

TSV05

TSV05 CTRL LT4
CVTSDCO

COPYRIGHT (c) 1982-84
RH-T1000-MC
FICHE 02 OF 02

APR 1985
digital
Made In USA

TSV05
CVTSDCO
CTRL LT4
FICHE 02 OF 02

NO	DATE	TIME	STATUS	REMARKS
1	1985	10:00	OK	INITIAL TEST
2	1985	10:05	OK	TEST 1
3	1985	10:10	OK	TEST 2
4	1985	10:15	OK	TEST 3
5	1985	10:20	OK	TEST 4
6	1985	10:25	OK	TEST 5
7	1985	10:30	OK	TEST 6
8	1985	10:35	OK	TEST 7
9	1985	10:40	OK	TEST 8
10	1985	10:45	OK	TEST 9
11	1985	10:50	OK	TEST 10
12	1985	10:55	OK	TEST 11
13	1985	11:00	OK	TEST 12
14	1985	11:05	OK	TEST 13
15	1985	11:10	OK	TEST 14
16	1985	11:15	OK	TEST 15
17	1985	11:20	OK	TEST 16
18	1985	11:25	OK	TEST 17
19	1985	11:30	OK	TEST 18
20	1985	11:35	OK	TEST 19
21	1985	11:40	OK	TEST 20
22	1985	11:45	OK	TEST 21
23	1985	11:50	OK	TEST 22
24	1985	11:55	OK	TEST 23
25	1985	12:00	OK	TEST 24
26	1985	12:05	OK	TEST 25
27	1985	12:10	OK	TEST 26
28	1985	12:15	OK	TEST 27
29	1985	12:20	OK	TEST 28
30	1985	12:25	OK	TEST 29
31	1985	12:30	OK	TEST 30
32	1985	12:35	OK	TEST 31
33	1985	12:40	OK	TEST 32
34	1985	12:45	OK	TEST 33
35	1985	12:50	OK	TEST 34
36	1985	12:55	OK	TEST 35
37	1985	13:00	OK	TEST 36
38	1985	13:05	OK	TEST 37
39	1985	13:10	OK	TEST 38
40	1985	13:15	OK	TEST 39
41	1985	13:20	OK	TEST 40
42	1985	13:25	OK	TEST 41
43	1985	13:30	OK	TEST 42
44	1985	13:35	OK	TEST 43
45	1985	13:40	OK	TEST 44
46	1985	13:45	OK	TEST 45
47	1985	13:50	OK	TEST 46
48	1985	13:55	OK	TEST 47
49	1985	14:00	OK	TEST 48
50	1985	14:05	OK	TEST 49
51	1985	14:10	OK	TEST 50
52	1985	14:15	OK	TEST 51
53	1985	14:20	OK	TEST 52
54	1985	14:25	OK	TEST 53
55	1985	14:30	OK	TEST 54
56	1985	14:35	OK	TEST 55
57	1985	14:40	OK	TEST 56
58	1985	14:45	OK	TEST 57
59	1985	14:50	OK	TEST 58
60	1985	14:55	OK	TEST 59
61	1985	15:00	OK	TEST 60
62	1985	15:05	OK	TEST 61
63	1985	15:10	OK	TEST 62
64	1985	15:15	OK	TEST 63
65	1985	15:20	OK	TEST 64
66	1985	15:25	OK	TEST 65
67	1985	15:30	OK	TEST 66
68	1985	15:35	OK	TEST 67
69	1985	15:40	OK	TEST 68
70	1985	15:45	OK	TEST 69
71	1985	15:50	OK	TEST 70
72	1985	15:55	OK	TEST 71
73	1985	16:00	OK	TEST 72
74	1985	16:05	OK	TEST 73
75	1985	16:10	OK	TEST 74
76	1985	16:15	OK	TEST 75
77	1985	16:20	OK	TEST 76
78	1985	16:25	OK	TEST 77
79	1985	16:30	OK	TEST 78
80	1985	16:35	OK	TEST 79
81	1985	16:40	OK	TEST 80
82	1985	16:45	OK	TEST 81
83	1985	16:50	OK	TEST 82
84	1985	16:55	OK	TEST 83
85	1985	17:00	OK	TEST 84
86	1985	17:05	OK	TEST 85
87	1985	17:10	OK	TEST 86
88	1985	17:15	OK	TEST 87
89	1985	17:20	OK	TEST 88
90	1985	17:25	OK	TEST 89
91	1985	17:30	OK	TEST 90
92	1985	17:35	OK	TEST 91
93	1985	17:40	OK	TEST 92
94	1985	17:45	OK	TEST 93
95	1985	17:50	OK	TEST 94
96	1985	17:55	OK	TEST 95
97	1985	18:00	OK	TEST 96
98	1985	18:05	OK	TEST 97
99	1985	18:10	OK	TEST 98
100	1985	18:15	OK	TEST 99
101	1985	18:20	OK	TEST 100
102	1985	18:25	OK	TEST 101
103	1985	18:30	OK	TEST 102
104	1985	18:35	OK	TEST 103
105	1985	18:40	OK	TEST 104
106	1985	18:45	OK	TEST 105
107	1985	18:50	OK	TEST 106
108	1985	18:55	OK	TEST 107
109	1985	19:00	OK	TEST 108
110	1985	19:05	OK	TEST 109
111	1985	19:10	OK	TEST 110
112	1985	19:15	OK	TEST 111
113	1985	19:20	OK	TEST 112
114	1985	19:25	OK	TEST 113
115	1985	19:30	OK	TEST 114
116	1985	19:35	OK	TEST 115
117	1985	19:40	OK	TEST 116
118	1985	19:45	OK	TEST 117
119	1985	19:50	OK	TEST 118
120	1985	19:55	OK	TEST 119
121	1985	20:00	OK	TEST 120
122	1985	20:05	OK	TEST 121
123	1985	20:10	OK	TEST 122
124	1985	20:15	OK	TEST 123
125	1985	20:20	OK	TEST 124
126	1985	20:25	OK	TEST 125
127	1985	20:30	OK	TEST 126
128	1985	20:35	OK	TEST 127
129	1985	20:40	OK	TEST 128
130	1985	20:45	OK	TEST 129
131	1985	20:50	OK	TEST 130
132	1985	20:55	OK	TEST 131
133	1985	21:00	OK	TEST 132
134	1985	21:05	OK	TEST 133
135	1985	21:10	OK	TEST 134
136	1985	21:15	OK	TEST 135
137	1985	21:20	OK	TEST 136
138	1985	21:25	OK	TEST 137
139	1985	21:30	OK	TEST 138
140	1985	21:35	OK	TEST 139
141	1985	21:40	OK	TEST 140
142	1985	21:45	OK	TEST 141
143	1985	21:50	OK	TEST 142
144	1985	21:55	OK	TEST 143
145	1985	22:00	OK	TEST 144
146	1985	22:05	OK	TEST 145
147	1985	22:10	OK	TEST 146
148	1985	22:15	OK	TEST 147
149	1985	22:20	OK	TEST 148
150	1985	22:25	OK	TEST 149
151	1985	22:30	OK	TEST 150
152	1985	22:35	OK	TEST 151
153	1985	22:40	OK	TEST 152
154	1985	22:45	OK	TEST 153
155	1985	22:50	OK	TEST 154
156	1985	22:55	OK	TEST 155
157	1985	23:00	OK	TEST 156
158	1985	23:05	OK	TEST 157
159	1985	23:10	OK	TEST 158
160	1985	23:15	OK	TEST 159
161	1985	23:20	OK	TEST 160
162	1985	23:25	OK	TEST 161
163	1985	23:30	OK	TEST 162
164	1985	23:35	OK	TEST 163
165	1985	23:40	OK	TEST 164
166	1985	23:45	OK	TEST 165
167	1985	23:50	OK	TEST 166
168	1985	23:55	OK	TEST 167
169	1985	00:00	OK	TEST 168
170	1985	00:05	OK	TEST 169
171	1985	00:10	OK	TEST 170
172	1985	00:15	OK	TEST 171
173	1985	00:20	OK	TEST 172
174	1985	00:25	OK	TEST 173
175	1985	00:30	OK	TEST 174
176	1985	00:35	OK	TEST 175
177	1985	00:40	OK	TEST 176
178	1985	00:45	OK	TEST 177
179	1985	00:50	OK	TEST 178
180	1985	00:55	OK	TEST 179
181	1985	01:00	OK	TEST 180
182	1985	01:05	OK	TEST 181
183	1985	01:10	OK	TEST 182
184	1985	01:15	OK	TEST 183
185	1985	01:20	OK	TEST 184
186	1985	01:25	OK	TEST 185
187	1985	01:30	OK	TEST 186
188	1985	01:35	OK	TEST 187
189	1985	01:40	OK	TEST 188
190	1985	01:45	OK	TEST 189
191	1985	01:50	OK	TEST 190
192	1985	01:55	OK	TEST 191
193	1985	02:00	OK	TEST 192
194	1985	02:05	OK	TEST 193
195	1985	02:10	OK	TEST 194
196	1985	02:15	OK	TEST 195
197	1985	02:20	OK	TEST 196
198	1985	02:25	OK	TEST 197
199	1985	02:30	OK	TEST 198
200	1985	02:35	OK	TEST 199
201	1985	02:40	OK	TEST 200
202	1985	02:45	OK	TEST 201
203	1985	02:50	OK	TEST 202
204	1985	02:55	OK	TEST 203
205	1985	03:00	OK	TEST 204
206	1985	03:05	OK	TEST 205
207	1985	03:10	OK	TEST 206
208	1985	03:15	OK	TEST 207
209	1985	03:20	OK	TEST 208
210	1985	03:25	OK	TEST 209
211	1985	03:30	OK	TEST 210
212	1985	03:35	OK	TEST 211
213	1985	03:40	OK	TEST 212
214	1985	03:45	OK	TEST 213
215	1985	03:50	OK	TEST 214
216	1985	03:55	OK	TEST 215
217	1985	04:00	OK	TEST 216
218	1985	04:05	OK	TEST 217
219	1985	04:10	OK	TEST 218
220	1985	04:15	OK	TEST 219
221	1985	04:20	OK	TEST 220
222	1985	04:25	OK	TEST 221
223	1985	04:30	OK	TEST 222
224	1985	04:35	OK	TEST 223
225	1985	04:40	OK	TEST 224
226	1985	04:45	OK	TEST 225
227	1985	04:50	OK	TEST 226
228	1985	04:55	OK	TEST 227
229	1985	05:00	OK	TEST 228
230	1985	05:05	OK	TEST 229
231	1985	05:10	OK	TEST 230
232	1985	05:15	OK	TEST 231
233	1985	05:20	OK	TEST 232
234	1985	05:25	OK	TEST 233
235	1985	05:30	OK	TEST 234
236	1985	05:35	OK	TEST 235
237	1985	05:40	OK	TEST 236
238	1985	05:45	OK	TEST 237
239	1985	05:50	OK	TEST 238
240	1985	05:55	OK	TEST 239
241	1985	06:00	OK	TEST 240
242	1985	06:05	OK	TEST 241
243	1985	06:10	OK	TEST 242
244	1985	06:15	OK	TEST 243
245	1985	06:20	OK	TEST 244
246	1985	06:25	OK	TEST 245
247	1985	06:30	OK	TEST 246
248	1985	06:35	OK	TEST 247
249	1985	06:40	OK	TEST 248
250	1985	06:45	OK	TEST 249
251	1985	06:50	OK	TEST 250
252	1985	06:55	OK	TEST 251
253	1985	07:00	OK	TEST 252
254	1985	07:05	OK	TEST 253
255	1985	07:10	OK	TEST 254
256	1985	07:15	OK	TEST 255
257	1985	07:20	OK	TEST 256
258	1985	07:25	OK	TEST 257
259	1985	07:30	OK	TEST 258
260	1985	07:35	OK	TEST 259
261	1985	07:40	OK	TEST 260
262	1985	07:45	OK	TEST 26

.REM

IDENTIFICATION

PRODUCT ID: AC-T099C-MC
PRODUCT TITLE: CV:SDCO TSV05 CTRL LT4
DECO/DEPO: 1.0
DEPARTMENT: COMPUTER SPECIAL SYSTEMS/PPG
DATE: JUNE 4, 1984

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1982,1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	COMMANDS
2.2	SWITCHES
2.3	FLAGS
2.4	HARDWARE QUESTIONS
2.5	SOFTWARE QUESTIONS
2.6	EXTENDED P-TABLE DIALOGUE
2.7	QUICK STARTUP PROCEDURE
3.0	ERROR INFORMATION
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES
7.0	MAINTENANCE HISTORY

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS A PDP-11/23 RESIDENT DIAGNOSTIC WHICH CHECKS THE FUNCTIONALITY OF A TSV05 MAGTAPE SUBSYSTEM WHILE CONNECTED TO A PDP-11/23 SYSTEM (Q-BUS). THE PROGRAM PROVIDES ERROR MESSAGES WHICH IDENTIFY FAILING FUNCTIONS THAT AID IN THE REPAIR OF THE DEVICE. THIS DIAGNOSTIC CONSIST OF EIGHT TEST WHICH ARE EXECUTED IN SEQUENCE.

THIS DIAGNOSTIC HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP., ACT, APT, SLIDE AND PAPER TAPE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP. USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

PDP-11/23 PROCESSOR AND MEMORY
CAUTION:DIAGNOSTIC REQUIRES 32K WORDS OF MEMORY
(28K USEABLE AND 4K RESERVED FOR I/O PAGE)
TSV05 MAGTAPE SUBSYSTEM (DRIVE AND CONTROLLER)
CONSOLE TERMINAL
PDP-11 DIAGNOSTIC SUPERVISOR (MSAAA.SYS VERSION 34 OR LATER)
PDP-11 DIAGNOSTIC LOADER/MONITOR (XXDP.)

1.3 RELATED DOCUMENTS AND STANDARDS

DIGITAL EQUIPMENT CORPORATION DOCUMENTS:

1. CHQUS XXDP. USERS MANUAL; DOCUMENT NUMBER AC-F348E-MC
DATE: 14 JULY 1980.
2. TSV05 TRANSPORT SUBSYSTEM USER'S GUIDE; DOCUMENT NUMBER EK-TSV05-UG-001
DATE: AUGUST 1982
3. TSV05 TRANSPORT SUBSYSTEM TECHNICAL MANUAL; DOCUMENT NUMBER EK-TSV05-TM-001
DATE: AUGUST 1982
4. TSV05 TRANSPORT SUBSYSTEM INSTALLATION MANUAL; DOCUMENT NUMBER EK-TSV05-IN 001
DATE: AUGUST 1982

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

FUNCTIONAL PDP-11/23 CENTRAL PROCESSOR AND MEMORY
 FUNCTIONAL CONSOLE TERMINAL
 FUNCTIONAL STANDALONE DIAGNOSTIC SUPERVISOR
 FUNCTIONAL DIAGNOSTIC LOADER/MONITOR (XXDP.)

1.5 ASSUMPTIONS

ALL HARDWARE EXCEPT THE HARDWARE UNDER TEST IS ASSUMED TO WORK PROPERLY OR FALSE ERRORS CAN BE REPORTED.
 THE TAPE BEING USED ON THE TS05 TRANSPORT IS A KNOWN GOOD REEL OF TAPE.
 CVTSAA, CVTSBA AND CVTSCA HAVE SUCCESSFULLY RUN.

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP. USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP. USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ^C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP. MONITOR (XXDP. OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

2.1.1 OPERATOR COMMANDS

THE TSV05 DIAGNOSTIC IS A PDP-11/23 DIAGNOSTIC SUPERVISOR COMPATIBLE PROGRAM. ALL LOADING AND RUNTIME INSTRUCTIONS CAN BE REFERENCED IN THE CHQUS XXDP. USERS MANUAL, DOCUMENT NUMBER AC-F348E MC. THE USER ENTRY IS IN QUOTES.

BOOT THE DIAGNOSTIC XXDP MEDIA

```
.R VTSD??
DIAG. RUN-TIME SERVICES REV D. APR 79
CVTSD-B 0
****TSV05 LOGIC DIAGNOSTIC****
UNIT IS TSV05
>DR
```

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

```
START/TESTS:1-5/PASS:1000/EOP:100
```

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST

*ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A "BELL" ON ERROR, YOU MAY USE THE FOLLOWING STRING:

```
/FLAGS:LOE:IER:BOE
```

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ?" YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 14 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL).

AFTER INITIAL STARTING OF THE PROGRAM (START COMMAND TO THE DIAGNOSTIC SUPERVISOR), THE PROGRAM WILL ISSUE THE "CHANGE HW?" QUESTION TO ASK IF THE HARDWARE PARAMETERS ARE TO BE CHANGED (BY THE OPERATOR).

ON A "N" (NO) RESPONSE TO THE "CHANGE HW?" QUESTION, THE DIAGNOSTIC WILL RUN USING THE DEFAULT VALUES FOR ALL QUESTIONS. THE DEFAULT ADDRESS AND VECTOR ARE:

```
TSBA/TSDB = 172520, VECTOR = 224
```

ON A "Y" (YES) RESPONSE TO THE QUESTION, THE FOLLOWING QUESTIONS WILL THEN BE ASKED TO ALLOW THE OPERATOR TO SELECT THE UNITS TO BE TESTED. A VALUE, IF PRESENT, LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN IF ONLY A CARRIAGE RETURN IS TYPED AS A RESPONSE. A "(D)" IN A QUESTION INDICATES THAT A DECIMAL NUMBER IS REQUIRED AS A RESPONSE. AN "(O)" INDICATES AN OCTAL NUMBER IS BEING SOLICITED. AN "(L)" INDICATES THAT A LOGICAL RESPONSE IS TO BE MADE: "Y" FOR YES, "N" FOR NO.

```
* UNITS (D) ? <ENTER THE NUMBER OF M7196 CONTROLLERS  
PRESENT TO BE TESTED>
```

```
UNIT 0
```

```
DEVICE ADDRESS (O) 172520 ? <ENTER THE ADDRESS OF THE  
TSBA/TSDB REGISTER>
```

```
VECTOR (O) 224 ? <ENTER ADDRESS OF INTERRUPT  
VECTOR>
```

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE "* UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING AS FOLLOWS:
UP TO 4 TSV05 CONTROLLERS PER 11/23 AND UP TO 2 DRIVES PER CONTROLLER

2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR CONTINUE COMMAND, THE RUNTIME SERVICES WILL ASK FOR SOFTWARE PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC OPERATION MODES. YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y". THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

THE FOLLOWING QUESTIONS ARE ASKED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES.

CHANGE SW (L) ? <TYPE Y TO CAUSE THE FOLLOWING
QUESTIONS TO BE ASKED>

INHIBIT ITERATIONS (L) N ? <TYPE "Y" TO PREVENT MULTIPLE
ITERATIONS OF CERTAIN TESTS.
THIS CAUSES EACH TEST PASS TO
RUN AS QUICKLY AS POSSIBLE.
ONLY QUICK-RUNNING LOGIC
TESTS USE MULTIPLE
ITERATIONS.>

2.6 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH UNIT TO BE TESTED. IF YOU HAVE A MULTIPLEXED DEVICE SUCH AS A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF THE ANSWERS ARE REPETITIOUS.

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING A DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT. THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

UNITS (D) ? 8<CR>

UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0<CR>
Q-FACTOR (0) 0 ? 1<CR>

UNIT 2
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 1<CR>
Q-FACTOR (0) 1 ? 0<CR>

UNIT 3
CSR ADDRESS (0) ? 160000<CR>
SUB DEVICE # (0) ? 2<CR>

Q-FACTOR (0) 0 ? <CR>

UNIT 4
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 3<CR>
Q-FACTOR (0) 0 ? <CR>

UNIT 5
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 4<CR>
Q-FACTOR (0) 0 ? <CR>

UNIT 6
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 5<CR>
Q-FACTOR (0) 0 ? <CR>

UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6<CR>
Q-FACTOR (0) 0 ? 1<CR>

UNIT 8
CSR ADDRESS (0) 160000<CR>
SUB-DEVICE # (0) ? 7<CR>
Q-FACTOR (0) 1 ? <CR>

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING MULTIPLE UNITS!

AS YOU CAN SEE FROM THE ABOVE EXAMPLE, THE HARDWARE PARAMETERS DO NOT VARY SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS NOT VERY EFFICIENT.

THE RUNTIME SERVICES CAN TAKE MULTIPLE UNIT SPECIFICATIONS HOWEVER. LET'S BUILD THE SAME TABLE USING THE MULTIPLE SPECIFICATION FEATURE.

UNITS (0) ? 8<CR>

UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0,1<CR>
Q-FACTOR (0) 0 ? 1,0<CR>

UNIT 3
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 2-5<CR>
Q-FACTOR (0) 0 ? 0<CR>

UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6,7<CR>
Q-FACTOR (0) 0 ? 1<CR>

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL

BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS, TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

```
* UNITS (0) ? 8<CR>
```

```
UNIT 1
```

```
CSR ADDRESS (0) ? 160000<CR>
```

```
SUB-DEVICE # (0) ? 0 7<CR>
```

```
Q-FACTOR (0) 0 ? 0.1.0....1.1<CR>
```

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING A NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

2.7 QUICK START-UP PROCEDURE (XXDP.)

TO START UP THIS PROGRAM:

1. BOCT XXDP.
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3). THE GENERAL ERROR MESSAGE IS OF THE FORM:

```
NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:XXXXXX  
ERROR MESSAGE
```

WHERE: NAME = DIAGNOSTIC NAME
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
NUMBER = ERROR NUMBER
UNIT NUMBER = 0 - N (N IS LAST UNIT IN FTABLE)
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS THE "IER" OR "IBR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS PRINTED UNLESS THE "IER", "IBR" OR "IXE" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES

3.2 SPECIFIC ERROR MESSAGES

BELOW ARE SAMPLE ERROR MESSAGES. EACH ERROR MESSAGE REPRESENTS DIFFERENT TYPES OF ERRORS DETECTED BY THIS DIAGNOSTIC.

ERROR MESSAGE EXAMPLE 1

THIS ERROR IS INDICATIVE OF AN INCORRECT REGISTER OR STATUS WORD RETURNED TO THE DIAGNOSTIC. THE FIRST PART DEFINES THE TEST FUNCTION AND UNIT THAT FAILED THE SECOND PART PROVIDES THE REGISTER BITS AND THEIR MNEMONICS FOR THE INCORRECT REGISTER OR STATUS WORDS THE THIRD PART IS THE EXPECTED AND RECEIVED DATA

TST 016 FIFO EXERCISER TEST
CVTSD WRD ERR 01610 ON UNIT 00 TST 016 SUB 002 PC 040624
FIFO STATUS (IN WORD 9) INCORRECT AFTER WRITE FIFO

TAPE BUS SIGNALS IN WORD #8: DESIGNATOR (BIT #)
PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>
IRESV2<14> IIDENT<11> IMER <8> IOML<5> IFB<1>
IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>

TAPE BUS SIGNALS IN WORD #9:
DATHIS<7> ILW<6> OUTRDY<5> INRDY<4>

MESSAGE BUFFER ADDRESS = 047352

MESSAGE BUFFER CONTENTS:

WORD	EXPD	RECV	XOR
WORD #0	EXPD: 100020	RECV: 100020	XOR: 000000
WORD #1	EXPD: 000012	RECV: 000012	XOR: 000000
WORD #2	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #3	EXPD: 000010	RECV: 000010	XOR: 000000
WORD #4	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #5	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #6	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #7	EXPD: 000000	RECV: 000000	XOR: 000000
WORD #8	EXPD: 070217	RECV: 070217	XOR: 000000
WORD #9	EXPD: 000074	RECV: 000034	XOR: 000040

ERROR MESSAGE EXAMPLE 2

THIS ERROR SHOWS A FATAL FUNCTION ERROR FROM THE TAPE DRIVE. IN THIS INSTANCE A UNRECOVERABLE ERROR OCCURED WHICH INDICATES THAT THE CONTROLLER MAY BE DEFECTIVE

CVTSD WRD ERR 00159 ON UNIT 00 TST 001 SUB 005 PC 026202
TSSR NOT CORRECT AFTER SPACE RECORDS COMMAND

TSSR = 100214

TSSR BITS SET: SC.SSR

TERMINATION CLASS CODE = UNRECOVERABLE ERROR

PACKET ADDRESS = 026420

PACKET WORD # = 140010

PACKET WORD # = 000010

PACKET WORD # = 000000

PACKET WORD # = 000024

ERROR MESSAGE EXAMPLE 3

THIS ERROR SHOWS THAT THE MOTION BIT DID NOT GET SET WHILE DOING A REWIND
WITH EXTENDED FEATURES MODE ENABLED.

CVTSD MRD ERR 00121 ON UNIT 00 TST 001 SUB 002 PC: 023306
MOT BIT (XST0) NOT SET DURING REWIND (EXTENDED FEATURES MODE)
EXPD: 000312 RECV: 000112 XOR: 000200

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE TOTAL NUMBER OF ERRORS REPORTED SINCE THE DIAGNOSTIC WAS STARTED. THE "EOP" SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END OF PASS MESSAGE IS PRINTED. SECTION 2.2 DESCRIBES SWITCHES.

