                ;***** ERROR 33 *****
1217 007166 032777 040000 017150      BIT      #B14,@SR
1218 007174 001007                BNE     .+20
1219 007176 012700 024152                MOV     #USOF E1,RO
1220 007202 012767 000033 006654      MOV     #33,ERRNUM
1221 007210 004767 006566                JSR    PC,DERR
1222                =          N+1
1223 007214                SCOPE  ST20
1224 007214 004567 006306                JSR    R5,SCPRTN
1225 007220 006642                ST20
1226 007222 012767 000000 017044 ST204:  MOV     #0,CNTR
1227 007230 012777 000020 017040      MOV     #20,@CSR
1228 007236 012777 000200 017032      MOV     #200,@CSR          ;SET 'CLRTPB'
1229 007244 012767 140000 006611      MOV     #B14+B15,GOOD      ;SET 'MAIN' WITH SAME FUN LOADED
1230 007252 017767 017020 006610 ST205:  MOV     @CSR,BAD          ;CHECK FOR 'FUNCOM' & 'INTR'
1231 007260 026767 006602 006602      CMP     GOOD,BAD          ;FUN COMPLETE?
1232 007266 001421                BEQ     ST206
  
```

```

1233 007270 005367 017000          DEC      CNTR          ;WAIT A BIT
1234 007274 001366          BNE      ST205
1235 007276          DATERR  \N,STEMF    ;ERROR:STATUS WRONG ON 'MAIN' FUNC
1236          ;***** ERROR 34 *****
1237 007276 032777 040000 017040    BIT      #B14,@SR
1238 007304 001007          BNE      .+20
1239 007306 012700 024213          MOV      #STEMF,RO
1240 007312 012767 000034 006544    MOV      #34,ERRNUM
1241 007320 004767 006456          JSR      PC,DERR
1242          =          N+1
1243 007324          SCOPE    ST204
1244 007324 004567 006176          JSR      R5,SCRPTN
1245 007330 004222          ST204
1246 007332 012777 001002 017000    ST206: MOV      #1002,@SPAR    ;POINT AT 'USOF'
1247 007340 012767 000001 006520    MOV      #1,GOOD      ;SHOULD SHOW SUCCESS
1248 007346 017767 016770 006514    MOV      @SPDR,BAD
1249 007354 026767 006506 006506    CMP      GOOD,BAD
1250          BEQ      ST207    ;USOF O.K.?
1251 007364          DATERR  \N,USOFE2  ;ERROR:BAD 'USOF' ON 'MAIN' FUNCTION
1252          ;***** ERROR 35 *****
1253 007364 032777 040000 016752    BIT      #B14,@SR
1254 007372 001007          BNE      .+20
1255 007374 012700 024244          MOV      #USOFE2,RO
1256 007400 012767 000035 006456    MOV      #35,ERRNUM
1257 007406 004767 006370          JSR      PC,DERR
1258          =          N+1
1259 007412          SCOPE    ST204
1260 007412 004567 006110          JSR      R5,SCRPTN
1261 007416 007222          ST204
1262 007420 012777 001001 016712    ST207: MOV      #1001,@SPAR    ;POINT AT DATA LENGTH
1263 007426 012767 000003 006432    MOV      #3,GOOD      ;SHOULD BE 3
1264 007434 017767 016702 006426    MOV      @SPDR,BAD
1265 007442 026767 006420 006420    CMP      GOOD,BAD
1266 007450 001416          BEQ      ST21
1267 007452          DATERR  \N,DATLER  ;ERROR:'MAIN' FUNC GAVE BAD DATA LEN.
1268          ;***** ERROR 36 *****
1269 007452 032777 040000 016664    BIT      #B14,@SR
1270 007460 001007          BNE      .+20
1271 007462 012700 024301          MOV      #DATLER,RO
1272 007466 012767 000036 006370    MOV      #36,ERRNUM
1273 007474 004767 006302          JSR      PC,DERR
1274          =          N+1
1275 007500          SCOPE    ST204
1276 007500 004567 006022          JSR      R5,SCRPTN
1277 007504 007222          ST204
1278 007506 012700 026266 016620    ST21:  MOV      #BAUD,RO    ;POINT AT BAUD TABLE
1279 007512 012777 001003          MOV      #1003,@SPAR  ;GET RESULTS OF EXAMINE
1280 007520 117720 016616          MOV      @SPDR,(RO)+
1281 007524 005277 016610          INC      @SPAR
1282 007530 117710 016606          MOV      @SPDR,(RO)   ;STORE BAUD RATES
1283
1284          ;CHECK RESULTS OF STARTING A REGULAR FUNCTION WITH 'MAIN'
1285          ; TRY DOING AN 'ACTIVATE REMOTE' FIRST USING 'MAIN', THEN 'STFUN'
1286
1287
1288 007534 012777 000020 016534    ST22:  MOV      #20,@CSR    ;CLEAR TOP BYTE OF CSR

```


1289	007542	012777	000000	016570		MOV	#0,@SPAR	
1290	007550	012777	000060	016564		MOV	#60,@SPDR	;LOAD 'ACTIVATE REMOTE' FUNC
1291	007556	005277	016556			INC	@SPAR	
1292	007562	012777	000002	016552		MOV	#2,@SPDR	;DATA LEN. = 2
1293	007570	005277	016544			INC	@SPAR	
1294	007574	012777	000001	016540		MOV	#1,@SPDR	;ONE REQUEST
1295	007602	005277	016532			INC	@SPAR	
1296	007606	012777	000001	016526		MOV	#1,@SPDR	;REMOTE #1
1297	007614	012777	000200	016454	ST220:	MOV	#200,@CSR	;SET 'MAIN'
1298	007622	032777	040000	016446	ST221:	BIT	#B14,@CSR	;FUNC. COMPL?
1299	007630	001774				BEQ	ST221	;NO, WAIT
1300	007632	012777	001000	016500		MOV	#1000,@SPAR	;ADDRESS RESULT
1301	007640	017767	016476	006222		MOV	@SPDR,BAD	;READ RESULT FUNC. CODE
1302	007646	012767	000060	006212		MOV	#60,GOOD	
1303	007654	026767	006206	006206		CMP	GOOD,BAD	;O.K.?
1304	007662	001416				BEQ	ST222	
1305	007664					DATERR	\N,RSPEMF	;ERROR:RES. FUNC. CODE WRONG FOR 'MAIN'
1306								;***** ERROR 37 *****
1307	007664	032777	040000	016452		BIT	#B14,@SR	
1308	007672	001007				BNE	+.20	
1309	007674	012700	024341			MOV	#RSPEMF,RO	
1310	007700	012767	000037	006156		MOV	#37,ERRNUM	
1311	007706	004767	006070			JSR	PC,DEPR	
1312		000040			N	=	N+1	
1313	007712					SCOPE	ST22	
1314	007712	004567	005610			JSR	R5,SCPRTN	
1315	007716	007534				ST22		
1316	007720	012777	001002	016412	ST222:	MOV	#1002,@SPAR	;ADDR 'USOF'
1317	007726	012767	000377	006132		MOV	#377,GOOD	;SHOULD SHOW TOTAL FAILURE
1318	007734	017767	016402	006126		MOV	@SPDR,BAD	
1319	007742	026767	006120	006120		CMP	GOOD,BAD	;USOF O.K.?
1320	007750	001416				BEQ	ST223	
1321	007752					DATERR	\N,USOFE3	;ERROR:'MAIN' ON REG FUNC GAVE WRONG USOF
1322								;***** ERROR 40 *****
1323	007752	032777	040000	016364		BIT	#B14,@SR	
1324	007760	001007				BNE	+.20	
1325	007762	012700	024403			MOV	#USOFE3,RO	
1326	007766	012767	000040	006070		MOV	#40,ERRNUM	
1327	007774	004767	006002			JSR	PC,DEPR	
1328		000041			N	=	N+1	
1329	010000					SCOPE	ST22	
1330	010000	004567	005522			JSR	R5,SCPRTN	
1331	010004	007534				ST22		
1332	010006	012777	000020	016262	ST223:	MOV	#20,@CSR	;CLR TOP BYTE
1333	010014	012777	000001	016254		MOV	#1,@CSR	;SET 'STFUN'
1334	010022	032777	040000	016246	ST224:	BIT	#B14,@CSR	;FUNCOM SET?
1335	010030	001774				BEQ	ST224	;WAIT
1336	010032	012777	001002	016300		MOV	#1002,@SPAR	;ADDRESS USOF
1337	010040	012767	000364	006020		MOV	#364,GOOD	;SHOULD SHOW REM. ERROR
1338	010046	017767	016270	006014		MOV	@SPDR,BAD	
1339	010054	026767	006006	006006		CMP	GOOD,BAD	;USOF O.K.?
1340	010062	001416				BEQ	ST23	
1341	010064					DATERR	\N,USOFE4	;ERROR:WRONG USOF ON REGULAR FUNCTION
1342								;***** ERROR 41 *****
1343	010064	032777	040000	016252		BIT	#B14,@SR	
1344	010072	001007				BNE	+.20	

```
1345 010074 012700 024446      MOV      #USOF E4,R0
1346 010100 012767 000041 005756  MOV      #41,ERRNUM
1347 010106 004767 005670      JSR      PC,DERR
1348      000042      N      =      N+1
1349 010112      SCOPE  ST223
1350 010112 004567 005410      JSR      R5,SCRPTN
1351 010116 010006      ST223
1352
1353      :END OF LOGIC TEST. PRINT END OF PASS AND VERSION STUFF
1354
1355
1356 010120 005767 016234      ST23:   TST      VERFLG      ;IS FLAG SET?
1357 010124 001402      BEQ      ST23A      ;NO
1358 010126 000167 000350      JMP      ST24      ;YES, JUST PRINT 'END PASS'
1359 010132 012767 177777 01622C  ST23A:  MOV      #-1,VERFLG  ;NO, SET FLAG
1360 010140 012701 026226      MOV      #VERSUN,R1  ;GET VERSION DATA
1361 010144 012702 026266      MOV      #PAUD,R2   ;AND BAUD RATES
1362 010150      PNTM    VERNUM      ;PRINT 'VERSION '
1363 010150 012700 021601      MOV      #VERSM,R0   ;PRINT MESSAGE
1364 010154 004767 006252      JSR      PC,TYPOUI  ;POINTED TO BY VERSM
1365 010160 012767 000021 016106  MOV      #21,CNTR
1366 010166 112100      ST230:  MOVB    (R1)+,R0    ;DUMP ALL VERSION INFO
1367 010170 004767 007504      JSR      PC,IT0
1368 010174 005367 016074      DEC      CNTR
1369 010200 001372      BNE     ST230
1370 010202      PNTM    BAUDM      ;PRINT 'BAUD RATES:'
1371 010202 012700 021634      MOV      #BAUDM,R0   ;PRINT MESSAGE
1372 010206 004767 006220      JSR      PC,TYPCUT  ;POINTED TO BY BAUDM
1373 010212      PNTM    PORTOM     ;PRINT 'PORT 0 = '
1374 010212 012700 021652      MOV      #PORTOM,R0  ;PRINT MESSAGE
1375 010216 004767 006210      JSR      PC,TYPOUT  ;POINTED TO BY PORTOM
1376 010222 111267 016134      MOV      (R2),WORD  ;
1377 010226 042767 177770 016126  BIC      #177770,WORD ;SORT OUT CORRECT BAUD RATE
1378 010234 012703 025022      MOV      #BDRTBL,R3 ;R3 POINTS AT TABLE OF RATES
1379 010240 006367 016116      ASL     WORD        ;DOUBLE OFFSET
1380 010244 066703 016112      ADD     WORD,R3     ;USE CORRECT OFFSET
1381 010250 011300      MOV      (R3),R0    ;GET BAUD
1382 010252 004767 006154      JSR      PC,TYPOUT
1383 010256      PNTM    PORT1M     ;NOW PRINT 'PORT 1 = '
1384 010256 012700 021666      MOV      #PORT1M,R0  ;PRINT MESSAGE
1385 010262 004767 006144      JSR      PC,TYPOUT  ;POINTED TO BY PORT1M
1386 010266 112267 016070      MOV      (R2)+,WORD ;GET NEXT BAUD RATE CODE
1387 010272 042767 177617 016062  BIC      #177617,WORD ;SORT OUT CORRECT PART
1388 010300 006267 016056      ASR     WORD
1389 010304 006267 016052      ASR     WORD
1390 010310 006267 016046      ASR     WORD
1391 010314 006267 016042      ASR     WORD
1392 010320 012703 025022      MOV      #BDRTBL,R3 ;
1393 010324 006367 016032      ASL     WORD        ;DOUBLE OFFSET
1394 010330 066703 016026      ADD     WORD,R3     ;USE IT AS TABLE OFFSET
1395 010334 011300      MOV      (R3),R0    ;
1396 010336 004767 006070      JSR      PC,TYPOUT  ;PRINT RATE FOR PORT 1
1397 010342      PNTM    PORT2M     ;PRINT 'PORT 2 = '
1398 010342 012700 021702      MOV      #PORT2M,R0  ;PRINT MESSAGE
1399 010346 004767 006060      JSR      PC,TYPOUT  ;POINTED TO BY PORT2M
1400 010352 111267 016004      MOV      (R2),WORD  ;GET BAUD RATE CODE
```

```

1401 010356 042767 177770 015776 BIC #177770,WORD ;SORT OUT CORRECT PART
1402 010364 012703 025022 MOV #BDRTBL,R3
1403 010370 006367 015766 ASL WORD ;DOUBLE OFFSET
1404 010374 066703 015762 ADD WORD,R3 ;USE IT AS TABLE OFFSET
1405 010400 011300 MOV (R3),R0
1406 010402 004767 006024 JSR PC,TYPOUT ;PRINT BAUD RATE FOR PORT 2
1407 010406 PNTM PORT3M ;PRINT 'PORT 3 = '
1408 010406 012700 021716 MOV #PORT3M,R0 ;PRINT MESSAGE
1409 010412 004767 006014 JSR PC,TYPOUT ;POINTED TO BY PORT3M
1410 010416 111267 015740 MOVB (R2),WORD
1411 010422 042767 177617 015732 BIC #177617,WORD ;GET CORRECT PART.
1412 010430 006267 015726 ASR WORD
1413 010434 006267 015722 ASR WORD
1414 010440 006267 015716 ASR WORD
1415 010444 006267 015712 ASR WORD
1416 010450 012703 025022 MOV #BDRTBL,R3
1417 010454 006367 015702 ASL WORD ;DOUBLE OFFSET
1418 010460 066703 015676 ADD WORD,R3 ;USE IT AS TABLE OFFSET
1419 010464 011300 MOV (R3),R0
1420 010466 004767 005740 JSR PC,TYPOUT ;PRINT RATE FOR PORT 3
1421 010472 004767 007060 JSR PC,NULLS ;NULLS AFTER PRINTOUT
1422 010476 000167 000016 JMP ST25
1423
1424 010502 005367 015576 ST24: DEC ITER ;UPDATE ITERATION
1425 010506 001404 BEQ ST25 ;DONE 10?
1426 010510 032777 004000 015626 BIT #B11,@SR ;SINGLE ITERATION?
1427 010516 001415 BEQ ST26
1428 010520 012767 000010 015556 ST25: MOV #10,ITER
1429 010526 005267 015560 INC PASCNT ;UPDATE PASS COUNT
1430 010532 032777 002000 015604 BIT #B10,@SR ;END PASS INHIBIT?
1431 010540 001004 BNE ST26
1432 010542 004767 000014 JSR PC,FNPRTN ;PRINT END PASS INFO
1433 010546 004767 007004 JSR PC,NULLS ;NULLS AFTER PRINTOUT
1434 010552 004767 005364 ST26: JSR PC,MONIT ;CHECK FOR CONTROL CHARACTERS
1435 010556 000167 172132 JMP ST0 ;START OVER
1436
1437
1438 010562 ENPRTN: PNTM ENDP5 ;PRINT 'END PASS # '
1439 010562 012700 021732 MOV #ENDP5,R0 ;PRINT MESSAGE
1440 010566 004767 005640 JSR PC,TYPOUT ;POINTED TO BY ENDP5
1441 010572 016700 015514 MOV PASCNT,R0 ;GET PASS COUNT
1442 010576 004767 006676 JSR PC,DECPNT ;PRINT IT IN DECIMAL
1443 010602 PNTM ERCTM ;PRINT 'ERROR COUNT = '
1444 010602 012700 021747 MOV #ERCTM,R0 ;PRINT MESSAGE
1445 010606 004767 005620 JSR PC,TYPOUT ;POINTED TO BY ERCTM
1446 010612 105767 015436 TSTB OFLOF ;CHECK FOR ERROR CNT OVERFLOW
1447 010616 001405 BEQ EN01 ;IT DID'T, PRINT COUNT
1448 010620 PNTM OVFLM ;IT DID, PRINT 'OVERFLOW'
1449 010620 012700 021765 MOV #OVFLM,R0 ;PRINT MESSAGE
1450 010624 004767 005602 JSR PC,TYPOUT ;POINTED TO BY OVFLM
1451 010630 000404 BR ENRET
1452 010632 016700 015444 EN01: MOV ERRCNT,R0 ;GET ERROR COUNT
1453 010636 004767 006636 JSR PC,DECPNT ;PRINT IT IN DECIMAL
1454 010642 000207 ENRET: RTS
1455
1456 .SBTTL FORT LOOP-BACK TEST
  
```