SUCCESSFUL RUN EXAMPLE (PDP-11/23)

```
DR>STA/FLA:PNT:MOE
UNITS (0) ? 1
UNIT 0
DEVICE ADDRESS (0) 172520 ? <CR>
VECTOR (0) 224 ? <CR>
CHANGE SW (L) ? N<CR>
```

THE ABOVE COMMAND WILL START THE DIAGNOSTIC. THE COMMAND HAS TWO SWITCHES ON WHICH ARE "PRINT EACH TEST NBR AS EXECUTED" AND "HALT ON ERROR".

```
TST: 001 SKIP TAPE MARKS TEST
TST: 002 NO-OP AND INITIALIZE TEST
TST: 003 ERASE AND OPERATION INCOMPLETE TEST
TST: 004 DATA PARITY TEST
TST: 005 TEST OF OPERATIONS AT EOT TEST
TST: 006 EXTENDED-MODE FUNCTIONS TEST
TST: 007 RECORD BUFFERING TEST
TST: 008 FUNCTION TIMING TEST
```

0 ERRORS

NOTE: THE DIAGNOSTIC WILL RUN CONTINUOUSLY UNLESS A PASS NUMBER LIMIT HAS BEEN SPECIFIED WITH THE "/PASS:" SWITCH.

PROGRAM RUN TIMES

THE AVERAGE RUN TIMES OF THE PROGRAM ARE LISTED BELOW. THESE FIGURES ARE TO BE USED AS A GUIDE. THE TIMING WAS DONE ON A PDP-11/23 PROCESSOR WITH A LA34 CONSOLE.

THE PROGRAM RUNS IN TWO MODES; NO ITERATIONS AND DEFAULT MODE. IN THE NO ITERATIONS MODE, EACH TEST IS RUN ONCE, WITH NO ITERATIONS. IN THE DEFAULT MODE EACH TEST IS REPEATED BY THE NUMBER OF TIMES INDICATED BY THE ITERATION COUNT. NO ITERATIONS MODE IS SELECTED BY ANSWERING THE INHIBIT ITERATIONS QUESTION WITH A "Y" (YES).

TEST NUMBER	N/I SECS.	NUMBER ITER	DEF SECS.
1	1	2	1
2	1	1	0

3	1	1	0
4	1	1	0
5	1	1	0
6	1	1	0
7	1	1	0
8	1	1	0

THE TIMES REQUIRED TO RUN TESTS 1 THROUGH 37 IN ONE COMMAND:

Q.V. 15 SECONDS
 DEFAULT 16 SECONDS

5.0 DEVICE INFORMATION TABLES

WHENEVER THE PROGRAM IS STARTED, VIA THE STA(RT) COMMAND, THE SUPERVISOR REQUESTS THE FOLLOWING P-TABLES PARAMETER CHANGES:

CHANGE MW (L) ?

UNITS (D) ? <ENTER THE NUMBER OF M7196 CONTROLLERS
 PRESENT TO BE TESTED>

UNIT 0

DEVICE ADDRESS (O) 172520 ? <ENTER THE ADDRESS OF THE
 TSBA/TSOB REGISTER>

VECTOR (O) 224 ? <ENTER ADDRESS OF INTERRUPT
 VECTOR>

THE ADDRESS AND VECTOR QUESTIONS WILL BE ASKED FOR EACH OF THE NUMBER OF UNITS (CONTROLLERS) SPECIFIED IN THE "# UNITS?" QUESTION. LOGICAL UNIT NUMBERS ARE ASSIGNED IN ORDER, BEGINNING AT 0. UP TO FOUR UNITS CAN BE SELECTED FOR TESTING.

IN ADDITION, ON A START, RESTART OR CONTINUE THE SUPERVISOR REQUESTS CHANGES TO THE SOFTWARE OPERATING PARAMETERS, AS FOLLOWS:

CHANGE SW (L) ?

INHIBIT ITERATIONS (L) N ?

6.0 TEST SUMMARIES

TEST 1: WRITE TAPE MARK RETRY

THIS TEST VERIFIES PROPER OPERATION OF THE WRITE TAPE MARK RETRY COMMAND (SPACE REVERSE, ERASE, WRITE TAPE MARK).

TEST 2: SKIP TAPE MARKS

THIS TEST VERIFIES PROPER OPERATION OF THE SKIP TAPE MARKS FORWARD AND SKIP TAPE MARKS REVERSE COMMANDS. PROPER OPERATION UNDER CONTROL OF ALL COMBINATIONS OF THE ENABLE SKIP TAPE MARKS STOP (ESS) AND ENABLE TAPE MARKS STOP OFF BOT (ENB) BITS SPECIFIED BY THE WRITE CHARACTERISTICS COMMAND.

TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

THIS TEST VERIFIES PROPER OPERATION OF THE NO-OP ("CLEAN TAPE") AND INITIALIZE COMMAND (SPACE REVERSE, ERASE, WRITE DATA)

TEST 4: ERASE AND OPERATION INCOMPLETE

VERIFIES THAT AN ERASE COMMAND ISSUED WHEN THE TAPE IS POSITIONED AT BOT OPERATES PROPERLY AND ACTUALLY ERASES TAPE.

TEST 5: DATA PARITY TEST

THIS TEST VERIFIES THAT THE DATA PARITY CIRCUITRY IN BOTH THE CONTROLLER AND THE TRANSPORT IS OPERATING PROPERLY BY FORCING DATA RECORDS WITH WRONG PARITY TO BE WRITTEN ONTO TAPE AND CHECKING THE RESULTS OBTAINED WHEN THE DATA IS READ.

TEST 6: OPERATIONS AT EOT

THIS TEST VERIFIES PROPER OPERATION OF THE WRITE DATA RETRY COMMAND (SPACE REVERSE, ERASE, WRITE DATA)

TEST 7: EXTENDED MODE FEATURES

THIS TEST VERIFIES THE OPERATION OF COMMANDS ONLY AVAILABLE WHEN THE CONTROLLER IS IN THE EXTENDED FEATURES MODE. THESE COMMANDS ARE:

REWIND WITH IMMEDIATE INTERRUPT

IF THE CONTROLLER IS NOT ALREADY IN EXTENDED FEATURES MODE, IT IS PLACED THERE VIA A WRITE SUBSYSTEM MEMORY COMMAND.

TEST 8: RECORD BUFFERING

THIS TEST VERIFIES THAT RECORD BUFFERING, USED FOR WRITE DATA AND READ NEXT COMMANDS, OPERATES PROPERLY AND IS PROPERLY CONTROLLED BY THE EXTENDED CHARACTERISTICS DATA WORD. IF THE M7196 CONTROLLER MODULE IS NOT ALREADY IN EXTENDED FEATURES MODE (AS CONTROLLED BY THE DIP SWITCH ON THE MODULE), IT IS PLACED INTO THAT MODE BY INVERTING THE SENSE OF THE SWITCH USING THE WRITE SUBSYSTEM MEMORY COMMAND. NOTE THAT RECORD BUFFERING HAS BEEN ENABLED IN PREVIOUS TESTS OF READ AND WRITE AND SO HAS BEEN PARTIALLY TESTED ALREADY. THIS TEST VERIFIES THAT BUFFERING IS ACTUALLY OPERATING.

TEST 9: FUNCTION TIMING

THIS TEST VERIFIES THAT THE TAPE TRANSPORT SEEMS TO BE WRITING RECORDS, GAPS, AND EXTENDED GAPS OF THE PROPER LENGTH. BOTH LOW AND HIGH SPEED MODES ARE TESTED. IT IS ALSO VERIFIED THAT A SKIP TAPE MARKS COMMAND WITH A COUNT OF 6 OR MORE, OPERATE THE TAPE IN HIGH-SPEED MODE. THIS TEST CAN ONLY BE RUN IF A REAL-TIME CLOCK IS AVAILABLE ON THE SYSTEM. THE TEST OPERATES BY TIMING VARIOUS TAPE-MOTION OPERATIONS, USING A NUMBER OF DIFFERENT TEST RECORD LENGTHS.

7.0 MAINTENANCE HISTORY

REVISION A - MARCH 1982

REVISION B APRIL 1983

- FIXED TWO PROBLEMS, ONE IN TEST 1 AND THE OTHER IN TEST 8.
REF. DOYLE TO GRASKY "TSV05 CVTSDA DIAGNOSTIC PATCH"; 23-DEC-82.

REVISION C - JUNE 1984

MINOR CHANGES FOR "ORION" CPU
ELIMINATED CPU ID MESSAGE.

```

2          .TITLE  TSV2  PROGRAM HEADER
3          .SBTTL  PROGRAM HEADER
4
10         .MCALL  SVC
11 000000  SVC          ; INITIALIZE SUPERVISOR MACROS
12         .ENABLE LC
13         .MLIST  BEX,CND
19 000000  .ENABL  ABS,AMA
20         .*2000
21 002000  BGNMOD  TSV2
   002000
22
23         ;**
24         ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
25         ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
26         ; -
27
28         POINTER BGNSW,BGNSFT,BGNAU,BGN DU,BGNRPT
29 002000  HEADER  CVTSD,C,0,655.,0
   002000  L$NAME::          ;DIAGNOSTIC NAME
   002000      103      .ASCII /C/
   002001      126      .ASCII /V/
   002002      124      .ASCII /T/
   002003      123      .ASCII /S/
   002004      104      .ASCII /D/
   002005      000      .BYTE  0
   002006      000      .BYTE  0
   002007      000      .BYTE  0
   002010  L$REV::          ;REVISION LEVEL
   002010      103      .ASCII /C/
   002011  L$DEPO::          ;0
   002011      060      .ASCII /0/
   002012  L$UNIT::          ;NUMBER OF UNITS
   002012  000000      .WORD  0
   002014  L$TIML::          ;LONGEST TEST TIME
   002014  001217      .WORD  655.
   002016  L$HPCP::          ;PTR. TO H.W. QUES.
   002016  105476      .WORD  L$HARD
   002020  L$SPCP::          ;PTR. TO S.W. QUES.
   002020  105630      .WORD  L$SOFT
   002022  L$HPTP::          ;PTR. TO DEF. H.W. PTABLE
   002022  002150      .WORD  L$HW
   002024  L$SPTP::          ;PTR. TO S.W. PTABLE
   002024  002160      .WORD  L$SW
   002026  L$LADP::          ;DIAG. END ADDRESS
   002026  ...404      .WORD  L$LAST
   002030  L$STA::          ;RESERVED FOR APT STATS
   002030  000000      .WORD  0
   002032  L$CO::          ;
   002032  000000      .WORD  0
   002034  L$DTYP::          ;DIAGNOSTIC TYPE
   002034  000000      .WORD  0
   002036  L$APT::          ;APT EXPANSION
   002036  000000      .WORD  0
   002040  L$DTP::          ;PTR. TO DISPATCH TABLE
   002040  002124      .WORD  L$DISPATCH
   002042  L$PRIO::          ;DIAGNOSTIC RUN PRIORITY

```

TSV2 - PROGRAM HEADER
PROGRAM HEADER

MACRO M1113 14 JUN-84 16:41

SEQ 0019

002042	000000		.WORD	0	
002044		L\$ENVI::	.WORD	0	;FLAGS DESCRIBE HOW IT WAS SETUP
002044	000000		.WORD	0	
002046		L\$EXP1::	.WORD	0	;EXPANSION WORD
002046	000000		.WORD	0	
002050		L\$MREV::	.WORD	0	;SVC REV AND EDIT #
002050	003		.BYTE	C\$REVISION	
002051	003		.BYTE	C\$EDIT	
002052		L\$EF::	.WORD	0	;DIAG. EVENT FLAGS
002052	000000		.WORD	0	
002054	000000		.WORD	0	
002056		L\$SPC::	.WORD	0	
002056	000000		.WORD	0	
002060		L\$DEVP::	.WORD	L\$DVTYP	; POINTER TO DEVICE TYPE LIST
002060	003374		.WORD	L\$DVTYP	
002062		L\$REPP::	.WORD	L\$RPT	;PTR. TO REPORT CODE
002062	022700		.WORD	L\$RPT	
002064		L\$EXP4::	.WORD	0	
002064	000000		.WORD	0	
002066		L\$EXP5::	.WORD	0	
002066	000000		.WORD	0	
002070		L\$AUT::	.WORD	L\$AU	;PTR. TO ADD UNIT CODE
002070	022366		.WORD	L\$AU	
002072		L\$DUT::	.WORD	L\$DU	;PTR. TO DROP UNIT CODE
002072	022464		.WORD	L\$DU	
002074		L\$LUN::	.WORD	0	;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD	0	
002076		L\$DESP::	.WORD	L\$DESC	;POINTER TO DIAG. DESCRIPTION
002076	003402		.WORD	L\$DESC	
002100		L\$LOAD::	EMT	E\$LOAD	;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		EMT	E\$LOAD	
002102		L\$ETP::	.WORD	0	;POINTER TO ERR_TBL
002102	000000		.WORD	0	
002104		L\$ICP::	.WORD	L\$INIT	;PTR. TO INIT CODE
002104	021572		.WORD	L\$INIT	
002106		L\$CCP::	.WORD	L\$CLEAN	;PTR. TO CLEAN-UP CODE
002106	022652		.WORD	L\$CLEAN	
002110		L\$ACP::	.WORD	L\$AUTO	;PTR. TO AUTO CODE
002110	022572		.WORD	L\$AUTO	
002112		L\$PRT::	.WORD	L\$PROT	;PTR. TO PROTECT TABLE
002112	021562		.WORD	L\$PROT	
002114		L\$TEST::	.WORD	0	;TEST NUMBER
002114	000000		.WORD	0	
002116		L\$DLY::	.WORD	0	;DELAY COUNT
002116	000000		.WORD	0	
002120		L\$HIME::	.WORD	0	;PTR. TO HIGH MEM
002120	000000		.WORD	0	

TSV2 - PROGRAM HEADER
DISPATCH TABLE

MACRO M1113 14-JUN-84 16:41

H,

SEQ 0020

.SBTTL DISPATCH TABLE

; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
; -

DISPATCH 9

.WORD 9

L#DISPATCH:;

.WORD T1

.WORD T2

.WORD T3

.WORD T4

.WORD T5

.WORD T6

.WORD T7

.WORD T8

.WORD T9

31
32
33
34
35
36
37
38 002122
002122 000011
002124
002124 023462
002126 032264
002130 041362
002132 046720
002134 052776
002136 055772
002140 063344
002142 073274
002144 101050
39

```
41          .SBTTL  DEFAULT HARDWARE P TABLE
42
43          ;**
44          ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
45          ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
46          ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
47          ;
48          BGNHW  DFPTBL          ;DEFAULT HARD-P-TABLE
          .WORD  L10000-L$HW/2
          L$HW::
          DFPTBL::
49
50          .WORD  172520          ; 1ST (OF 2) REGISTERS.
51          .WORD  224            ; INTERRUPT VECTOR
52          .WORD  PRI04          ; INTERRUPT PRIORITY.
53          ENDPW
          L:0000:
```

```
55          .SBTTL SOFTWARE P TABLE
56
57          ;**
58          ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
59          ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
60          ;
61          BGNSW   SFPTBL
           .WORD   L10001-L$SW/2
           L$SW::
           SFPTBL::
62
63          TRANSTST::      .WORD   0           ; ENABLE TEST OF TRANSPORT(S) IF =1
64          NOITS::        .WORD   0           ; INHIBIT ITERATION OPTION.
65                                     ; ... 0 = ITERATE.
66                                     ; ...NZ = INHIBIT ITERATE.
67          LERRMAX::      .WORD   15.        ; LOCAL (PER TEST) ERROR LIMIT
68          GERRMAX::      .WORD   200.       ; GLOBAL (PER UNIT) ERROR LIMIT
69          ENDSW
           L10001:
70
71          ENDMOD
```


7
8
13
19
20 002170
002170
21
22
23
24
25
26
27
28
32 002170

.TITLE TSV3 - GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD TSV3
TSV3::

.SBTTL GLOBAL EQUATES SECTION

; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
; ARE USED IN MORE THAN ONE TEST.
;

EQUALS ; GET STANDARD EQUATES.

; BIT DIFINITIONS

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1

001000	BIT9== BIT09
000400	BIT8== BIT08
000200	BIT7== BIT07
000100	BIT6== BIT06
000040	BIT5== BIT05
000020	BIT4== BIT04
000010	BIT3== BIT03
000004	BIT2== BIT02
000002	BIT1== BIT01
000001	BIT0== BIT00

; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040	EF.START== 32.	; START COMMAND WAS ISSUED
000037	EF.RESTART== 31.	; RESTART COMMAND WAS ISSUED
000036	EF.CONTINUE== 30.	; CONTINUE COMMAND WAS ISSUED
000035	EF.NEW== 29.	; A NEW PASS HAS BEEN STARTED
000034	EF.PWR== 28.	; A POWER-FAIL/POWER-UP OCCURRED

; PRIORITY LEVEL DEFINITIONS

```

000340      PRI07== 340
000300      PRI06== 300
000240      PRI05== 240
000200      PRI04== 200
000140      PRI03== 140
000100      PRI02== 100
000040      PRI01== 40
000000      PRI00== 0
    
```

```

; OPERATOR FLAG BITS
;
000004      EVL==      4
000010      LOT==     10
000020      ADR==     20
000040      IDU==     40
000100      ISR==    100
000200      UAM==    200
000400      BOE==    400
001000      PNT==   1000
002000      PRI==   2000
004000      IXE==   4000
010000      IBE==  10000
020000      IER==  20000
040000      LOE==  40000
100000      HOE== 100000
    
```

33
34 002170

```

;DEFINE MEMORY MANAGEMENT REGISTERS
KT11
.SBTTL MEMORY MANAGEMENT DEFINITIONS
;*KT11 VECTOR ADDRESS
000250      MMVEC= 250
;*KT11 STATUS REGISTER ADDRESSES
177572      SR0= 177572
177574      SR1= 177574
177576      SR2= 177576
172516      SR3= 172516
;IF NB
;*USER "I" PAGE DESCRIPTOR REGISTERS
UIPDR0= 177600
UIPDR1= 177602
UIPDR2= 177604
UIPDR3= 177606
UIPDR4= 177610
UIPDR5= 177612
UIPDR6= 177614
UIPDR7= 177616
;IF NB
;*USER "D" PAGE DESCRIPTOR REGISTERS
UDPDR0= 177620
UDPDR1= 177622
UDPDR2= 177624
UDPDR3= 177626
UDPDR4= 177630
UDPDR5= 177632
UDPDR6= 177634
UDPDR7= 177636
.ENDC
    
```

;*USER "I" PAGE ADDRESS REGISTERS

UIPAR0= 177640
UIPAR1= 177642
UIPAR2= 177644
UIPAR3= 177646
UIPAR4= 177650
UIPAR5= 177652
UIPAR6= 177654
UIPAR7= 177656

.IF NB

;*USER "D" PAGE ADDRESS REGISTERS

UDPAR0= 177660
UDPAR1= 177662
UDPAR2= 177664
UDPAR3= 177666
UDPAR4= 177670
UDPAR5= 177672
UDPAR6= 177674
UDPAR7= 177676

.ENDC

.ENDC

.IF NB

;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS

SIPDR0= 172200
SIPDR1= 172202
SIPDR2= 172204
SIPDR3= 172206
SIPDR4= 172210
SIPDR5= 172212
SIPDR6= 172214
SIPDR7= 172216

.IF NB

;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS

SDPDR0= 172220
SDPDR1= 172222
SDPDR2= 172224
SDPDR3= 172226
SDPDR4= 172230
SDPDR5= 172232
SDPDR6= 172234
SDPDR7= 172236

.ENDC

;*SUPERVISOR "I" PAGE ADDRESS REGISTERS

SIPAR0= 172240
SIPAR1= 172242
SIPAR2= 172244
SIPAR3= 172246
SIPAR4= 172250
SIPAR5= 172252
SIPAR6= 172254
SIPAR7= 172256

.IF NB

;*SUPERVISOR "D" PAGE ADDRESS REGISTERS

SDPAR0= 172260
SDPAR1= 172262
SDPAR2= 172264
SDPAR3= 172266

```
SDPAR4= 172270
SDPAR5= 172272
SDPAR6= 172274
SDPAR7= 172276
.ENDC
.ENDC
;*KERNEL "I" PAGE DESCRIPTOR REGISTERS
172300 KIPDR0= 172300
172302 KIPDR1= 172302
172304 KIPDR2= 172304
172306 KIPDR3= 172306
172310 KIPDR4= 172310
172312 KIPDR5= 172312
172314 KIPDR6= 172314
172316 KIPDR7= 172316
.IF NB
;*KERNEL "D" PAGE
DESCRIPTOR REGISTERS
KDPDR0= 172320
KDPDR1= 172322
KDPDR2= 172324
KDPDR3= 172326
KDPDR4= 172330
KDPDR5= 172332
KDPDR6= 172334
KDPDR7= 172336
.ENDC
;*KERNEL "I" PAGE ADDRESS REGISTERS
172340 KIPAR0= 172340
172342 KIPAR1= 172342
172344 KIPAR2= 172344
172346 KIPAR3= 172346
172350 KIPAR4= 172350
172352 KIPAR5= 172352
172354 KIPAR6= 172354
172356 KIPAR7= 172356
.IF NB
;*KERNEL "D" PAGE ADDRESS REGISTERS
KDPAR0= 172360
KDPAR1= 172362
KDPAR2= 172364
KDPAR3= 172366
KDPAR4= 172370
KDPAR5= 172372
KDPAR6= 172374
KDPAR7= 172376
.ENDC
```

```

39          .SBTTL  TSV05 REGISTER AND PACKET DEFINITIONS
40
41          ;
42          ; SOME GENERAL EQUATES.
43          ;
44
45          000004      ERRVEC==      4          ; PCINTER TO ERROR VECTOR FOR BUS TIME OUT.
46          000060      TTIVEC==     60          ; INTERRUPT VECTOR FOR CONSOLE INPUT
47          177560      TTICSR==     177560      ; BUS ADDRESS OF CONSOLE INPUT
48          177562      TTIBFR==     177562      ; CONSOLE INPUT DATA BUFFER
49          177520      BDVPCR==     177520      ; BDV11 PAGE CONTROL REGISTER
50
51          ;
52          ;BIT DEFINITIONS FOR TSSR REGISTER
53          ;
54
55          100000      SC=      BIT15      ;SPECIAL CONDITION
56          040000      BIE=     BIT14      ;BUS INTERFACE ERROR
57          020000      SCE=     BIT13      ;SANITY CHECK ERROR
58          010000      RMR=     BIT12      ;MODIFICATION REFUSED
59          004000      NXM=     BIT11      ;NONEXISTANT MEMORY ERROR
60          002000      NBA=     BIT10      ;NEED BUFFER ADDRESS
61          001400      HIAADR= BIT9:BIT8   ;EXTENDED ADDRESS BITS
62          000200      SSR=     BIT7       ;SUB SYSTEM READY
63          000100      OFL=     BIT6       ;OFF LINE BIT
64          000060      FATERR= BIT4:BITS   ;FATAL TERMINATION ERROR CODES
65          000016      TERCLS= BIT3:BIT2:BIT1 ;TERMINATION CODES
66
67          ;
68          ;
69          ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 0
70          ;(XST0)
71          ;
72          ;
73
74          100000      XSOTMK= BIT15      ;TAPE MARK DETECTED
75          040000      XSORLS= BIT14      ;RECORD LENGTH SHORT
76          020000      XSOLET= BIT13      ;LOGICAL END OF TAPE
77          010000      XSORLL= BIT12      ;RECORD LENGTH LONG
78          004000      XSOWLE= BIT11      ;WRITE LOCK ERROR
79          002000      XSONEF= BIT10      ;NON EXECUTABLE FUNCTION
80          001000      XSOILC= BIT9       ;ILLEGAL COMMAND
81          000400      XSOILA= BIT8       ;ILLEGAL ADDRESS
82          000200      XSOMOT= BIT7       ;TAPE IN MOTION
83          000100      XSOONL= BIT6       ;TRANSPORT ON LINE
84          000040      XSOIE=  BITS      ;INTERRUPT ENABLE
85          000020      XSOVCK= BIT4       ;VOLUME CHECK BIT
86          000010      XSOPED= BITS      ;PHASE ENCODED DRIVE
87          000004      XSOWLK= BIT2       ;WRITE LOCKED
88          000002      XS0BOT= BIT1       ;BEGINNING OF TAPE
89          000001      XS0EOT= BIT0       ;END OF TAPE

```

```

91      ;
92      ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 1
93      ;(XST1)
94      ;
95      100000 X1.DLT = BIT15          ;DATA LATE
96      040000 X1.SPARE= BIT14          ;NOT USED
97      020000 X1.COR = BIT13          ;CORRECTABLE DATA ERROR
98      017375 X1.MBZ = BIT12·BIT11·BIT10·BIT9·BIT7·BIT6·BIT5·BIT4·BIT3·BIT2·BIT0 ;ALWAYS 0
99      000400 X1.RBP = BIT8          ;READ BUS PARITY ERROR
100     000002 X1.UNC = BIT1          ;UNCORRECTABLE DATA OR HARD ERROR
101
102     ;
103     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 2
104     ;(XST2)
105     ;
106     100000 X2.OPM = BIT15          ;OPERATION IN PROGRESS (TAPE MOVING)
107     040000 X2.RCE = BIT14          ;RAM CHECKSUM ERROR
108     035400 X2.SPARE= BIT13·BIT12·BIT11·BIT9·BIT8 ;NOT USED BY TSV05 (ALWAYS=0)
109     002000 X2.WCF = BIT10          ;WRITE CLOCK FAILURE (FIFO NOT EMPTIED BY TRANSPORT)
110     000200 X2.EXTF = BIT7          ;IF WRITE CHAR CMD THEN = EXTENDED FEATURES ENABLED
111     000100 X2.BUFE = BIT6          ;IF WRITE CHAR CMD THEN = BUFFERING ENABLED
112     000077 X2.REV = 000077        ;IF WRITE CHAR CMD THEN = MICROCODE REVISION LEVEL
113     000007 X2.UNIT = BIT2·BIT1·BIT0 ;IF GET STATUS THEN = CURRENTLY SELECTED UNIT NO.
114
115     ;
116     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 3
117     ;(XST3)
118     ;
119     177400 X3.MDE = 177400          ;MICRO-DIAGNOSTIC ERROR CODE
120     000200 X3.SPARE= BIT7          ;NOT USED BY TSV05
121     000100 X3.OPI = BIT6          ;OPERATION INCOMPLETE
122     000040 X3.REV = BIT5          ;REVERSE
123     000020 X3.TRF = BIT4          ;TRANSPORT RESPONSE FAILURE
124     000010 X3.DCK = BIT3          ;DENSITY CHECK
125     000006 X3.MBZ =BIT2·BIT1      ;NOT USED ALWAYS 0
126     000001 X3.RIB = BIT0          ;REVERSE INTO BOT
127
128     ;
129     ;BIT DEFINITIONS FOR EXTENDED STATUS REGISTER 4
130     ;(XST4)
131     ;
132     100000 X4.HSP = BIT15          ;HIGH SPEED
133     040000 X4.RCE = BIT14          ;RETRY COUNT EXCEEDED
134     020000 X4.TSM = BIT13          ;TRANSPORT SPECIAL MODE
135     017400 X4.MBZ = BIT12·BIT11·BIT10·BIT9·BIT8 ;NOT USED ALWAYS 0
136     000377 X4.WRC = 000377        ;WRITE RETRY COUNT FIELD
137
138     ;
139     ;
140     ;TSSR TERMINATION CODES (BIT 0 2)
141     ;
142     ;
143     ;
144     000006 TSREJ= 3*2              ;COMMAND REJECTED
145     000006 UNREC= 6                ;UNRECOVERABLE ERROR
    
```

```

147      ;*
148      ;
149      ;DEVICE REGISTER OFFSETS
150      ;
151      ;
152      ;
153      000000      TSBA== 0
154      000000      TSDB== 0      ;TSDB/TSBA REGISTER
155      000001      TSBAH== 1
156      000001      TSDBH== 1      ;TSDB/TSBA REGISTER HIGH BYTE
157      000002      TSSR== 2      ;TSSR REGISTER
158      000003      TSSRH== 3      ;TSSR REGISTER HIGH BYTE
159      ;
160      ;*
161      ; TSDB ADDRESS BIT DEFINITIONS
162      ;
163      000003      A1716 = BIT1:BIT0      ;ADDRESS BITS 17:16 ARE IN 1:0
164      ;
165      ;*
166      ; COMMAND DEFINITIONS
167      ;
168      000017      P.GETSTAT      = 17      ;GET STATUS
169      000013      P.INIT        = 13      ;INITIALIZE
170      000012      P.CONTROL     = 12      ;CONTROL COMMANDS
171      000011      P.FORMAT      = 11      ;FORMAT
172      000010      P.POSITION    = 10      ;POSITION
173      000006      P.WRTSUB      = 6       ;SUBSYSTEM WRITE
174      000005      P.WRITE       = 5       ;WRITE
175      000004      P.WRTCHAR     = 4       ;WRITE CHARACTERISTICS
176      000001      P.READ        = 1       ;READ
177      ;
178      ;*
179      ; COMMAND PACKET HEADER WORD BIT DEFINITIONS
180      ;
181      100000      P.ACK          = BIT15    ;BUFFER AVAIL FOR CONTROLLER
182      040000      P.CVC         = BIT14    ;CLEAR VOLUME CHECK
183      020000      P.OPP         = BIT13    ;REVERSE SEQUENCE OF DATA BITS
184      010000      P.SWB         = BIT12    ;SWAP BYTES IN MEMORY
185      007400      P.MODE        = BIT11:BIT10:BIT9:BIT8 ;EXTENDED COMMAND MODE FIELD
186      000200      P.IE          = BIT7     ;INTERRUPT ENABLE
187      000140      P.FMT         = BIT6:BIT5 ;PACKET HEADER TYPE (ALWAYS=0)
188      000037      P.CMD         = 37      ;MAJOR COMMAND FIELD
189      ;
190      ;*
191      ; CONTROL COMMAND MODE CODES
192      000000      PC.RELEASE     = 0*256.   ;RELEASE BUFFER
193      000400      PC.REWIND     = 1*256.   ;REWIND
194      001000      PC.NOOP       = 2*256.   ;NO-OP
195      002000      PC.IEREW      = 4*256.   ;REWIND IMMEDIATE INTERRUPT
196      002400      PC.ERASE      = 5*256.   ;SECURITY ERASE

```

```

198      ;*
199      ; CONTROLLER RAM DEFINITIONS
200      ;
201      000167      RMCHBEG = 167      ;CHARACTERISTICS IO DATA BEGIN RAM ADDRESS
202      000200      RMCHEND = 200      ;CHARACTERISTICS IO DATA END RAM ADDRESS
203      000201      RMPKTBEG= 201      ;COMMAND PACKET BEGIN RAM ADDRESS
204      000210      RMPKTEND= 210      ;COMMAND PACKET END RAM ADDRESS
205      000215      RMMSGBEG= 215      ;MESSAGE BUFFER BEGIN RAM ADDRESS
206      000234      RMMSGEND= 234      ;MESSAGE BUFFER END RAM ADDRESS
207      ;*
208      ;
209      ;REGISTER DEFINITIONS IN THE MESSAGE BUFFER
210      ;
211      ;
212      ;
213      000006      XST0== 6      ;EXTENDED STATUS REGISTER 0 (WORD 4)
214      000010      XST1== 8      ;EXTENDED STATUS REGISTER 1 (WORD 5)
215      000012      XST2== 10     ;EXTENDED STATUS REGISTER 2 (WORD 6)
216      000014      XST3== 12     ;EXTENDED STATUS REGISTER 3 (WORD 7)
217      000016      XST4== 14     ;EXTENDED STATUS REGISTER 4 (WORD 8)
218      ;
219      ;*
220      ;
221      ;OFFSETS TO WORD LOCATIONS IN PACKET DEFINITIONS
222      ;
223      ;
224      ;
225      000002      PKLOW  = 2      ;LOW ORDER CHARACTERISTIC DATA POINTER
226      000004      PKHI   = 4      ;HIGH ORDER CHARACTERISTIC DATA POINTER
227      000006      PKBCNT = 6      ;NUMBER OF BYTES IN DATA PACKET
228      ;
229      000010      EXBCNT=10      ;NUMBER OF BYTES IN EXTENDED DATA PACKET
230      ;
231      ;*
232      ;DATA PACKET OFFSETS FOR WRITE SUBSYSTEM COMMAND
233      ;
234      000000      BSEL0  = 0      ;BYTE 0
235      000001      BSEL1  = 1      ;BYTE 1
236      000002      SEL2   = 2      ;WORD 2
237      000004      SELDATA = 4      ;WORD 3

```



```

239      ;*
240      ;BSELO SELECT CODES FOR WRITE SUBSYSTEM COMMAND
241      ;-
242      000000      PW.NOP          * 0          ;NO OP
243      000001      PW.RDRAM        * 1          ;READ RAM
244      000002      PW.WTRAM        * 2          ;WRITE RAM
245      000003      PW.RFIFO        * 3          ;READ FIFO
246      000004      PW.WFIFO        * 4          ;WRITE FIFO
247      000005      PW.RDSTAT       * 5          ;READ STATUS
248      000006      PW.WCTL         * 6          ;WRITE TAPE CONTROL
249      000007      PW.WFMT         * 7          ;WRITE TAPE FORMAT
250      000010      PW.WMISC        * 10         ;WRITE MISCELLANEOUS
251      000011      PW.WNPR         * 11         ;WRITE NPR CONTROL
252      000020      PW.D22          * 20         ;DO MICROTEST 22
253      000021      PW.D11          * 21         ;DO MICROTEST 11
254      000022      PW.D13          * 22         ;DO MICROTEST 13
255      000023      PW.NO1311       * 23         ;DISABLE MICROTEST 11 AND 13
256      000024      PW.RDXT         * 24         ;READ EXT. TAPE STATUS (NOT SUPPORTED BY ALL TRANSPORTS)
257
258      ;*
259      ;BSEL1 CODES FOR WRITE TAPE CONTROL
260      ;-
261      000200      WC.IFAD          * BIT7       ;IFAD - FORMATTER ADDRESS
262      000100      WC.IOTAD        * BIT6       ;ITAD0 - TRANSPORT ADDRESS BIT 0
263      000040      WC.I1TAD        * BIT5       ;ITAD1 - TRANSPORT ADDRESS BIT 1
264      000020      WC.ISRESV       * BIT4       ;IRESV5 - RESERVED #5
265      000010      WC.IREW         * BIT3       ;IREW - REWIND
266      000004      WC.IRWU         * BIT2       ;IRWU - REWIND AND UNLOAD
267      000002      WC.IFEN         * BIT1       ;IFEN - FORMATTER ENABLE
268      000001      WC.IGO          * BIT0       ;GO
269
270      ;*
271      ;BSEL1 CODES FOR WRITE FORMAT
272      ;-
273      000200      WF.IHISP         * BIT7       ;IHISP - HIGH SPEED
274      000100      WF.IWRT         * BIT6       ;IWRT - WRITE
275      000040      WF.IREV         * BIT5       ;IREV - REVERSE
276      000020      WF.IWFM         * BIT4       ;IWFM - WRITE FILE MARK
277      000010      WF.IEDIT        * BIT3       ;IEDIT - EDIT
278      000004      WF.IERASE       * BIT2       ;IERASE - ERASE
279      000002      WF.I3RESV       * BIT1       ;IRESV3 - RESERVED #3
280      000001      WF.I4RESV       * BIT0       ;IRESV4 - RESERVED #4
281
282      ;*
283      ;BSEL1 CODES FOR WRITE MISCELLANEOUS SUBCOMMAND
284      ;-
285      000200      MS.EXT           * BIT7       ;INVERT SENSE OF EXTENDED FEATURES SWITCH
286      000020      MS.RSFIFO       * BIT4       ;RESET FIFO AND INPUT PARITY ERRORR
287      000010      MS.RSTAPE       * BIT3       ;RESET TAPE STATUS IN 2 FLIP-FLOPS
288      000006      MS.ATTN         * BIT2!BIT1 ;ATTENTION TRIGGER FIELD
289      000001      MS.RSD           * BIT0       ;RESET TIMER A,B THEN DELAY TIMES IN SEL2
    
```

```

291      ;*
292      ; MS.ATTN SUBCODES
293      ;
294      000000      MSA.NOP = 0*2      ;NO-OP (NOTHING TRIGGERED)
295      000002      MSA.VOL = 1*2     ;SIMULATE ON-LINE/OFF-LINE TRANSITION
296      000004      MSA.NRAM= 2*2    ;FORCE NON-FATAL RAM ERROR (FORCES ERRCODE 54)
297      000006      MSA.FRAME= 3*2   ;FORCE FATAL RAM ERROR (CAUSES SCE TO SET)
298      ;*
299      ; WRITE SUBSYSTEM WRITE NPR BSEL1 BIT DEFINITIONS
300      ;
301      000200      NP.IR      = BIT7      ;INTERRUPT REQUEST (0-1 TRANSITION)
302      000100      NP.OUT     = BIT6      ;TAPE DATA DIRECTION OUT (0= IN)
303      000040      NP.LOOP   = BITS      ;ENABLE TRANSPORT LOOPBACK
304      000020      NP.WRP    = BIT4      ;WRITE CORRECT PARITY (SET=0 TO WRITE WRONG)
305      ;*
306      ; READ STATUS MESSAGE BUFFER BIT DEFINITIONS
307      ;
308      ;
309      000200      S2.DIM      = BIT7      ;WORD #9 BYTE 2 DATA IN MISS
310      000100      S2.ILW     = BIT6      ;
311      000040      S2.OUTRDY  = BITS      ;
312      000020      S2.INRDY   = BIT4      ;
313      000010      S2.ATIMR   = BIT3      ;
314      000004      S2.BTIMR   = BIT2      ;
315      000003      S2.UNDEF   = BIT1+BIT0 ;(UNDEFINED)
316      100000      S1.PARIN   = BIT15     ;WORD #8 BYTE 1 PARIN H
317      040000      S1.I2RESV  = BIT14     ;
318      020000      S1.I1RESV  = BIT13     ;
319      010000      S1.IEOT    = BIT12     ;
320      004000      S1.IIDENT  = BIT11     ;
321      002000      S1.ICER    = BIT10     ;
322      001000      S1.IFMK    = BIT9      ;
323      000400      S1.IHER    = BIT8      ;
324      000200      S0.ISPEED  = BIT7      ;WORD #8 BYTE 0 ISPEED H
325      000100      S0.IRDY    = BIT6      ;
326      000040      S0.IONL    = BITS      ;
327      000020      S0.ILDP    = BIT4      ;
328      000010      S0.IDBY    = BIT3      ;
329      000004      S0.IRWD    = BIT2      ;
330      000002      S0.IFBY    = BIT1      ;
331      000001      S0.IFPT    = BIT0      ;

```

```

333             .SBTTL SPECIAL MACROS AND OPDEFS.
334
335             ;*
336             ;SAVE GENERAL REGS 1 TO 5
337             ;
338
339             .MACRO SAVREG
340             JSR   R5,REGSAV
341             .ENDM
342
343             ;*
344             ; MACRO TO FORCE AN ERROR
345             ;-
346             .MACRO FORCERROR TAG,NOTSSR
347             .NLIST
348             .IIF NDF LISTALL, .NLIST
349             .LIST
350             .IF B NOTSSR
351             MOV   TSSR(R5),R1      ;READ TSSR
352             .ENDC
353             MOV   FORCER,FORCER    ;IS FORCER SET? (LEAVE C BIT ALONE)
354             BNE  TAG               ;BR IF YES
355             .NLIST
356             .IIF NDF LISTALL, .LIST
357             .LIST
358             .ENDM
359
360             ;*
361             ; MACRO TO FORCE AN EXIT TO AVOID SECTION ITERATIONS
362             ; WILL EXIT TO A LABEL IF FORCER IS NEGATIVE
363             ; SO TO FORCE ERRORS AND EXIT ON 1 ERROR SET
364             ; FORCER TO 177777
365             ; TO FORCE ERRORS AND ITERATIONS SET FORCER TO 1.
366             ;-
367             .MACRO FORCEEXIT TAG
368             .NLIST
369             .IIF NDF LISTALL, .NLIST
370             .LIST
371             MOV   FORCER,FORCER    ;IS FORCER NEGATIVE?
372             BMI  TAG               ;BR IF YES
373             .NLIST
374             .IIF NDF LISTALL, .LIST
375             .LIST
376             .ENDM
377             ;*
378             ; MACRO TO INCREMENT ERROR COUNTS
379             ;-
380             .MACRO NEXT.ERRNO
381             .NLIST
382             ;;;.IIF NDF LISTALL, .NLIST
383             ERRNO=ERRNO+1
384             ;;;.IIF NDF LISTALL, .LIST
385             .LIST
386             .ENDM

```

TSV3 GLOBAL AREAS MACRO M1113 14 JUN-84 16:41
SPECIAL MACROS AND OPDEFS.

SEQ 0034

```

388      ;*
389      ;MACRO TO PERFORM XOR
390      ;
391
392      .MACRO XOR      A,B
393      MOV      A,-(SP)
394      BIC      B,(SP)
395      BIC      A,B
396      BIS      (SP),B
397      .ENDM
398
399      000000      EN=0      ; INITIALIZE ERROR NUMBER
400      .SBTTL FORCER - FORCE ERROR FLAG
401
402      ;
403      ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
404      ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
405      ;
406
407 002170 000000 FORCER::      0      ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED
408      ; - BY THE MACRO "IFERROR"). AN ERROR NEED NOT
409      ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.

```

.SBTTL GLOBAL DATA SECTION

```

411
412
413
414
415
416
417
418
419
420
421
422 002172 000000
423 002174 000000
424 002176 000000
425 002200 000000
426 002202 000224
427 002204 000200
428 002206 000000
429 002210 000000
430 002212 000000
431 002214 000000
432 002216 000000
433 002220 000000
434 002222 000000
435 002224 000000
436 002226 000000
437 002230 000000
438 002232 000000
439 002234
440 002274 000000
441 002276 000000
442 002300 000000
443 002302 000000
444 002304 000000
445 002306 000000
446 002310 000000
447 002312 000000
448 002314
449 002460
450 002624

;***
;THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
;IN MORE THAN ONE TEST.
;
;
;THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
;SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
;
EPRTSW::      .WORD  0      ;PRINT SWITCH
UNITN::      .WORD  0      ;UNIT # UNDER TEST.
QVP::        .WORD  0      ;QUICK VERIFY FLAG.
CSRADDR::    .WORD  0      ;ADDRESS OF CSR FOR CURRENT DEVICE
IVEC::       .WORD  224    ;INTERRUPT VECTOR
IPRI::       .WORD  PRI04   ;INTERRUPT PRIORITY.
TSTCNT::     .WORD  0      ;NUMBER OF TESTS RUN IN THIS PASS
LOOPCNT::    .WORD  0      ;REMAINING ITERATION COUNT FOR TEST
DEVCNT::     .WORD  0      ;NUMBER OF DEVICE UNDER TEST
FATFLG::     .WORD  0      ;SET IF FATAL ERROR IS DETECTED IN TEST
INTRECV::    .WORD  0      ;SET IF TAPE INTERRUPT WAS RECEIVED
EXTFEA::     .WORD  0      ;EXTENDED FEATURES SOFTWARE SW 0-OFF;1-ON
BENBSW::     .WORD  0      ;BUFFER ENABLE SWITCH SW 0-OFF;1-ON
EXPD::       .WORD  0      ;EXPECTED RAM DATA FOR PRAMPKT ROUTINE
RECV::       .WORD  0      ;RECEIVED RAM DATA FOR PRAMPKT ROUTINE
ERRHI::      .WORD  0      ;HIGH ADDRESS MEMORY ERROR
ERRLO::      .WORD  0      ;LOW ADDRESS MEMORY ERROR
RAMDATA::    .BLKW  16.    ;DATA READ FROM RAM PACKET OR MESSAGE BUF AREA
RAMSIZ::     .WORD  0      ;RAM DATA SIZE FOR PRAMPKT ROUTINE
RCVHIADD::   .WORD  0      ;RECEIVED BUFFER HIGH ADDRESS
RCVLOADD::   .WORD  0      ;RECEIVED BUFFER LOW ADDRESS
COUNT::     .WORD  0      ;TEST COUNT PATTERN
DATA::       .WORD  0      ;TEST DATA
TSTFLAG::    .WORD  0      ;TEST FLAG WORD
TSTPTR::     .WORD  0      ;TSTBLK POINTER
PRMNO::      .WORD  0      ;PRINT ROUTINE TEMP
EXPMSG::     .BLKB  100.   ;EXPECTED MESSAGE BUFFER DATA
RECMMSG::    .BLKB  100.   ;RECEIVED MESSAGE BUFFER DATA
TMPBFR::     .BLKB  80.    ;TEMPORARY STORAGE FOR PRINT

```

```

452 .SBTTL TSTBLK TEST DATA TABLE
453
454 ;*
455 ; THIS TABLE CONTAINS TEST DATA USED IN SEVERAL TESTS
456 ;
457 ; IN SEQUENCE THE DATA IS:
458 ;
459 ; ALL ZEROS
460 ; ALL ONES
461 ; WALKING ONES
462 ; WALKING ZEROS
463 ; ALTERNATING ONES AND ZEROS
464 ;
465 ;
466 ;
467
468 TSTBLK::
469 002744 000000 .WORD 0 ;ALL ZEROS
470 002744 177777 .WORD 177777 ;ALL ONES
471 002750 000001 .WORD BIT0 ;DATA FOR WALKING ONES
472 002752 000002 .WORD BIT1
473 002754 000004 .WORD BIT2
474 002756 000010 .WORD BIT3
475 002760 000020 .WORD BIT4
476 002762 000040 .WORD BIT5
477 002764 000100 .WORD BIT6
478 002766 000200 .WORD BIT7
479 002770 000400 .WORD BIT8
480 002772 001000 .WORD BIT9
481 002774 002000 .WORD BIT10
482 002776 004000 .WORD BIT11
483 003000 010000 .WORD BIT12
484 003002 020000 .WORD BIT13
485 003004 040000 .WORD BIT14
486 003006 100000 .WORD BIT15
487 003010 177776 .WORD +CBIT0 ;DATA FOR WALKING ZEROS
488 003012 177775 .WORD +CBIT1
489 003014 177773 .WORD +CBIT2
490 003016 177767 .WORD +CBIT3
491 003020 177757 .WORD +CBIT4
492 003022 177737 .WORD +CBIT5
493 003024 177677 .WORD +CBIT6
494 003026 177577 .WORD +CBIT7
495 003030 177377 .WORD +CBIT8
496 003032 176777 .WORD +CBIT9
497 003034 175777 .WORD +CBIT10
498 003036 173777 .WORD +CBIT11
499 003040 167777 .WORD +CBIT12
500 003042 157777 .WORD +CBIT13
501 003044 137777 .WORD +CBIT14
502 003046 077777 .WORD +CBIT15
503 003050 125252 .WORD 125252 ;ALTERNATING ONES, ZEROS
504 003052 052525 .WORD 052525 ;ALTERNATING ONES, ZERO OPPOSITE FROM ABOVE
505 TBLEND==.