```

1457
1458
1459
1460
1461
1462 010644 012767 000000 015422 PORTLP: MOV #0,CNTR ;BETTER CHECK FOR PWR-UP COMPL
1463 010652 027727 015420 102000 PRLC: CMP @CSR,#102000 ;POWER UP COMPLETE?
1464 010660 001420 BEQ POK ;YES, CARRY ON
1465 010662 005367 015406 DEC CNTR ;NO, WAIT A BIT
1466 010666 001371 BNE PRLC
1467 010670 ERROR \N,NOGO ;ERROR:HOST MICRO FAILED TO START
1468 ;***** ERROR 42 *****
1469 010670 032777 040000 015446 BIT #B14,@SR
1470 010676 001007 BNE .+20
1471 010700 012700 022552 MOV #NOGO,R0
1472 010704 012767 000042 005152 MOV #42,ERRNUM
1473 010712 004767 005000 JSR PC,ERR
1474 000043 N = N+1
1475 010716 HLT ;RUN LOGIC TEST.
1476 010716 004767 004564 JSR PC,SWHLT
1477 010722 012767 000010 015354 POK: MOV #10,ITER ;LOAD ITERATION COUNT
1478 010730 005067 015330 CLR ALLFLG ;CLEAR 'ALL' FLAG
1479 010734 005067 015352 CLR PASCNT ;CLEAR PASS COUNT
1480 010740 005067 015336 CLR ERRCNT ;CLEAR ERROR COUNT
1481 010744 105067 015304 CLRBF OFLOF ;CLR OVERFLOW FLAG
1482 010750 PNTM PLTM ;PRINT 'PORT LOOP-BACK TEST'
1483 010750 012700 022002 MOV #PLTM,R0 ;PRINT MESSAGE
1484 010754 004767 005452 JSR PC,TYPOUT ;POINTED TO BY PLTM
1485 010760 PSTO: PNTM PORTN ;PRINT 'PORT # (<CR>=ALL) '
1486 010760 012700 022030 MOV #PORTN,R0 ;PRINT MESSAGE
1487 010764 004767 005442 JSR PC,TYPOUT ;POINTED TO BY PORTN
1488 010770 012767 000004 006350 MOV #4,KBBUF ;DEFAULT IS 'ALL'
1489 010776 012767 010760 015312 MOV #PSTO,PCONT ;LOAD RETURN ADDRESS
1490 011004 004767 005464 JSR PC,INPKB ;GET KBD INPUT
1491 011010 005067 015302 CLR PCONT
1492 011014 026727 006326 000004 CMP KBBUF,#4 ;WAS IT DEFAULT (OR 4)?
1493 011022 001007 BNE PST1 ;NO
1494 011024 012767 177777 015232 MOV #-1,ALLFLG ;YES, SET 'ALL' FLAG
1495 011032 005067 015264 CLR PORT ;START WITH PORT 0
1496 011036 000167 000034 JMP PST2A
1497
1498 011042 101004 PST1: BHI PSTE ;WAS IT HIGHER THAN 4?
1499 011044 005067 015214 CLR ALLFLG ;NO, MUST BE A PORT #
1500 011050 000167 000014 JMP PST2
1501 011054 PSTE: PNTM AGAIN ;WRONG!, TRY AGAIN
1502 011054 012700 020371 MOV #AGAIN,R0 ;PRINT MESSAGE
1503 011060 004767 005346 JSR PC,TYPOUT ;POINTED TO BY AGAIN
1504 011064 000167 177670 JMP PSTO
1505 011070 016767 006252 015224 PST2: MOV KBBUF,PORT ;SAVE PORT #
1506 011076 012777 000020 015172 PST2A: MOV #20,@CSR ;CLR TOP BYTE
1507 011104 005077 015230 CLR @SPAR ;ADDRESS REQ. SCR. PAD
1508 011110 012777 000212 015224 MOV #212,@SPDR ;LOAD MAINT FUNC.
1509 011116 005277 015216 INC @SPAR ;TO DO PORT LOOP-BACK
1510 011122 012777 000023 015212 MOV #23,@SPDR ;DATA LEN. = 23
1511 011130 005277 015204 INC @SPAR
1512 011134 012777 000001 015200 MOV #1,@SPDR ;ONE REQUEST

```

1513	011142	005277	015172			INC	@SPAR	
1514	011146	016777	015150	015166		MOV	PORT,@SPDR	:LOAD PORT # (XMTR)
1515	011154	005277	015160			INC	@SPAR	
1516	011160	016777	015136	015154		MOV	PORT,@SPDR	:LOAD PORT # (RCVR)
1517	011166	012767	000020	015100		MOV	#20,CNTR	:CHARACTER COUNTER
1518	011174	012701	025144			MOV	#DATBL,R1	:LOOP DATA FROM 'DATBL'
1519	011200	005277	015134		PST3:	INC	@SPAR	
1520	011204	012177	015132			MOV	(R1)+,@SPDR	:LOAD DATA BYTE
1521	011210	005367	015050			DEC	CNTR	:LOADED ALL?
1522	011214	001371				BNE	PST3	:NO, KEEP LOADING
1523	011216				PSTS:	PNTM	TYPGR	:PRINT 'TYPE <CR> WHEN READY'
1524	011216	012700	022062			MOV	#TYPGR,RO	:PRINT MESSAGE
1525	011222	004767	005204			JSR	PC,TYPGR	:POINTED TO BY TYPGR
1526	011226	012767	011216	015062	PST4:	MOV	#PSTS,PCONT	:INDICATE PROMPT PENDING
1527	011234	012767	001236	006104		MOV	#1236,KBBUF	:LOAD ANY DEFAULT VALUE
1528	011242	004767	005226			JSR	PC,INPKB	:GET KEYBOARD INPUT
1529	011246	005067	015044			CLR	PCONT	
1530	011252	026727	006070	001236		CMP	KBBUF,#1236	:<CR> ONLY TYPED?
1531	011260	001356				BNE	PSTS	:IF NOT, KEEP LOOKING FOR IT
1532								
1533								
1534								:START MAINTENANCE FUNCTION
1535	011262	012777	000020	015006	PST5:	MOV	#20,@CSR	:CLR TOP BYTE
1536	011270	012777	000200	015000		MOV	#200,@CSR	:SET 'MAIN'
1537	011276	005777	014774		PST6:	TST	@CSR	:LOOK FOR INTERRUPT BIT
1538	011302	00403				BMI	PST60	
1539	011304	004767	004632			JSR	PC,MONIT	:AND WATCH FOR CNTRL CHARS.
1540	011310	000772				BR	PST6	
1541	011312	012777	001002	015020	PST60:	MOV	#1002,@SPAR	:NOW CHECK USOF
1542	011320	027727	015016	000001		CMP	@SPDR,#1	:DONE SUCCESSFULLY?
1543	011326	001004				BNE	PST7	:NO, CHECK ERROR TYPE
1544	011330	004767	004606			JSR	PC,MONIT	
1545	011334	000167	000430			JMP	ENDCHK	:YES, DONE A PASS
1546	011340	027727	014776	000375	PST7:	CMP	@SPDR,#375	:FUNCTION FAIL?
1547	011346	103434				BLO	PST8	
1548	011350					ERROR	W,PLFFE	:ERROR:TOTAL FAILURE OF 'MAIN' FUNCTION
1549								:***** ERROR 43 *****
1550	011350	032777	040000	014766		BIT	#B14,@SR	
1551	011356	001007				BNE	.+20	
1552	011360	012700	024503			MOV	#PLFFE,RO	
1553	011364	012767	000043	004472		MOV	#43,ERRNUM	
1554	011372	004767	004320			JSR	PC,ERR	
1555		000044			N	=	N+1	
1556	011376	032777	040000	014740		BIT	#B14,@SR	:PRINT ALLOWED?
1557	011404	001010				BNE	PST7S	:NO, SKIP IT
1558	011406					PNTM	PORTMS	:PRINT 'PORT # '
1559	011406	012700	022331			MOV	#PORTMS,RO	:PRINT MESSAGE
1560	011412	004767	005014			JSR	PC,TYPGR	:POINTED TO BY PORTMS
1561	011416	016700	014700			MOV	PORT,RO	
1562	011422	004767	005746			JSR	PC,OCPTNT	
1563	011426				PST7S:	SCOPE	PST5	
1564	011426	004567	004074			JSR	RS,SCRPTN	
1565	011432	011262				PST5		
1566	011434	000167	000330			JMP	ENDCHK	
1567	011440	027727	014676	000374	PST8:	CMP	@SPDR,#374	:XMTR HUNG UP?
1568	011446	103434				BLO	PST80	

1569	011450				ERROR	\N,PLTXTO		:ERROR:XMTR TIMEOUT ERROR
1570								:***** ERROR 44 *****
1571	011450	032777	040000	014666	BIT	#B14,@SR		
1572	011456	001007			BNE	.+20		
1573	011460	012700	024546		MOV	#PLTXTO,RO		
1574	011464	012767	000044	004372	MOV	#44,ERRNUM		
1575	011472	004767	004220		JSR	PC,ERR		
1576		000045			=	N+1		
1577	011476	032777	040000	014640	BIT	#B14,@SR		:PRINT ALLOWED?
1578	011504	001010			BNE	PST8S		:NO, SKIP IT
1579	011506				PNTM	PORTMS		:PRINT 'PORT # ''
1580	011506	012700	022331		MOV	#PORTMS,RO		:PRINT MESSAGE
1581	011512	004767	004714		JSR	PC,TYPOUT		:POINTED TO BY PORTMS
1582	011516	016700	014600		MOV	PORT,RO		
1583	011522	004767	005646		JSR	PC,OCTPNT		
1584	011526				PST8S:	SCOPE		
1585	011526	004567	003774		JSR	R5,SCRPTN		
1586	011532	011262			PST5			
1587	011534	000167	000230		JMP	ENDCHK		
1588	011540	027727	014576	000373	PST80:	MOV @SPDR,#373		:RCVR HUNG UP?
1589	011546	103434			BLO	PST9		
1590	011550				ERROR	\N,PLRCTO		:ERROR:RCVR TIMEOUT ERROR
1591								:***** ERROR 45 *****
1592	011550	032777	040000	014566	BIT	#B14,@SR		
1593	011556	001007			BNE	.+20		
1594	011560	012700	024574		MOV	#PLRCTO,RO		
1595	011564	012767	000045	004272	MOV	#45,ERRNUM		
1596	011572	004767	004120		JSR	PC,ERR		
1597		000046			=	N+1		
1598	011576	032777	040000	014540	BIT	#B14,@SR		:PRINT ALLOWED?
1599	011604	001010			BNE	PST80S		:NO, SKIP IT
1600	011606				PNTM	PORTMS		:PRINT 'PORT # ''
1601	011606	012700	022331		MOV	#PORTMS,RO		:PRINT MESSAGE
1602	011612	004767	004614		JSR	PC,TYPOUT		:POINTED TO BY PORTMS
1603	011616	016700	014500		MOV	PORT,RO		
1604	011622	004767	005546		JSR	PC,OCTPNT		
1605	011626				PST80S:	SCOPE		
1606	011626	004567	003674		JSR	R5,SCRPTN		
1607	011632	011262			PST5			
1608	011634	000167	000130		JMP	ENDCHK		
1609	011640	012777	001001	014472	PST9:	MOV #1001,@SPAR		:ADDRESS DATA TO FIND BAD WORD
1610	011646	067777	014470	014464	ADD	@SPDR,@SPAR		
1611	011654	017767	014462	004206	MOV	@SPDR,BAD		:GET BAD WORD
1612	011662	042777	001000	014450	BIC	#B09,@SPAR		
1613	011670	062777	000002	014442	ADD	#2,@SPAR		:GET GOOD WORD
1614	011676	017767	014440	004162	MOV	@SPDR,GOOD		:ERROR:LOOP TEST DETECTED DATA ERROR
1615	011704				DATERR	\N,PLDERR		:***** ERROR 46 *****
1616								
1617	011704	032777	040000	014432	BIT	#B14,@SR		
1618	011712	001007			BNE	.+20		
1619	011714	012700	024622		MOV	#PLDERR,RO		
1620	011720	012767	000046	004136	MOV	#46,ERRNUM		
1621	011726	004767	004050		JSR	PC,DERR		
1622		000047			=	N+1		
1623	011732	032777	040000	014404	BIT	#B14,@SR		:PRINT ALLOWED?
1624	011740	001010			BNE	PST9S		

```
1625 011742          PNTRM  PORTMS      ;PRINT 'PORT # ''
1626 011742 012700 022331      MOV    #PORTMS,R0 ;PRINT MESSAGE
1627 011746 004767 004460      JSR    PC,TYPOUT   ;POINTED TO BY PORTMS
1628 011752 016700 014344      MOV    PORT,R0
1629 011756 004767 005412      JSR    PC,OCTPNT
1630 011762          PST9S: SCOPE  PST5      ;PRINT #
1631 011762 004567 003540      JSR    R5,SCPRTN
1632 011766 011262          PST5
1633
1634
1635 011770 052777 000020 014300 ENDCHK: BIS    #B04,@CSR      ;CLR TOP BYTE
1636 011776 005767 014262          TST    ALLFLG      ;DOING ALL PORTS?
1637 012002 001441          BEQ    PNTEND      ;NO, DONE A PASS
1638 012004 026727 014312 000003      CMP    PORT,#3     ;YES, DONE ALL?
1639 012012 103416          BLO    ENDO        ;NO, KEEP GOING
1640 012014 005067 014302          CLR    PCRT       ;CLEAR PORT #
1641 012020 012777 000003 014312      MOV    #3,@SPAR    ; IN REQ. SCPD
1642 012026 016777 014270 014306      MOV    PORT,@SPDR
1643 012034 005277 014300          INC    @SPAR
1644 012040 016777 014256 014274      MOV    PORT,@SPDR
1645 012046 000417          BR     PNTEND
1646 012050 005267 014246          ENDO: INC    PORT   ;NEXT PORT
1647 012054 012777 000003 014256      MOV    #3,@SPAR   ;PREPARE TO LOAD NEXT PORT #
1648 012062 016777 014234 014252      MOV    PORT,@SPDR
1649 012070 005277 014244          INC    @SPAR
1650 012074 016777 014222 014240      MOV    PORT,@SPDR
1651 012102 000167 177154          JMP    PST5
1652
1653 012106 005367 014172          PNTEND: DEC   ITER  ;UPDATE ITERATION COUNT
1654 012112 001404          BEQ    PNT0
1655 012114 032777 004000 014222      BIT    #B11,@SR   ;SW 11 =, INH ITERATION
1656 012122 001415          BEQ    PNTRTN
1657 012124 012767 000010 014152      PNT0: MOV    #10,ITER ;RELOAD ITERATION COUNT
1658 012132 005267 014154          PNT1: INC    PASCNT ;UPDATE PASS COUNT
1659 012136 032777 002000 014200      BIT    #B10,@SR  ;SW 10 = INH END PASS PRINT
1660 012144 001004          BNE    PNTRTN
1661 012146 004767 176410          JSR    PC,ENPRTN  ;PRINT END PASS
1662 012152 004767 005400          JSR    PC,NULLS  ;PRINT NULLS
1663 012156 004767 003760          PNTRTN: JSR   PC,MONIT
1664 012162 000167 177074          JMP    PST5
1665
1666          .SBTTL  HOST-REMOTE EXERCISE
1667
1668
1669          ;START BACKGROUND SCAN, WAIT FOR 'THRU'
1670          ;REPORT ALARMS WHILE WAITING.
1671          ;WHEN 'THRU' SETS, DO A 'PORT CONFIGURATION STATUS' REPORT
1672          ;AND THEN RUN 'DIAGNOSTIC' FUNCTION ON ALL
1673          ;  AVAILABLE REMOTES
1674          ;  REPORT ALL CHANGES OF REMOTE STATUS
1675
1676
1677 012166          CMREXC: MTPS  #340      ;INTERRUPT OFF
1678 012166 012746 000340          MOV    #340,-(SP)
1679 012172 012746 012200          MOV    #648,-(SP)
1680 012176 000002          RTI
```

```

1681 012200
1682 012200 005067 014106
1683 012204 012767 000010 014072
1684 012212 105067 014036
1685 012216 005067 014060
1686 012222 012767 000000 014044
1687 012230 027727 014042 102000 CELP:
1688 012236 001420
1689 012240 005367 014030
1690 012244 001371
1691 012246
1692
1693 012246 032777 040000 014070
1694 012254 001007
1695 012256 012700 022552
1696 012262 012767 000047 003574
1697 012270 004767 003422
1698
1699 012274 N
1700 012274 004767 003206
1701 012300 005067 013764 EXS1:
1702 012304 005077 013766
1703 012310 012777 000020 013760
1704 012316 012777 000044 013752
1705 012324
1706 012324 012700 022112
1707 012330 004767 004076
1708 012334 105067 013720
1709 012340 105767 013715
1710 012344 005067 014002
1711 012350 005067 012630
1712 012354 005067 013024
1713 012360 005067 013220
1714 012364 005067 013414
1715
1716
1717 :EXERCISER MAIN LOOP
1718 012370 004767 000554 EXS2:
1719 012374 004767 003542 JSR PC,SCALM :GO SCAN ALARMS
1720 012400 032777 040000 013670 BIT #B14,@CSR :CHECK FOR CNTRL CHARS.
1721 012406 001410 BEQ EXS2B :IS FUN COMPL SET?
1722 012410 052777 000020 013660 BIS #20,@CSR :NO.
1723 012416 005767 013646 TST CHNGFL :CLR TOP BYTE
1724 012422 001022 BNE EXS3 :CHANGE FLAG SET?
1725 012424 004767 003512 JSR PC,MONIT :IF YES, DO PCS
1726 012430 105767 013625 EXS2B: TSTB THRU :CHECK FOR CNTRL CHARS
1727 012434 001015 BNE EXS3 :IS 'THRU' SET?
1728 012436 105767 013616 TSTB ONCE :IF YES, DO PORT CONFIG
1729 012442 001752 BEQ EXS2 :DONE PORT CONFIG ONCE?
1730 012444 005767 013702 TST TALLY :NO. WAIT FOR THRU
1731 012450 001747 BEQ EXS2 :YES, ANY REMOTES?
1732 012452 005767 013612 TST CHNGFL :NO. WAIT FOR CHANGE OR THRU
1733 012456 001004 BNE EXS3 :YES, ANY CHANGES PENDING?
1734 012460 004767 000222 JSR PC,DIAGFU :YES, NEW PORT CONFIG
1735 012464 000167 000400 JMP SKPD2 :NO. GO DO DIAG FUNC ON AVAIL REMOTES
1736 012470 004767 000072 EXS3: JSR PC,PRTCNF :THEN PRINT END PASS STUFF
:DO A PORT CONFIG FUNC.

```



```

1737 012474 005067 013570          CLR      CHNGFL      ;CLR CHANGE FLAG
1738 012500 112777 000074 013570  MOVB     #74,@CSR    ;FNAB BKG SCN,CHGST,THRU INTR
1739 012506 005767 013640          TST      TALLY      ;ANY REMOTES?
1740 012512 001406                BEQ      EXS3A       ;NO. IDLE
1741 012514 105767 013540          TSTB     ONCE        ;ALREADY DONE REPORT?
1742 012520 001014                BNE     EXS3B       ;YES, SKIP IT THIS TIME
1743 012522 004767 001314          JSR      PC,REPORT   ;REPORT ALL REGULAR REMOTES
1744 012526 000411                BR      EXS3B
1745 012530 105067 013524          EXS3A:  CLRB     ONCE        ;NO REMOTES. WAIT FOR THRU
1746 012534                PNTM     NOREM       ;PRINT 'NO REMOTES...WAITING'
1747 01254  012700 022223          MOV      #NOREM,RO  ;PRINT MESSAGE
1748 012540 004767 003666          JSR      PC,TYP0UT  ;POINTED TO BY NOREM
1749 012544 105067 013511          CLRB    THRUFL      ;CLEAR THRU FLAG
1750 012550 000707                BR      EXS2
1751 012552 105067 013503          EXS3B:  CLRB     THRUFL ;CLEAR THRU FLAG
1752 012556 112767 177777 013474  MOVB     #-1,ONCE   ;SET 'ONCE' FLAG
1753 012564 000701                BR      EXS2
1754
1755          ;PORT CONFIGURATION STATUS SUBROUTINE
1756          ;PERFORMS A P.C.S. ON ALL 4 PORTS AND SAVES THE
1757          ;REMOTE #'S IN TABLES
1758
1759 012566 005067 013476          PRTCNF: CLR      CHNGFL ;CLEAR 'CHANGE FLAG'
1760 012572 005067 013554          CLR      TALLY      ;CLEAR REMOTE TALLY
1761 012576 012767 025204 013550  MOV      #EXTBL1,TBLAD ;SET UP TO CONFIG PORT 0
1762 012604 012767 000000 013506  MOV      #0,PORN     ;
1763 012612 004767 000722                JSR      PC,PORCNF   ;DO PORT CONFIG ON PORT 0
1764 012616 012767 025404 013530  MOV      #EXTBL2,TBLAD ;
1765 012624 012767 000001 013466  MOV      #1,PORN     ;
1766 012632 004767 000702                JSR      PC,PORCNF   ;DO PORT CONFIG ON PORT 1
1767 012636 012767 025604 013510  MOV      #EXTBL3,TBLAD ;
1768 012644 012767 000002 013446  MOV      #2,PORN     ;
1769 012652 004767 000662                JSR      PC,PORCNF   ;DO PORT CONFIG ON PORT 2
1770 012656 012767 026004 013470  MOV      #EXTBL4,TBLAD ;
1771 012664 012767 000003 013426  MOV      #3,PORN     ;
1772 012672 004767 000642                JSR      PC,PORCNF   ;DO PORT CONFIG ON PORT 3
1773 012676 005767 013366          TST      CHNGFL     ;ANY CHANGES PENDING?
1774 012702 001331                BNE     PRTCNF      ;IF SO, DO IT ALL OVER
1775 012704 000207                RTS      PC
1776
1777
1778          ;SUBROUTINE TO DO THE DIAGNOSTIC FUNCTION ON
1779          ;ALL REMOTES WHICH ARE ON LINE AND REGULAR
1780
1781 012706 012701 025204          DIAGFU: MOV      #EXTBL1,R1 ;GET TABLE FOR PORT 0
1782 012712 012167 013356          MOV      (R1)+,CNTR ;GET # OF REMOTES ON PORT
1783 012716 001002                BNE     SKP1        ;NONE, NEXT PORT
1784 012720 000167 000016          JMP      SKP2
1785 012724 012167 013400          SKP1:  MOV      (R1)+,REMO ;GET 1ST REMOTE #
1786 012730 004767 001376          JSR      PC,DIAGNF  ;DO DIAGNOSTIC FUNC. ON REM.
1787 012734 005367 013334          DEC     CNTR        ;DO ALL REMOTES ON THIS PORT
1788 012740 001371                BNE     SKP1
1789
1790 012742 012701 025404          SKP2:  MOV      #EXTBL2,R1 ;GET TABLE FOR PORT 1
1791 012746 012167 013322          MOV      (R1)+,CNTR ;1ST ENTRY IS # OF REMOTES
1792 012752 001002                BNE     SKP3

```

```

1793 012754 000167 000016
1794 012760 012167 013344
1795 012764 004767 001342
1796 012770 005367 013300
1797 012774 001371
1798 012776 012701 025604
1799 013002 012167 013266
1800 013006 001002
1801 013010 000167 000016
1802 013014 012167 013310
1803 013020 004767 001306
1804 013024 005367 013244
1805 013030 001371
1806 013032 012701 026004
1807 013036 012167 013232
1808 013042 001002
1809 013044 000167 000016
1810 013050 012167 013254
1811 013054 004767 001252
1812 013060 005367 013210
1813 013064 001371
1814 013066 000207
1815
1816
1817 013070 005367 013210
1818 013074 001404
1819 013076 032777 004000 013240
1820 013104 001415
1821 013106 012767 000010 013170
1822 013114 005267 013172
1823 013120 032777 002000 013216
1824 013126 001004
1825 013130 004767 175426
1826 013134 004767 004416
1827 013140 004767 002776
1828 013144 000167 177220
1829
1830
1831
1832
1833
1834
1835 013150 032777 010000 013120
1836 013156 001415
1837 013160 012700 000007
1838 013164 004767 004510
1839 013170 004767 004504
1840 013174 004767 004500
1841 013200 004767 004474
1842 013204 052777 000020 013064
1843 013212 000240
1844 013214 000240
1845 013216 000240
1846 013220 032777 020000 013050
1847 013226 001531
1848 013230 012777 001000 013102

```

```

      JMP      SKP4
SKP3:  MOV      (R1)+,REMNO
      JSR      PC,DIAGNF
      DEC      CNTR
      BNE      SKP3
SKP4:  MOV      #EXTBL3,R1
      MOV      (R1)+,CNTR
      BNE      SKP5
      JMP      SKP6
SKP5:  MOV      (R1)+,REMNO
      JSR      PC,DIAGNF
      DEC      CNTR
      BNE      SKP5
SKP6:  MOV      #EXTBL4,R1
      MOV      (R1)+,CNTR
      BNE      SKP7
      JMP      SKP8
SKP7:  MOV      (R1)+,REMNO
      JSR      PC,DIAGNF
      DEC      CNTR
      BNE      SKP7
SKP8:  RTS      PC

SKPD2: DEC      ITER
      BEQ      SKPX0
      BIT      #B11,@SR
      BEQ      SKPX1
SKPX0: MOV      #10,ITER
      INC      PASCNT
      BIT      #B10,@SR
      BNE      SKPX1
      JSR      PC,ENPRTN
      JSR      PC,NULLS
SKPX1: JSR      PC,MONIT
      JMP      EXS2

      ;SUBROUTINE TO CHECK FOR ALARMS AND CHANGES OF REMOTE STATE.
      ; IF ALARM...RING 4 BELLS
      ; IF CHANGE OF REMOTE STATE...SET A FLAG

SCALM: BIT      #B12,@CSR
      BEQ      SCA1
      MOV      #7,R0
      JSR      PC,TTO
      JSR      PC,TTO
      JSR      PC,TTO
      JSR      PC,TTO
      BIS      #B04,@CSR
      SCA1:  NOP
      NOP
      NOP
      BIT      #B13,@CSR
      BEQ      SCA3
      MOV      #1000,@SPAR

```

```

      ;IF NONE, NEXT PORT
      ;GET REMOTE NO.
      ;DO DIAGNOSTIC FUNCTION ON REM.

      ;GET TABLE FOR PORT 2
      ;1ST ENTRY IS # OF REMOTES

      ;IF NONE, NEXT PORT
      ;GET REMOTE #
      ;DO DIAGNOSTIC FUNC. ON REMOTE

      ;GET TABLE FOR PORT 3
      ;1ST ENTRY IS # OF REMOTES

      ;IF NONE, DONE!
      ;GET REMOTE #
      ;DO DIAGNOSTIC FUNCTION ON REM.
      ;FINISH ALL REMOTES ON THIS PORT

      ;RETURN

      ;UPDATE ITERATION COUNT
      ;IF 0, END PASS.
      ;SW 11 = 1 == INH ITER.

      ;RELOAD ITERATION COUNT
      ;UPDATE PASS COUNT
      ;SW 10 = INH END PASS PRINT

      ;PRINT END PASS STUFF

      ;KEEP GOING

```

```

      ;ALARMS?
      ;NO, CHECK FOR CHG ST.
      ;ALARM! RING TTY BELLS

      ;CLR ALARM

      ;CHANGE OF STATE?
      ;NO, CHECK THRU
      ;SEE IF THIS IS A SYS ERR

```

D 4

1849	013236	027727	013100	000310		CMP	@SPDR,#310	: OR REMOTE CHANGE
1850	013244	001415				BEQ	SCA2	: REMOTE CHANGE IS O.K.
1851	013246					ERROR	\N,YSER	: ERROR:HOST SYS ERROR. RUN LOGIC TEST.
1852								:***** ERROR 50 *****
1853	013246	032777	040000	013070		BIT	#B14,@SR	
1854	013254	001007				BNE	.+20	
1855	013256	012700	024662			MOV	#YSER,RO	
1856	013262	012767	000050	002574		MOV	#50,ERRNUM	
1857	013270	004767	002422			JSR	PC,ERR	
1858		000051			N	=	N+1	
1859	013274					HLT		
1860	013274	004767	002206			JSR	PC,SWHLT	
1861	013300				SCA2:	PNTM	REMCHG	: PRINT 'REMOTE STATUS CHANGE'
1862	013300	012700	022400			MOV	#REMCHG,RO	: PRINT MESSAGE
1863	013304	004767	003122			JSR	PC,TYPOUT	: POINTED TO BY REMCHG
1864	013310					PNTM	REMSG	: PRINT 'REMOTE #'
1865	013310	012700	022342			MOV	#REMSG,RO	: PRINT MESSAGE
1866	013314	004767	003112			JSR	PC,TYPOUT	: POINTED TO BY REMSG
1867	013320	012777	001003	013012		MOV	#1003,@SPAR	
1868	013326	017700	013010			MOV	@SPDR,RO	: GET THE REMOTE NUMBER
1869	013332	004767	004036			JSR	PC,OCTPNT	: PRINT IT
1870	013336					PNTM	REMSTM	: PRINT 'REMOTE STATUS NOW ''
1871	013336	012700	022436			MOV	#REMSTM,RO	: PRINT MESSAGE
1872	013342	004767	003064			JSR	PC,TYPOUT	: POINTED TO BY REMSTM
1873	013346	005277	012766			INC	@SPAR	
1874	013352	032777	000040	012762		BIT	#B05,@SPDR	: NOW REGULAR?
1875	013360	001405				BEQ	SCA2A	: NO
1876	013362					PNTM	REGMG	: YES, PRINT 'REGULAR'
1877	013362	012700	022462			MOV	#REGMG,RO	: PRINT MESSAGE
1878	013366	004767	003040			JSR	PC,TYPOUT	: POINTED TO BY REGMG
1879	013372	000416				BR	SCA2C	
1880	013374	032777	000100	012740	SCA2A:	BIT	#B06,@SPDR	: IRREGULAR?
1881	013402	001405				BEQ	SCA2B	
1882	013404					PNTM	IRRGMG	: YES, PRINT 'IRREGULAR'
1883	013404	012700	022473			MOV	#IRRGMG,RO	: PRINT MESSAGE
1884	013410	004767	003016			JSR	PC,TYPOUT	: POINTED TO BY IRRGMG
1885	013414	000410				BR	SCA2D	
1886	013416	017700	012720		SCA2B:	MOV	@SPDR,RO	: OTHERWISE, JUST PRINT THE STATUS
1887	013422	004767	003746			JSR	PC,OCTPNT	
1888	013426	000410				BR	SCA2E	: AND LEAVE THE TALLY ALONE
1889	013430	005267	012716		SCA2C:	INC	TALLY	: UPDATE TALLY FOR ADDED REMOTE
1890	013434	000405				BR	SCA2E	
1891	013436	005767	012710		SCA2D:	TST	TALLY	: IS TALLY ALREADY =0?
1892	013442	001402				BEQ	SCA2E	: YES, FORGET IT
1893	013444	005367	012702			DEC	TALLY	: NO, MAKE IT ONE LESS
1894	013450	005767	002676		SCA2E:	TST	TALLY	: IS TALLY NOW 0?
1895	013454	001006				BNE	SCA2F	: NO, O.K.
1896	013456	105067	012576			CLRB	ONCE	: YES, HAVE TO START OVER
1897	013462					PNTM	ALROFF	: PRINT 'ALL REMOTES ARE NOW IRREGULAR
1898	013462	012700	022506			MOV	#ALROFF,RO	: PRINT MESSAGE
1899	013466	004767	002740			JSR	PC,TYPOUT	: POINTED TO BY ALROFF
1900	013472	012767	177777	012570	SCA2F:	MOV	#-1,CHNGFL	: SET CHANGE FLAG
1901	013500	052777	000020	012570		BIS	#20,@CSR	: CLEAR TOP BYTE
1902	013506	000167	177500			JMP	SCA1	: GO SEE IF ANY MORE CHANGES.
1903	013512	032777	004000	012556	SCA3:	BIT	#B11,@CSR	: IS TALLY SET?
1904	013520	001406				BEQ	SCA1	: NO, LEAVE

```
1905 013522 112767 177777 012531      MOVB    #-1,THRUF      ;YES, SET THE 'THRUF' FLAG
1906 013530 052777 000020 012540      BIS     #20,@CSR      ;CLEAR TOP BYTE
1907 013536 000207                SCART:  RTS     PC
1908
1909
1910      ;SUBROUTINE TO PERFORM 'PORT CONFIGURATION STATUS' ON A GIVEN
1911      ;PORT AND STORE THE RESULTS IN A GIVEN TABLE. ALSO
1912      ;STORED IS THE STATUS OF EACH REMOTE
1913      ;FOR EACH REGULAR REMOTE FOUND, A TALLY IS INCREMENTED
1914
1915      ;TABLES ARE IN THE FOLLOWING FORMAT:
1916      :      [ # OF ENTRIES ]
1917      :      [REM STAT][REM # ]
1918      :      [ " " ][ " " ]
1919
1920 013540 016701 012610      PORCNF: MOV     TBLAD,R1      ;TABLE ADDR GIVEN BY CALLER
1921 013544 052777 000020 012524      BIS     #B04,@CSR    ;CLR TOP BYTE OF CSR
1922 013552 005077 012562      CLR     @SPAR        ;ADDR REQ. SCPD
1923 013556 012777 000140 012556      MOV     #140,@SPDR   ;LOAD PORT CONF. ST. FUNL.
1924 013564 005277 012550      INC     @SPAR
1925 013570 012777 000002 012544      MOV     #2,@SPDR    ;DATA LEN. = 2
1926 013576 005277 012536      INC     @SPAR
1927 013602 012777 000001 012532      MOV     #1,@SPDR    ; ONE REQUEST
1928 013610 005277 012524      INC     @SPAR
1929 013614 016777 012500 012520      MOV     PORN,@SPDR   ;PORT # GIVEN BY CALLER
1930 013622 004767 177322      JSR     PC,SCALM     ;CLEAR ALARMS & CHANGES
1931 013626 052777 000001 012442      BIS     #B00,@CSR    ;START FUNCTION
1932 013634 032777 040000 012434      PORO:  BIT     #B14,@CSR ;FUNCTION COMPLETE?
1933 013642 001003                BNE     PORNO        ;NO, WAIT
1934 013644 004767 177300      JSR     PC,SCALM     ;SCAN ALARMS AGAIN
1935 013650 000771                BR      PORO
1936 013652 004767 002264      POROO: JSR     PC,MONIT
1937 013656 012777 001000 012454      MOV     #1000,@SPAR ;ADDR RESULT SCPD
1938 013664 027727 012452 000140      CMP     @SPDR,#140  ;THIS RESULT FROM PORT CONFIG?
1939 013672 001415                BEG     POR1
1940 013674                ERROR   \N,SYSER      ;ERROR:HOST SYS ERROR. RUN LOGIC TEST
1941
1942      ;***** ERROR 51 *****
1943 013674 032777 040000 012442      BIT     #B14,@SR
1944 013702 001007                BNE     .+20
1945 013704 012700 024662      MOV     #YSER,RO
1946 013710 012767 000051 002146      MOV     #51,ERRNUM
1947 013716 004767 001774      JSR     PC,ERR
1948 013722 000052                N      N+1
1949 013722 004767 001560      HLT
1950 013726 005277 012406      POR1:  JSR     PC,SMLT
1951 013732 017700 012404      INC     @SPAR        ;GET # OF REMOTES
1952 013736 006200      MOV     @SPDR,RO    ;DIVIDE BY 4
1953 013740 006200      ASR     RO
1954 013742 010021      ASR     RO
1955 013744 001002      MOV     RO,(R1)+    ;SAVE AS # OF ENTRIES
1956 013746 000167 000042      BNE     PORLD       ;IF MORE THAN 0, LOAD THEM
1957 013752 010067 012316      JMP     PORXT       ;IF NONE, EXIT
1958 013756 012777 001003 012354      PORLD: MOV     RO,CNTR ;GET # OF ENTRIES
1959 013764 117721 012352      MOV     #1003,@SPAR
1960 013770 005277 012344      PORCN: MOV     @SPDR,(R1)+ ;LOAD REMOTE #
      INC     @SPAR
```