```

```

507 .SBTTL GLOBAL ENVIRONMENT STORAGE
508 ;
509 ; STORAGE FOR DEVICE REGISTERS
510 ;
511 003054 000000 100000 000000 DUMMY: 0,100000,0,0 ; DUMMY DEVICE REGISTERS...
512 003064 000000 000000 000000 0,0,0,0,0,0,0,0 ; ...FOR MULTI-UNIT CHECKOUT.
513 ;
514 ;
515 003104 000000 DUFLG:: .WORD 0 ; "DROPPED UNIT" FLAG.
516 ; INHIBITS CODE IN "CLEAN-UP".
517 003106 000000 NODEV:: .WORD 0 ; FLAG TO SAY NO DEVICE.
518 ;
519 003110 000000 TEMP1:: .WORD 0 ; SOME TEMP LOCATIONS.
520 003112 000000 TEMP2:: .WORD 0
521 003114 000000 XXCOMM:: .WORD 0 ; XXDP+ COMM BLOCK POINTER.
522 003116 000000 FREE:: .WORD 0 ; 1ST FREE MEMORY ADDRESS...
523 003120 000000 FRESIZ:: .WORD 0 ; ...AND SIZE (IN WORDS).
524 003122 000000 FREEHI: .WORD 0 ; LAST WORD IN FREE SPACE
525 003124 000000 KTFLG:: .WORD 0 ; KT11, MEM AVAIL FLAG -
526 ; - .WORD 0 = <24K OR NO KT
527 ; - NZ = >24K AND KT.
528 003126 000000 KTENABLE:: .WORD 0 ; SET BY TEST ROUTINES TO FLAG >28K UNDER TEST
529 003130 000000 NXMFLG:: .WORD 0 ; SET IF WE CAN TEST CLEARED OTHERWISE
530 003132 000000 NXMLO:: .WORD 0 ; NXM LO ADDRESS BITS
531 003134 000000 NXMHI:: .WORD 0 ; NXM HI ADDRESS BITS FOR DAL'S 16 21
532 003136 000000 T23A:: .WORD 0 ; 11/23A FLAG
533 003140 000000 T23B:: .WORD 0 ; 11/23B FLAG
534 003142 000000 T3BFLG:: .WORD 0 ; TEST 3B FLAG +0
535 003144 002000 PST32W:: .WORD 2000 ; 32W BLOCK ADDRESS FOR 32K START
536 003146 000000 SIFLAG:: .WORD 0 ;
537 003150 000000 BADDAT:: .WORD 0 ; ACTUAL DATA
538 003152 000000 GDDAT:: .WORD 0 ; EXPECTED DATA
539 003154 000000 LOOPFL:: .WORD 0 ;
540 003156 CTAB:: .WORD 0 ; CONFIGURATION TABLES.
541 003156 000000 CTABM:: .WORD 0 ; CONFIG WORK.
542 003160 .WORD 0
543 003162 .WORD 0
544 003164 .WORD 0
545 003166 177777 .WORD 1 ; END OF MEM TABLE.
546 003170 CTABE::
547 ; ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
548 ;
549 ; 0 = UNIT NOT TESTED
550 ; 100000 = UNIT ONLINE, NO ERRORS
551 ; 10XXXX = UNIT ONLINE, ENCOUNTERED XXXX ERRORS
552 ; 160000 = UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
553 ; 160001 = UNIT DROPPED, NOT IDLE AT START
554 ; 14XXXX = UNIT DROPPED, ENCOUNTERED XXXX ERRORS
555 ;
556 003170 ERTABL: .BLKW 64.
557 003370 000000 ERTABE: .WORD 0
558 ;
559 003372 000000 SKIPT: .WORD 0 ; 1=SKIP SUBTEST 0=NO SKIP OF SUBTEST

```

```

                    .SBTTL GLOBAL TEXT MESSAGES
561
562                ;**
563                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
564                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
565                ; MORE THAN ONE TEST.
566                ;
567                ;*
568                ; NAMES OF DEVICES SUPPORTED
569                ;
570 003374          DEVTYP <TSV05>
003374          L#DEV TYP::
003374          .ASCIZ /TSV05/
                    .EVEN
571
572                ;*
573                ; TEST DESCRIPTION
574                ;
575 003402          DESCRIPT <**** TSV05 LOGIC DIAGNOSTIC CHECK TRANSPORT IF ERROR ****>
003402          L#DESC::
003402          .ASCIZ /**** TSV05 LOGIC DIAGNOSTIC CHECK TRANSPORT IF ERROR ****/
                    .EVEN
597
598                ;*
599                ; BIT TO ASCII CONVERSION FOR TSSR REGISTER
600                ;
601 003476 003536 003541 003545 TSSRBIT:: .WORD 1#,2#,3#,4#,5#,6#,7#,8#
602 003516 003577 003573 003607 .WORD 9#,10#,11#,12#,13#,14#,15#,16#
603 003536 123 103 000 1#: .ASCIZ 'SC'
604 003541 102 111 105 2#: .ASCIZ 'BIE'
605 003545 123 103 105 3#: .ASCIZ 'SCE'
606 003551 122 115 122 4#: .ASCIZ 'RMR'
607 003555 116 130 115 5#: .ASCIZ 'NXM'
608 003561 116 102 101 6#: .ASCIZ 'NBA'
609 003565 102 111 124 7#: .ASCIZ 'BIT9'
610 003572 102 111 124 8#: .ASCIZ 'BIT8'
611 003577 123 123 122 9#: .ASCIZ 'SSR'
612 003603 117 106 114 10#: .ASCIZ 'OFL'
613 003607 102 111 124 11#: .ASCIZ 'BIT5'
614 003614 102 111 124 12#: .ASCIZ 'BIT4'
615 003621 102 111 124 13#: .ASCIZ 'BIT3'
616 003626 102 111 124 14#: .ASCIZ 'BIT2'
617 003673 102 111 124 15#: .ASCIZ 'BIT1'
618 003640 102 111 124 16#: .ASCIZ 'BIT0'
619                .EVEN
620 003646 124 123 123 SFIERR: .ASCIZ 'TSSR ERROR AFTER SOFT INIT'
621 003701 124 123 123 SFHERR: .ASCIZ 'TSSR ERROR AFTER BUS RESET'
622 003734 040 040 116 NXR: .ASCIZ / NON-EXISTANT DEVICE REGISTER/
623 003773 045 101 040 NXR: .ASCIZ /#A ADDRESS: #06/
624 004014 045 101 040 TSSX: .ASCII /#A TSBA,TSSR EXP'D: #06#A,#06#N/
625 004054 045 101 040 TSSX: .ASCIZ /#A TSBA,TSSR REC'D: #06#A,#06/
626 004113 045 116 045 FUSI: .ASCII /#N#A/
627 004117 040 040 125 USI: .ASCIZ / UNEXPECTED INTERRUPT/
628 004146 040 040 111 NSI: .ASCIZ / INTERRUPT EXPECTED, NOT RECEIVED/
629 004211 045 116 045 FNOINTR: .ASCII /#N#A/
630 004215 040 040 116 NOINTR: .ASCIZ / NO INTERRUPT WAS GENERATED/
631 004252 040 040 111 IFALT: .ASCIZ / INTERRUPT FAULT/
632 004274 045 101 040 INTX: .ASCIZ /#A CPU PC: #06#A TSBA: #06/

```



```

633 004331 040 040 042 NOINIT: .ASCIZ / "BUS INIT" DIDN T INITIALIZE CONTROLLER
634 004403 040 040 042 NSINIT: .ASCIZ / "SOFT INIT" DIDN T INITIALIZE THE DP'
635 004453 040 040 042 BRINIT: .ASCIZ / "BUS-RESET" DIDN T INITIALIZE THE DP'
636 004523 000 .ASCIZ //
637 004524 045 116 000 NULCR: .ASCIZ /#N/
638 004527 045 101 040 EXPGOT: .ASCIZ /#A EXP'D: #06#A, REC'D: #06/
639 004563 045 116 045 EXPGT2: .ASCIZ /#N#A EXP'D: #06#A, #06#N#A REC'D: #0#A, #06/
640 004637 045 101 040 DUAD12: .ASCIZ /#A REG(W) WRITTEN TO: #06#A REG(R) READ; EXP D: #06#A, REC D: #06/
641 004741 122 101 115 PKTRAM: .ASCIZ 'RAM Contents Do Not Match Packet Sent
642 005007 040 040 103 SCME: .ASCIZ / CONFIG DOESN'T MATCH MFG. MASTER/
643 005052 127 122 111 WRTMSG: .ASCIZ 'WRITE CHARACTERISTICS Failed'
644 005107 124 123 123 WRTERR: .ASCIZ 'TSSR Incorrect After WRITE Command, More Bits Set Than SSR
645 005202 124 123 123 RDERR: .ASCIZ 'TSSR Incorrect After READ Command, More Bits Set Than SSR
646 005274 106 101 124 SCHERR: .ASCIZ 'FATAL ERROR IN SUBTEST - CHECK TAPE,CABLES,TRANSPORT etc
647 005366 105 122 122 RETERR: .ASCIZ 'ERROR IN SUBTEST - WRITE DATA RETRY FIVE TIMES FAILED'
648 005454 045 116 045 NOEM: .ASCIZ '#N#A ***** NO NXM ADDRESS--CANNOT TEST NXM TIMEOUT. *****#N
649 005550 045 116 045 M8186: .ASCIZ '#N#A ***** 11/23A SYSTEM *****#N
650 005641 045 116 045 M8189: .ASCIZ '#N#A ***** 11/23B SYSTEM *****#N
651 .EVEN
652 .SBTTL GLOBAL ERROR REPORT SECTION
653
654
655 ;**
656 ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
657 ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
658 ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
659 ;
660 ; BGNMSG NXRERR ;NON-EXISTANT DEVICE REGISTER.
661 ;
662 ; PRINTX #NXRX,NODEV ;NODEV = NXM ADDRESS.
663 ; MOV NODEV,-(SP)
664 ; MOV #NXRX,-(SP)
665 ; MOV #2,-(SP)
666 ; MOV SP,RO
667 ; TRAP C#PNTX
668 ; ADD #6,SP
669 ; JSR PC,EXTEND ; PRINT EXTENSION IF REQUIRED.
670 ; ENDMSG
671 ;
672 ; L10002: TRAP C#MSG
673 ;
674 ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
675 ; TO ANY OF THE ABOVE ERROR SIGNATURES.
676 ;
677 ; EXTEND: TST (PC)+
678 ; EXTA: 0 ; 0 = NO EXTENSION.
679 ; BEQ 1$
680 ; JSR PC,EXTA ; APPEND EXTENSION TEXT.
681 ; 1$: PRINTX #NULCR ; PRINT A BLANK LINE
682 ; MOV #NULCR,-(SP)
683 ; MOV #1,-(SP)
684 ; MOV SP,RO
685 ; TRAP C#PNTX
686 ; ADD #4,SP
687 ; RTS PC

```

674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692 006020
693 006020
694 006024 010104
695 006026
006026 010446
006030 012746 006473
006034 012746 000002
006040 010600
006042 104414
006044 062706 000006
696 006050 010400
697 006052 004737 016124
698 006056 103410
699 006060
006060 012746 006713
006064 012746 000001
006070 010600
006072 104415
006074 062706 000004
700 006100 010403
701 006102 042703 001476
702 006106 001434
703 006110 012702 002624
704 006114 012701 003476
705 006120 005703
706 006122 001413
707 006124 000241
708 006126 006103
709 006130 103006
710 006132 011100
711 006134 112022
712 006136 001376
713 006140 112762 000054 177777
714 006146 005721
715 006150 000763
716 006152 105042
717 006154
006154 012746 002624
006160 012746 006664

```

.SBTTL PRITSSR PRINT TSSR CONTENTS
;
; ROUTINE TO DISPLAY THE CONTENTS, AND BIT DEFINITIONS, OF
; THE TSSR REGISTER. THIS ROUTINE IS NORMALLY CALLED ONLY
; BY A MESSAGE PRINTING ROUTINE
;
; INPUTS:
;
; R1 CONTENTS OF TSSR
;
; SUBORDINATE ROUTINES:
;
; CHKAMB CHECK FOR AMBIGUOUS CONTENTS
;
;
PRITSSR:
  SAVREG ;SAVE GENERAL REGISTERS
  MOV R1,R4 ;SAVE THE TSSR CONTENTS
  PRINTB @TSSRFOR,R4 ;PRINT THE CONTENTS OF TSSR
  MOV R4,-(SP)
  MOV @TSSRFOR,-(SP)
  MOV @2,-(SP)
  MOV SP,R0
  TRAP C@PNTB
  ADD @6,SP
  MOV R4,R0 ;GET TSSR BACK FOR CHKAMB
  JSR PC,CHKAMB ;ARE CONTENTS AMBIGUOUS ?
  BCS 5@ ;BRANCH IF NOT
  PRINTX @AMBTSSR ;SHOW CONTENTS ARE AMBIGUOUS
  MOV @AMBTSSR,-(SP)
  MOV @1,-(SP)
  MOV SP,R0
  TRAP C@PNTX
  ADD @4,SP
5@: MOV R4,R3 ;CONTENTS OF TSSR
  BIC @HIADDR!FATERR!TERCLS,R3 ;CLEAR ALL MULTIPLE BIT FIELDS
  BEQ 20@ ;NO BITS ARE SET
  MOV @TMPBFR,R2 ;TEMPORARY ASCII BUFFER
  MOV @TSSRBIT,R1 ;ASCII EQUIVALENT OF BITS
10@: TST R3 ;REMAINING BITS TO CONVERT
  BEQ 15@ ;BRANCH WHEN ALL ARE DONE
  CLC ;CLEAR CARRY FOR SHIFT
  ROL R3 ;SHIFT NEXT BIT TO CARRY
  BCC 13@ ;BRANCH IF BIT NOT SET
  MOV (R1),R0 ;POINTER TO BIT DEFINITION
11@: MOVB (R0),R2 ;MOVE ASCII TO BUFFER
  BNE 11@ ;MOVE ALL BITS
  MOVB #' ,,-1(R2) ;INSERT A COMMA TO TERMINATE
13@: TST (R1) ;POINT TO NEXT DESCRIPTION
  BR 10@ ;GET THE REMAINING BITS
15@: CLRB -(R2) ;TERMINATE THE LINE
  PRINTX @TSSDEF,@TMPBFR ;PRINT THE BIT DEFINITIONS
  MOV @TMPBFR,-(SP)
  MOV @TSSDEF,-(SP)

```

```

006164 012746 000002      MOV      #2, (SP)
006170 010600      MOV      SP,R0
006172 104415      TRAP    C:PNTX
006174 062706 000006      ADD     #6,SP

718
719 006200 010403      201:    MOV      R4,R3          ;GET THE TSSR CONTENTS
720 006202 042703 177761      BIC     #+CTERCIS,R3    ;CLEAR ALL BUT TERMINATION
721 006206 016303 006754      MOV     TCOCOD(R3),R3   ;GET THE TERMINATION CODE MEANING
722 006212      PRINTX #TCOASC,R3      ;PRINT THE TERMINATION CODE
      MOV     R3,-(SP)
      MOV     #TCOASC,-(SP)
      MOV     #2,-(SP)
      MOV     SP,R0
      TRAP    C:PNTX
      ADD     #6,SP

723 006234 010403      MOV     R4,R3          ;TSSR CONTENTS AGAIN
724 006236 042703 177717      BIC     #+CFATERR,R3   ;CLEAR ALL BUT FATAL TERMINATION
725 006242 001416      BEQ     251            ;DON'T PRINT IF ZERO
726 006244 006203      ASR     R3
727 006246 006203      ASR     R3
728 006250 006203      ASR     R3          ;ALINE TERMINATION CODE FOR INDEX
729 006252 016303 007314      MOV     TSFCOD(R3),R3  ;GET THE FATAL TERMINATION CODE
730 006256      PRINTX #TFCASC,R3      ;PRINT THE FATAL TERMINATION CODE
      MOV     R3,-(SP)
      MOV     #TFCASC,-(SP)
      MOV     #2,-(SP)
      MOV     SP,R0
      TRAP    C:PNTX
      ADD     #6,SP

731 006300 042704 176377      251:    BIC     #+CHIADDR,R4   ;CLEAR ALL BUT EXTENDED ADDRESS
732 006304 001411      BEQ     301            ;DON'T PRINT IF ZERO
733 006306      PRINTX #TEXASC,R4      ;PRINT THE EXTENDED ADDRESS BITS
      MOV     R4,-(SP)
      MOV     #TEXASC,-(SP)
      MOV     #2,-(SP)
      MOV     SP,R0
      TRAP    C:PNTX
      ADD     #6,SP

734 006330 013703 002172      301:    MOV     EPRTSW,R3      ;PRINT MEASGE BUFFER ADDRESS
735 006334      PRINTX R3              ;PRINT PROPER MESSAGE
      MOV     R3,-(SP)
      MOV     #1,-(SP)
      MOV     SP,R0
      TRAP    C:PNTX
      ADD     #4,SP

736 006352 000207      RTS     PC              ;RETURN TO CALLER

```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 16:41
 PRITSSR PRINT TSSR CONTENTS

SEQ 0042

752	006354	045	116	045	EPRT1:	.ASCIZ	'#NSA *****CHECK TRANSPORT*****'
753	006413	045	116	045	EPRT2:	.ASCIZ	'#NSA *****CHECK PARITY SWITCH IN TRANSPORT*****'
755	006473	045	116	045	TSSRFOR:	.ASCIZ	'#NSA TSSR = #06'
756	006513	045	116	045	TEXASC:	.ASCIZ	'#NSA Extended Address Bits = #06'
757	006554	045	116	045	TCOASC:	.ASCIZ	'#NSA Termination Class Code = #T'
758	006615	045	116	045	TFCASC:	.ASCIZ	'#NSA Fatal Termination Class Code = #T'
759	006664	045	116	045	TSSDEF:	.ASCIZ	'#NSA TSSR Bits Set: #T'
760	006713	045	116	045	AMBTSSR:	.ASCIZ	'#NSA TSSR Contents Are Ambiguous'
761						.EVEN	
762	006754	006774	007017	007045	TCOCOD:	.WORD	1#,2#,3#,4#,5#,6#,7#,8#
763	006774	116	157	162	1#:	.ASCIZ	'Normal Termination'
764	007017	124	145	162	2#:	.ASCIZ	'Termination Condition'
765	007045	124	141	160	3#:	.ASCIZ	'Tape Status Alert'
766	007067	106	165	156	4#:	.ASCIZ	'Function Reject'
767	007107	122	145	143	5#:	.ASCIZ	'Recoverable Error - Tape Position One Record Down'
768	007171	122	145	143	6#:	.ASCIZ	'Recoverable Error - Tape Was Not Moved'
769	007240	125	156	162	7#:	.ASCIZ	'Unrecoverable Error'
770	007264	106	141	164	8#:	.ASCIZ	'Fatal Controller Error'
771						.EVEN	
772							
773	007314	007324	007360	007371	TSFCOD:	.WORD	1#,2#,3#,4#
774	007324	111	156	164	1#:	.ASCIZ	'Internal Diagnostic Failure'
775	007360	122	145	163	2#:	.ASCIZ	'Reserved'
776	007371	102	165	163	3#:	.ASCIZ	'Bus Interface or Sanity Check Error'
777	007435	122	145	163	4#:	.ASCIZ	'Reserved'
778						.EVEN	

```

780 .SBTTL PRIPKT - PRINT THE ADDRESS/CONTENTS OF COMMAND PACKET
781
782 ;*
783 ;THIS ROUTINE PRINTS THE ADDRESS AND CONTENTS OF A COMMAND PACKET.
784 ;THIS ROUTINE IS NORMALLY ONLY CALLED FROM A PRINT ROUTINE.
785 ;
786 ;INPUT:
787 ;
788 ; R0 NUMBER OF WORDS IN PACKET
789 ; R3 HIGH ORDER COMMAND PACKET ADDRESS
790 ; R4 ADDRESS OF COMMAND PACKET
791 ;
792 ; NOTE: R3 IS IGNORED IF THE KTENABLE FLAG IS CLEAR.
793 ;
794
795 PRIPKT::
796 SAVREG ;SAVE THE REGISTERS
797 MOV R0,R5 ;SAVE NO. OF WORDS IN PACKET
798 TST KTENABLE ;ABOVE 28K UNDER TEST?
799 BNE 10# ;BR IF YES
800 CLR R3 ;SET HIGH ORDER ADDRESS TO 0
801 10#: MOV R3,R1 ;COPY HIGH ORDER ADDRESS
802 MOV R4,R0 ;GET LOWER ADDRESS
803 ROL R0 ;SHIFT BIT 15 INTO C BIT
804 ROL R1 ;AND INTO HIGH ORDER.
805 PRINTB @PKTADD,R1,R4 ;PRINT PACKET ADDRESS
      MOV R4,-(SP)
      MOV R1,-(SP)
      MOV @PKTADD,-(SP)
      MOV @3,-(SP)
      MOV SP,R0
      TRAP C#PNTB
806 15#: ADD @10,SP
      MOV R3,R0 ;GET HIGH ORDER ADDRESS
807 BEQ 20# ;BR IF NOT ABOVE 28K.
808 MOV R4,R1 ;GET LOW ORDER ADDRESS
809 JSR PC,SETMAP ;SETUP PAR6 MAPPING FOR 18 BIT ADDRESS
810 MOV R0,R4 ;GET RETURNED PAR6 ADDRESS BIAS
811 20#: CLR R1 ;SAVE WORD NUMBER
812 25#: MOV (R4)+,R2 ;GET PACKET CONTENTS
813 PRINTB @PKTFRM,R1,R2 ;PRINT THE DATA
      MOV R2,-(SP)
      MOV R1,-(SP)
      MOV @PKTFRM,-(SP)
      MOV @3,-(SP)
      MOV SP,R0
      TRAP C#PNTB
814 007560: ADD @10,SP
      INC R1 ;NEXT WORD NUMBER
815 CMP R1,R5 ;DONE ALL PACKET WORDS?
816 BLT 25# ;LOOP TILL ALL DONE
817 RTS PC ;RETURN
818
819 007574 045 116 045 PKTFRM: .ASCIZ '#N#A Packet Word #D1#A = #06'
820 007632 045 116 045 PKTADD: .ASCIZ '#N#A Packet Address = #01#05'
821 .EVEN
    
```

```

823 .SBTTL PRIBXOR PRINT EXPD, RECV AND XOR BYTE
824
825 ;*
826 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE DATA BYTE
827 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
828 ;
829 ;INPUTS:
830 ;
831 ; R1 RECEIVED DATA
832 ; R2 EXPECTED DATA
833 ;
834 ;OUTPUT:
835 ;
836 ; R0 XOR OF EXPECTED/RECEIVED DATA
837 ;
838 PRIBXOR::
839 SAVREG ;SAVE THE REGISTERS
840 MOV R2,R3 ;EXPECTED DATA
841 XOR R1,R3 ;FORM THE EXCLUSIVE OR
842 MOV #C<377>,R0 ;BYTE MASK
843 BIC R0,R1 ;SAVE LOW BYTE RECV
844 BIC R0,R2 ;SAVE LOW BYTE EXPD
845 BIC R0,R3 ;SAVE LOW BYTE XOR
846 PRINTB #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
847 MOV R3,-(SP)
848 MOV R1,-(SP)
849 MOV R2,-(SP)
850 MOV #XORFOR,-(SP)
851 MOV #4,-(SP)
852 MOV SP,R0
853 TRAP C#PNTB
854 ADD #12,SP
855 MOV R3,R0 ;R0 HAS XOR ON RETURN
856 RTS PC ;RETURN TO CALLER
857
858 007670
859 007670 010203
860 007674 010203 177400
861 007676
862 007706 012700 177400
863 007712 040001
864 007714 040002
865 007716 040003
866 007720
867 007720 010346
868 007722 010146
869 007724 010246
870 007726 012746 007752
871 007732 012746 000004
872 007736 010600
873 007740 104414
874 007742 062706 000012
875 007746 010300
876 007750 000207
877
878 007752 045 116 045 XORFOR: .ASCIZ '###A EXPD: ###A RECV: #03#A XOR. #03
879 .EVEN
880 .SBTTL PRIBXOR - PRINT EXPD, RECV AND XOR
881
882 ;*
883 ;PRINT EXPECTED DATA, RECEIVED DATA, AND XOR OF THE TWO
884 ;THIS ROUTINE IS NORMALLY CALLED ONLY FOR PRINT ROUTINES.
885 ;
886 ;INPUTS:
887 ;
888 ; R1 RECEIVED DATA
889 ; R2 EXPECTED DATA
890 ;
891 ;OUTPUT:
892 ;
893 ; R0 XOR OF EXPECTED/RECEIVED DATA
894 ;
895 PRIBXOR::
896 SAVREG ;SAVE THE REGISTERS
897 MOV R2,R3 ;EXPECTED DATA
898 XOR R1,R3 ;FORM THE EXCLUSIVE OR
899 PRINTB #XORFOR,R2,R1,R3 ;PRINT THE MESSAGE
900
901 010020
902 010020 010203
903 010024 010203
904 010026
905 010036
  
```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN 84 16:41
PRI XOR - PRINT EXPD, RECV AND XOR

SEQ 0045

```

010036 010346            MOV    R3, (SP)
010040 010146            MOV    R1, -(SP)
010042 010246            MOV    R2, (SP)
010044 012746 010070    MOV    @XORFOR, (SP)
010050 012746 000004    MOV    @4, -(SP)
010054 010600            MOV    SP, R0
010056 104414            TRAP   C:PNTB
010060 062706 000012    ADD    @12, SP
872 010064 010300            MOV    R3, R0                    ;R0 HAS XOR ON RETURN
873 010066 000207            RTS    PC                        ;RETURN TO CALLER
874
875 010070        045        116        045 XORFOR: .ASCIZ 'N#A EXPD: #06#A RECV: #06#A XOR: #06'
876                            .EVEN

```

TSV3 - GLOBAL AREAS MACRO M1113 14 JUN 84 16:41
 PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT

SEQ 0046

```

878                                    SBTTL PRIEQU - PRINT BIT NUMBERS AS ASCII EQUIVALENT
879
880                                    ;*
881                                    ;
882                                    ;ROUTINE TO CONVERT BIT VALUES TO ASCII AND PRINT THE STRING
883                                    ;THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
884                                    ;
885                                    ;INPUTS:
886                                    ;
887                                    ;        R0        OCTAL VALUE TO CONVERT
888                                    ;        R1        TABLE OF POINTERS TO ASCII EQUIVALENT
889                                    ;
890                                    ;
891
892 010136                             PRIEQU:
893 010136                             SAVREG                             ;SAVE THE REGISTERS
894 010142 000207                     RTS        PC                         ;RETURN TO CALLER
895
896                                    .SBTTL PRIRAM - PRINT RAM ADDRESS
897                                    ;*
898                                    ;
899                                    ;PRINT CONTROLLER RAM ADDRESS.
900                                    ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
901                                    ;
902                                    ;INPUTS:
903                                    ;
904                                    ;        R4        RAM ADDRESS
905                                    ;
906                                    ;-
907 010144                             PRIRAM:
908 010144                             SAVREG                             ;SAVE R1-R5 UNTIL NEXT RETURN
909 010150                             PRINTB    #RAMFOR,R4                 ;PRINT RAM ADDRESS IN ERROR
                                     MOV        R4,-(SP)
                                     MOV        #RAMFOR,-(SP)
                                     MOV        #2,-(SP)
                                     MOV        SP,R0
                                     TRAP       C#PNTB
                                     ADD        #6,SP
910 010172 000207                     RTS        PC                         ;RETURN
911
912 010174                             045    RAMFOR: .ASCIZ 'NMA CONTROLLER RAM ADDRESS = #06'
913                                    .EVEN