```

1961 013774 117721 012342          MOVB   @SPDR,(R1)+      ;LOAD REM. STATUS
1962 014000 062777 000003 012332  ADD    #3,@SPAR
1963 014006 005367 012262          DEC    CNTR            ;GOT THEM ALL?
1964 014012 001364          BNE    PORCN          ;NO, GET THE REST
1965 014014 005767 012300  PORXT: TST   PORN      ;PORT 0?
1966 014020 001002          BNE    1$            ;NO, NEVER MIND
1967 014022 005067 012324          CLR    TALLY        ;YES, CLEAR TALLY
1968 014026 060067 012320  1$:   ADD    R0,TALLY   ;UPDATE TALLY
1969 014032 052777 000020 012236  BIS    #B04,@CSR    ;CLR TOP BYTE
1970 014040 000207          RTS    PC            ;AND RETURN
1971
1972          ;SUBROUTINE TO REPORT THE STATUS OF EACH PORT WHICH HAS ANY
1973          ;REMOTES ATTACHED AND REGULAR.
1974
1975 014042 012767 000012 012236  REPORT: MOV   #12,LINLEN ;ALLOW 11(DEC) REM'S PER LINE.
1976 014050 012701 025204          MOV   #EXTBL1,R1    ;GET PORT TABLE
1977 014054 012167 012214          MOV   (R1)+,CNTR   ;GET # OF REMOTES
1978 014060 001405          BEQ   REPO         ;NONE HERE, NEXT PORT
1979 014062 012767 000000 012230  MOV   #0,PORN      ;LOAD PORT NO.
1980 014070 004767 000076          JSR   PC,PRNREM    ;GO PRINT REMOTES ON THIS PORT
1981 014074 012701 025404  REPO:  MOV   #EXTBL2,R1
1982 014100 012167 012170          MOV   (R1)+,CNTR   ;GET # OF REMOTES
1983 014104 001405          BEQ   REP1         ;NONE HERE, NEXT PORT
1984 014106 012767 000001 012204  MOV   #1,PORN      ;GO PRINT REMOTES ON THIS PORT
1985 014114 004767 000052          JSR   PC,PRNREM
1986 014120 012701 025604  REP1:  MOV   #EXTBL3,R1
1987 014124 012167 012144          MOV   (R1)+,CNTR   ;GET # OF REMOTES
1988 014130 001405          BEQ   REP2         ;NONE HERE, NEXT PORT
1989 014132 012767 000002 012160  MOV   #2,PORN      ;GO PRINT REMOTES ON THIS PORT
1990 014140 004767 000026          JSR   PC,PRNREM
1991 014144 012701 026004  REP2:  MOV   #EXTBL4,R1
1992 014150 012167 012120          MOV   (R1)+,CNTR   ;GET # OF REMOTES
1993 014154 001405          BEQ   REP3         ;NONE WE'RE DONE
1994 014156 012767 000003 012134  MOV   #3,PORN      ;GO PRINT REMOTES ON THIS PORT
1995 014164 004767 000002          JSR   PC,PRNREM
1996 014170 000207          RTS    PC            ;RETURN WHEN DONE
1997
1998          ;SUBROUTINE TO DO THE PRINTING PART OF PORT CONFIGURATION REPORT.
1999          ;R1 POINTS AT FIRST REMOTE #
2000          ;PORN = PORT #
2001          ;CNTR = # OF REMOTES ON THIS PORT
2002
2003
2004 014172          PRNREM: PNTM   PORTMS ;PRINT 'PORT #'
2005 014172 012700 022331          MOV   #PORTMS,R0  ;PRINT MESSAGE
2006 014176 004767 002230          JSR   PC,TYP0UT   ;POINTED TO BY PORTMS
2007 014202 016700 012112          MOV   PORN,R0     ;GET PORT #
2008 014206 004767 003162          JSR   PC,0CTPNT   ;PRINT IT
2009 014212 012100          PREMO: MOV   (R1)+,R0 ;GET REMOTE #
2010 014214 042700 177400          BIC   #177400,R0  ;CLR HIGH BYTE
2011 014220 004767 003176          JSR   PC,0CTJSP   ;PRINT IT
2012 014224 004767 001712          JSR   PC,MONIT    ;CHECK FOR CNTRL CHARS
2013 014230 005367 012040          DEC   CNTR
2014 014234 001426          BEQ   PREML       ;EXIT IF ALL DONE
2015 014236 005367 012044          DEC   LINLEN
2016 014242 001363          BNE   PREMC

```

```

2017 014244 012767 000012 012034      MOV      #12,LINLEN      ;YES, RESTORE LINLEN
2018 014252 012700 000015      MOV      #15,R0
2019 014256 004767 003416      JSR      PC,TTO        ;CARRIAGE RETURN
2020 014262 012767 000010 012002      MOV      #10,CLK      ;SPACE PAST PORT #
2021 014270 012700 000040      MOV      #40,R0
2022 014274 004767 003400      PREM1:  JSR      PC,TTO
2023 014300 005367 011766      DEC      CLK
2024 014304 001373      BNE     PREM1
2025 014306 000167 177700      JMP     PREMO
2026 014312 012700 000015      PREML:  MOV      #15,R0      ;ANOTHER <CR>
2027 014316 004767 003356      JSR      PC,TTO
2028 014322 012767 000017 011756      MOV      #12,LINLEN
2029 014330 000207      RTS      PC              ;AND RETURN
2030
2031      ;SUBROUTINE TO DO DIAGNOSTIC FUNCTION ON A GIVEN REMOTE
2032      ;ENTER WITH REMOTE # IN 'REMNO'
2033
2034
2035 014332 012767 000020 011732  DIAGNF: MOV      #20,CLK      ;SET UP TO LOAD REQUEST
2036 014340 012702 025144      MOV      #DATBL,R2
2037 014344 005077 011770      CLR     @SPAR          ;ADDRESS REQ SCR PAD
2038 014350 012777 000100 011764      MOV      #100,@SPDR    ;LOAD 'DIAG' FUNC
2039 014356 005277 011756      INC     @SPAR
2040 014362 012777 000022 011752      MOV      #22,@SPDR    ;DATA LEN. = 22
2041 014370 005277 011744      INC     @SPAR
2042 014374 012777 000001 011740      MOV      #1,@SPDR     ;ONE REQUEST
2043 014402 005277 011732      INC     @SPAR
2044 014406 016777 011716 011726  DIAGO: MOV      REMNO,@SPDR  ;LOAD REMOTE #
2045 014414 005277 011720      INC     @SPAR
2046 014420 012277 011716      MOV      (R2)+,@SPDR  ;LOAD DATA PATTERN
2047 014424 005367 011642      DEC     CLK
2048 014430 001371      BNE     DIAGO
2049 014432 004767 176512      DIAG00: JSR      PC,SCALM   ;CLEAR ALARMS ETC.
2050 014436 052777 000020 011632      BIS     #B04,@CSR     ;CLR TOP BYTE
2051 014444 052777 000001 011624      BIS     #B00,@CSR     ;SET START FUNC.
2052 014452 032777 040000 011616  DIAG1: BIT     #B14,@CSR  ;FUN COMPL?
2053 014460 001003      BNE     DIAG10
2054 014462 004767 176462      JSR      PC,SCALM
2055 014466 000771      BR     DIAG1
2056 014470 012777 001000 011642  DIAG10: MOV      #1000,@SPAR ;ADDRESS RESULT SCR. PAD
2057 014476 012767 000100 001362      MOV      #100,GOOD    ;RIGHT FUNCTION?
2058 014504 017767 011632 001356      MOV      @SPDR,BAD
2059 014512 026767 001350 001350      CMP     GOOD,BAD
2060 014520 001416      BEQ     DIAG2
2061 014522      DATERR \N,DIAGF      ;ERROR:DIAG FUNC GAVE WRONG FUNC CODE
2062      ;***** ERROR 52 *****
2063 014522 032777 040000 011614      BIT     #B14,@SR
2064 014530 001007      BNE     .+20
2065 014532 012700 024723      MOV      #DIAGF,R0
2066 014536 012767 000052 001320      MOV      #52,ERRNUM
2067 014544 004767 001232      JSR      PC,DERR
2068      = N+1
2069 014550      SCOPE  DIAG00
2070 014550 004567 000752      JSR      R5,SCRPTN
2071 014554 014432
2072 014556 062777 000002 011554  DIAG2: ADD     #2,@SPAR      ;CHECK FUNCTION STATUS

```

```

2073 014564 012767 000001 001274      MOV      #1,GOOD          ;SHD SHOW SUCCESS
2074 014572 017767 011544 001270      MOV      @SPDR,BAD
2075 014600 026767 001262 001262      CMP      GOOD,BAD
2076 014606 001460          RFD      DIAG4           ;O.K.
2077 014610          DATERR  \N,DIAGUF       ;ERROR:USOF WRONG ON 'DIAG' FUNC
2078                                     ;***** ERROR 53 *****
2079 014610 032777 040000 011526      BIT      #B14,@SR
2080 014616 001007          BNE      .+20
2081 014620 012700 024766      MOV      #DIAGUF,RO
2082 014624 012767 000053 001232      MOV      #53,ERRNUM
2083 014632 004767 001144          JSR      PC,DERR
2084          000054          =      N+1
2085 014636 032777 040000 011500      BIT      #B14,@SR       ;PRINT ALLOWED?
2086 014644 001036          BNE      DIAG3S        ;NO, SKIP IT
2087 014646          PNTM    REMSG         ;PRINT 'REMOTE #'
2088 014646 012700 022342      MOV      #REMSG,RO      ;PRINT MESSAGE
2089 014652 004767 001554          JSR      PC,TYP0UT     ;POINTED TO BY REMSG
2090 014656 005277 011456          INC      @SPAR
2091 014662 017700 011454          MOV      @SPDR,RO      ;GET REM. NO.
2092 014666 004767 002502          JSR      PC,OC1PNT     ;PRINT IT
2093 014672          PNTM    STATEM       ;PRINT 'STATE '
2094 014672 012700 022355      MOV      #STATEM,RO     ;PRINT MESSAGE
2095 014676 004767 001530          JSR      PC,TYP0UT     ;POINTED TO BY STATEM
2096 014702 005277 011432          INC      @SPAR
2097 014706 017700 011430          MOV      @SPDR,RO      ;GET STAE OF REMOTE
2098 014712 004767 002456          JSR      PC,OC1PNT
2099 014716          PNTM    USORM         ;PRINT 'USOR = '
2100 014716 012700 022367      MOV      #USORM,RO      ;PRINT MESSAGE
2101 014722 004767 001504          JSR      PC,TYP0UT     ;POINTED TO BY USORM
2102 014726 005277 011406          INC      @SPAR
2103 014732 017700 011404          MOV      @SPDR,RO
2104 014736 004767 002432          JSR      PC,OC1PNT
2105 014742          DIAG3S: SCOPE      DIAG00
2106 014742 004567 000560          JSR      R5,SCRPTN
2107 014746 014432          DIAG00
2108 014750 052777 000020 011320  DIAG4: BIS      #B04,@CSR  ;CLEAR TOP BYTE WHEN DONE
2109 014756 000207          RTS      PC

```

.SBTTL MAINTENANCE ROUTINE

```

2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128

```

;THIS ROUTINE WORKS LIKE ODT BASICALLY BUT
 ;ONLY ON THE CMR-11 SCRATCHPAD MEMORY.
 ;THE REQUEST SCR. PAD MAY BE LOADED/READ
 ;TYPING 'G' CAUSES EXECUTION OF THE FUNCTION
 ;LOADED. COMPLETION OF THE FUNCTION CAUSES
 ;THE FIRST LOCATION OF THE RESULT SCR. PAD
 ; TO BE PRINTED.
 ;THE CMR-11 CSR MAY BE EXAMINED BY TYPING
 ; IT'S BUS ADDRESS FOLLOWED BY SLASH (/)
 ; EXIT IS ACHIEVED BY TYPING CNTRL-C
 ;OR 'P' TO PROCEED TO THE POINT AT WHICH THE
 ;MAIN PROGRAM WAS INTERRUPTED.

```

2127 014760 112767 177777 011263  ODTPR: MOV      #-1,DBFLG  ;SET DEBUG FLAG
2128 014766 017767 011304 011342  MOV      @CSR,SAVEC    ;SAVE CSR FOR LOOKING AT

```

2129	014774				PNTM	DBAID	:PRINT 'CMR-11 DEBUG AID'
2130	014774	012700	020507		MOV	#DBAID,RO	:PRINT MESSAGE
2131	015000	004767	001426		JSR	PC,TYPOUT	:POINTED TO BY DBAID
2132	015004	005067	011252		CLR	ADDR	
2133	015010	016767	011246	002330	MOV	ADDR,KBBUF	:GET ADDRESS
2134	015016	004767	001452		JSR	PC,INPKB	: FROM KEYBOARD
2135	015022	016767	002320	011232	MOV	KBBUF,ADDR	
2136	015030	026727	011226	001777	CMP	ADDR,#1777	:WITHIN LIMIT?
2137	015036	101423			BLOS	\$2\$:YES, O.K.
2138	015040	026767	011216	011230	CMP	ADDR,CSR	:IS IT THE CSR?
2139	015046	001012			BNE	\$10\$:NO
2140	015050	017700	011222		MOV	@CSR,RO	
2141	015054	004767	002342		JSR	PC,OCTJSP	:YES, SHOW CONTENTS OF CSR
2142	015060	012700	000015		MOV	#15,RO	
2143	015064	004767	002610		JSR	PC,TTO	
2144	015070	000167	177710		JMP	\$0\$	
2145	015074					\$10\$:	
2146	015074	012700	020371		PNTM	AGAIN	
2147	015100	004767	001326		MOV	#AGAIN,RO	:PRINT MESSAGE
2148	015104	000741			JSR	PC,TYPOUT	:POINTED TO BY AGAIN
2149	015106	105767	011144		BR	\$1\$	
2150	015112	001042				\$2\$:	
2151	015114	105767	011132		TSTB	SLFL	:SLASH HIT?
2152	015120	001462			BNE	\$11\$:YES DUMP DATA
2153	015122	052777	000020	011146	TSTB	GOFLG	: 'G' TYPED?
2154	015130	005077	011204		BEQ	\$20\$:NO
2155	015134	027727	011202	000151	BIS	#B04,@CSR	:INSURE TOP BYTE CLEAR
2156	015142	101404			CLR	@SPAR	:ADDRESS FUNC. CODE
2157	015144	052777	000200	011124	CMP	@SPDR,#151	:IS FUNC NOT MAINT?
2158	015152	000403			BLOS	\$3\$:IT'S NOT MAINT. DO IT
2159	015154	052777	000001	011114	BIS	#B07,@CSR	:IT IS MAINT, SET 'MAIN'
2160	015162	032777	100000	011106	BR	\$4\$	
2161	015170	001003			BIS	#B00,@CSR	:SET 'STFUN'
2162	015172	004767	000744		BIT	#B15,@CSR	:IS THERE AN INTR?
2163	015176	000771			BNE	\$5\$:YES, SEE WHAT IT IS
2164	015200	012767	001000	011054	JSR	PC,MONIT	
2165	015206	012700	000015		BR	\$4\$	
2166	015212	004767	002462		MOV	#1000,ADDR	:ADDRESS RESULT
2167	015216	000430			MOV	#15,RO	
2168	015220	016777	011036	011112	JSR	PC,TTO	:DO <CR>
2169	015226	017700	011110		BR	\$21\$	
2170	015232	004767	002164		MOV	ADDR,@SPAR	:LOAD 'ADDRESS'
2171	015236				MOV	@SPDR,RO	:GET 'CONTENTS'
2172	015236	012700	020636		JSR	PC,OCTJSP	:AND PRINT IT
2173	015242	004767	001164		PNTM	TWOSP	:SPACE AND PROMPT (:)
2174	015246	017767	011070	002072	MOV	#TWOSP,RO	:PRINT MESSAGE
2175	015254	004767	001214		JSR	PC,TYPOUT	:POINTED TO BY TWOSP
2176	015260	016777	002062	011054	MOV	@SPDR,KBBUF	
2177	015266	105767	010761		JSR	PC,INPKB	:GET KBD INPUT
2178	015272	001421			MOV	KBBUF,@SPDR	:LOAD ANY NEW INPUT
2179	015274	005267	010762			\$20\$:	:LINE FEED?
2180	015300	016700	010756		TSTB	INCF	
2181	015304	020027	001777		BEQ	\$13\$	
2182	015310	101402			INC	ADDR	:IF YES, INCREMENT ADDRESS
2183	015312	000167	177556		MOV	ADDR,RO	
2184	015316	004767	002052		CMP	RO,#1777	:VALID ADDRESS?
					BLOS	\$22\$:YES, O.K.
					JMP	\$10\$:NO GOOD. GET A NEW ONE
					JSR	PC,OCTPNT	:PRINT NEW ADDRESS


```

2185 015322 012700 000057      MOV      #'/,R0      ;PRINT A SLASH
2186 015326 004767 002346      JSR      PC,ITO
2187 015332 000167 177662      JMP      $11$      ;GO PRINT CONTENTS
2188 015336 105767 010714      $13$: TSTB     SLFL      ;SLASH TYPED?
2189 015342 001402      BEQ      $23$      ;NO, CHECK IF IT'S 'P'
2190 015344 000167 177650      JMP      $11$      ;YES, SHOW CONTENTS
2191 015350 105767 010703      $23$: TSTB     CONFLG    ;'P' TYPED?
2192 015354 001004      BNE      $25$      ;NO, MUST BE <CR>
2193 015356 105767 010670      TSTB     GOFLG
2194 015362 001257      BNE      $2A$
2195 015364 000611      BR       $1$      ;YES, START FUNCTION
2196 015366 105067 010657      $25$: CLRB     DBFLG    ;CLEAR DEBUG FLAG
2197 015372 005767 010720      TST      PCONT
2198 015376 001402      BEQ      $24$      ;ANY PROMPTS PENDING?
2199 015400 016716 010712      MOV      PCONT,(SP) ;NO, JUST RETURN TO CALLER
2200 015404 000207      $24$: RTS      PC      ;YES, RETYPE PROMPT.
                                ;UNOFFICIAL RETURN

```

.SBTTL UTILITY ROUTINES

```

;ROUTINES TO SAVE AND RESTORE GENERAL REGISTERS
; CALLED BY JSR R5,REGSAV .
; AND JSR R5,REGRES

```

```

2209 015406 010446      REGSAV: MOV      R4,-(SP)
2210 015410 010346      MOV      R3,-(SP)
2211 015412 010246      MOV      R2,-(SP)
2212 015414 010146      MOV      R1,-(SP)
2213 015416 010046      MOV      R0,-(SP)
2214 015420 000115      JMP      @R5
2216 015422 030026      REGRES: BIT      R0,(SP)+ ;THROW AWAY OLD R5 VALUE
2217 015424 012600      MOV      (SP)+,R0
2218 015426 012601      MOV      (SP)+,R1
2219 015430 012602      MOV      (SP)+,R2
2220 015432 012603      MOV      (SP)+,R3
2221 015434 012604      MOV      (SP)+,R4
2222 015436 000205      RTS      R5

```

;ROUTINE TO SET LSI-11 FLAG (SOFTWARE)

```

2226 015440 112767 177777 010602 LSTST: MOVB     #-1,LSIFLG ;SET LSI-11 SOFTWARE FLAG
2227 015446 000002      RTI      ;RETURN

```

;ROUTINE TO SET SOFTWARE SWITCH REGISTER FLAG

```

2231 015450 012767 026346 010666 SRTST: MOV      #SSWR,SR ;NO HARDWARE SWR, USE
2232 015456 012767 177777 010664      MOV      #-1,SWRFLG ;MEMORY LOCATION
2233 015464 000002      RTI      ;AND SET FLAG

```

;ROUTINE TO PRINT ILLEGAL DEVICE ADDRESS

```

2237 015466 012706 002000      DVATST: MOV      #ISP,SP ;FIX THE STACK
2238 015472      PNTM     INVLAD ;PRINT NON-EXST ADDR MSG
2239 015472 012700 020534      MOV      #INVLAD,R0 ;PRINT MESSAGE
2240 015476 004767 000730      JSR      PC,TYPOUT ;POINTED TO BY INVLAD

```

```

2241 015502 000167 164404          JMP      PROMT          ;GO GET CORRECT ADDRESS
2242
2243          ;ROUTINE TO REPLACE HALT WITH CALL TO SW MONITOR
2244          ;ON NON SWR PROCESSORS (CALLED BY MACRO 'HLT')
2245
2246 015506 005767 010636          SWHLT:  TST      SWRFLG      ;ANY HARDWARE SWR?
2247 015512 001403                    BEQ      1$                ;IF YES, GO HALT
2248 015514 004767 000454          JSR      PC,SWDMP        ;IF NO, DUMP SWITCHES
2249 015520 000207                    RTS      PC                ;CONTINUE
2250 015522 000000          1$:      HALT          ;REAL HALT
2251 015524 000207                    RTS      PC                ;CONTINUE
2252
2253          ;ROUTINE USED FOR LOOPING ON ERROR ROUTINE (CALLED BY MACRO 'SCOPE')
2254
2255 015526 004767 000410          SCPRTN: JSR      PC,MONIT    ;CHECK FOR ^S, OR ^C
2256 015532 005777 010606                    TST      @SR              ;IS SW 15 SET?
2257 015536 100402                    BMI      SBAK            ;YES, DON'T HALT
2258 015540                    HLT                      ;COMMON ERROR HALT
2259 015540 004767 177742          JSR      PC,SWHLT
2260 015544 012500          SBAK:  MOV      (R5)+,R0    ;GET DIRECTION FOR SCOPE LOOP
2261 015546 032777 020000 010570    BIT      #B13,@SR        ;SW 13 SET?
2262 015554 001402                    BEQ      SCONT           ;NO, DON'T LOOP
2263 015556 012605                    MOV      (SP)+,R5        ;YES, RESTORE R5
2264 015560 000110                    JMP      (R0)            ;AND LOOP
2265 015562 005267 010514          SCONT:  INC      ERRCNT    ;UPDATE ERR CNT IF 'FALL THRU'
2266 015566 001003                    BNE      SCORTN         ;WATCH FOR ERRCNT OFLO
2267 015570 112767 177777 010456    MOVB    #-1,OFLOF       ;SET OVERFLO FLAG.
2268 015576 000205          SCORTN: RTS      R5      ;AND RETURN
2269
2270          ;ROUTINE TO GENERATE CSC DEVICE ADDRESSES FROM 'DEVADR'
2271
2272
2273 015600 016700 010562          DEVGEN: MOV      DEVADR,R0 ;GET DEVICE CSR ADDRESS
2274 015604 010067 010466                    MOV      R0,CSR         ;GENERATE CSR
2275 015610 062700 000002                    ADD      #2,R0
2276 015614 010067 010520                    MOV      R0,SPAR        ;GENERATE SPAR
2277 015620 062700 000002                    ADD      #2,R0
2278 015624 010067 010512                    MOV      R0,SPDR        ;GENERATE SPDR
2279 015630 016700 010530                    MOV      VECTOR,R0
2280 015634 012760 000340 000002    MOV      #340,2(R0)
2281 015642 000207          RTS      PC            ;RETURN
2282
2283          ;ROUTINE TO CATCH ERROR TRAPS TO LOC 4
2284
2285 015644 011667 010506          HDER:  MOV      (SP),TEMP  ;SAVE STACK FOR ADDR OF TRAP
2286 015650 012706 002000                    MOV      #ISP,SP        ;RENEW STACK BEFORE USING
2287 015654                    PNTM    TRP4            ;PRINT 'TRAPPED TO 4 FROM LOC '
2288 015654 012700 020600                    MOV      #TRP4,R0       ;PRINT MESSAGE
2289 015660 004767 000546                    JSR      PC,TYPOUT      ;POINTED TO BY TRP4
2290 015664 162767 000002 010464    SUB      #2,TEMP
2291 015672 016700 010460                    MOV      TEMP,R0
2292 015676 004767 001472                    JSR      PC,OCTPNT
2293 015702 004767 001650                    JSR      PC,NULLS
2294 015706 012706 002000                    MOV      #ISP,SP
2295 015712 000167 164542          JMP      RESTRT
2296

```

```

2297
2298
2299
2300
2301 015716 011667 000140
2302 015722 162767 000026 000132
2303 015730 010046
2304 015732
2305 015732 012700 016072
2306 015736 004767 000470
2307 015742 016700 000116
2308 015746 004767 001422
2309 015752
2310 015752 012700 016105
2311 015756 004767 000450
2312 015762 016700 000074
2313 015766 004767 001402
2314 015772 012600
2315 015774 004767 000432
2316 016000 000207
2317
2318
2319 016002 011667 000054
2320 016006 162767 000026 000046
2321 016014 004767 177710
2322 016020
2323 016020 012700 016117
2324 016024 004767 000402
2325 016030 016700 000032
2326 016034 004767 001334
2327 016040
2328 016040 012700 016133
2329 016044 004767 000362
2330 016050 016700 000014
2331 016054 004767 001314
2332 016060 000207
2333
2334
2335
2336
2337 016062 000000
2338 016064 000000
2339 016066 000000
2340 016070 000000
2341
2342
2343
2344
2345 016072 023046 025052 051105
2346 016100 047522 020122 100
2347 016105 040 052101 046040
2348 016112 041517 020056 100
2349 016117 046 044123 052517
2350 016124 042114 041040 020105
2351 016132 100
2352 016133 054 053440 051501

```

;ERROR ROUTINES: REGULAR ERRORS AND DATA ERRORS.

```

ERR:  MOV (SP),ERRAD ;GET ADDRESS OF ERROR CALL
      SUB #26,ERRAD ;CORRECT IT
ERR1: MOV RO,-(SP) ;STACK ERROR MSG ADDR
      PNTM ERRM ;PRINT '**ERROR**'
      MOV #ERRM,RO ;PRINT MESSAGE
      JSR PC,TYPOUT ;POINTED TO BY ERRM
      MOV ERRNUM,RO
      JSR PC,OCTPNT ;PRINT ERROR NO. (P)
      PNTM WDAT ;PRINT ' AT LOC. '
      MOV #WDAT,RO ;PRINT MESSAGE
      JSR PC,TYPOUT ;POINTED TO BY WDAT
      MOV ERRAD,RO
      JSR PC,OCTPNT ;PRINT ADDRESS OF ERROR
      MOV (SP)+,RO ;GET ADDR OF ERROR MSG
      JSR PC,TYPOUT ;PRINT ERR MSG
      RTS PC ;RETURN

```

```

DERR: MOV (SP),ERRAD ;GET ADDRESS OF ERROR CALL
      SUB #26,ERRAD ;CORRECT IT
      JSR PC,ERR1 ;PRINT REG. ERROR MSG
      PNTM WDSDB ;PRINT 'SHOULD BE '
      MOV #WDSDB,RO ;PRINT MESSAGE
      JSR PC,TYPOUT ;POINTED TO BY WDSDB
      MOV GOOD,RO
      JSR PC,OCTPNT ;PRINT GOOD DATA
      PNTM WDWAS ;PRINT ' WAS '
      MOV #WDWAS,RO ;PRINT MESSAGE
      JSR PC,TYPOUT ;POINTED TO BY WDWAS
      MOV BAD,RO
      JSR PC,OCTPNT ;PRINT BAD DATA
      RTS PC ;RETURN

```

;RELATED STORAGE:

```

ERRAD: .WORD 0
ERRNUM: .WORD 0
GOOD: .WORD 0
BAD: .WORD 0

```

;ASSOCIATED ASCII (ONLY SOME OF IT)

```

ERRM: .ASCII /**ERROR @/
WDAT: .ASCII / AT LOC. @/
WDSDB: .ASCII /SHOULD BE @/
WDWAS: .ASCII /, WAS @/

```

```

2353 016140 040040
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365 016142 005000
2366 016144 105777 010224
2367 016150 100402
2368 016152 000167 000220
2369 016156 017700 010210
2370 016162 042700 177600
2371 016166 020027 000023
2372 016172 001056
2373 016174
2374 016174 012700 020642
2375 016200 004767 000226
2376 016204 017700 010134
2377 016210 004767 001160
2378 016214
2379 016214 012700 020636
2380 016220 004767 000206
2381 016224 017767 010114 001114
2382 016232 012767 016174 010056
2383 016240 004767 000230
2384 016244 005067 010046
2385 016250 016777 001072 010066
2386 016256
2387 016256 012700 020652
2388 016262 004767 000144
2389 016266 105777 010102
2390 016272 100375
2391 016274 017700 010072
2392 016300 042700 177600
2393 016304 020027 000023
2394 016310 001731
2395 016312 020027 000020
2396 016316 001363
2397 016320 012700 000015
2398 016324 004767 001350
2399 016330 020027 000024
2400 016334 001004
2401 016336 012706 002000
2402 016342 000167 164166
2403 016346 020027 000017
2404 016352 001002
2405 016354 000167 176400
2406
2407 016360 020027 000003
2408 016364 001004

.EVEN
.SBTTL 'SWITCH' MONITOR ROUTINE

:ENTER AT MONIT FROM EVERY SUB-TEST TO SEE IF CNTRL-S OR CNTRL-C WAS TYPED
:ENTER AT SWDMP FROM ERROR HALTS IF SW 15 = 0
:ALSO MONITORS THE FOLLOWING CONTROL FUNCTIONS:
:   CNTRL-P CONTINUE (PROCEED)
:   CNTRL-O GO TO CMR-11 DEBUG ROUTINE

MONIT: CLR      R0
        TSTB   @KBS           ;CHECK KEYBOARD FLAG
        BMI    MONIC         ;IF SET, CHECK WHAT CHAR.
        JMP    EX4           ;OTHERWISE, EXIT
MONIC: MOV    @KBD,R0
MONCH: BIC    #177600,R0     ;TRIM OFF PARITY BIT
        CMP    R0,#23       ;WAS IT ^S?
        BNE    EX1          ;NO, EXIT
SWDMP: PNTM   SWRMSG        ;PRINT 'SWR = '
        MOV    #SWRMSG,R0   ;PRINT MESSAGE
        JSR    PC,TYPDUT    ;POINTED TO BY SWRMSG
        MOV    @SR,R0       ;GET CONTENTS OF SR
        JSR    PC,OCTPNT    ;PRINT IT
        PNTM   TWOSP        ;SPACE AND PROMPT (:)
        MOV    #TWOSP,R0    ;PRINT MESSAGE
        JSR    PC,TYPDUT    ;POINTED TO BY TWOSP
        MOV    @SR,KBBUF    ;LOAD OLD SWITCHES
        MOV    #SWDMP,PCONT ;LOAD RETURN ADDRESS
        JSR    PC,INPKB     ;GET KBD INPUT
        CLR    PCONT
        MOV    KBBUF,@SR    ;LOAD NEW SWITCHES
CCRTN: PNTM   TYPCTP        ;PRINT 'CNTRL-P TO CONTINUE'
        MOV    #TYPCTP,R0   ;PRINT MESSAGE
        JSR    PC,TYPDUT    ;POINTED TO BY TYPCTP
CONTW1: TSTB  @KBS
        BPL    CONTW1
        MOV    @KBD,R0
        BIC    #177600,R0   ;TRIM OFF PARITY BIT
        CMP    R0,#23       ;^S?
        BEQ    SWDMP        ;YES, GET SWR AGAIN
        CMP    R0,#20       ;^P?
        BNE    CONTW1      ;NO, KEEP LOOKING
        MOV    #15,R0       ;RETURN LINE
        JSR    PC,TTO
EX1:   CMP    R0,#24       ;WAS A ^T TYPED?
        BNE    EX2          ;NO, EXIT
        MOV    #ISP,SP      ;YES, RENEW STACK
        JMP    BCONT        ;BACK TO DISPATCHER
EX2:   CMP    R0,#17       ;WAS CNTRL-O TYPED?
        BNE    EX3          ;NO
        JMP    ODTPH        ;AND GO TO DEBUG ROUTINE
EX3:   CMP    R0,#3        ;WAS CNTRL-C TYPED?
        BNE    EX4          ;NO.

```

2409	016366	012706	002000		MOV	#ISP,SP	:YES, REFRESH STACK
2410	016372	000167	164136		JMP	BCONT	:AND RESTART
2411	016376	020027	000006	EX4:	CMP	RO,#6	:WAS IT ^F?
2412	016402	001012			BNE	EX5	:NO, EXIT
2413	016404	105767	007641		TSTB	D3FLG	:IS DEBUG FLAG SET?
2414	016410	001007			BNE	FX5	:YES, JUST RETURN
2415	016412	105767	007637		TSTB	SAFE	:ALREADY IN FILL RTN?
2416	016416	001004			BNE	EX5	:YES, JUST RETURN
2417	016420	005067	007672		CLK	PCONT	:CLEAR PROMPT CARRIER
2418	016424	004767	000316		JSR	PC,KBRET	:SNEAK INTO FILL COUNT ROUTINE
2419	016430	000207		EX5:	RTS	PC	

.SBTTL MESSAGE PRINT ROUTINE

:MESSAGE TYPOTUT ROUTINE (CALLED BY MACRO PNTM A)
 :MESSAGES ARE IN THE FORMAT:
 : MSG: .ASCII /&MESSAGE&@/
 :
 :WHERE: & IS TRANSLATED INTO CR. AND LF.
 :
 :USES THE SUBROUTINE 'TTO'
 :WHICH PRINTS CR. & LF. UPON SEEING A CR. CODE.
 :AND @ IS MESSAGE TERMINATOR
 :
 :ENTER WITH ADDRESS OF MESSAGE IN RO

2437	016432	010046		TYPOTUT:	MOV	RO,-(SP)	:STACK ADDRESS OF MESSAGE
2438	016434	117600	000000	TPOFCH:	MOVB	@(SP),RO	:FETCH ASCII BYTE
2439	016440	022700	000100		CMP	#100,RO	:IS IT @ (TERMINATOR)?
2440	016444	001411			BEQ	TPOUTX	:YES-EXIT
2441	016446	022700	000046		CMP	#46,RO	:IS IT CRLF FLAG?
2442	016452	001002			BNE	TPCONT	:NO-TYPE CHARACTER
2443	016454	012700	000015		MOV	#15,RO	:YES, CHANGE DATA TO CR
2444	016460	004767	001214	TPCONT:	JSR	PC,TTO	:TYPE IT
2445	016464	005216			INC	(SP)	:MOVE POINTER TO NEXT BYTE
2446	016466	000762			BR	TPOFCH	:FETCH NEXT CHARACTER
2447	016470	005726		TPOUTX:	TST	(SP)+	:POP STACK TO REACH RETURN VECTOR
2448	016472	000207			RTS	PC	

.SBTTL KEYBOARD INPUT ROUTINE

:KEYBOARD INPUT ROUTINE CALLED BY JSR PC,INPKB
 :ENTERED WITH OLD CONTENTS IN KBBUF
 :IF JUST CR TYPED, EXIT WITH SAME CONTENTS IN KBBUF
 :IF NEW NUMBER TYPED, EXIT WITH NEW CONTENTS IN KBBUF

2458	016474	005067	000644	INPKB:	CLR	NOKEFL	:CLEAR NO NUMBER FLAG
2459	016500	010146			MOV	R1,-(SP)	:STACK OLD R1
2460	016502	016746	000640		MOV	KBBUF,-(SP)	:STACK 'OLD CONTENTS'
2461	016506	005067	000634		CLR	KBBUF	:CLEAR INPUT BUFFER
2462	016512	004767	000216	GETCHR:	JSR	PC,KBRE	:FETCH A CHARACTER IN RO
2463	016516	004767	001156		JSR	PC,TTO	:ECHO IT
2464	016522	020027	000012		CMP	RO,#12	:WAS IT A CR OR LF?