```


TSV3 - GLOBAL AREAS MACRO M1113 14-JUN 84 16:41
 PRIADD - PRINT MEMORY ERROR ADDRESS

SEQ 0047

```

915          .SBTTL PRIADD PRINT MEMORY ERROR ADDRESS
916          ;*
917          ;
918          ;PRINT MEMORY ADDRESS
919          ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
920          ;
921          ; IMPLICIT INPUTS
922          ;
923          ;     ERRHI - HIGH ORDER ADDRESS
924          ;     ERRLO - LOW ORDER ADDRESS
925          ;
926          ;
927 010236    PRIADD:
928 010236    SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
929 010242    013700 002230    MOV ERRHI,R0          ;GET HIGH ADDRESS
930 010246    013701 002232    MOV ERRLO,R1          ;GET LOW ADDRESS
931 010252    010102          MOV R1,R2            ;COPY LOW ADDRESS
932 010254    006101          ROL R1                ;SHIFT BIT 15 TO C BIT
933 010256    006100          ROL R0                ;SHIFT INTO HIGH ORDER
934 010260    PRINTB #PRIA0,R0,R2          ;PRINT MEMORY ADDRESS IN ERROR
          010260    010246    MOV R2,-(SP)
          010262    010046    MOV R0,-(SP)
          010264    012746    010306    MOV #PRIA0,-(SP)
          010270    012746    000003    MOV #3,-(SP)
          010274    010600    MOV SP,R0
          010276    104414    TRAP C#PNTB
          010300    062706    000010    ADD #10,SP
935 010304    000207    RTS PC                ;RETURN
936
937 010306    045      116      045 PRIA0: .ASCIZ 'ANNA MEMORY ERROR ADDRESS = #01#05'
938          .EVEN
939
940          .SBTTL PRITADD - PRINT MEMORY TEST ADDRESS
941          ;*
942          ;
943          ;PRINT MEMORY ADDRESS
944          ;THIS ROUTINE IS NORMALLY CALLED ONLY FROM PRINT ROUTINES.
945          ;
946          ; IMPLICIT INPUTS
947          ;
948          ;     ERRHI - HIGH ORDER ADDRESS
949          ;     ERRLO - LOW ORDER ADDRESS
950          ;
951          ;
952 010352    PRITADD:
953 010352    SAVREG                                ;SAVE R1-R5 UNTIL NEXT RETURN
954 010356    013702 002230    MOV ERRHI,R2          ;GET HIGH ADDRESS
955 010362    013701 002232    MOV ERRLO,R1          ;GET LOW ADDRESS
956          ;MOV R1,R2            ;COPY LOW ADDRESS
957          ;ROL R1                ;SHIFT BIT 15 TO C BIT
958          ;ROL R0                ;SHIFT INTO HIGH ORDER
959 010366    PRINTB #PRIT0,R1          ;PRINT MEMORY ADDRESS LOW IN ERROR
          010366    010146    MOV R1,-(SP)
          010370    012746    010434    MOV #PRIT0,-(SP)
          010374    012746    000002    MOV #2,-(SP)
          010400    010600    MOV SP,R0
          010402    104414    TRAP C#PNTB

```

TSV3 GLOBAL AREAS MACRO M1113 14 JUN 84 16:41
 PRITADD PRINT MEMORY TEST ADDRESS

SEQ 0048

960	010404	062706	000006			ADD	#6,SP	
	010410					PRINTB	#PRIT1,R2	;PRINT MEMORY ADDRESS HIGH IN ERROR
	010410	010246				MOV	R2,-(SP)	
	010412	012746	010477			MOV	#PRIT1,(SP)	
	010416	012746	000002			MOV	#2,-(SP)	
	010422	010600				MOV	SP,R0	
	010424	104414				TRAP	C:PNTB	
	010426	062706	000006			ADD	#6,SP	
961	010432	000207				RTS	PC	;RETURN
962								
963	010434	045	116	045	PRIT0:	.ASCIZ	'#N/A MEMORY TEST ADDRESS LOW = #06'	
964	010477	045	116	045	PRIT1:	.ASCIZ	'#N/A MEMORY TEST ADDRESS HIGH = #06'	
965						.EVEN		

```

        .SBTTL SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND
967
968
969      ;*
970      ;
971      ;ROUTINE TO ISSUE A SPACE RECORDS
972      ;COMMAND (FORWARD OR REVERSE)
973      ;
974      ;INPUT:
975      ;
976      ;       R3      NUMBER OF RECORDS TO BE SPACED OVER
977      ;               BIT15 CONTROLS DIRECTION
978      ;               BIT15 = 0 IS FORWARD
979      ;               BIT15 = 1 IS REVERSE
980      ;       R5      FIRST DEVICE UNIBUS ADDRESS
981      ;
982      ;       REQUIRES A WRITE CHARACTERISTICS DONE PREVIOUSLY
983      ;
984      ;OUTPUT:
985      ;
986      ;       CARRY   SET - SPACE RECORDS COMMAND OK
987      ;               CLR - SPACE RECORDS FAILED
988      ;
989      ;
990      ;       R0      THE CONTENTS OF R4 IS MOVED TO R0
991      ;
992      ;
993      ;IMPLICIT OUTPUT:
994      ;
995      ;       TAPE HAS BEEN MOVED
996      ;
997      ;SIDE EFFECTS:
998      ;
999      ;
1000     ;-
1001
1002     010544     SPACE::
1003     010544     SAVREG                                ;SAVE THE GENERAL REGISTERS
1004     010550     012737 000764 010740     MOV      #500.,SDELAY      ;SET UP DELAY
1005     010556     012737 140010 010730     MOV      #140010,80$     ;SET UP COMMAND, SPACE FORWARD
1006     010564     005703                    TST      R3              ;CHECK FOR DIRECTION
1007     010566     100403                    BMI      5$             ;BR, IF REVERSE INDICATED
1008     010570     010337 010732     MOV      R3,90$         ;LOAD UP NUMBER OF RECORDS TO SPACE
1009     010574     000407                    BR       10$           ;GO DO COMMAND
1010     010576     042703 100000     5$:    BIC      #BIT15,R3     ;CLEAR DIRECTION BIT
1011     010602     010337 010732     MOV      R3,90$         ;LOAD UP NUMBER OF RECORDS TO SPACE
1012     010606     052737 000400 010730     BIS      #BIT8,80$     ;SET REVERSE BIT IN COMMAND PACKET
1013     010614     012704 C10730     10$:   MOV      #80$,R4      ;SET UP R4 WITH PACKET ADDRESS
1014     010620     010465 000000     MOV      R4,TSDB(R5)   ;SEND OUT COMMAND
1015     010624     004737 016330     15$:   JSR      PC,WAITF   ;WAIT FOR SSR
1016     010630     103420                    BCS     20$             ;BR, IF SSR IS SET AND OK
1017     010632                    DELAY   250             ;DELAY ABOUT .25 SECONDS
        010632     012727 000250     MOV      #250,(PC)+
        010636     000000                    .WORD   0
        010640     013727 002116     MOV      L#DLY,(PC)+
        010644     000000                    .WORD   0
        010646     005367 177772     DEC      -6(PC)
        010652     001375                    BNE     .-4
    
```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 16:41
 SPACE - SPACE RECORDS (FORWARD AND REVERSE) COMMAND

SEQ 0050

```

010654 005367 177756          DEC    -22(PC)
010660 001367                BNE    .-20
1018 010662 005337 010740     DEC    SDELAY          ;BUMP DELAY COUNTER DOWN
1019 010666 001356                BNE    15$            ;BR, IF MORE DELAY
1020 010670 000411                BR     60$            ;BR IF TROUBLE CARRY = CLEAR
1021 010672 016501 000002 20$:  MOV    TSSR(R5),R1    ;READ TSSR
1022 010676 012702 000200     MOV    #SSR,R2        ;SET UP EXPECTED
1023 010702 020201 25$:  CMP    R2,R1          ;ARE THEY OK
1024 010704 001401                BEQ    40$            ;BR, IF EQUAL = OK
1025 010706 000402                BR     60$            ;TROUBLE EXIT
1026 010710 000261 40$:  SEC                    ;SET CARRY NO TROUBLE
1027 010712 000401                BR     70$            ;EXIT
1028 010714 000241 60$:  CLC                    ;CARRY CLEAR = ERROR
1029 010716 70$:
1030 010716 010400     MOV    R4,R0          ;PASS PACKET ADDRESS
1031 010720 000207     RTS    PC             ;RETURN
1032
1033
1034
1035     ;PACKET FOR SPACE COMMAND
1036
1037
1038     010730     .=<.10>&177770
1039
1040
1041     ;COMMAND WORD
1042 010730 000000 80$:  .WORD
1043     ;NUMBER OF RECORDS TO BE SPACED OVER WORD
1044 010732 000000 90$:  .WORD
1045 010734 000000     .WORD
1046 010736 000000     .WORD
1047 010740 000000 SDELAY: .WORD 0          ;DELAY COUNTER
1048     .EVEN
1049     .SBTTL WRTCHR - WRITE CHARACTERISTICS COMMAND

```

TSV3 - GLOBAL AREAS MACRO M1113 14 JUN 84 16:41
 WRTCHR - WRITE CHARACTERISTICS COMMAND

SEQ 0051

```

1051 ;
1052 ;ROUTINE TO ISSUE A WRITE CHARACTERISTICS
1053 ;COMMAND SO THAT OTHER COMMANDS WILL BE ACCEPTED
1054 ;
1055 ;INPUT:
1056 ; R4 ADDRESS OF PACKET FROM TEST
1057 ; R5 FIRST DEVICE UNIBUS ADDRESS
1058 ; REQUIRES A CALL TO SOFINIT BE DONE PREVIOUSLY
1059 ;
1060 ;OUTPUT:
1061 ; R0 TSSR CONTENTS
1062 ; CARRY SET - WRITE CHARACTERISTICS COMMAND OK
1063 ; CLR WRITE CHARACTERISTICS FAILED
1064 ;
1065 ;IMPLICIT OUTPUT:
1066 ;
1067 ; MESSAGE BUFFER AND OTHER BUFFERS ALL SET UP
1068 ; SOFTWARE SWITCHES SET AS FOLLOWS:
1069 ; EXTFEA = EXTENDED FEATURES PRESENT
1070 ; BENBSW = BUFFER ENABLE SWITCH ON OR OFF
1071 ;
1072 ;SIDE EFFECTS:
1073 ;
1074 010742 WRTCHR::
1075 010742 SAVREG ;SAVE THE GENERAL REGISTERS
1076 010746 005037 002222 CLR BENBSW ;CLEAR BUFFER ENABLE SWITCH
1077 010752 005037 002220 CLR EXTFEA ;CLEAR EXTENDED FEATURES SW SWITCH
1078 010756 010465 000000 10$: MOV R4,TSDB(R5) ;SEND OUT COMMAND
1079 010762 004737 016416 JSR PC,CHKTSSR ;WAIT FOR SSR
1080 010766 103401 BCS 20$ ;BR, IF SSR IS SET AND OK
1081 010770 000435 BR 60$ ;BR IF TROUBLE CARRY = CLEAR
1082 010772 016501 000002 20$: MOV TSSR(R5),R1 ;READ TSSR
1083 010776 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
1084 011002 032701 00010C BIT #OFL,R1 ;WAS OFF LINE SET IN TSSR
1085 011006 001402 BEQ 25$ ;BR, IF NO OFL SET
1086 011010 052702 000100 BIS #OFL,R2 ;MAKE THEM LOOK ALIKE
1087 011014 020201 25$: CMP R2,R1 ;ARE THEY OK
1088 011016 001401 BEQ 40$ ;BR, IF EQUAL = OK
1089 011020 000421 BR 60$ ;TROUBLE EXIT
1090 011022 062704 000010 40$: ADD #8,R4 ;POINT TO WRT CHARA DATA PACKET
1091 011026 011403 MOV (R4),R3 ;GET ADDRESS OF MESSAGE BUFFER
1092 011030 032763 000200 000012 BIT #X2.EXTF,XST2(R3) ;EXTENDED FEATURES BIT SET?
1093 011036 001402 BEQ 45$ ;BR IF NO
1094 011040 005237 002220 INC EXTFEA ;SET EXTENDED FEATURES SW SWITCH
1095 011044 45$:
1096 011044 032763 000100 000012 BIT #X2.BUFE,XST2(R3) ;BUFFER ENABLE SWITCH SET
1097 011052 001402 BEQ 50$ ;BR, IF SWITCH NOT SET
1098 011054 005237 002222 INC BENBSW ;SET SOFTWARE SWITCH FOR ENABLED
1099 011060 50$:
1100 011060 000261 SEC ;SET CARRY NO TROUBLE
1101 011062 000401 BR 70$ ;EXIT
1102 011064 000241 CLC ;CARRY CLEAR = ERROR
1103 011066 016500 000002 70$: MOV TSSR(R5),R0 ;RETURN TSSR CONTENTS
1104 011072 000207 RTS PC ;RETURN

```

4

SBTTL REWIND POSITION TAPE (REWIND) COMMAND

1106
 1107
 1108
 1109
 1110
 1111
 1112
 1113
 1114
 1115
 1116
 1117
 1118
 1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1130
 1131
 1132
 1133 011074
 1134 011074
 1135 011100 012704 011170
 1136 011104 010465 000000
 1137 011110 012703 000550
 1138 011114 004737 016330
 1139 011120 103417
 1140 011122
 011122 012727 000372
 011126 000000
 011130 013727 002116
 011134 000000
 011136 005367 177772
 011142 001375
 011144 005367 177756
 011150 001367
 1141 011152 005303
 1142 011154 001357
 1143 011156 000241
 1144 011160 010400
 1145 011162 000207
 1146
 1148 011170
 1150 011170
 1151 011170 102010
 1152 011172 000000

```

: THIS ROUTINE WILL REWIND THE SELECTED TAPE.
: CAUTION: THE ROUTINE DOES NOT WAIT FOR BOT
:           TO ARRIVE ALSO THE CALLER MUST CHECK FOR
:           SSR TO SET IN THE TSSR
: CALLING SEQUENCE:
: DO A SOFT INIT
: DO A WRITE CHARACTERISTICS
: JSR PC,REWIND
: INPUT:
: R5 FIRST DEVICE UNIBUS ADDRESS
: OUTPUT
: R0 THE CONTENTS OF R4 IS PASSED TO R0
: -
REWIND::
  SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
  MOV @RWPACK,R4 ;GET PACKET ADDRESS
  MOV R4,TSDB(R5) ;SEND PACKET ADDRESS TO EXECUTE
  MOV @360,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
10$: JSR PC,WAITF ;WAIT FOR SSR TO SET
  BCS 20$ ;LEAVE WHEN SSR IS SET
  DELAY 250. ;WAIT FOR .25 SECONDS
  MOV @250.,(PC)+
  .WORD 0
  MOV L#DLY,(PC)+
  .WORD 0
  DEC -6(PC)
  BNE .-4
  DEC -22(PC)
  BNE .-20
  DEC R3 ;BUMP COUNTER DOWN
  BNE 10$ ;KEEP GOING
  CLC ;CLEAR CARRY TO SET ERROR
20$: MOV R4,R0 ;PASS THE PACKET ADDRESS
  RTS PC ;RETURN
RWPACK: .=<. +10>E177770
  .WORD 102010 ;POSTION COMMAND (REWIND)
  .WORD 0 ;NOT USED

```

```

1154 .SBTTL CKRAM - COMPARE RAM TO I/O PACKET
1155 ;*
1156 ;
1157 ;ROUTINE TO READ THE FIRST 8 BYTES FROM RAM
1158 ;MEMORY AND COMPARE THIS DATA TO A COMMAND PACKET.
1159 ;
1160 ;INPUT:
1161 ;
1162 ; R4 ADDRESS OF THE COMMAND PACKET
1163 ; R5 FIRST DEVICE UNIBUS ADDRESS
1164 ;
1165 ;OUTPUT:
1166 ;
1167 ; CARRY SET - RAM MATCHES PACKET
1168 ; CLR - RAM DOES NOT MATCH PACKET
1169 ;
1170 ;IMPLICIT OUTPUT:
1171 ;
1172 ; THE TABLE RAMDATA IS FILLED WITH THE
1173 ; DATA HELD IN RAM.
1174 ; RAMSIZ IS SET TO 8. FOR PRAMPKT ROUTINE
1175 ;
1176 ;SIDE EFFECTS:
1177 ;
1178 ; THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1179 ;
1180 ;
1181 ;
1182 CKRAM::
1183 SAVREG ;SAVE THE GENERAL REGISTERS
1184 MOV #RAMDATA,R1 ;ADDRESS TO SAVE THE RAM DATA
1185 MOV #RMPKTBEG,R2 ;BYTE ADDRESS OF FIRST RAM DATA
1186 CLR R3 ;CLEAR THE ERROR FLAG
1187 JSR PC,CHKTSSR ;WAIT FOR SSR
1188 MOVB #0,TSDB(R5) ;SET MAINTENANCE MODE
1189 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
1190 MOV R2,TSDB(R5) ;SELECT NEXT RAM ADDRESS
1191 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
1192 MOVB TSBA(R5),(R1) ;READ THE RAM DATA
1193 CHPB (R1),,(R4) ;COMPARE TO EXPECTED
1194 BEQ 20 ;BRANCH IF OK
1195 INC R3 ;SET ERROR FLAG
1196 INC R2 ;ADDRESS OF NEXT RAM LOCATION
1197 CMP R2,#RMPKTEND ;REACHED END YET ?
1198 BLE 10 ;BRANCH TILL ALL READ
1199 TST R3 ;WAS AN ERROR FOUND ?
1200 BEQ 30 ;BRANCH IF NOT
1201 CLC ;CLEAR CARRY TO SHOW ERROR
1202 BR 50 ;AND EXIT
1203 SEC ;SHOW GOOD COMPARE
1204 MOV #8,,RAMSIZ ;SETUP RAMSIZ FOR PRAMPKT ROUTINE
1205 RTS PC ;RETURN
    
```

```

1207 .SBTTL CKRAM2 COMPARE RAM TO I/O CHARACTERISTICS DATA
1208 ;
1209 ;
1210 ;ROUTINE TO READ THE FIRST 8 OR 10 BYTES FROM RAM
1211 ;MEMORY AND COMPARE THIS DATA TO A CHARACTERISTICS DATA BLOCK.
1212 ;
1213 ;INPUT:
1214 ;
1215 ; R4 ADDRESS OF THE CHARACTERISTICS DATA
1216 ; R5 FIRST DEVICE UNIBUS ADDRESS
1217 ;
1218 ;OUTPUT:
1219 ;
1220 ; CARRY SET - RAM MATCHES PACKET
1221 ; CLP - RAM DOES NOT MATCH PACKET
1222 ;
1223 ;IMPLICIT OUTPUT:
1224 ;
1225 ; THE TABLE RAMDATA IS FILLED WITH THE
1226 ; DATA HELD IN RAM.
1227 ; RAMSIZ IS SET TO 8. OR 10. FOR PRAMPKT ROUTINE
1228 ;
1229 ;SIDE EFFECTS:
1230 ;
1231 ; THE SUBSYSTEM IS LEFT IN MAINTENANCE MODE
1232 ;
1233 CKRAM2::
1234 SAVREG ;SAVE THE GENERAL REGISTERS
1235 MOV #RAMDATA,R1 ;ADDRESS TO SAVE THE RAM DATA
1236 MOV #RMCHBEG,R2 ;BYTE ADDRESS OF FIRST RAM DATA
1237 CLR R3 ;CLEAR THE ERROR FLAG
1238 JSR PC,CHKTSSR ;WAIT FOR SSR
1239 MOVB #0,TSDB(R5) ;SET MAINTENANCE MODE
1240 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
1241 MOV R2,TSDB(R5) ;SELECT NEXT RAM ADDRESS
1242 JSR PC,CHKTSSR ;WAIT FOR SSR TO SET
1243 MOVB TSBA(R5),(R1) ;READ THE RAM DATA
1244 CMPB (R1),.(R4) ;COMPARE TO EXPECTED
1245 BEQ 20# ;BRANCH IF OK
1246 INC R3 ;SET ERROR FLAG
1247 INC R2 ;ADDRESS OF NEXT RAM LOCATION
1248 MOV #8.,RAMSIZ ;ASSUME EXTFEA NOT SET
1249 TST EXTFEA ;IS THE SOFTWARE EXTENDED FEATURES SET
1250 BEQ 25# ;BR, IF NOT SET
1251 MOV #10.,RAMSIZ ;SET RAMSIZ FOR EXTEND FEATURES
1252 CMP R2,#RMCHEND ;AT END OF EXTENDED BUFFER
1253 BLE 10# ;BR, IF NOT AT END YET
1254 BR 27# ;AT END BRANCH
1255 CMP R2,#RMCHEND-2 ;REACHED END YET ?
1256 BLE 10# ;BRANCH TILL ALL READ
1257 TST R3 ;WAS AN ERROR FOUND ?
1258 BEQ 30# ;BRANCH IF NOT
1259 CLC ;CLEAR CARRY TO SHOW ERROR
1260 BR 50# ;AND EXIT
1261 SEC ;SHOW GOOD COMPARE
1262 RTS PC ;RETURN

```


()^r,

```

1264          .SBTTL  CKMSG      COMPARE WRITE CHAR. MESSAGE BUFFERS
1265          ;*
1266          ;
1267          ;ROUTINE TO COMPARE A WRITE CHARACTERISTICS EXPD AND RECV
1268          ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1269          ;ERROR PRINT ROUTINES.
1270          ;
1271          ;INPUT:
1272          ;
1273          ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1274          ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1275          ;      R2      EXPD MESSAGE BUFFER ADDRESS
1276          ;OUTPUT:
1277          ;
1278          ;      CARRY   SET - MESSAGE BUFFERS MATCH
1279          ;              CLR -MESSAGE BUFFERS DON'T MATCH
1280          ;
1281          ;IMPLICIT OUTPUT:
1282          ;
1283          ;      EXPMSG   BUFFER IS SET TO EXPD DATA
1284          ;      RECMMSG  BUFFER IS SET TO RECV DATA
1285          ;      RCVHIADD SET TO HIGH ORDER ADDRESS OF RECV
1286          ;      RCVLOADD SET TO LOW ORDER ADDRESS OF RECV
1287          ;
1288          ;-
1289          CKMSG::
1290          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
1291          MOV      R0,RCVHIADD ;SAVE RECV HIGH ADDRESS
1292          MOV      R1,RCVLOAD  ;SAVE RECV LOW ADDRESS
1293          TST     KTENABLE    ;TESTING ABOVE 28K?
1294          BEQ     101        ;BR IF NO
1295          JSR     PC,SETMAP   ;RETURN ADDRESS BIASED TO PAR6 IN R0
1296          MOV     R0,R1      ;GET RETURNED ADDRESS BIASED TO PAR6
1297          101:  CLR     R4      ;WORD IN BUFFER
1298          CLR     R3          ;CLEAR ERROR SEEN FLAG
1299          MOV     R2,R5      ;GET EXPD BUFFER ADDRESS
1300          151:  MOV     (R2),EXPMSG(R4) ;SAVE EXPD FOR ERROR REPORT
1301          MOV     (R1),RECMMSG(R4) ;SAVE RECV FOR ERROR REPORT
1302          CMP     (R2),.(R1) ;EXPD EQUAL RECV?
1303          BEQ     251        ;BR IF YES
1304          INC     R3          ;SET ERROR SEEN FLAG
1305          251:  ADD     @2,R4   ;POINT TO NEXT WORD ADDRESS
1306          CMP     R4,@14     ;DONE FIRST 7 WORDS?
1307          BLE     151        ;BR IF NO
1308          BIT     @X2.EXTF,XST2(R5) ;IS EXTENDED FEATURES SET IN EXPD?
1309          BEQ     501        ;BR IF NO
1310          CMP     R4,@16     ;DONE EXTENDED FEATURES WORD?
1311          BLE     151        ;BR IF NO
1312          501:  TST     R3      ;ANY ERRORS SEEN?
1313          BEQ     551        ;BR IF NO
1314          CLC          ;SET FAILURE
1315          BR     601        ;
1316          551:  SEC          ;SET SUCCESS
1317          601:  RTS     PC    ;RETURN

```

```

1319          .SBTTL CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS
1320          ;*
1321          ;ROUTINE TO COMPARE AN EXPECTED AND RECEIVED MESSAGE
1322          ;BUFFER. THE EXPECTED AND RECEIVED BUFFERS ARE STORED FOR
1323          ;ERROR PRINT ROUTINES.
1324          ;
1325          ;INPUT:
1326          ;      R0      RECV MESSAGE BUFFER HIGH ORDER ADDRESS
1327          ;      R1      RECV MESSAGE BUFFER LOW ORDER ADDRESS
1328          ;      R2      EXPD MESSAGE BUFFER ADDRESS
1329          ;      R3      NUMBER OF BYTES TO COMPARE
1330          ;
1331          ;OUTPUT:
1332          ;      CARRY   SET - MESSAGE BUFFERS MATCH
1333          ;              CLR - MESSAGE BUFFERS DON'T MATCH
1334          ;
1335          ;IMPLICIT OUTPUT:
1336          ;      EXPMSG   BUFFER IS SET TO EXPD DATA
1337          ;      RECVMSG  BUFFER IS SET TO RECV DATA
1338          ;      RCVHIADD  SET TO HIGH ORDER ADDRESS OF RECV
1339          ;      RCVLOADD  SET TO LOW ORDER ADDRESS OF RECV
1340          ;-
1341          CKMSG2::
1342          SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
1343          CMP            R3,#RECVMSG-EXPMSG;R3D IS COUNT ABOVE MAX ALLOWED?
1344          BLE           50;R3D BR IF NO
1345          MOV           #RECVMSG-EXPMSG,R3;R3D
1346          PRINTF        #DEBUGMSG          ;R3D
1347          MOV           #DEBUGMSG,-(SP)
1348          MOV           #1,-(SP)
1349          MOV           SP,R0
1350          TRAP         C#PRINTF
1351          ADD           #4,SP
1352          MOV           R0,RCVHIADD        ;SAVE RECV HIGH ADDRESS
1353          MOV           R1,RCVLOAD        ;SAVE RECV LOW ADDRESS
1354          TST          KTENABLE          ;TESTING ABOVE 28K?
1355          BEQ          10;              ;BR IF NO
1356          JSR         PC,SETMAP          ;RETURN ADDRESS BIASED TO PAR6 IN R0
1357          MOV         R0,R1              ;GET RETURNED ADDRESS BIASED TO PAR6
1358          CLR         R4                ;WORD IN BUFFER
1359          CLR         R5                ;CLEAR ERROR SEEN FLAG
1360          MOV         (R2),EXPMSG(R4)    ;SAVE EXPD FOR ERROR REPORT
1361          MOV         (R1),RECVMSG(R4)   ;SAVE RECV FOR ERROR REPORT
1362          CMP         (R2),.(R1)         ;EXPD EQUAL RECV?
1363          BEQ         25;              ;BR IF YES
1364          INC         R5                ;SET ERROR SEEN FLAG
1365          ADD         #1,R4              ;POINT TO NEXT BYTE
1366          CMP         R4,R3              ;DONE ALL BYTES?
1367          BGE         50;              ;BR IF YES
1368          BR          15;              ;DO NEXT BYTE
1369          TST         R5                ;ANY ERRORS SEEN?
1370          BEQ         55;              ;BR IF NO
1371          CLC          CLC                ;SET FAILURE
1372          BR          60;
1373          SEC         SEC                ;SET SUCCESS
1374          RTS         RTS                ;RETURN