```

2465 016526 001002      BNE      1$      ;NO
2466 016530 000167 000154  JMP      NRTRN   ;YES, RETURN WITH PROPER KBBUF
2467 016534 010001      MOV      R0,R1   ;SET UP TO CHECK FOR A NUMBER
2468 016536 042701 177407  BIC      #1774(7,R1) ;MASK ALL BUT # CODE
2469 016542 020127 000060  CMP      R1,#6L  ;IS IT A # FROM 0-7?
2470 016546 001441      BEW      3$      ;YES, PACK IT
2471 016550 020027 000177  CMP      R0,#17/ ;WAS IT A DELETE/RUBOUT?
2472 016554 001024      BNE      2$      ;NO, MUST BE GARBAGE
2473 016556 012700 000057  MOV      #57,R0  ;YES, BUT PRINT '^'
2474 016562 004767 001112  JSR      PC,TTO  ;
2475 016566 000241      CLC                    ;CLEAR THE C-BIT
2476 016570 006067 000552  ROR      KBBUF    ;DELETE LAST DIGIT
2477 016574 000241      CLC                    ;
2478 016576 006067 000544  ROR      KBBUF    ; THAT WAS STUFFED
2479 016602 000241      CLC                    ;
2480 016604 006067 000536  ROR      KBBUF    ; INTO KBBUF
2481 016610 005767 000532  TST      KBBUF    ;HAVE WE DELETED EVERYTHING?
2482 016614 001002      BNE      11$     ;NO
2483 016616 005067 000522  CLR      NOKEFL   ;YES, BACK TO NO NUMBER INPUT
2484 016622 000167 177664  JMP      GETCHR   ;GO FOR MORE INPUT
2485 016626 012700 000207  MOV      #207,R0 ;ECHO 'BELL' FOR ERRONEOUS INPUT
2486 016632 004767 001042  JSR      PC,TTO  ;
2487 016636 012700 000077  MOV      #77,R0  ;THEN '?'
2488 016642 004767 001032  JSR      PC,TTO  ;
2489 016646 000167 177640  JMP      GETCHR  ;AND GET ANOTHER CHARACTER
2490 016652 012767 177777 000464 3$: MOV      #-1,NOKEFL ;GOT A DIGIT. SET FLAG
2491 016660 042700 177770  BIC      #177770,R0 ;GET THE DIGIT PART OF THE CHARACTER
2492 016664 006367 000456  ASL      KBBUF    ;SHIFT KBBUF BUFFER
2493 016670 006367 000452  ASL      KBBUF    ; TO ACCEPT THE
2494 016674 006367 000446  ASL      KBBUF    ; NEW DIGIT.
2495 016700 050067 000442  BIS      R0,KBBUF ;ADD THE NEW DIGIT
2496 016704 000167 177602  JMP      GETCHR  ;GO FOR MORE INPUT
2497
2498 016710 005767 000430  NRTRN: TST      NOKEFL ;WAS THERE NEW DATA?
2499 016714 001004      BNE      NEK     ;YES, GO BACK WITH IT
2500 016716 012667 000424  MOV      (SP)+,KBBUF ;NO, RETRIEVE OLD DATA
2501 016722 012601      MOV      (SP)+,R1 ;RESTORE R1
2502 016724 000207      RTS      PC      ;AND RETURN
2503 016726 005726  NEK: TST      (SP)+   ;DUMP OLD DATA
2504 016730 012601      MOV      (SP)+,R1 ;RESTORE R1
2505 016732 000207      RTS      PC      ;AND RETURN
2506
2507
2508 ;KEYBOARD READ ROUTINE
2509 ;CHECKS FOR THE FOLLOWING SPECIAL CHARACTERS:
2510 ;
2511 ; CNTRL-C RESTART PROG.
2512 ; CNTRL-F CHANGE/SHOW FILL COUNT
2513 ; CNTRL-O ENTER 'DE-BUG' ROUTINE
2514 ; UNLESS ALREADY THERE
2515 ; OR UNLESS IN INITIAL DIALOGUE.
2516 ; IF IN 'DE-BUG' ROUTINE:
2517 ; <CR> CLOSE LOCATION
2518 ; <LF> CLOSE LOC. AND OPEN NEXT
2519 ; '.' OPEN LOCATION
2520 ; 'G' SET START FUNC OR MAINT
; 'P' RETURN AND CONTINUE

```

2521									
2522	016734	105777	007434	KBRD:	TSTB	@KBS			:WAIT FOR INPUT FROM CONSOLE
2523	016740	100375			BPL	KBRD			
2524	016742	017700	007424		MOV	@KBD,RO			:PUT THE CHAR INTO RO
2525	016746	105067	007301	KBRET:	CLRB	INCF			:CLEAR FLAGS
2526	016752	105067	007274		CLRB	GOFLG			
2527	016756	105067	007274		CLRB	SLFL			
2528	016762	105067	007271		CLRB	CONFLG			
2529	016766	042700	177500		BIC	#177600,RO			:TRIM PARITY
2530	016772	020027	000073		COMP	RO,#3			:WAS ^C TYPED?
2531	016776	001004			BNE	1\$:YES, RESTART PROGRAM
2532	017000	012706	002000		MOV	#ISP,SP			
2533	017004	000167	163524		JMP	BCONT			
2534	017010	105767	007235	1\$:	TSTB	DBFLG			:DON'T ALLOW ^F IN 'DEBUG'
2535	017014	001060			BNE	4\$			
2536	017016	020027	000006		COMP	RO,#6			:WAS CNTRL-F TYPED?
2537	017022	001055			BNE	4\$:NO
2538	017024	105767	007225	2\$:	TSTB	SAFE			:ALREADY IN FILL CNT RTN?
2539	017030	001401			BEQ	20\$:NO, SAFE TO ENTER
2540	017032	000543			BR	KRTN			:YES, IGNORE CHARACTER
2541	017034			20\$:	PNTM	FILMSG			:PRINT 'FILL COUNT = '
2542	017034	012700	017354		MOV	#FILMSG,RO			:PRINT MESSAGE
2543	017040	004767	177366		JSR	PC,TYPOUT			:POINTED TO BY FILMSG
2544	017044	112767	177777	007203	MOV	#-1,SAFE			:SET FLAG
2545	017052	016700	000272		MOV	FILL,RO			
2546	017056	004767	000312		JSR	PC,OCTPNT			:SHOW FILL COUNT
2547	017062				PNTM	TWOSP			:SPACE AND PROMPT
2548	017062	012700	020636		MOV	#TWOSP,RO			:PRINT MESSAGE
2549	017066	004767	177340		JSR	PC,TYPOUT			:POINTED TO BY TWOSP
2550	017072	016767	000252	000246	MOV	FILL,KBBUF			
2551	017100	004767	177370		JSR	PC,INPKB			:GET NEW COUNT
2552	017104	026727	000236	000500	COMP	KBBUF,#500			:TOO BIG?
2553	017112	101406			BLOS	3\$			
2554	017114				PNTM	AGAIN			:NO GOOD, GET IT AGAIN
2555	017114	012700	020771		MOV	#AGAIN,RO			:PRINT MESSAGE
2556	017120	004767	177:36		JSR	PC,TYPOUT			:POINTED TO BY AGAIN
2557	017124	000167	1777:4		JMP	20\$			
2558	017130	016767	0002:2	000212	3\$:	MOV	KBBUF,FILL		:LOAD NEW COUNT
2559	017136	005767	007154		TST	PCONT			:RETURNING TO A PROMPT?
2560	017142	001477			BEQ	KRTN			:NO, CLEAN RETURN
2561	017144	062706	000006		ADD	#6,SP			:YES, BREAK UP WITH INPKB
2562	017150	016716	007142		MOV	PCONT,(SP)			:MODIFY RETURN ADDR
2563	017154	000472			BR	KRTN			
2564	017156	105767	007067	4\$:	TSTB	DBFLG			:IS 'DEBUG' FLAG SET?
2565	017162	001021			BNE	6\$:YES, CHECK FOR OTHER FLAGS
2566	017164	020027	000017		COMP	RO,#17			:CNTRL-O TYPED?
2567	017170	001064			BNE	KRTN			:NO, EXIT
2568	017172	005767	007120		TST	PCONT			:TEST IF ODT IS ALLOWED
2569	017176	001404			BEQ	5\$:YES, IF PCONT = 0 OR
2570	017200	026727	007112	002356	COMP	PCONT,#PRMT4			: IF GREATER THAN PRIORITY
2571	017206	103407			BLO	6\$: PROMPT.
2572	017210	105767	007041	5\$:	TSTB	SAFE			:ALSO, NOT WHILE IN FILL
2573	017214	001052			BNE	KRTN			: COUNT ROUTINE.
2574	017216	062706	000006		ADD	#6,SP			:BREAK ALL TIES WITH 'INPKB'
2575	017222	000167	175532		JMP	GO'PR			:A-O.K. GO TO DEBUG-AID
2576	017226	020027	000012	6\$:	COMP	RO,#16			:CHECK FOR L.F.

```

2577 017232 001007          BNE      7$          ;NOT A LINE FEED.
2578 017234 012700 000015    MOV      #15,RO      ;CHANGE L.F. TO C.R.
2579 017240 112767 177777 007005  MOVB    #-1,INCF    ;IT WAS, SET INCR FLAG
2580 017246 000167 000070          JMP      KRTN        ;AND LEAVE
2581 017252 020027 000057    7$:    CMP      RO,#'/    ;WAS IT A SLASH?
2582 017256 001007          BNE      8$          ;NO.
2583 017260 112767 177777 006770  MOVB    #-1,SLFL    ;YES, SET SLASH FLAG
2584 017266 004767 000406    JSR     PC,ITO      ;ECHO THE SLASH
2585 017272 012716 016710    MOV     #NRTRN,(SP) ;AND EXIT.
2586 017276 020027 000107    8$:    CMP      RO,#'G    ;WAS 'G' TYPED?
2587 017302 001007          BNE      9$          ;NO.
2588 017304 112767 177777 006740  MOVB    #-1,GOFLG   ;YES, SET 'GO' FLAG
2589 017312 004767 000362    JSR     PC,ITO      ;ECHO THE 'G'
2590 017316 012716 016710    MOV     #NRTRN,(SP)
2591 017322 020027 000120    9$:    CMP      RO,#'P    ;WAS 'P' TYPED?
2592 017326 001005          BNE      KRTN        ;NO, EXIT
2593 017330 112767 177777 006721  MOVB    #-1,CONFLG  ;YES, SET 'CONTINUE' FLAG
2594 017336 012716 016710    MOV     #NRTRN,(SP)
2595 017342 000207    KRTN:  RTS      PC
  
```

:ASSOCIATED VARIABLE STORAGE:

```

2600 017344 000000          NOKEFL: .WORD 0
2601 017346 000000          KBBUF:  .WORD 0
2602 017350 000010          FILL:   .WORD 10
2603 017352 000000          FLCNT: .WORD 0
2604 017354 043046 046111 020114  FILMSG: .ASCII /&FILL COUNT - @/
2605 017362 047503 047125 020124
2606 017370 020075          100
2607          017374
  
```

.EVEN

.SBTTL BINARY TO ASCII CONVERSION ROUTINES

```

2611          ;CONVERTS BINARY TO BINARY, BINARY TO
2612          ;OCTAL, AND BINARY TO DECIMAL; EITHER
2613          ;UNJUSTIFIED WITH LEADING ZERO'S SUPPRESSED
2614          ;OR RIGHT JUSTIFIED WITH LEADING 0'S
2615          ;SUPPRESSED
  
```

:REGULAR BIN-OCTAL UNJUSTIFIED:

```

2620 017374          OCTPNT: REGSAV
2621 017374 004567 176006    JSR     R5,REGSAV
2622 017400 005067 000360    CLR     RJFLG      ;CLEAR RIGHT JUSTIFY FLAG
2623 017404 012701 000010    MOV     #10,R1     ;SET RADIX FOR OCTAL
2624 017410 004767 000156    JSR     PC,NUMPNT  ;CONVERT & PRINT
2625 017414          REGRES
2626 017414 004567 176002    JSR     R5,REGRES
2627 017420 000207    RTS      PC        ;RETURN
  
```

:BIN-OCTAL JUSTIFIED:

```

2631 017422          OCTJSP: REGSAV
2632 017422 004567 175760    JSR     R5,REGSAV
  
```



```

2633 017426 012767 177777 000330      MOV    #-1,RJFLG      ;SET RIGHT JUSTIFY FLAG
2634 017434 012701 000010      MOV    #10,R1         ;SET RADIX FOR OCTAL
2635 017440 004767 000126      JSR    PC,NUMPNT      ;CONVERT & PRINT
2636 017444      REGRES
2637 017444 004567 175752      JSR    R5,REGRES
2638 017450 000207      RTS    PC
2639
2640      ;BIN-BIN
2641
2642 017452      BINPNT: REGSAV
2643 017452 004567 175730      JSR    R5,REGSAV
2644 017456 005067 000302      CLR    RJFLG          ;CLEAR RIGHT JUSTIFY FLAG
2645 017462 012701 000002      MOV    #2,R1         ;SET RADIX FOR BINARY
2646 017466 004767 000100      JSR    PC,NUMPNT      ;CONVERT & PRINT
2647 017472      REGRES
2648 017472 004567 175724      JSR    R5,REGRES
2649 017476 000207      RTS    PC
2650
2651      ;BIN-DECIMAL UNJUSTIFIED:
2652
2653 017500      DECPNT: REGSAV
2654 017500 004567 175702      JSR    R5,REGSAV
2655 017504 005067 000254      CLR    RJFLG          ;CLEAR RIGHT JUSTIFY FLAG
2656 017510 012701 000012      MOV    #12,R1        ;SET RADIX FOR DECIMAL
2657 017514 004767 000052      JSR    PC,NUMPNT      ;CONVERT & PRINT
2658 017520      REGRES
2659 017520 004567 175676      JSR    R5,REGRES
2660 017524 000207      RTS    PC
2661
2662      ;BIN-DECIMAL JUSTIFIED (6 PLACES)
2663
2664 017526      DECJSP: REGSAV
2665 017526 004567 175654      JSR    R5,REGSAV
2666 017532 012767 177777 000224      MOV    #-1,RJFLG      ;SET RIGHT JUSTIFY FLAG
2667 017540 012701 000012      MOV    #12,R1        ;SET RADIX FOR DECIMAL
2668 017544 004767 000022      JSR    PC,NUMPNT      ;CONVERT & PRINT
2669 017550      REGRES
2670 017550 004567 175646      JSR    R5,REGRES
2671 017554 000207      RTS    PC
2672
2673
2674 017556 005000      NULLS: CLR    R0
2675 017560 004767 000114      ISR    PC,TTO
2676 017564 004767 000110      JSR    PC,TTO
2677 017570 000207      NULOUT: RTS    PC
2678
2679      ;UNSIGNED CONVERT-PRINT ROUTINE (BIN - ASCII)
2680
2681 017572 010167 000170      NUMPNT: MOV    R1,RADIX      ;SAVE RADIX
2682 017576 005002      CLR    R2            ;CLEAR TAB COUNTER
2683 017600 005001      DIVSET: CLR    R1      ;CLEAR WORK REGISTER
2684 017602 020067 000160      DIVID: CMP    R0,RADIX    ;IS NUMBER BELOW RADIX?
2685 017606 103404      BLO    GEYDG         ;IF YES, STORE DIGIT
2686 017610 166700 000152      SUB    RADIX,R0      ;ELSE, KEEP SUBTRACTING
2687 017614 005201      INC    R1            ;AND KEEP COUNT
2688 017616 000771      BR     DIVID

```

```
2689 017620 010046 GETDG: MOV R0,-(SP) ;STACK REMAINDER
2690 017622 010100 MOV R1,R0
2691 017624 001403 BEQ PNTEXT ;PRINT IF HIGHEST ORDER STACKED
2692 017626 005202 INC R2 ;ELSE COUNT DIGITS FOR R. JUSTIFY
2693 017630 004767 177744 JSR PC, DIVSET
2694
2695 017634 012703 000006 PNTEXT: MOV #6,R3 ;GET DIGIT COUNT CONSTANT
2696 017640 160203 SUB R2,R3 ;HAVE WE PRODUCED 6 DIGITS?
2697 017642 003413 BLE PNT ;YES, JUSTIFICATION UNNECESSARY
2698 017644 005767 000114 TST RJFLG ;IS THE JUSTIFY FLAG SET?
2699 017650 001410 BEQ PNT ;NO-DON'T JUSTIFY
2700 017652 012700 000040 JUST: MOV #40,R0 ;YES, PRINT LEADING SPACES
2701 017656 004767 000016 JSR PC, TTO
2702 017662 005303 DEC R3
2703 017664 001372 BNE JUST
2704 017666 005067 000072 CLR RJFLG ;CLEAR JUSTIFY FLG WHEN DONE
2705 017672 012600 PNT: MOV (SP)+,R0 ;GET REST OF DIGITS OFF STACK
2706 017674 052700 000060 BIS #'0,R0 ;MAKE THEM ASCII
2707 ;TYPE OUT ROUTINE
2708 ;PRINTS A CHARACTER WHICH IS IN R0
2709 ;IF THE CHARACTER IS 'CR.', ALSO PRINT A 'LF.'.
2710
2711
2712 017700 010077 006474 TTO: MOV R0,@TTB ;PRINT CONTENTS OF R0
2713 017704 105777 006472 TTOLP: TSTB @TTS ;WAIT TILL PRINT DONE
2714 017710 100375 BPL TTOLP
2715 017712 022700 000015 CMP #15,R0 ;WAS IT A CR?
2716 017716 001401 BEQ TTOLF ;YES, ECHO A LF AS WELL
2717 017720 000207 RTS PC ;NO, JUST RETURN
2718 017722 012700 000012 TTOLF: MOV #12,R0
2719 017726 004767 177746 JSR PC, TTO ;ECHO A LINE FEED
2720 017732 016767 177412 177412 MOV FILL,FLCNT ;PRINT NULLS FOR FILL
2721 017740 012700 000000 MOV #0,R0
2722 017744 004767 177730 1S: JSR PC, TTO
2723 017750 005367 177376 DEC FLCNT
2724 017754 003373 BGT 1S
2725 017756 012700 000012 MOV #12,R0 ;THEN RESTORE R0
2726 017762 000207 RTS PC ;AND LEAVE
2727
2728
2729 ;ASSOCIATED VARIABLE STORAGE:
2730
2731 017764 000000 RJFLG: .WORD 0
2732 017766 000000 RADIX: .WORD 0
2733
2734 .SBTTL CONSTANTS, VARIABLES & ASCII
2735
2736
2737 ;ASCII MESSAGES:
2738
2739 017770 041446 041532 045115 TSTHDR: .ASCII /BCZCMJAO CMR-11 HOST EXERCISER 14-JAN-80/
2740 017776 030101 041440 051115
2741 020004 030455 020061 047510
2742 020012 052123 042440 042530
2743 020020 041522 051511 051105
2744 020026 020040 030440 026464
```

CZCMJAO CMR-11 HOST EXERCISER
CZCMJA.P11 06-MAR-80 15:13

MACY11 30A(1052) 10-MAR-80 14:09 PAGE 50
CONSTANTS, VARIABLES & ACSII

SEQ 0058

2745 020034 040512 026516 030070
2746 020042 023046 020040 020040
2747 020050 020040 025052 020052
2748 020056 040503 052125 047511
2749 020064 020116 025052 052
2750 020071 046 044124 051511
2751 020076 050040 047522 051107
2752 020104 046501 044440 020123
2753 020112 040503 040520 046102
2754 020120 020105 043117 040440
2755 020126 052114 051105 047111
2756 020134 020107 044124 020105
2757 020142 052123 052101 105
2758 020147 046 043117 040440
2759 020154 054516 043040 042511
2760 020162 042114 042440 052521
2761 020170 050111 042515 052116
2762 020176 040440 052124 041501
2763 020204 042510 020104 047524
2764 020212 040440 054516 047440
2765 020220 020106 044124 105
2766 020225 046 042522 047515
2767 020232 042524 020123 047117
2768 020240 052040 044510 020123
2769 020246 046503 026522 030461
2770 020254 056
2771 020255 046 042046 020117
2772 020262 047516 020124 051120
2773 020270 041517 042505 020104
2774 020276 043111 043040 042511
2775 020304 042114 044040 051101
2776 020312 053504 051101 020105
2777 020320 051511 051440 044524
2778 020326 046114 047440 026516
2779 020334 044514 042516 023041
2780 020342 040046
2781 020344 044446 052116 051105
2782 020352 040506 042503 040440
2783 020360 042104 042522 051523
2784 020366 020072 100
2785 020371 046 044124 052101
2786 020376 053440 047117 052047
2787 020404 042040 027117 052040
2788 020412 054522 040440 040507
2789 020420 047111 040046
2790 020424 044446 052116 051105
2791 020432 040506 042503 053040
2792 020440 041505 047524 035122
2793 020446 040040
2794 020450 044446 052116 051105
2795 020456 040506 042503 050040
2796 020464 044522 051117 052111
2797 020472 020131 032050 052040
2798 020500 020117 024467 020072
2799 020506 100
2800 020507 046 046503 026522

.ASCII /&& *** CAUTION ***

.ASCII /&THIS PROGRAM IS CAPABLE OF ALTERING THE STATE/

.ASCII /&OF ANY FIELD EQUIPMENT ATTACHED TO ANY OF THE/

.ASCII /&REMOTES ON THIS CMR-11./

.ASCII /&&DO NOT PROCEED IF FIELD HARDWARE IS STILL ON-LINE &&/

FRAD: .ASCII /&INTERFACE ADDRESS: @/

AGAIN: .ASCII /&THAT WON'T DO. TRY AGAIN&/

DVECT: .ASCII /&INTERFACE VECTOR: @/

DVPRI: .ASCII /&INTERFACE PRIORITY (4 TO 7): @/

DBAID: .ASCII /&CMR-11 DE-BUG AID&/

2801	020514	030461	042040	026505	
2802	020522	052502	020107	044501	
2803	020530	023104	040046		
2804	020534	044446	053116	046101	INVLAD: .ASCII /&INVALID ADDRESS...(IT'S NOT THERE)@/
2805	020542	042111	040440	042104	
2806	020550	042522	051523	027056	
2807	020556	024056	052111	051447	
2808	020564	047040	052117	052040	
2809	020572	042510	042522	040051	
2810	020600	023046	051124	050101	TRP4: .ASCII /&&TRAPPED TO 4 FROM LOCATION @/
2811	020606	042520	020104	047524	
2812	020614	032040	043040	047522	
2813	020622	020115	047514	040503	
2814	020630	044524	047117	040040	
2815	020636	020040	040072		TWCSP: .ASCII / :@/
2816	020642	051446	051127	036440	SWRMSG: .ASCII /&SWR = @/
2817	020650	040040			
2818	020652	052046	050131	020105	TYPCTP: .ASCII /&TYPE CNTRL-P TO CONTINUE@/
2819	020660	047103	051124	026514	
2820	020666	020120	047524	041440	
2821	020674	047117	044524	052516	
2822	020702	040105			
2823	020704	051446	046105	041505	TSTSEL: .ASCII /&SELECT TEST (4=HELP) @/
2824	020712	020124	042524	052123	
2825	020720	024040	036464	042510	
2826	020726	050114	020051	100	
2827	020733	046	042523	042514	HELPM: .ASCII /&SELECT ONE OF THE FOLLOWING:&/
2828	020740	052103	047440	042516	
2829	020746	047440	020106	044124	
2830	020754	020105	047506	046114	
2831	020762	053517	047111	035107	
2832	020770	046			
2833	020771	046	051103	047440	.ASCII /&CR OR 1 = LOGIC TEST/
2834	020776	020122	020061	020075	
2835	021004	047514	044507	020103	
2836	021012	042524	052123		
2837	021016	020046	020040	020040	.ASCII /& 2 = PORT LOOP-BACK TEST/
2838	021024	031040	036440	050040	
2839	021032	051117	020124	047514	
2840	021040	050117	041055	041501	
2841	021046	020113	042524	052123	
2842	021054	020046	020040	020040	.ASCII /& 3 - CMR-11 EXERCISER/
2843	021062	031440	036440	041440	
2844	021070	051115	030455	020061	
2845	021076	054105	051105	044503	
2846	021104	042523	122		
2847	021107	046	020040	020040	.ASCII /& 4 = THIS HELP INFO./
2848	021114	020040	020064	020075	
2849	021122	044124	051511	044040	
2850	021130	046105	020120	047111	
2851	021136	047506	056		
2852	021141	046	047103	051124	.ASCII /&CNTRL-C = TEST SELECT/
2853	021146	026514	020103	020075	
2854	021154	042524	052123	051440	
2855	021162	046105	041505	124	
2856	021167	046	047103	051124	.ASCII /&CNTRL-S = SWITCH OPTION SELECT/

2857	021174	026514	020123	020075	
2858	021202	053523	052111	044103	
2859	021210	047440	052120	047511	
2860	021216	020116	042523	042514	
2861	021224	052103			
2862	021226	041446	052116	046122	.ASCII /&CNTRL-F = CONSOLE FILL COUNT SELECT/
2863	021234	043055	036440	041440	
2864	021242	047117	047523	042514	
2865	021250	043040	046111	020114	
2866	021256	047503	047125	020124	
2867	021264	042523	042514	052103	
2868	021272	041446	052116	046122	.ASCII /&CNTRL-O - DEBUG ROUTINE WHICH ALLOWS/
2869	021300	047455	036440	042040	
2870	021306	041105	043525	051040	
2871	021314	052517	044524	042516	
2872	021322	053440	044510	044103	
2873	021330	040440	046114	053517	
2874	021336	123			
2875	021337	046	020040	020040	.ASCII /& LOADING SCR. PAD AND STARTING/
2876	021344	020040	020040	020040	
2877	021352	047514	042101	047111	
2878	021360	020107	041523	027122	
2879	021366	050040	042101	040440	
2880	021374	042116	051440	040524	
2881	021402	052122	047111	107	
2882	021407	046	020040	020040	.ASCII /& FUNCTIONS AND EXAMINING THE/
2883	021414	020040	020040	020040	
2884	021422	052506	041516	044524	
2885	021430	047117	020123	047101	
2886	021436	020104	054105	046501	
2887	021444	047111	047111	020107	
2888	021452	044124	105		
2889	021455	046	020040	020040	.ASCII /& RESULTS/
2890	021462	020040	020040	020040	
2891	021470	042522	052523	052114	
2892	021476	123			
2893	021477	046	047103	051124	.ASCII /&CNTRL-P = PROCEED FROM SW. MONITOR@/
2894	021504	026514	020120	020075	
2895	021512	051120	041517	042505	
2896	021520	020104	051106	046517	
2897	021526	051440	027127	046440	
2898	021534	047117	052111	051117	
2899	021542	100			
2900	021543	046	047514	044507	LGKTSM: .ASCII /&LOGIC TEST@/
2901	021550	020103	042524	052123	
2902	021556	040046			
2903	021560	051446	051103	050056	SCPAD: .ASCII /&SCR.PAD ADDR = @/
2904	021566	042101	040440	042104	
2905	021574	020122	020075	100	
2906	021601	046	044046	051517	VERSM: .ASCII /&HOST FIRMWARE VERSION : @/
2907	021606	020124	044506	046522	
2908	021614	040527	042522	053040	
2909	021622	051105	044523	047117	
2910	021630	035040	040040		
2911	021634	023046	040502	042125	BAUDM: .ASCII &&BAUD RATES:@/
2912	021642	051040	052101	051505	