```

```

1371 011712      120      122      117  DEBUGMSG: .ASCIZ 'PROGRAM INTERNAL ERROR -CKMSG2 MESSAGE BUFFER EXCEEDED ;@@D
1372 012002      045      116      045  FERCM: .ASCII /MNA ***/
1373 012013      040      040      124  ERCH: .ASCIZ / TSSR ERROR CODE REC'D = /
1374 012046      056      056      056  SIMSG: .ASCIZ /.... AFTER DOING SOFT INIT/
1375 012101      124      105      123  TINERR: .ASCIZ /TEST: .../
1376                                     .EVEN
1377                                     ;*
1378                                     ;
1379                                     ;PRINT ROUTINE TO FATAL SOFT INIT ERRORS
1380                                     ;
1381                                     ;INPUT:
1382                                     ;
1383                                     ;      R1      CONTENTS OF TSSR AT ERROR
1384                                     ;
1385                                     ;SIDE EFFECTS:
1386                                     ;
1387                                     ;      EXECUTES DROP UNIT TO CEASE TESTING
1388                                     ;
1389                                     ;
1390                                     ;
1391 012114      BGNMSG  SFIMSG
1392 012114      004737  006020  SFIMSG:: JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1393 012120      004737  017262  JSR      PC,CKDROP        ;DROP UNIT, IF ALLOWED
1394 012124      ENDMSG
1395 012124      104423  L10003: TRAP      C#MSG
1396                                     ;*
1397                                     ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1398                                     ;TSSR AND A COMMAND PACKET OTHER THAN GET STATUS COMMAND PACKET.
1399                                     ;
1400                                     ;INPUTS:
1401                                     ;
1402                                     ;      R1      TSSR CONTENTS
1403                                     ;      R4      ADDRESS OF COMMAND PACKET
1404                                     ;
1405                                     ;-
1406                                     ;
1407 012126      BGNMSG  PKTSSR
1408 012126      004737  006020  PKTSSR:: JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1409 012132      012700  000004  MOV      #4,R0            ;NO. OF WORDS IN PACKET
1410 012136      004737  007446  JSR      PC,PRIPKT        ;PRINT THE CONTENTS OF COMMAND PACKET
1411 012142      ENDMSG
1412 012142      104423  L10004: TRAP      C#MSG

```

TSV3 - GLOBAL AREAS MACRO M1113 14-JUN-84 16:41
 CKMSG2 - COMPARE EXPD RECV MESSAGE BUFFERS

SEQ 0058

```

1413      ;*
1414      ;PRINT ROUTINE TO PRINT THE CONTENTS OF
1415      ;TSSR AND A GET STATUS COMMAND PACKET.
1416      ;
1417      ;INPUTS:
1418      ;
1419      ;       R1       TSSR CONTENTS
1420      ;       R4       ADDRESS OF COMMAND PACKET
1421      ;
1422      ;-      BGNMSG  PKTGETS
1423      PKTGETS:
1424      JSR      PC,PRITSSR      ;PRINT THE CONTENTS OF TSSR REGISTER
1425      MOV      #2,R0          ;NO. OF WORDS IN GET STATUS PACKET
1426      JSR      PC,PRIPKT      ;PRINT THE CONTENTS OF COMMAND PACKET
1427      ENDMSG
1428      L10005:
1429      TRAP     C#MSG
1430      ;*
1431      ;PRINT TSSR ERRORS FOR INITIALIZATION TESTS
1432      ;
1433      ;INPUTS:
1434      ;       R1       TSSR CONTENTS
1435      ;       R4       ADDRESS OF COMMAND PACKET
1436      ;
1437      ;-      BGNMSG  SFFMSG
1438      SFFMSG:
1439      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR REGISTER
1440      ENDMSG
1441      L10006:
1442      TRAP     C#MSG
1443      .SBTTL   PKTMES  - PRINT TSSR AND MESSAGE BUFFER
1444      ;*
1445      ;PRINT ROUTINE TO PRINT THE CONTENTS OF TSSR AND MESSAGE
1446      ;BUFFER FOR ERROR REPORTS
1447      ;
1448      ;INPUTS:
1449      ;
1450      ;       R1       CONTENTS OF TSSR
1451      ;       R2       LOW ORDER MESSAGE BUFFER
1452      ;       R3       HIGH ORDER MESSAGE BUFFER ADDRESS
1453      ;       NOTE: R3 IS IGNORED IF KTENABLE FLAG IS CLEAR
1454      ;
1455      ;-      BGNMSG  PKTMES
1456      PKTMES:
1457      JSR      PC,PRITSSR      ;PRINT CONTENTS OF TSSR
1458      MOV      R2,R0          ;LOW ORDER ADDRESS
1459      MOV      R3,R1          ;HIGH ORDER ADDRESS
1460      JSR      PC,PRMESS      ;PRINT THE MESSAGE BUFFER
1461      ENDMSG
1462      L10007:
1463      TRAP     C#MSG

```

H^c,

```

1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468 012206
      012206
1469 012206 004737 010352
1470 012212 016501 000002
1471 012216 004737 006020
1472 012222
      012222
      012222 104423
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486 012224
      012224
1487 012224 012700 000007
1488 012230 005737 002220
1489 012234 001402
1490 012236 012700 000010
1491 012242 004737 014632
1492 012246
      012246
      012246 104423

```

```

.SBTTL ADDSSR PRINT TEST ADDRESS AND TSSR
;*
;PRINT ROUTINE TO PRINT THE CONTENTS OF
;TSSR AND A MEMORY TEST ADDRESS
;
;INPUTS:
;
; R5 FIRST DEVICE UNIBUS ADDRESS
; ERRHI HIGH ORDER MEMORY TEST ADDRESS
; ERRLO LOW ORDER MEMORY TEST ADDRESS
;
      BGNMSG ADDSSR
ADDSSR::
      JSR PC,PRITADD ;PRINT MEMORY TEST ADDRESS
      MOV TSSR(R5),R1 ;GET CURRENT TSSR
      JSR PC,PRITSSR ;PRINT THE CONTENTS OF TSSR REGISTER
      ENDMSG
L10010:
      TRAP C#MSG

.SBTTL MSGEXP - PRINT WRITE CHAR. EXPD-RECV MESSAGE BUFFERS
;*
;PRINT ROUTINE TO PRINT WRITE CHARACTERISTIC MESSAGE BUFFER
;
;IMPLICIT INPUTS:
;
; EXPMSG - EXPECTED MESSAGE BUFFER
; RECMMSG - RECEIVED MESSAGE BUFFER
; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
;
      BGNMSG MSGEXP
MSGEXP::
      MOV #7,R0 ;ASSUME NO EXT FEATURES
      TST EXTFEA ;EXT FEATURES SET?
      BEQ 5$ ;BR IF NO
      MOV #8.,R0 ;EXT FEATURE BUFFER IS 8 WORDS
      JSR PC,PRMSGEXP ;PRINT EXPD/RECV MESSAGE BUFFERS
      ENDMSG
5$:
L10011:
      TRAP C#MSG

```

1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506 012250
012250
1507 012250
012250 010146
012252 012746 012322
012256 012746 000002
012262 010600
012264 104415
012266 062706 000006
1508 012272
012272 012746 012371
012276 012746 000001
012302 010600
012304 104415
012306 062706 000004
1509 012312 010100
1510 012314 004737 015202
1511 012320
012320
012320 104423
1512 012322 045 116
1513 012371 045 116
1514

```

.SBTTL FIFEXP PRINT FIFO EXP/RECV DATA
:
:PRINT ROUTINE TO PRINT FIFO EXP/RECV DATA
:
: R1 BYTE COUNT
:
:IMPLICIT INPUTS:
:
: EXPMSG - EXPECTED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
: RECMMSG - RECEIVED MESSAGE BUFFER (CONTAINS FIFO DATA ONLY)
:
: BGNMSG FIFEXP
FIFEXP::
PRINTX #FIF1MSG,R1 ;PRINT BYTES TRANSFERRED
MOV R1,-(SP)
MOV #FIF1MSG,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #6,SP
PRINTX #FIF2MSG ;PRINT HEADER MSG
MOV #FIF2MSG,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
MOV R1,R0 ;GET BYTE COUNT
JSR PC,PRBYTEXP ;PRINT FIFO BYTES IN ERROR
ENDMSG
L10012:
TRAP C#MSG
045 FIF1MSG: .ASCIZ '#N#A NUMBER OF BYTES TRANSFERRED = #02
045 FIF2MSG: .ASCIZ '#N#A FIFO DATA BYTES IN ERROR:'
.EVEN

```

```

1516 .SBTTL MSGSTAT PRINT STATUS HEADER AND MESSAGE BUFFERS
1517 :
1518 :
1519 :PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1520 :
1521 :
1522 :IMPLICIT INPUTS:
1523 :
1524 : EXPMSG - EXPECTED MESSAGE BUFFER
1525 : RECMMSG - RECEIVED MESSAGE BUFFER
1526 : RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1527 : RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1528 :
1529 012430 BGNMSG MSGSTAT
012430 MSGSTAT:
1530 012430 012701 012472 MOV #STATCOD,R1 ;ASCII ADDRESS TABLE
1531 012434 012100 10$: MOV (R1)+,RO ;DONE ALL MSG LINES?
1532 012436 001410 BEQ 20$ ;BR IF YES
1533 012440 PRINTX RO ;PRINT STATUS BIT NAMES
012440 010046 MOV RO,-(SP)
012442 012746 000001 MOV #1,-(SP)
012446 010600 MOV SP,RO
012450 104415 TRAP C#PNTX
012452 062706 000004 ADD #4,SP
1534 012456 000766 BR 10$ ;DO ANOTHER MSG LINE
1535 012460 012700 000012 20$: MOV #10.,RO ;NUMBER OF WORDS IN A READ STATUS BUFFER
1536 012464 004737 014632 JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1537 012470 ENDMMSG
012470 L10013:
012470 104423 TRAP C#MSG
1538
1539 012472 012510 012552 012643 STATCOD: .WORD 1$,2$,3$,4$,5$,6$,0
1540 012510 045 116 045 1$: .ASCIZ 'ANSA Tape Bus Signals in Word #8:'
1541 012552 045 116 045 2$: .ASCIZ 'ANSA PARERR<15> IEOT <12> IFMK <9> IRDY<6> IRWD<2>'
1542 012643 045 116 045 3$: .ASCIZ 'ANSA IRESV2<14> IIDENT<11> IHER <8> IONL<5> IFBY<1>'
1543 012734 045 116 045 4$: .ASCIZ 'ANSA IRESV1<13> ICER <10> ISPEED<7> ILDP<4> IFPT<0>'
1544 013025 045 116 045 5$: .ASCIZ 'ANSA Tape Bus Signals in Word #9:'
1545 013067 045 116 045 6$: .ASCIZ 'ANSA DATMIS<7> ILW<6> OUTRDY<5> INRDY<4>'
1546 .EVEN
1547

```

```

1549 .SBTTL MSGLOOP - PRINT LOOPBACK HEADER AND MESSAGE BUFFERS
1550 ;*
1551 ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RECV
1552 ;
1553 ;IMPLICIT INPUTS:
1554 ;
1555 ; EXPMSG - EXPECTED MESSAGE BUFFER
1556 ; RECMMSG - RECEIVED MESSAGE BUFFER
1557 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1558 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1559 ;
1560 ;
1561 013144 BGNMSG MSGLOOP
013144 MSGLOOP:
1562 013144 012701 013206 MOV #LOOPCOD,R1 ;ASCII ADDRESS TABLE
1563 013150 012100 10$: MOV (R1)+,R0 ;DONE ALL MSG LINES?
1564 013152 001410 BEQ 20$ ;BR IF YES
1565 013154 PRINTX R0 ;PRINT STATUS BIT NAMES
013154 010046 MOV R0,-(SP)
013156 012746 000001 MOV #1,-(SP)
013162 010600 MOV _P,R0
013164 104415 TRAP C#PNTX
013166 062706 000004 ADD #4,SP
1566 013172 000766 BR 10$ ;DO ANOTHER MSG LINE
1567 013174 012700 000012 20$: MOV #10.,R0 ;NUMBER OF WORDS IN A READ STATUS BUFFER
1568 013200 004737 014632 JSR PC,PRMSGEXP ;PRINT EXPD/RECV MESSAGE BUFFERS
1569 013204 ENDMSG
013204 L10014:
013204 104423 TRAP C#MSG
1570
1571 013206 013226 013301 013400 LOOPCOD: .WORD 1$,2$,3$,4$,5$,6$,7$,0
1572 013226 045 116 045 1$: .ASCIZ 'ANSA Tape Bus Loopback Signals in Word #8:'
1573 013301 045 116 045 2$: .ASCIZ 'ANSA PARERR<15> IRESV2<14> IRESV1<13>'
1574 013400 045 116 045 3$: .ASCIZ 'ANSA IHISP=>IEOT<12> IWRT=>IDENT<11> IREV =>ICER <10>'
1575 013477 045 116 045 4$: .ASCIZ 'ANSA IWM =>IFMK<09> IEDIT=>IHER <08> IFAD =>ISPEED<07>'
1576 013576 045 116 045 5$: .ASCIZ 'ANSA ITADO=>IRDY<06> ITAD1=>IONL <05> IERASE=>ILDPA <04>'
1577 013675 045 116 045 6$: .ASCIZ 'ANSA IREW =>IDBY<03> IRWU =>IRWD <02> IFEN =>IFBY <01>'
1578 013774 045 116 045 7$: .ASCIZ 'ANSA IGO =>IFPT<00>'
1579 .EVEN

```



```

1581          .SBTTL MSGSUB PRINT WRITE SUBSYSTEM MESSAGE BUFFER
1582          ;*
1583          ;PRINT ROUTINE TO PRINT MESSAGE BUFFER EXPD/RCV
1584          ;
1585          ;IMPLICIT INPUTS:
1586          ;
1587          ;   EXPMSG - EXPECTED MESSAGE BUFFER
1588          ;   RECMMSG - RECEIVED MESSAGE BUFFER
1589          ;   RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1590          ;   RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1591          ;
1592          ;-
1593          BGNMSG MSGSUB
1594 014022 MSGSUB:
1595 014022 012700 000012 MOV #10.,R0 ;SIZE OF WRITE SUBSYSTEM BUFFER
1596 014026 004737 014632 JSR PC,PRMSGEXP ;PRINT EXPD/RCV MESSAGE BUFFERS
1597 014032 ENDMMSG
1598          L10015:
1599          TRAP C#MSG
1600          .SBTTL MEMADD PRINT MEMORY ADDRESS DATA ERROR
1601          ;*
1602          ;PRINT ROUTINE TO PRINT MEMORY ADDRESS DATA COMPARE ERROR
1603          ;
1604          ;IMPLICIT INPUTS:
1605          ;
1606          ;   LRRHI - MEMORY ERROR HIGH ORDER ADDRESS
1607          ;   ERRLO - MEMORY ERROR LOW ORDER ADDRESS
1608          ;   EXP - EXPECTED DATA
1609          ;   RECV - RECEIVED DATA
1610          ;
1611          ;-
1612          BGNMSG MEMADD
1613 014034 MEMADD:
1614 014034 004737 010236 JSR PC,PRIADD ;PRINT MEMORY ADDRESS IN ERROR
1615 014040 013701 002224 MOV EXPD,R1 ;GET EXPD DATA
1616 014044 013702 002226 MOV RECV,R2 ;GET RECEIVED DATA
1617 014050 004737 010020 JSR PC,PRIXOR ;PRINT EXPD/RCV
1618 014054 ENDMMSG
1619          L10016:
1620          TRAP C#MSG

```

```

1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634 014056
1635 014056
1636 014062 012701 002234
1637 014066 0C5002
1638 014070 122124
1639 014072 001005
1640 014074
1641 014104 000436
1642 014106 116105 177777
1643 014112 116403 177777
1644 014116
1645 014126 042703 177400
1646 014132 116137 177777 002226
1647 014140 116437 177777 002224
1648 014146
    014146 010346
    014150 013746 002224
    014154 013746 002226
    014160 010246
    014162 012746 014236
    014166 012746 000005
    014172 010600
    014174 104414
    014176 062706 000014
1649 014202 005202
1650 014204 005737 002274
1651 014210 001404
1652 014212 020237 002274
1653 014216 003724
1654 014220 000403
1655 014222 020227 000010
1656 014226 002720
1657 014230 005037 002274
1658 014234 000207
1659
1660 014236 045 116 045 RAMASC: .ASCIZ '##A BYTE: #02#A RAM: #03#A Packet: #03#A XOR:#03
1661

```

```

.SBTTL PRAMPKT PRINT RAM AND PACKET DATA
;
;PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
;WHEN THE RAM DATA DOES NOT MATCH.
;
;INPUTS:
;
; R4 POINTER TO COMMAND PACKET
;IMPLICIT INPUTS:
; RAMDATA DATA AS READ FROM THE RAM
; RAMSIZ NUMBER OF BYTES IN PACKET
; IF RAMSIZ=0 THEN DEFAULT TO 8.
;
;IMPLICIT OUTPUTS:
; RAMSIZ SET TO 0
;
PRAMPKT:
    SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
    MOV #RAMDATA,R1 ;DATA FROM THE RAM
    CLR R2 ;INIT BYTE NUMBER
5:    CMPB (R1),.(R4). ;COMPARE EXPECTED, RECEIVED
    BNE 7: ;BR IF NO MATCH
    FORCERROR 7: ,NOTSSR
    BR 10: ;BND
7:    MOVB -1(R1),R5 ;GET RECV RAM DATA
    MOVB -1(R4),R3 ;GET EXPD PACKET DATA
    XOR R5,R3 ;XOR EXPD/RECV
    BIC #177400,R3 ;LOW BYTE ONLY
    MOVB -1(R1),RECV ;GET RECEIVED RAM DATA
    MOVB -1(R4),EXPD ;GET EXPECTED RAM DATA
    PRINTB #RAMASC,R2,RECV,EXPD,R3
    MOV R3,-(SP)
    MOV EXPD,-(SP)
    MOV RECV,-(SP)
    MOV R2,-(SP)
    MOV #RAMASC,-(SP)
    MOV #5,-(SP)
    MOV SP,R0
    TRAP C#PNTB
    ADD #14,SP
10:    INC R2 ;UPDATE BYTE COUNT
    TST RAMSIZ ;DEFAULT TO 8.?
    BEQ 15: ;BR IF YES
    CMP R2,RAMSIZ ;DONE ALL BYTES?
    BLE 5: ;BR IF NO
    BR 25: ;
15:    CMP R2,#8. ;DONE DEFAULT NUMBER OF BYTES?
20:    BLT 5: ;BR IF NO
25:    CLR RAMSIZ ;SET DEFAULT RAMSIZ
    RTS PC ;RETURN

```

NE

```

1663 .SBTTL PRMESS PRINT CONTENTS OF MESSAGE BUFFER
1664 :
1665 : THIS ROUTINE PRINTS THE CONTENTS OF
1666 : THE 7 OR 8 WORD MESSAGE BUFFER RETURNED BY THE TSV 05.
1667 :
1668 : INPUT:
1669 : R0 LOW ORDER ADDRESS OF MESSAGE BUFFER
1670 : R1 HIGH ORDER ADDRESS OF MESSAGE BUFFER
1671 : NOTE: R1 IS IGNORED IF KTENABLE FLAG IS CLEAR
1672 : THIS ROUTINE IS NORMALLY CALLED FROM A PRINT ROUTINE
1673 :
1674 014322 PRMESS: SAVREG ;SAVE THE REGISTERS
1675 014326 010005 MOV R0,R5 ;SAVE LOW ORDER ADDRESS
1676 014330 005737 003126 TST KTENABLE ;ADDRESS ABOVE 28K?
1677 014334 001001 BNE 10$ ;BR IF YES
1678 014336 005001 CLR R1 ;SET HIGH ORDER ADDRESS TO 0
1679 014340 010103 10$: MOV R1,R3 ;SAVE HIGH ORDER ADDRESS
1680 014342 006100 ROL R0 ;SHIFT BIT15 TO C BIT
1681 014344 006101 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
1682 014346 PRINTX @PROASC,R1,R5 ;PRINT MESSAGE BUFFER ADDRESS
1683 014346 010546 MOV R5,-(SP)
1684 014350 010146 MOV R1,-(SP)
1685 014352 012746 014500 MOV @PROASC,(SP)
1686 014356 012746 000003 MOV @3,-(SP)
1687 014362 010600 MOV SP,R0
1688 014364 104415 TRAP C$PNTX
1689 014366 062706 000010 ADD @10,SP
1690 014372 PRINTX @PRIASC ;PRINT HEADER FOR CONTENTS
1691 014372 012746 014545 MOV @PRIASC,-(SP)
1692 014376 012746 000001 MOV @1,-(SP)
1693 014402 010600 MOV SP,R0
1694 014404 104415 TRAP C$PNTX
1695 014406 062706 000004 ADD @4,SP
1696 014412 005004 CLR R4 ;NUMBER OF THE NEXT WORD
1697 014414 010501 MOV R5,R1 ;COPY LOW ORDER ADDRESS
1698 014416 010300 MOV R3,R0 ;COPY HIGH ORDER ADDRESS
1699 014420 001403 BEQ 20$ ;BR IF NOT ABOVE 28K
1700 014422 004737 017376 JSR PC,SETMAP ;SETUP PAR ADDRESS IN R0
1701 014426 010005 MOV R0,R5 ;GET PAR FORMAT ADDRESS ABOVE 28K
1702 014430 20$: PRINTX @PRASC,R4,(R5)+ ;PRINT THE CONTENTS OF MEMORY BUFFER
1703 014430 012546 MOV (R5)+,-(SP)
1704 014432 010446 MOV R4,-(SP)
1705 014434 012746 014603 MOV @PRASC,-(SP)
1706 014440 012746 000003 MOV @3,-(SP)
1707 014444 010600 MOV SP,R0
1708 014446 104415 TRAP C$PNTX
1709 014450 062706 000010 ADD @10,SP
1710 014454 005204 INC R4 ;NUMBER OF THE NEXT
1711 014456 020427 000007 CMP R4,@7 ;DONE ALL YET ?
1712 014462 003005 BGT 50$ ;BRANCH IF ALL DONE
1713 014464 002761 BLT 20$ ;PRINT FIRST 7 WORDS
1714 014466 032763 000200 000012 BIT @X2.EXTF,XST2(R3);EXTENDED FEATUTES ON ?
1715 014474 001355 BNE 20$ ;PRINT EXTENDED STATUS WORD
1716 014476 000207 50$: RTS PC ;RETURN
1717 014500 045 116 045 PROASC: .ASCIZ '#N$A Message Buffer Address = #01#05'
1718 014545 045 116 045 PRIASC: .ASCIZ '#N$A Message Buffer Contents:'
1719 014603 045 116 045 PRASC: .ASCIZ '#N$A Word#D1$A: #0'