```
2913 021650 040072
2914 021652 050046 051117 020124 PORT0M: .ASCII /&PORT #0 = @/
2915 021660 030043 036440 040040
2916 021666 050046 051117 020124 PORT1M: .ASCII /&PORT #1 = @/
2917 021674 030443 036440 040040
2918 021702 050046 051117 020124 PORT2M: .ASCII /&PORT #2 = @/
2919 021710 031043 036440 040040
2920 021716 050046 051117 020124 PORT3M: .ASCII /&PORT #3 = @/
2921 021724 031443 036440 040040
2922 021732 042446 042116 050040 ENDPS: .ASCII /&END PASS # @/
2923 021740 051501 020123 020043
2924 021746 100
2925 021747 040 042440 051122 ERCTM: .ASCII / ERR. CNT = @/
2926 021754 020056 047103 020124
2927 021762 020075 100
2928 021765 052 047452 042526 OVFLM: .ASCII /**OVERFLOW**@/
2929 021772 043122 047514 025127
2930 022000 040052
2931 022002 050046 051117 020124 PLTM: .ASCII /&PORT LOOP-BACK TEST&/
2932 022010 047514 050117 041055
2933 022016 041501 020113 042524
2934 022024 052123 040046
2935 022030 052046 050131 020105 PORTN: .ASCII /&TYPE PORT # (<CR>=ALL): @/
2936 022036 047520 052122 021440
2937 022044 024040 041474 037122
2938 022052 040475 046114 035051
2939 022060 040040
2940 022062 052046 050131 020105 TYPGR: .ASCII /&TYPE <CR> WHEN READY: @/
2941 022070 041474 037122 053440
2942 022076 042510 020116 042522
2943 022104 042101 035131 040040
2944 022112 041446 051115 030455 EXCM: .ASCII /&CMR-11 EXERCISE ROUTINE&/
2945 022120 020061 054105 051105
2946 022126 044503 042523 051040
2947 022134 052517 044524 042516
2948 022142 046
2949 022143 046 043450 053111 .ASCII /&(GIVE ME A FEW MINUTES ALONE WITH THE REMOTES)@/
2950 022150 020105 042515 040440
2951 022156 043040 053505 046440
2952 022164 047111 052125 051505
2953 022172 040440 047514 042516
2954 022200 053440 052111 020110
2955 022206 044124 020105 042522
2956 022214 047515 042524 024523
2957 022222 100
2958 022223 046 025052 020052 NOREM: .ASCII /&*** NO REMOTES ON HOST ***&/
2959 022230 047516 051040 046505
2960 022236 052117 051505 047440
2961 022244 020116 047510 052123
2962 022252 025040 025052 046
2963 022257 046 020050 020111 .ASCII /&( I WILL KEEP LOOKING FOR ANY CHANGES )&/
2964 022264 044527 046114 045440
2965 022272 042505 020120 047514
2966 022300 045517 047111 020107
2967 022306 047506 020122 047101
2968 022314 020131 044103 047101
```

2969	022322	042507	020123	023051	
2970	022330	100			
2971	022331	046	047520	052122	PORTMS: .ASCII /&PORT # @/
2972	022336	021440	040040		
2973	022342	051046	046505	052117	REMSG: .ASCII /&REMOTE # @/
2974	022350	020105	020043	100	
2975	022355	046	052123	052101	STATEM: .ASCII /&STATE = @/
2976	022362	020105	020075	100	
2977	022367	046	051525	051117	USORM: .ASCII /&USOR = @/
2978	022374	036440	040040		
2979	022400	025046	025052	051040	REMCHG: .ASCII /&*** REMOTE STATUS CHANGE ***@/
2980	022406	046505	052117	020105	
2981	022414	052123	052101	051525	
2982	022422	041440	040510	043516	
2983	022430	020105	025052	040052	
2984	022436	051046	046505	052117	REMSTM: .ASCII /&REMOTE STATUS NOW @/
2985	022444	020105	052123	052101	
2986	022452	051525	047040	053517	
2987	022460	040040			
2988	022462	051040	043505	046125	REGMG: .ASCII / REGULAR@/
2989	022470	051101	100		
2990	022473	040	051111	042522	IRRGMG: .ASCII / IRREGULAR@/
2991	022500	052507	040514	040122	
2992	022506	025046	025052	040440	ALROFF: .ASCII /&*** ALL REMOTES NOW IRREGULAR ***@/
2993	022514	046114	051040	046505	
2994	022522	052117	051505	047040	
2995	022530	053517	044440	051122	
2996	022536	043505	046125	051101	
2997	022544	025040	025052	040046	
2998					
2999					:ERROR MESSAGES
3000					
3001	022552	044040	051517	020124	NOGO: .ASCII / HOST MICRO FAILED TO START.@/
3002	022560	044515	051103	020117	
3003	022566	040506	046111	042105	
3004	022574	052040	020117	052123	
3005	022602	051101	027124	100	
3006	022607	040	053520	026522	NOTUP: .ASCII / PWR-UP SEQUENCE NOT COMPLETE.(BAD CSR).@/
3007	022614	050125	051440	050505	
3008	022622	042525	041516	020105	
3009	022630	047516	020124	047503	
3010	022636	050115	042514	042524	
3011	022644	024056	040502	020104	
3012	022652	051503	024522	040056	
3013	022660	041440	051123	051040	CSRRST: .ASCII / CSR REG. NOT RESET.@/
3014	022666	043505	020056	047516	
3015	022674	020124	042522	042523	
3016	022702	027124	100		
3017	022705	040	040503	023516	SPRCLR: .ASCII / CAN'T CLR SPAR REG.@/
3018	022712	020124	046103	020122	
3019	022720	050123	051101	051040	
3020	022726	043505	040056		
3021	022732	041040	042101	042040	SPRDAT: .ASCII / BAD DATA IN SPAR REG.@/
3022	022740	052101	020101	047111	
3023	022746	051440	040520	020122	
3024	022754	042522	027107	100	

3025	022761	040	042522	020121	ISATE: .ASCII / REQ SCPD ADR TST ERROR.@/
3026	022766	041523	042120	040440	
3027	022774	051104	052040	052123	
3028	023002	042440	051122	051117	
3029	023010	040056			
3030	023012	051040	051505	046125	OSDTE: .ASCII / RESULT SCPD DATA TST ERROR.@/
3031	023020	020124	041523	042120	
3032	023026	042040	052101	020101	
3033	023034	051524	020124	051105	
3034	023042	047522	027122	100	
3035	023047	040	042522	052523	OSATE: .ASCII / RESULT SCPD ADR TST ERROR.@/
3036	023054	052114	051440	050103	
3037	023062	020104	042101	020122	
3038	023070	051524	020124	051105	
3039	023076	047522	027122	100	
3040	023103	040	042522	020121	ISDTE: .ASCII / REQ SCPD DATA TST ERROR.@/
3041	023110	041523	042120	042040	
3042	023116	052101	020101	051524	
3043	023124	020124	051105	047522	
3044	023132	027122	100		
3045	023135	040	042522	027121	RQSPCL: .ASCII / REQ. SCPD NOT ALL 0'S.@/
3046	023142	051440	050103	020104	
3047	023150	047516	020124	046101	
3048	023156	020114	023460	027123	
3049	023164	100			
3050	023165	040	042522	027123	RSSPCL: .ASCII / RES. SCPD NOT ALL 0'S.@/
3051	023172	051440	050103	020104	
3052	023200	047516	020124	046101	
3053	023206	020114	023460	027123	
3054	023214	100			
3055	023215	040	041442	051114	CLTBER: .ASCII / 'CLRTPB' FAILED TO CLR TOP BYTE.@/
3056	023222	050124	021102	043040	
3057	023230	044501	042514	020104	
3058	023236	047524	041440	051114	
3059	023244	052040	050117	041040	
3060	023252	052131	027105	100	
3061	023257	040	051127	047117	DIERR: .ASCII / WRONG CSR FOR 'DIE' FUNC.@/
3062	023264	020107	051503	020122	
3063	023272	047506	020122	042042	
3064	023300	042511	020042	052506	
3065	023306	041516	040056		
3066	023312	041040	042101	041440	CSRWE: .ASCII / BAD CSR BITS WHILE HUNG.@/
3067	023320	051123	041040	052111	
3068	023326	020123	044127	046111	
3069	023334	020105	052510	043516	
3070	023342	040056			
3071	023344	041440	047101	052047	CSRCE: .ASCII / CAN'T CLR LOW CSR WHILE HUNG.@/
3072	023352	041440	051114	046040	
3073	023360	053517	041440	051123	
3074	023366	053440	044510	042514	
3075	023374	044040	047125	027107	
3076	023402	100			
3077	023403	040	041501	042503	SPAVME: .ASCII / ACCESSED REQ SCPD WITH 'MAIN' SET.@/
3078	023410	051523	042105	051040	
3079	023416	050505	051440	050103	
3080	023424	020104	044527	044124	

3081	023432	021040	040515	047111	
3082	023440	020042	042523	027124	
3083	023446	100			
3084	023447	040	041501	042503	SPAVFE: .ASCII / ACCESSED REQ SCPD WITH 'STFUN' SET.@/
3085	023454	051523	042105	051040	
3086	023462	050505	051440	050103	
3087	023470	020104	044527	044124	
3088	023476	021040	052123	052506	
3089	023504	021116	051440	052105	
3090	023512	040056			
3091	023514	044040	051517	020124	NOINTR: .ASCII / HOST CAN'T INTERRUPT UNIBUS.@/
3092	023522	040503	023516	020124	
3093	023530	047111	042524	051122	
3094	023536	050125	020124	047125	
3095	023544	041111	051525	040056	
3096	023552	042440	051122	047117	ERINTR: .ASCII / ERRONEOUS HOST INTERRUPT.@/
3097	023560	047505	051525	044040	
3098	023566	051517	020124	047111	
3099	023574	042524	051122	050125	
3100	023602	027124	100		
3101	023605	040	051127	047117	WPODI: .ASCII / WRONG PRIORITY. DEV. IS HIGHER.@/
3102	023612	020107	051120	047511	
3103	023620	044522	054524	020056	
3104	023626	042504	027126	044440	
3105	023634	020123	044510	044107	
3106	023642	051105	040056		
3107	023646	050040	047522	020107	HITER: .ASCII / PROG INIT. FAILED TO INIT CSR.@/
3108	023654	047111	052111	020056	
3109	023662	040506	046111	042105	
3110	023670	052040	020117	047111	
3111	023676	052111	041440	051123	
3112	023704	040056			
3113	023706	046440	041511	047522	VISF: .ASCII / MICRO DIDN'T STORE VERSION NO.@/
3114	023714	042040	042111	023516	
3115	023722	020124	052123	051117	
3116	023730	020105	042526	051522	
3117	023736	047511	020116	047516	
3118	023744	040056			
3119	023746	031040	042116	053040	VIDER: .ASCII / 2ND VERSION INFO DIFFERS FROM PWR-UP.@/
3120	023754	051105	044523	047117	
3121	023762	044440	043116	020117	
3122	023770	044504	043106	051105	
3123	023776	020123	051106	046517	
3124	024004	050040	051127	052455	
3125	024012	027120	100		
3126	024015	040	041501	042503	SPAVIC: .ASCII / ACCESSED RES. SCPD WITH INTR CLR.@/
3127	024022	051523	042105	051040	
3128	024030	051505	020056	041523	
3129	024036	042120	053440	052111	
3130	024044	020110	047111	051124	
3131	024052	041440	051114	040056	
3132	024060	053440	047522	043516	STESTF: .ASCII / WRONG CSR WITH 'STFUN'.@/
3133	024066	041440	051123	053440	
3134	024074	052111	020110	051442	
3135	024102	043124	047125	027042	
3136	024110	100			

CZCMJAO CMR-11 HOST EXERCISER
CZCMJA.P11 06-MAR-80 15:13

MACY11 30A(1052) 10-MAR-80 14:09 N 5
CONSTANTS, VARIABLES & ACSII PAGE 57

SEQ 0065

3137	024111	040	046442	044501	RSPEF: .ASCII / 'MAIN' RESULT FUNC. CODE WRONG.@/
3138	024116	021116	051040	051505	
3139	024124	046125	020124	052506	
3140	024132	041516	020056	047503	
3141	024140	042504	053440	047522	
3142	024146	043516	040056		
3143	024152	041040	042101	052440	USOFE1: .ASCII / BAD USOF FROM 'STFUN' ON MAINT.@/
3144	024160	047523	020106	051106	
3145	024166	046517	021040	052123	
3146	024174	052506	021116	047440	
3147	024202	020116	040515	047111	
3148	024210	027124	100		
3149	024213	040	051127	047117	STEMF: .ASCII / WRONG CSR WITH 'MAINT'.@/
3150	024220	020107	051503	020122	
3151	024226	044527	044124	021040	
3152	024234	040515	047111	021124	
3153	024242	040056			
3154	024244	041040	042101	052440	USOFE2: .ASCII / BAD JSOF FROM 'MAINT' FUNC.@/
3155	024252	047523	020106	051106	
3156	024260	046517	021040	040515	
3157	024266	047111	021124	043040	
3158	024274	047125	027103	100	
3159	024301	040	046442	044501	DATLER: .ASCII / 'MAIN' FUNC GAVE BAD DATA LEN.@/
3160	024306	021116	043040	047125	
3161	024314	020103	040507	042526	
3162	024322	041040	042101	042040	
3163	024330	052101	020101	042514	
3164	024336	027116	100		
3165	024341	040	042522	027123	RSPEMF: .ASCII / RES. FUNC CODE WRONG FOR 'MAIN'.@/
3166	024346	043040	047125	020103	
3167	024354	047503	042504	053440	
3168	024362	047522	043516	043040	
3169	024370	051117	021040	040515	
3170	024376	047111	027042	100	
3171	024403	040	040502	020104	USOFE3: .ASCII / BAD USOF FROM 'MAIN' ON REG FUNC.@/
3172	024410	051525	043117	043040	
3173	024416	047522	020115	046442	
3174	024424	044501	021116	047440	
3175	024432	020116	042522	020107	
3176	024440	052506	041516	040056	
3177	024446	053440	047522	043516	USOFE4: .ASCII / WRONG USOF ON REGULAR FUNC.@/
3178	024454	052440	047523	020106	
3179	024462	047117	051040	043505	
3180	024470	046125	051101	043040	
3181	024476	047125	027103	100	
3182	024503	040	047524	040524	PLFFE: .ASCII / TOTAL FAILURE ON 'MAIN' FUNCTION.@/
3183	024510	020114	040506	046111	
3184	024516	051125	020105	047117	
3185	024524	021040	040515	047111	
3186	024532	020042	052506	041516	
3187	024540	044524	047117	040056	
3188	024546	050040	051117	020124	PLTXTO: .ASCII / PORT XMTR TIMED OUT.@/
3189	024554	046530	051124	052040	
3190	024562	046511	042105	047440	
3191	024570	052125	040056		
3192	024574	050040	051117	020124	PLRCTO: .ASCII / PORT RCVR TIMED OUT.@/

3193	024602	041521	051126	052040	
3194	024610	046511	042105	047440	
3195	024616	052125	040056		
3196	024622	050040	051117	026524	PLDERR: .ASCII / PORT-LOOP DETECTED DATA ERROR.@/
3197	024630	047514	050117	042040	
3198	024636	052105	041505	042524	
3199	024644	020104	040504	040524	
3200	024652	042440	051122	051117	
3201	024660	040056			
3202	024662	044040	051517	020124	YSER: .ASCII / HOST SYS ERROR. RUN LOGIC TEST.@/
3203	024670	054523	020123	051105	
3204	024676	047522	027122	051040	
3205	024704	047125	046040	043517	
3206	024712	041511	052040	051505	
3207	024720	027124	100		
3208	024723	040	042042	040511	DIAGF: .ASCII / 'DIAG' FUNC GAVE WRONG FUNC CODE.@/
3209	024730	021107	043040	047125	
3210	024736	020103	040507	042526	
3211	024744	053440	047522	043516	
3212	024752	043040	047125	020103	
3213	024760	047503	042504	040056	
3214	024766	052440	047523	020106	DIAGUF: .ASCII / USOF WRONG ON 'DIAG' FUNC.@/
3215	024774	051127	047117	020107	
3216	025002	047117	021040	044504	
3217	025010	043501	020042	052506	
3218	025016	041516	040056		
3219					
3220					.EVEN
3221					
3222					;TABLE STORAGE
3223					
3224					
3225					;TABLE OF BAUD RATE MESSAGE POINTERS
3226					
3227	025022	025042			BDRTBL: .WORD BD1
3228	025024	025047			.WORD BD2
3229	025026	025054			.WORD BD3
3230	025030	025062			.WORD BD4
3231	025032	025070			.WORD BD5
3232	025034	025076			.WORD BD6
3233	025036	025104			.WORD BD7
3234	025040	025122			.WORD BD8
3235					
3236					;TABLE OF ASCII STORAGE FOR BAUD RATES
3237					
3238	025042	031440	030060	100	BD1: .ASCII / 300@/
3239	025047	040	030066	040060	BD2: .ASCII / 600@/
3240	025054	030440	030062	040060	BD3: .ASCII / 1200@/
3241	025062	031040	030064	040060	BD4: .ASCII / 2400@/
3242	025070	032140	030070	040060	BD5: .ASCII / 4800@/
3243	025076	034440	030066	040060	BD6: .ASCII / 9600@/
3244	025104	021440	044510	044107	BD7: .ASCII / 'HIGH' SPEED@/
3245	025112	020042	050123	042505	
3246	025120	040104			
3247	025122	021040	050123	041505	BD8: .ASCII / 'SPECIAL' SPEED@/
3248	025130	040511	021114	051440	

3249	025136	042520	04211	100	
3250		025144			.EVEN
3251					
3252					;DATA TABLE FOR MAINTENANCE AND DIAGNOSTIC FUNCTION
3253					
3254	025144	000001			DATBL: .WORD 001
3255	025146	000003			.WORD 003
3256	025150	000007			.WORD 007
3257	025152	000017			.WORD 017
3258	025154	000037			.WORD 037
3259	025156	000077			.WORD 077
3260	025160	000177			.WORD 177
3261	025162	000377			.WORD 377
3262	025164	000376			.WORD 376
3263	025166	000374			.WORD 374
3264	025170	000370			.WORD 370
3265	025172	000360			.WORD 360
3266	025174	000340			.WORD 340
3267	025176	000300			.WORD 300
3268	025200	000200			.WORD 200
3269	025202	000000			.WORD 000
3270					
3271					
3272					;PORT CONFIGURATION STATUS TABLES.
3273					
3274	025204	000100			EXTBL1: .BLKW 64.
3275	025404	000100			EXTBL2: .BLKW 64.
3276	025604	000100			EXTBL3: .BLKW 64.
3277	026004	000100			EXTBL4: .BLKW 64.
3278					
3279					;TABLE FOR ORIGINAL (PWR-UP) VERSION DATA
3280					
3281	026204	000011			VERSPL: .BLKW 9.
3282					
3283					;TABLE FOR 2ND VERSION DATA
3284					
3285	026226	000011			VERSUN: .BLKW 9.
3286					
3287					;SOFTWARE FLAG BYTES
3288					
3289	026250	000			!FLG: .BYTE 0
3290	026251	000			FLG: .BYTE 0
3291	026252	000			GOFLG: .BYTE 0
3292	026253	000			INCF: .BYTE 0
3293	026254	000			IFLOF: .BYTE 0
3294	026255	000			SAFE: .BYTE 0
3295	026256	000			SLFL: .BYTE 0
3296	026257	000			CONFLG: .BYTE 0
3297	026260	000			ONCE: .BYTE 0
3298	026261	000			THRU: .BYTE 0
3299					.EVEN
3300					
3301					;VARIABLES:
3302					
3303	026262	000000			ADDR: .WORD 0
3304	026264	000000			ALLFLG: .WORD 0

3305	026266	000000	BAUD:	.WORD	0
3306	026270	000000	CHNGFL:	.WORD	0
3307	026272	000000	CLK:	.WORD	0
3308	026274	000000	CNTR:	.WORD	0
3309	026276	000000	CSR:	.WORD	0
3310	026300	000000	DEVER:	.WORD	0
3311	026302	000000	ERRCNT:	.WORD	0
3312	026304	000000	ITER:	.WORD	0
3313	026306	000000	LINLEN:	.WORD	0
3314	026310	000000	MASK:	.WORD	0
3315	026312	000000	PASCNT:	.WORD	0
3316	026314	000000	PAT:	.WORD	0
3317	026316	000000	PCONT:	.WORD	0
3318	026320	000000	PORN:	.WORD	0
3319	026322	000000	PORT:	.WORD	0
3320	026324	000000	PRI0:	.WORD	0
3321	026326	000000	REMFLG:	.WORD	0
3322	026330	000000	REMNO:	.WORD	0
3323	026332	000000	REMTN:	.WORD	0
3324	026334	000000	REMST:	.WORD	0
3325	026336	000000	SAVEC:	.WORD	0
3326	026340	000000	SPAR:	.WORD	0
3327	026342	000000	SPDR:	.WORD	0
3328	026344	000000	SR:	.WORD	0
3329	026346	000000	SWREG:	.WORD	0
3330	026350	000000	SWRFLG:	.WORD	0
3331	026352	000000	TALLY:	.WORD	0
3332	026354	000000	TBLAD:	.WORD	0
3333	026356	000000	TEMP:	.WORD	0
3334	026360	000000	VERFLG:	.WORD	0
3335	026362	000000	WORD:	.WORD	0
3336					
3337					
3338			;CONSTANTS:		
3339					
3340	026364	000170	VECTOR:	.WORD	170
3341	026366	164000	DEVADR:	.WORD	164000
3342	026370	000004	FKPRI0:	.WORD	4
3343	026372	177562	KBD:	.WORD	177562
3344	026374	177560	KBS:	.WORD	177560
3345	026376	177546	LCS:	.WORD	177546
3346	026400	177566	TTB:	.WORD	177566
3347	026402	177564	TTS:	.WORD	177564
3348					
3349	000001		.END		

ST15	005114	871	889#	920			
ST150	005270	915#	916				
ST16	005312	904	924#	939			
ST160	005400	928	940#	955			
ST161	005466	944	956#	974			
ST162	005574	963	975#	993			
ST17	005702	982	997#	1033			
ST170	005726	1005#	1020	1034			
ST170R	006010	1021#	1035				
ST18	006202	1048	1062	1067#	1088		
ST180	006334	1077	1089#	1090	1106		
ST181	006362	1094#	1128				
ST182	006424	1095	1107#	1123			
ST183	006514	1112	1124#				
ST19	006534	1126	1134#	1152			
ST2	003130	564	578#				
ST20	006642	1141	1162#	1193	1209	1225	
ST200	006752	1176#					
ST201	006766	1178#	1182				
ST202	007046	1180	1194#				
ST203	007134	1198	1210#				
ST204	007222	1214	1226#	1245	1261	1277	
ST205	007252	1230#	1234				
ST206	007332	1232	1246#				
ST207	007420	1250	1262#				
ST21	007506	1266	1278#				
ST22	007534	1288#	1315	1331			
ST220	007614	1297#					
ST221	007622	1298#	1299				
ST222	007720	1304	1316#				
ST223	010006	1320	1332#	1351			
ST224	010022	1334#	1335				
ST23	010120	1340	1356#				
ST23A	010132	1357	1359#				
ST230	010166	1366#	1369				
ST24	010502	1358	1424#				
ST25	010520	1422	1425	1428#			
ST26	010552	1427	1431	1434#			
ST3	003214	582	594#				
ST4	003234	597#	601				
ST6	003254	599	608#	622			
ST60	003330	611	623#				
ST61	003344	625#	643	650			
ST62	003454	632	644#				
ST7	003514	645	656#				
ST70	003524	658#	666				
ST71	003576	663	667#				
ST80	003606	669#	690	695			
ST80S	003710	682	688#				
ST81	003716	672	691#				
ST82	003740	692	696#				
ST90	003754	698#	718	722			
ST90S	004050	710	716#				
ST91	004056	700	719#				
SWDMP	016174	489	2248	2373#	2382	2394	
SWHL T	015506	1476	1700	1860	1949	2246#	2259

\$2\$	015106	2137	2149#																	
\$20\$	015266	2152	2177#																	
\$21\$	015300	2167	2180#																	
\$22\$	015316	2182	2184#																	
\$23\$	015350	2189	2191#																	
\$24\$	015404	2198	2200#																	
\$25\$	015366	2192	2196#																	
\$3\$	015154	2156	2159#																	
\$4\$	015162	2158	2160#	2163																
\$5\$	015200	2161	2164#																	
.	= 026404	122#	123	131	133	135	137	139	141	143	145	147	149	151						
		153	155	157	159	161	163	165	167	169	171	173	175	177						
		179	181	183	185	187	189	191	193	195	197	199	201	203						
		205	207	209	211	213	215	217	219	221	223	225	227	229						
		231	233	235	237	239	241	243	245	247	249	251	253	255						
		257	259	261	263	265	267	269	271	273	275	277	279	281						
		283	285	287	289	291	293	295	297	299	301	303	305	307						
		309	311	313	315	317	319	321	323	325	327	329	331	333						
		335	337	339	341	343	345	347	349	351	353	355	357	359						
		361	363	365	367	369	371	373	375	377	379	381	386#	392#						
		552	570	586	615	636	676	704	744	772	816	843	875	909						
		932	948	967	986	1012	1026	1054	1081	1099	1116	1145	1186	1202						
		1218	1238	1254	1270	1308	1324	1344	1470	1551	1572	1593	1618	1694						
		1854	1943	2064	2080	2607#	3250#	3274#	3275#	3276#	3277#	3281#	3285#							

LZCMJAO CMR-11 HOST EXERCISER
CZCMJA.P11 06-MAR-80 15:13

MACY11 30A(1052) 10-MAR-80 14:09 PAGE 73
CROSS REFERENCE TABLE -- MACRO NAMES

B 7

SEQ 0079

DATERR	72#	567	583	612	633	673	701	741	769	813	840	872	906	929	945
	1078	1113	1183	1199	1215	1235	1251	1267	1305	1321	1341	1615	2061	2077	
ERROR	42#	549	964	983	1009	1023	1051	1096	1142	1467	1548	1569	1590	1691	1851
	1940														
HLT	82#	1475	1699	1859	1948	2258									
MTPS	102#	396	481	998	1040	1677									
PNTM	88#	411	414	424	432	441	445	453	459	504	520	536	683	711	751
	779	823	850	1362	1370	1373	1383	1397	1407	1438	1443	1448	1482	1485	1501
	1523	1558	1579	1600	1625	1705	1746	1861	1864	1870	1876	1882	1897	2004	2087
	2093	2099	2129	2145	2171	2238	2287	2304	2309	2322	2327	2373	2378	2386	2541
	2547	2554													
REGRES	116#	2625	2636	2647	2658	2669									
REGSAV	112#	2620	2631	2642	2653	2664									
SCOPE	95#	557	575	591	620	641	688	716	756	784	828	855	880	918	937
	953	972	991	1018	1031	1059	1086	1104	1121	1150	1191	1207	1223	1243	1259
	1275	1313	1329	1349	1563	1584	1605	1630	2069	2105					

. ABS. 026404 000

ERRORS DETECTED: 0

CZCMJA.BIN,CZCMJA.LST/CRF/SOL/NL:TOC=CZCMJA.P1*
RUN-TIME: 9 20 3 SECONDS
RUN-TIME RATIO: 71/34=2.1
CORE USED: 10K (19 PAGES)