```

```

1702 .EVEN
1703 .SBTTL PRMSGEXP PRINT EXPD/RCV MESSAGE BUFFERS
1704
1705 ;ROUTINE TO PRINT EXPECTED AND RECEIVED MESSAGE BUFFERS
1706 ; RO - NUMBER OF WORDS IN BUFFER
1707 ;IMPLICIT INPUTS:
1708 ; EXPMSG - EXPECTED MESSAGE BUFFER
1709 ; RECMSG - RECEIVED MESSAGE BUFFER
1710 ; RCVHIADD- RECEIVED MESSAGE BUFFER HIGH ORDER ADDRESS
1711 ; RCVLOADD- RECEIVED MESSAGE BUFFER LOW ORDER ADDRESS
1712 ;
1713 PRMSGEXP::
1714 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1715 MOV RO,R5 ;SAVE NUMBER OF WORDS
1716 MOV RCVLOADD,RO ;GET RCV LOW ADDRESS
1717 MOV RO,R4 ;COPY LOW ADDRESS
1718 MOV RCVHIADD,R1 ;GET RCV HIGH ADDRESS
1719 ROL RO ;SHIFT BIT15 TO C BIT
1720 ROL R1 ;SHIFT TO HIGH ORDER FOR PRINTOUT
1721 PRINTX @PRMSG0,R1,R4 ;PRINT MESSAGE BUFFER ADDRESS
1722 MOV R4,-(SP)
1723 MOV R1,-(SP)
1724 MOV @PRMSG0,-(SP)
1725 MOV @3,-(SP)
1726 MOV SP,RO
1727 TRAP C:PNTX
1728 ADD #10,SP
1729 PRINTX @PRMSG1 ;PRINT HEADER FOR CONTENTS
1730 MOV @PRMSG1,-(SP)
1731 MOV #1,-(SP)
1732 MOV SP,RO
1733 TRAP C:PNTX
1734 ADD #4,SP
1735 CLR R4 ;NUMBER OF THE CURRENT WORD
1736 MOV @EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1737 MOV @RECMSG,R2 ;GET RCV BUFFER ADDRESS
201: MOV (R1),R0 ;GET EXPD
1738 MOV (R2),R3 ;GET RCV
1739 XOR R0,R3 ;XOR EXPD/RCV
1740 PRINTX @PRMSG2,R4,(R1),.(R2),.R3
1741 MOV R3,-(SP)
1742 MOV (R2),.-(SP)
1743 MOV (R1),.-(SP)
1744 MOV R4,-(SP)
1745 MOV @PRMSG2,-(SP)
1746 MOV #5,-(SP)
1747 MOV SP,RO
1748 TRAP C:PNTX
1749 ADD #14,SP
1750 INC R4 ;NUMBER OF THE NEXT
1751 CMP R4,R5 ;DONE ALL YET?
1752 BGE 501 ;BR IF YES
1753 BR 201 ;DO ANOTHER
1754 501: RTS PC ;RETURN
1755 045 PRMSG0: .ASCIZ '#N#A Message Buffer Address = #01#05'
1756 045 PRMSG1: .ASCIZ '#N#A Message Buffer Contents:'
1757 045 PRMSG2: .ASCIZ '#N#A WORD #D2#A EXPD: #06#A RCV: #06#A XOR: #06#A'
    
```

```

1730 .EVEN
1740 .SBTTL PRBYTEXP - PRINT ERROR BYTES IN EXP/REC MESSAGE BUFFER
1741 ;
1742 ;
1743 ; ROUTINE TO PRINT ERROR BYTES IN MESSAGE BUFFERS
1744 ; ONLY THE FIRST 8 ERRORS ENCOUNTERED ARE PRINTED DUE TO SCREEN SPACE
1745 ;
1746 ; R0 - NUMBER OF BYTES IN BUFFER
1747 ;
1748 ; IMPLICIT INPUTS:
1749 ;
1750 ; EXPMSG - EXPECTED MESSAGE BUFFER
1751 ; RECMSG - RECEIVED MESSAGE BUFFER
1752 ;
1753 015202 PRBYTEXP::
1754 015202 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
1755 015206 010005 MOV R0,R5 ;SAVE NUMBER OF BYTES
1756 015210 005037 002312 CLR PRMNO ;INIT ERROR COUNT
1757 015214 005004 CLR R4 ;NUMBER OF THE CURRENT BYTE
1758 015216 012701 002314 MOV @EXPMSG,R1 ;GET EXPD BUFFER ADDRESS
1759 015222 012702 002460 MOV @RECMSG,R2 ;GET RECV BUFFER ADDRESS
1760 015226 111100 201: MOV (R1),R0 ;GET EXPD BYTE
1761 015230 042700 177400 BIC @C<377>,R0 ;CLEAR UPPER BYTE
1762 015234 110037 015550 MOV R0,PRBEXP ;SAVE FOR ERROR REPORT
1763 015240 111203 MOV (R2),R3 ;GET RECV BYTE
1764 015242 042703 177400 BIC @C<377>,R3 ;CLEAR UPPER BYTE
1765 015246 110337 015552 MOV R3,PRBREC ;FOR ERROR REPORT
1766 015252 XOR R0,R3 ;XOR EXPD/RECV
1767 015262 122122 CMPB (R1),R3 ;EXPD = RECV?
1768 015264 001431 BEQ 301 ;BR IF YES
1769 015266 005237 002312 INC PRMNO ;UPDATE ERROR COUNT
1770 015272 023727 002312 000010 CMP PRMNO,#8 ;PRINTED 8?
1771 015300 101023 BHI 301 ;BR IF YES
1772 015302 271: PRINTX @PRBMSG,R4,PRBEXP,PRBREC,R3
015302 010346 MOV R3,-(SP)
015304 013746 015552 MOV PRBREC,-(SP)
015310 013746 015550 MOV PRBEXP,-(SP)
015314 010446 MOV R4,-(SP)
015316 012746 015416 MOV @PRBMSG,-(SP)
015322 012746 000005 MOV #5,-(SP)
015326 010600 MOV SP,R0
015330 104415 TRAP C#PNTX
015332 062706 000014 ADD #14,SP
1773 015336 FORCEXIT 501 ;BBD
1774 015346 000404 BR 351 ;BBD
1775 015350 301:
1776 015350 FORCERROR 271,NOTSSR ;BBD
1777 015360 351:
1778 015360 005204 INC R4 ;NUMBER OF THE NEXT
1779 015362 020405 CMP R4,R5 ;DONE ALL YET?
1780 015364 002001 BGE 501 ;BP IF YES
1781 015366 000717 BR 201 ;DO ANOTHER
1782 015370 501: PRINTX @PRBTOT,PRMNO ;PRINT TOTAL ERROR COUNT
015370 013746 002312 MOV PRMNO,-(SP)
015374 012746 015503 MOV @PRBTOT,-(SP)
015400 012746 000002 MOV #2,-(SP)
015404 010600 MOV SP,R0

```


TSV3 - GLOBAL AREAS MACRO M1113 14 JUN-84 16:41
 RAMERR - PRINT RAM AND PACKET DATA

SEQ 0069

```

1832 ;IMPLICIT INPUTS:
1833 ;
1834 ; RAMDATA DATA AS READ FROM THE RAM
1835 ; RAMSIZ NUMBER OF BYTES IN PACKET
1836 ; IF RAMSIZ=0 THEN DEFAULT TO 8.
1837 ;
1838 ;IMPLICIT OUTPUTS:
1839 ;
1840 ; RAMSIZ SET TO 0
1841 ;
1842 ;
1843 015570 BGNMSG RAMERR
015570 RAMERR::
1844 015570 004737 014056 JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA
1845 015574 ENDMSG
015574 L10021:
015574 104423 TRAP C#MSG
;
; .SBTTL RAMTADD - PRINT TEST ADDRESS, RAM AND PACKET DATA
;
; PRINT ROUTINE TO DISPLAY RAM/PACKET DATA
;
; INPUTS:
;
; R4 POINTER TO COMMAND PACKET
;
;IMPLICIT INPUTS:
;
; RAMDATA DATA AS READ FROM THE RAM
; RAMSIZ NUMBER OF BYTES IN PACKET
; IF RAMSIZ=0 THEN DEFAULT TO 8.
; ERRHI HIGH ORDER TEST ADDRESS
; ERRLO LOW ORDER TEST ADDRESS
;
;IMPLICIT OUTPUTS:
;
; RAMSIZ SET TO 0
;-
;
; BGNMSG RAMTADD
RAMTADD::
1870 015576 004737 010352 JSR PC,PRITADD ;PRINT TEST ADDRESS
1871 015602 004737 014056 JSR PC,PRAMPKT ;PRINT RAM/PACKET DATA
1872 015606 ENDMSG
015606 L10022:
015606 104423 TRAP C#MSG
;
; .SBTTL RAMEXP - PRINT RAM EXPD/RECV DATA
;
; PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
;
; INPUTS:
;
; R1 RECEIVED DATA
; R2 EXPECTED DATA

```

```

1883          :      R4      CONTROLLER RAM ADDRESS
1884          :
1885
1886 015610    BGNMSG  RAMEXP
          RAMEXP::
1887 015610    042701  177400    BIC      #C<377>,R1      ;SAVE EXPD RAM DATA BYTE
1888 015614    042702  177400    BIC      #C<377>,R2      ;SAVE EXPD RAM DATA BYTE
1889 015620    004737  010144    JSR      PC,PRIRAM      ;PRINT THE RAM ADDRESS
1890 015624    004737  010020    JSR      PC,PRIXOR      ;PRINT THE DATA
1891 015630    ENDMSG
          L10023:
          015630  104423    TRAP      C#MSG
1892
1893          .SBTTL  TIMEXP  PRINT TIMER A,B AND EXP/REC
1894          ;*
1895          ;
1896          ;PRINT ROUTINE TO DISPLAY EXPD/RECV DATA
1897          ;AND TIMER A,B HEADER MESSAGE
1898          ;
1899          ;INPUTS:
1900          ;
1901          ;      R1      RECEIVED DATA
1902          ;      R2      EXPECTED DATA
1903          ;-
1904
1905 015632    BGNMSG  TIMEXP
          TIMEXP::
1906 015632    PRINTX  #TIMSGO      ;PRINT HEADER
          015632  012746  015660    MOV      #TIMSGO,-(SP)
          015636  012746  000001    MOV      #1,-(SP)
          015642  010600    MOV      SP,R0
          015644  104415    TRAP      C#PNTX
          015646  062706  000004    ADD      #4,SP
1907 015652  004737  010020    JSR      PC,PRIXOR      ;PRINT THE DATA
1908 015656    ENDMSG
          L10024:
          015656  104423    TRAP      C#MSG
1909
1910 015660    045      116      045  TIMSGO: .ASCIZ  'ANNA TIMER A STATUS IS IN BIT 3ANNA TIMER B STATUS IS IN BIT 2'
1911          .EVEN
1912          .SBTTL  BADSSR - PRINT TSSR ERRORS ON DATA TRANSFERS
1913
1914          ;*
1915          ;
1916          ;PRINT ROUTINE FOR TSSR ERRORS ON DATA TRANSFERS
1917          ;
1918          ;INPUTS:
1919          ;
1920          ;      R1      CONTENTS OF TSSR
1921          ;      R2      DATA WRITTEN (8 BITS)
1922          ;
1923          ;
1924
1925 015760    BGNMSG  BADSSR
          BADSSR::
1926 015760  010246    MOV      R2,-(SP)      ;SAVE DATA TRANSFERRED
1927 015762  042702  177400    BIC      #177400,R2    ;GET JUST ONE BYTE
    
```



```

1928 015766          PRINTB  #XFERASC,R2
      015766 010246      MOV     R2,-(SP)
      015770 012746 016020  MOV     #XFERASC,-(SP)
      015774 012746 000002  MOV     #2,(SP)
      016000 010600      MOV     SP,R0
      016002 104414      TRAP   C#PNTB
      016004 062706 000006  ADD     #6,SP
1929 016010          MOV     (SP)+,R2          ;RESTORE R2
1930 016012 004737 006020  JSR    PC,PRITSSR      ;DECODE TSSR CONTENTS
1931 016016          ENDMSG
      016016          L10025:
      016016 104423      TRAP   C#MSG
1932 01602C 045 116 045 XFERASC: .ASCIZ 'N/A Data Transferred - #03'

```

1934
 1935
 1936
 1937
 1938
 1939
 1940
 1941
 1942
 1943
 1944
 1945
 1946
 1947
 1948
 1949
 1950
 1951
 1952
 1953
 1954
 1955
 1956
 1957
 1958
 1959
 1960
 1961
 1962
 1963
 1964
 1965
 1966
 1967
 1968 016054
 1969 016054
 1970 016060 012765 000000 000002
 1971 016066 004737 016330
 1972 016072 016500 000002
 1973 016076 010004
 1974 016100 042704 176277
 1975 016104 052704 002200
 1976 016110 020400
 1977 016112 001402
 1978 016114 000241
 1979 016116 000401
 1980 016120 000261
 1981 016122 000207

```
.SBTTL GLOBAL SUBROUTINES SECTION
; **
; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
; THAT ARE USED IN MORE THAN ONE TEST.
; --
        .SBTTL SOFINIT - SOFT INITIALIZE OF CONTROLLER
; *
;
; ROUTINE TO DO A SOFT INITIALIZE OF THE CONTROLLER
; BY WRITING INTO THE TSSR REGISTER. AFTER THE INIT,
; THE TSSR REGISTER IS TESTED FOR ERRORS. ANY ERRORS
; DETECTED SHOULD BE TREATED AS DEVICE FATAL ERRORS.
;
; INPUTS:
;
;     R5      ADDRESS OF FIRST REGISTER
;
; OUTPUTS:
;
;     R0      CONTENTS OF TSSR, IF ERROR
;     CARRY   SET IF INIT WAS OKAY
;             CLEAR IF FATAL ERROR
;
; CALLING SEQUENCE:
;
;     MOV     @ADDRESS,R5
;     JSR     PC,SOFINIT
;     BCS     CONTINUE
;     ERROF          ;REPORT FATAL ERROR
; --
SOFINIT::
        SAVREG          ; SAVE THE REGISTERS
        MOV     @0,TSSR(R5) ; DO THE INIT.
        JSR     PC,WAITF  ; WAIT FOR SSR
        MOV     TSSR(R5),R0 ; GET THE TSSR REGISTER
        MOV     R0,R4     ; TSSR CONTENTS
        BIC     @+C<HIADDR!OFL>,R4
        BIS     @SSR!NBA,R4 ; R4 HAS EXPECTED CONTENTS
        CMP     R4,R0     ; ONLY EXPECTED BITS SET ?
        BEQ     5$        ; BRANCH IF OKAY
        CLC          ; CLEAR THE CARRY FOR ERROR
        BR     10$       ; GO TO EXIT
5$:     SEC          ; SET THE CARRY BIT
10$:    RTS     PC      ; RETURN TO CALLER
```

```

1983 .SBTTL CHKAMB - CHECK TSSR FOR AMBIGUITY
1984
1985 ;*
1986 ;
1987 ;THIS ROUTINE TESTS THE CONTENTS OF THE TSSR REGISTER
1988 ;FOR AMBIGUITY
1989 ;
1990 ;INPUT:
1991 ;
1992 ;      RO      CONTENTS OF TSSR
1993 ;
1994 ;OUTPUT:
1995 ;
1996 ;      RO      CONTENTS OF TSSR
1997 ;
1998 ;      CARRY   SET - NO AMBIGUITY
1999 ;              CLR - AMBIGUOUS CONTENTS
2000 ;
2001 ;-
2002
2003 CHKAMB:
2004 SAVREG          ;SAVE THE GENERAL REGISTERS
2005 MOV            RO,R4          ;CONTENTS OF TSSR
2006 BIT           #SC,RO         ;IS BIT 15 SET ?
2007 BNE           5$            ;BRANCH IF YES
2008 BIT           #+C<NBA!OFL!SSR!HIADDR>,RO ;ANY OTHER BITS SET ?
2009 BNE           40$          ;MUST BE AN ERROR
2010 BR           45$          ;RETURN WITH SUCCESS
2011 BIT           #SSR,RO       ;IS READY BIT SET ?
2012 BNE           10$          ;BRANCH IF READY BIT IS SET.
2013 BIT           #BITS,RO     ;IS FATAL ERROR BIT SET ?
2014 BEQ           40$          ;ERROR IF NOT
2015 BIC           #+CTERCLS,R4 ;CLEAR ALL BUT TERMINATION CODE
2016 CMP           R4,#16       ;ALL THREE BITS MUST BE SET
2017 BNE           40$          ;ERROR IF NOT SET
2018 BR           45$          ;OK IF ALL ARE SET
2019 BIT           #BITS,RO     ;IS FATAL ERROR BIT SET ?
2020 BEQ           45$          ;ERROR IF BIT IS SET WITH SSR
2021 BIT           #BIT2!BIT1,RO ;IS THIS A FUNCTION REJECT
2022 BNE           45$          ;BR, IF TSSR IS OK
2023 CLC           40$          ;AMBIGUOUS CONTENTS
2024 BR           50$
2025 SEC           45$          ;SHOW SUCCESS - NO AMBIGUITY
2026 RTS           50$          ;RETURN TO CALLER

```

```

2028 .SBTTL ENAINT,DSBINT - ENABLE/DISABLE INTERRUPTS
2029 ;
2030 ; DEFAULT DISPLAY INTERRUPT HANDLERS.
2031 ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
2032 ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
2033 ;
2034 ;
2035 ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
2036 ;
2037 ;         IOKCKIN=BIT7      ; DON'T CHECK FOR BAD INTERRUPTS - TEST WILL.
2038 ;         IOKSTP=BIT0      ; EXPECT "STOP" INTERRUPT.
2039 ;
2040 ; INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
2041 INTMASK: .BYTE 0
2042 ; INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
2043 INTFLAG: .BYTE 0
2044 ;
2045 ; SAVED INTERRUPT VECTOR:
2046 INTVEC: .WORD 0
2047 ; SAVE CPU PC
2048 INTCPC: .WORD 0
2049 ;
2050 ; SUBROUTINE TO ENABLE INTERRUPTS:
2051 ENAINT: MOV     RO,-(SP)      ;SAVE RO
2052         MOV     IVEC,RO      ;GET POINTER TO VECTORS
2053         MOV     @INTR,(RO)+  ;SET UP INTERRUPT VECTOR
2054         MOV     @PRI07,(RO)+
2055         MOV     (SP)+,RO      ;RESTORE RO
2056         MOV     (SP),-(SP)
2057         MOV     @0,2(SP)     ;SET CPU TO LEVEL 0
2058         RTI
2059 ;
2060 ; SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
2061 DSBINT: MOV     (SP),-(SP)
2062         MOV     @PRI07,2(SP)
2063         RTI
2064 ;
2065 ; SBTTL INTR - INTERRUPT HANDLERS
2066 BGNSRV INTR      ;DEFINE INTERRUPT ENTRY
2067 INTR::
2068         MOV     @1,INTRECV   ;SET FLAG TO SHOW INTERRUPT RECEIVED
2069         CLRB    INTFLAG      ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2070         BITB    @IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2071         BNE     1$           ;BR IF YES
2072         BISB    @IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2073 ;
2074 ; SAVE REGISTERS, MSG BUFFER, ETC.
2075 1$:
2076         ENDSRV
2077 L10026:
2078         RTI
  
```

<f>

```

2077 .SBTTL WAITF WAIT FOR SUBSYSTEM READY
2078 ;
2079 ; SUBROUTINE TO WAIT FOR THE SUBSYSTEM READY FLAG
2080 ;
2081 ; INPUTS:
2082 ;
2083 ; R5 ADDRESS OF FIRST DEVICE REGISTER
2084 ;
2085 ; OUTPUTS:
2086 ;
2087 ; R0 CONTENTS OF LAST TSSR READ
2088 ; CARRY SET - READY BIT SET
2089 ; CLR - TIMEOUT WAITING FOR READY
2090 ;
2091 016330 000401 WAITF:: BR 1# ;NOP WHEN SUPER FIXED
2092 016332 BREAK ; DO A SUPVSR BREAK FIRST.
016332 104422 TRAP C#BRK
2093 016334 012746 011000 1# : MOV #11000,-(SP) ;25-APRIL-83 REV B - 1100 MSEC TIMER
2094 016340 016500 000002 2# : MOV TSSR(R5),R0 ;READ THE TSSR REGISTER
2095 016344 105700 TSTB R0 ;TEST FOR READY BIT SET
2096
2097 016346 100420 BMI 3# ; EXIT ON STOP FLAG.
2098 016350 DELAY 1 ; WAIT 100 USEC
016350 012727 000001 MOV #1,(PC)+
016354 000000 .WORD 0
016356 013727 002116 MOV L#DLY,(PC)+
016362 000000 .WORD 0
016364 005367 177772 DEC -6(PC)
016370 001375 BNE .-4
016372 005367 177756 DEC -22(PC)
016376 001367 BNE .-20
2099 016400 005316 DEC (SP) ;REDUCE DELAY COUNT
2100 016402 001356 BNE 2# ;RETRY UNTIL TIMER EXPIRES
2101 016404 000241 CLC ; C = 0, CONTROLLER STILL RUNNING...
2102 016406 000401 BR 4# ;...OR HUNG-UP AFTER 300 MSEC.
2103 016410 000261 3# : SEC ; C = 1, CONTROLLER IS STOPPED.
2104 016412 005326 4# : DEC (SP)+ ;RESTORE STACK WITHOUT CHANGING CARRY BIT
2105 016414 000207 RTS PC
  
```

```

2107 .SBTTL  CHK TSSR - CHECK TSSR FOR READY
2108 ;*
2109 ; THIS ROUTINE WAITS FOR READY IN THE TSSR
2110 ; AND TESTS FOR AMBIGUOUS BIT SETTINGS IN TSSR.
2111 ;
2112 ; INPUT:
2113 ;       R5      ADDRESS OF CSR REGISTERS
2114 ;
2115 ; OUTPUT:
2116 ;       R0      CONTENTS OF TSSR
2117 ;       CARRY   SET - OKAY
2118 ;              CLR - NOT READY AMBIGUOUS, OR SC SET
2119 ;
2120 CHK TSSR:
2121 016416 JSR    PC, WAITF      ; WAIT FOR READY
2122 016422 BCC    20$          ; BRANCH IF TIME OUT
2123 016424 JSR    PC, CHKAMB ; TSSR AMBIGUOUS?
2124 016430 BCC    10$          ; BR IF YES
2125 016432 BIT    #SC, R0   ; SPECIAL CONDITION SET?
2126 016436 BEQ    15$          ; BR IF NO
2127 016440 BIT    #<SCE!BIE!RMR!NXM>, R0 ; ANY ERROR BITS SET?
2128 016444 BEQ    15$          ; BR IF NO
2129 016446 10$: CLC          ; SET FAILURE
2130 016450 BR     20$          ;
2131 016452 15$: SEC          ; SET SUCCESS
2132 016454 20$: RTS    PC    ; RETURN TO CALLER
2133 .SBTTL  XNXM - CHECK FOR NONEXISTENT MEMORY
2134 ;*
2135 ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2136 ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
2137 ; "C" = 0, ALL ADDRESSES OK.
2138 ;
2139 ; CALL:  MOV  ADR1, R1
2140 ;        MOV  ADR2, R2
2141 ;        JSR  PC, NXM
2142 ;        RETURN ; TEST "C" AND PROCEED.
2143 016456 XNXM: MOV    #2$, #4   ; SET BUSERR VECTOR.
2144 016464 MOV    #PRI04, #6
2145 016472 CLR    R3           ; FLAG.
2146 016474 1$: TST    (R1)    ; TEST THE ADDRESS(ES).
2147 ; IF ANY TRAP, CONTINUE AT 2$.
2148 016476 CMP    R1, R2    ; OTHERWISE, CONTINUE HERE.
2149 016500 BEQ    3$           ; BR IF FINISHED (NO NEXM'S).
2150 016502 ADD    #2, R1    ; SET NEXT ADDRESS...
2151 016506 BR     1$           ; ...AND CONTINUE.
2152 016510 2$: COM    R3     ; GOT ONE, SET FLAG...
2153 016512 MOV    #3$, (SP)
2154 016516 RTI          ; ...AND DISMISS INTERRUPT...
2155 016520 3$: CLRVEC #4    ; ...AND GIVE BACK THE VECTOR.
2156 016526 MOV    #4, R0
2157 016530 TRAP  C#CVEC
2158 016532 TST    R3           ; DID WE CATCH ONE ??
2159 016534 BEQ    .+4        ; NO, "C" = 0, SKIP NEXT.
2159 016534 SEC          ; YES, "C" = 1, (R1) = NEXM ADDR.
2159 016534 RTS    PC

```

```

2161          .SBTTL TSTLOOP CHECK ITERATION COUNT
2162          ;*
2163          ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
2164          ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
2165          ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
2166          ;
2167          ; CALL: LOOPTO ARG
2168          ;
2169          TSTLOOP::
2170          016536 005737 002162      TST      NOITS      ; ITERATIONS INHIBITED?
2171          016542 001006              BNE      1$          ; YES.
2172          016544 005737 002176      TST      QVP          ; NO.
2173          016550 100403              BMI      1$          ; LOOPS DISALLOWED IN QUICK PASS.
2174          016552 005337 002210      DEC      LOOPCNT     ; BUMP LOOP COUNTER.
2175          016556 001002              BNE      2$
2176          016560 000241      1$:    CLC              ; LOOP DISALLOWED, OR DONE.
2177          016562 000401              BR       3$
2178          016564 000261      2$:    SEC              ; LOOP ENABLED.
2179          016566 000207      3$:    RTS      PC
2180
2181          .SBTTL TSTSETUP - PRINT TEST NAME AND INIT ERROR COUNTS
2182          ;*
2183          ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
2184          ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
2185          ; IN THE CURRENT RUN SEQUENCE.
2186          ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
2187          ;
2188          ; INPUT:
2189          ;
2190          ;      RO      POINTER TO TEST ID ASCIZ STRING
2191          ;
2192          ; OUTPUT:
2193          ;
2194          ;      RS      ADDRESS OF FIRST DEVICE REGISTER
2195          ;
2196          ; IMPLICIT OUTPUTS:
2197          ;
2198          ;      TSTCNT  UPDATED TO COUNT TESTS PERFORMED SINCE START OR RESTART
2199          ;
2200          ; SIDE EFFECTS:
2201          ;
2202          ;      INTERRUPT LEVEL IS RASIED TO LEVEL OF
2203          ;      THE DEVICE UNDER TEST
2204          ;
2205          ;
2206          ;
2207          TSTSETUP::
2208          016570 010046      MOV      RO, (SP)      ; SAVE THE TEST ID MESSAGE
2209          016572 005037 003146      CLR      SIFLAG      ; CLEAR "SOFT INIT" FLAG
2210          016576 005037 017036      CLR      ERRK        ; CLEAR LOCAL ERROR COUNTER
2211          016602 005037 005766      CLR      EXTA        ; CLEAR ERROR EXTENSION FLAG
2212          016606 105037 016224      CLR      INTMASK     ; CLEAR INTERRUPT MASK (CHECK ERROR)
2213          016612 013700 002174      MOV      UNITN,RO    ; GET THE UNIT NUMBER.
2214          016616 006300              ASL      RO           ; ... AND MAKE IT A WORD OFFSET.
2215          016620 005737 003106      TST      NODEV       ; DID STARTUP FIND THE DEVICE?
2216          016624 001430              BEQ     4$           ; BR IF YES
2217          016626 100010              BPL     3$           ; BR IF NOT IDLE

```

```

2218 016630 052760 160000 003170      BIS      #160000,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2219 016636      ERRDF    1,NXR,NXRERR ; NO DEVICE HERE PRINT IT
      016636 104455      TRAP     C#ERRDF
      016640 000001      .WORD    1
      016642 003734      .WORD    NXR
      016644 005732      .WORD    NXRERR
2220 016646 000407      BR       2$
2221 016650 052760 160001 003170 3$:  BIS      #160001,ERTABL(RO) ; FLAG ERROR IN THE ERROR TABLE
2222 016656      ERRDF    2,NOINIT ; DEVICE NOT IDLE
      016656 104455      TRAP     C#ERRDF
      016660 000002      .WORD    2
      016662 004331      .WORD    NOINIT
      016664 000000      .WORD    0
2223 016666 012737 177777 003104 2$:  MOV      #1,DUFLG ; DROP THE UNIT
2224 016674      DDDU     UNITN
      016674 013700 002174      MOV      UNITN,RO
      016700 104451      TRAP     C#DDDU
2225 016702      DDCLN    ; ABORT THE PASS
      016702 104444      TRAP     C#DCLN
2226 016704 000423      BR       5$
2227
2228 016706      RFLAGS   RO ; GET THE OPERATOR FLAGS.
      016706 104421      TRAP     C#RFLA
2229 016710 032700 001000      BIT      #PNT,RO ; PRINT THE TEST NUMBERS?
2230 016714 001412      BEQ     1$ ; BR IF NO
2231 016716 011600      MOV      (SP),RO ; GET THE ID MESSAGE
2232 016720      PRINTF  #TNAM,RO ; DISPLAY THE TEST ID
      016720 010046      MOV      RO,-(SP)
      016722 012746 016764      MOV      #TNAM,-(SP)
      016726 012746 000002      MOV      #2,-(SP)
      016732 010600      MOV      SP,RO
      016734 104417      TRAP     C#PNTF
      016736 062706 000006      ADD     #6,SP
2233 016742 005237 002206      1$: INC     TSTCNT ; BUMP TEST COUNTER.
2234 016746      SETPRI  IPRI ; PRIORITY THAT OF DEVICE
      016746 013700 002204      MOV      IPRI,RO
      016752 104441      TRAP     C#SPRI
2235 016754 005726      5$: TST     (SP)+ ; FIX UP THE STACK
2236 016756 013705 002200      MOV      CSRADDR,R5 ; ADDRESS OF TSV REGISTERS ON UNIBUS
2237 016762 000207      RTS     PC
2238 016764 045 123 045 TNAM: .ASCIZ '#S#T#A Test'
2239      .EVEN
2240      .SBTTL TSTEND - PRINT ERRORS RECEIVED
2241      ;
2242      ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
2243      ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
2244      ;
2245      TSTEND: RFLAGS   RO
      017000      TRAP     C#RFLA
2246 017002 030027 020000      BIT      RO,#IER
2247 017006 001412      BEQ     1$ ; BR IF "IER" NOT SET.
2248 017010      PRINTF  #ESUM,ERRK ; PRINT ERROR COUNT.
      017010 013746 017036      MOV      ERRK,-(SP)
      017014 012746 017040      MOV      #ESUM,-(SP)
      017020 012746 000002      MOV      #2,-(SP)
      017024 010600      MOV      SP,RO
      017026 104417      TRAP     C#PNTF

```


TSV3 . GLOBAL AREAS MACRO M1113 14-JUN-84 16:41
TSTEND - PRINT ERRORS RECEIVED

SEQ 0079

2249	017030	062706	000006			ADD	#6.SP
2250	017034	000207			18:	RTS	PC
2251	017036	000000				ERRK:	0 ; LOCAL ERROR COUNT.
2252	017040	045	101	040		ESUM:	.ASCIZ /#A #D#A ERRORS/
2253	017057	105	122	122		EMAXDU:	.ASCIZ /ERROR LIMIT REACHED -- DROPPING UNIT/
2254							.EVEN

```

2256                                     .SBTTL INCERK INCREMENT LOCAL ERROR COUNT
2257                                     ;
2258                                     ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
2259                                     ;
2260 017124 005237 017036 INCERK: INC ERRK ; INCREMENT LOCAL ERROR COUNT
2261 017130 010046 MOV RO,-(SP) ; SAVE RO
2262 017132 013700 002174 MOV UNITN,RO ; GET UNIT NUMBER,
2263 017136 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET.
2264 017140 062700 003170 ADD @ERTABL,RO ; RO GETS ADDRESS OF ERROR TABLE ENTRY.
2265 017144 005210 INC (RO) ; INCREMENT THE DEVICE ERROR COUNT
2266 017146 032710 007777 BIT @7777,(RO) ; DID WE OVERFLOW THE FIELD?
2267 017152 001001 BNE 1$ ; BR IF NO.
2268 017154 005310 DEC (RO) ; YES -- BACK IT UP TO 7777.
2269 017156 012600 1$: MOV (SP)+,RO ; RESTORE RO
2270 017160 000207 RTS PC ; RETURN TO CALLER.
2271
2272 017162 010046 CKEMAX: MOV RO,-(SP) ; SAVE RO
2273 017164 013700 002174 MOV UNITN,RO ; GET UNIT NUMBER
2274 017170 006300 ASL RO ; ... AND MAKE IT A WORD OFFSET
2275 017172 016000 003170 MOV ERTABL(RO),RO ; GET ERROR TABLE ENTRY
2276 017176 042700 170000 BIC @170000,RO ; EXTRACT ERROR COUNT FIELD
2277 017202 020037 002166 CMP RO,GERRMAX ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
2278 017206 103004 BHIS 1$ ; BR IF YES
2279 017210 023737 017036 002164 CMP ERRK,LERRMAX ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
2280 017216 103417 BLO 2$ ; BR IF NO
2281 017220 1$: RFLAGS RO ; GET OPERATOR FLAGS
2282 017222 032700 000040 TRAP C@RFLA ; IS DROPPING INHIBITED?
2283 017226 001013 BNE 2$ ; BR IF YES.
2284 017230 012737 177777 003104 MOV @-1,DUFLG ; NO -- DROP THE UNIT
2285 017236 104455 ERDF 4,EMAXDU
2286 017246 013700 002174 TRAP C@ERDF
2287 017254 104444 .WORD 4
2288 017256 012600 .WORD EMAXDU
2289 017260 000207 .WORD 0
DODU UNITN
MOV UNITN,RO
TRAP C@DODU
DOCLN
TRAP C@DCLN
2$: MOV (SP)+,RO ; RESTORE RO
RTS PC ; RETURN TO CALLER

```

```

2291          .SBTTL CKDROP CHECK IF UNIT SHOULD BE DROPPED
2292          ;
2293          ; CHECK IF UNIT SHOULD BE DROPPED
2294          ;
2295 017262 010046 CKDROP: MOV RO, -(SP)
2296 017264          FORCERROR 1$,NOTSSR
2297 017274          RFLAGS RO
2298 017276 104421 TRAP C$RFLA
2299 017302 001010 BIT @IDU,RO
2300 017304 011600 BNE 1$
2301 017306 012737 177777 003104 MOV (SP),RO
2302 017314          MOV @-1,DUFLG
2303 017322 013700 002174 DODU UNITN
2304 017324 104444 MOV UNITN,RO
2305 017326 000207 TRAP C$DODU
2306          DOCLN          ;ABORT THE PASS
2307          TRAP C$DCLN
2308          MOV (SP)+,RO
2309          RTS PC
2310          .SBTTL CONFIG - DETERMINE CONFIGURATION OF SYSTEM
2311          ;
2312          ; SUBROUTINE - DETERMINE CONFIGURATION OF TSV05 SYSTEM.
2313          ;
2314          ; CONFIG:
2315          JSR PC,SOFINIT
2316          RTS PC
2317          .SBTTL KTON,KTOFF - ENABLE/DISABLE MEMORY MANAGEMENT
2318          ;
2319          ; SUBROUTINE - ENABLE MEM MGT.
2320          ;
2321          ; KTON: TST KTF LG ; GOT KT?
2322          BEQ 1$ ; NO.
2323          MOV @1,SRO ; YES. ENABLE K'11.
2324          1$: RTS PC
2325          ;
2326          ; SUBROUTINE - DISABLE MEM MGT.
2327          ;
2328          ; KTOFF: TST KTF LG ; GOT KT11?
2329          BEQ 1$ ; NO.
2330          NOP
2331          NOP
2332          MOV @0,SRO ; DISABLE KT.
2333          1$: RTS PC

```

```

2334 .SBTTL SETMAP SETUP PAR6 MAPPING
2335
2336 ;*
2337 ;
2338 ; THIS ROUTINE SETS UP KERNEL PAR6 TP HANDLE
2339 ; AN 18 BIT ADDRESS. THE OFFSET INTO THE PAGE
2340 ; IS RETURNED BIASED TO PAR6.
2341 ;
2342 ; INPUTS:
2343 ;
2344 ; R0 HIGH ORDER ADDRESS BITS
2345 ; R1 LOW ORDER ADDRESS BITS
2346 ;
2347 ; OUTPUTS:
2348 ;
2349 ; R0 OFFSET INTO BLOCK WITH PAR6 BIAS (I.E. THE ADDRESS)
2350 ; CARRY SET IF SUCCESS
2351 ; CLR IF ERROR
2352 ;
2353 SETMAP:
2354 SAVREG ;SAVE R1-R4 UNTIL NEXT RETURN
2355 TST KTF LG ;SYSTEM HAVE ABOVE 28K?
2356 BEQ 10$ ;BR IF NO
2357 MOV R1,R2 ;SAVE LOW ORDER BITS
2358 .REPT 6
2359 ASR R0 ;CONVERT WORD ADDRESS TO 32W BLOCKS
2360 ROR R1 ;MAKE IT DOUBLE PRECISION
2361 .ENDR
2362 BIC #177,R1 ;ALINE FOR LOWER 4K BOUNDARY
2363 CMP R1,KTF LG ;HIGHER THAN EXISTING MEMORY?
2364 BHIS 10$ ;BR IF YES
2365 MOV R1,#KIPAR6 ;SETUP MAPPING REGISTER PAR6
2366 BIC #160000,R2 ;SETUP DISPLACEMENT IN PAGE
2367 ADD #140000,R2 ;ADD IN PAR6 BIAS
2368 MOV R2,R0 ;RETURN IN R0
2369 SEC ;SET SUCCESS
2370 BR 15$
2371 10$: CLC ;SET FAILURE
2372 15$: RTS PC ;RETURN
2373 .SBTTL FILLMEM - FILL MEMORY WITH BACKGROUND PATTERN
2374 ;*
2375 ; FILL MEMORY WITH A BACKGROUND PATTERN
2376 ;
2377 ; INPUTS:
2378 ;
2379 ; R0 = BACKGROUND PATTERN
2380 ; FREE = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2381 ; KTF LG = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2382 ;
2383 ; OUTPUTS:
2384 ;
2385 ; NONE
2386 ;
2387 ;
2388 FILLMEM:
2389 SAVREG ;SAVE R1-R5 UNTIL NEXT RETURN
2390 JSR PC,KTOFF ;DISABLE KT.
    
```

2391	017512	010003				MOV	R0,R3	;COPY TEST PATTERN
2392	017514	013701	003116			MOV	FREE,R1	;GET FIRST FREE LOCATION
2393	017520	013702	003120			MOV	FRESIZ,R2	;SIZE OF FREE SPACE BELOW 28K.
2394	017524	010321		10#:		MOV	R3,(R1)+	;STORE A BACKGROUND WORD
2395	017526	005302				DEC	R2	;DONE ALL MEMORY IN FREE SPACE?
2396	017530	003375				BGT	10#	;BR IF NO
2397	017532	005737	003124			TST	KTFLG	; GOT KT?
2398	017536	001477				BEQ	55#	; NO. GET OUT.
2399	017540	004737	017336			JSR	PC,KTON	; YES. ENABLE KT.
2400	017544	005000				CLR	R0	;HIGH ORDER ADDRESS START
2401	017546	013701	003144			MOV	PST32W,R1	;GET >28K START ADDRESS (IN 32W BLOCKS)
2402		000006				.REPT	6	
2403						CLC		;CLEAR C BIT
2404						ROL	R1	;CONVERT BLOCKS TO WORDS
2405						ROL	R0	;MAKE IT DOUBLE PRECISION
2406						.ENDR		
2407	017616	004737	017376			JSR	PC,SETMAP	;SETUP PAR6 MAPPING REGISTER
2408	017622	010320		30#:		MOV	R3,(R0)+	;STORE TEST PATTERN IN >28K ADDRESS
2409	017624	020027	160000			MOV	R0,#160000	;END OF PAR6 MAPPING AREA?
2410	017630	103774				BLO	30#	;BR IF NO
2411	017632	162700	020000			SUB	#20000,R0	;BACKUP INTO PAR6 MAPPING BEGIN
2412	017636	062737	000200	172354		ADD	#200,#KIPAR6	;POINT TO NEXT 4K BLOCK >28K.
2413	017644	023737	172354	003124		CMP	#KIPAR6,KTFLG	;END OF MEMORY?
2414	017652	001427				BEQ	50#	;BR IF YES
2415	017654	005737	003136			TST	T23A	;11/23A?
2416	017660	001407				BEQ	35#	;NO KEEP GOING
2417	017662	013704	177572			MOV	SRO,R4	;GET SRO CONTENTS
2418	017666	042704	177761			BIC	#177761,R4	;CLEAR ALL BUT PAGE NUMBER
2419	017672	022704	000016			CMP	#16,R4	;SEE IF PAGE 7
2420	017676	001415				BEQ	50#	;EXIT IF THERE
2421	017700	005737	003140		35#:	TST	T23B	;11/23B?
2422	017704	001410				BEQ	45#	;NO KEEP GOING
2423	017706	023727	172354	007600		CMP	#KIPAR6,#7600	;REACHED 18 BITS?
2424	017714	103001				BHIS	40#	;YES
2425	017716	000403				BR	45#	;NO KEEP GOING
2426	017720	012737	000020	172516	40#:	MOV	#20,SR3	;SET 22 BIT RELOCATION
2427	017726	000137	017622		45#:	JMP	30#	;KEEP GOING ON ETC.
2428	017732	004737	017354		50#:	JSR	PC,KTOFF	; DISABLE KT.
2429	017736	000207			55#:	RTS	PC	

```

2431          .SBTTL CMPMEM - COMPARE MEMORY TO BACKGROUND PATTERN
2432          ;
2433          ; COMPARE MEMORY WITH A BACKGROUND PATTERN
2434          ;
2435          ; INPUTS:
2436          ;
2437          ;     RO = BACKGROUND PATTERN
2438          ;     FREE  = FIRST LOCATION AVAILABLE TO DIAGNOSTIC
2439          ;     KTFLG  = SET TO HIGHEST MEMORY LOCATION IF > 28K.
2440          ;
2441          ; OUTPUTS:
2442          ;
2443          ;     CARRY  - SET IF NO ERROR
2444          ;     CARRY  - CLR IF ERROR
2445          ;
2446          ; IMPLICIT OUTPUTS:
2447          ;
2448          ;     ERRHI  - ERROR HIGH ADDRESS
2449          ;     ERRLO  - ERROR LOW ADDRESS
2450          ;     EXPD   - EXPECTED DATA
2451          ;     RECV   - RECEIVED DATA
2452          ;
2453          ; CMPMEM:
2454          ; SAVREG          ;SAVE R1-R5 UNTIL NEXT RETURN
2455          ; MOV R0,R3      ;COPY TEST PATTERN
2456          ; JSR PC,KTOFF   ;DISABLE KT.
2457          ; MOV FREE,R1    ;GET FIRST FREE LOCATION
2458          ; MOV FRESIZ,R2  ;SIZE OF FREE SPACE BELOW 28K.
2459          ; CMP R3,(R1)    ;FREE SPACE LOCATION EQUAL TO EXPD?
2460          ; BEQ 15$        ;BR IF YES
2461          ; MOV R1,ERRLO   ;SAVE ADDRESS IN ERROR
2462          ; CLR ERRHI      ;NO HIGH ADDRESS
2463          ; MOV R3,EXPD    ;SAVE EXPD FOR ERROR REPORT
2464          ; MOV (R1),RECV  ;SAVE RECV FOR ERROR REPORT
2465          ; BR 50$         ;
2466          ; 15$: TST (R1)  ;POINT TO NEXT ADDRESS
2467          ; DEC R2         ;DONE ALL MEMORY IN FREE SPACE?
2468          ; BGT 10$       ;BR IF NO
2469          ; TST KTFLG     ; GOT KT?
2470          ; BEQ 55$       ; NO. GET OUT.
2471          ; JSR PC,KTON   ; YES. ENABLE KT.
2472          ; CLR R0        ;HIGH ORDER ADDRESS START
2473          ; MOV PST32W,R1 ;GET >28K START ADDRESS (IN 32W BLOCKS)
2474          ; .REPT 6
2475          ; ROL R1        ;CONVERT BLOCKS TO WORDS
2476          ; ROL R0        ;MAKE IT DOUBLE PRECISION
2477          ; .ENDR
2478          ; BIC #177,R1   ;ALINE 4K BOUNDARY
2479          ; MOV R0,-(SP)   ;SAVE HIGH ORDER
2480          ; MOV R1,-(SP)   ;SAVE LOW ORDER
2481          ; JSR PC,SETMAP  ;SETUP PAR6 MAPPING REGISTER
2482          ; MOV R0,R4      ;COPY ADDRESS BIASED TO PAR6
2483          ; MOV (SP)+,R1   ;RESTORE LOW ORDER IN NON PAR6 FORMAT
2484          ; MOV (SP)+,R0   ;RESTORE HIGH ORDER IN NON PAR6 FORMAT
2485          ; 30$: CMP R3,(R4) ;ABOVE 28K LOCATION EQUAL EXPD?
2486          ; BEQ 32$       ;BR IF YES
2487          ; MOV R0,ERRHI  ;SAVE HIGH ORDER IN EPROR

```

```

2488 020120 010137 002232      MOV     R1,ERRLO      ;SAVE LOW ORDER IN ERROR
2489 020124 010337 002224      MOV     R3,EXPD      ;SAVE EXPD FOR ERROR REPORT
2490 020130 011437 002226      MOV     (R4),RECV    ;SAVE RECV FOR ERROR REPORT
2491 020134 000421              BR      50$          ;
2492 020136 062701 000002      32$:   ADD     @2,R1      ;UPDATE NON PAR6 ADDRESS
2493 020142 005500              ADC     R0           ;MAKE IT DOUBLE PRECISION ADD
2494 020144 062704 000002      ADD     @2,R4        ;UPDATE PAR FORMAT ADDRESS
2495 020150 020427 160000      CMP     R4,@160000   ;END OF PAR6 MAPPING AREA?
2496 020154 103755              BLO    30$          ;BR IF NO
2497 020156 162704 020000      SUB     @20000,R4    ;BACKUP INTO PAR6 MAPPING BEGIN
2498 020162 062737 000200 172354  ADD     @200,@*KIPAR6 ;POINT TO NEXT 4K BLOCK >28K.
2499 020170 023737 172354 003124  CMP     @*KIPAR6,KTFLG ;END OF MEMORY?
2500 020176 101744              BLOS   30$          ;BR IF NO
2501 020200 004737 017354      50$:   JSR     PC,KTOFF   ;TURN OFF MEMORY MAPPING
2502 020204 000241              CLC                    ;SET FAILURE
2503 020206 000403              BR      60$          ;
2504 020210 004737 017354      55$:   JSR     PC,KTOFF   ;TURN OFF MEMORY MAPPING
2505 020214 000261              SEC                    ;SET SUCCESS
2506 020216 000207      60$:   RTS     PC
2507              .SBTTL REGSAV - SAVE R1-R5 ON STACK
2508              ;*
2509              ;
2510              ;ROUTINE TO
2511              ;SAVE R1 THROUGH R5 ON THE STACK
2512              ;
2513              ;CALLING SEQUENCE:
2514              ;
2515              ;      JSR     R5,REGSAV
2516              ;
2517              ;THIS IS A COOROUTINE WHICH TRANSFER CONTROL BACK TO
2518              ;THE CALLING ROUTINE. AT THE END OF THE CALLING ROUTINE,
2519              ;THE RTS PC RETURNS CONTROL TO THIS ROUTINE TO RESTORE
2520              ;REGISTERS.
2521              ;
2522              ;THIS ROUTINE SHOULD ONLY BE CALLED FROM ROUTINES WHICH ARE
2523              ;CALLED VIA A JSR PC INSTRUCTION
2524              ;
2525              ;-
2526
2527 020220      REGSAV:
2528 020220 010446      MOV     R4,-(SP)
2529 020222 010346      MOV     R3,-(SP)
2530 020224 010246      MOV     R2,-(SP)
2531 020226 010146      MOV     R1,(SP)
2532 020230 010546      MOV     R5,-(SP)
2533 020232 016605 000012      MOV     10.(SP),R5
2534 020236 004736      JSR     PC,@(SP)+
2535 020240 012601      MOV     (SP)+,R1
2536 020242 012602      MOV     (SP)+,R2
2537 020244 012603      MOV     (SP)+,R3
2538 020246 012604      MOV     (SP)+,R4
2539 020250 012605      MOV     (SP)+,R5
2540 020252 000207      RTS     PC
  
```

```

2542 .SBTTL GETPAT GET 8 BIT PATTERN FROM OPERATOR
2543 ;*
2544 ;ROUTINE TO REQUEST AN 8 BIT DATA PATTERN FROM THE OPERATOR
2545 ;
2546 ;INPUTS: NONE.
2547 ;
2548 ;OUTPUTS:
2549 ; RO OCTAL NUMBER FROM THE OPERATOR
2550 ;
2551 ;CALLING SEQUENCE:
2552 ; JSR PC,GETPAT
2553 ;-
2554 GETPAT::
2555 1$: SAVREG ;SAVE THE GENERAL REGISTERS
2556 1$: GMANID DATASC,PATDAT,0,377,0,377,NO
    TRAP C$GMAN
    BR 10000$
    .WORD PATDAT
    .WORD T$CODE
    .WORD DATASC
    .WORD 377
    .WORD T$LOLIM
    .WORD T$HILIM
2557 10000$: BNCOMPLETE 1$ ;RETRY IF ERROR
    BCC 1$
2558 MOV PATDAT,RO ;DATA PATTERN FROM OPERATOR
2559 RTS PC ;RETURN TO CALLER
2560
2561 ;*
2562 ;LOCAL DATA AREA
2563 ;-
2564
2565 PATDAT: .WORD 0 ;TEMPORARY STORAGE FOR DATA
2566 DATASC: .ASCIZ 'ENTER DATA PATTERN'
2567 .EVEN
  
```



```

2569 .SBTTL GETSEL - ISSUE MENU AND GET OPERATOR RESPONSE
2570 ;*
2571 ;ROUTINE TO ISSUE A MENU AND GET THE OPERATOR'S RESPONSE.
2572 ;
2573 ;INPUTS:
2574 ; R0 ADDRESS OF ASCIZ STRING OF MENU
2575 ; R1 MAXIMUM ALLOWABLE OPERATOR RESPONSE
2576 ;
2577 ;OUTPUTS:
2578 ; R0 NUMBER OF THE OPERATOR'S SELECTION
2579 ;
2580 GETSEL::
2581 SAVREG ;SAVE GENERAL REGISTERS
2582 MOV R0,R2 ;SAVE THE MENU ADDRESS
2583 MOV R2,R3 ;START OF MENU STRING
2584 TST (R3) ;END OF ASCII ?
2585 BEQ 3$ ;BRANCH IF ALL LINES DISPLAYED
2586 PRINTF #SELASC,(R3)+ ;DISPLAY THE MENU
      MOV (R3)+, -(SP)
      MOV #SELASC, -(SP)
      MOV #2, -(SP)
      MOV SP,R0
      TRAP C#PNTF
      ADD #6, SP
      BR 2$
2587 3$: GMANID MENASC,MENRES,D,-1,0,-1,NO
      TRAP C#GMAN
      BR 10001$
      .WORD MENRES
      .WORD T#CODE
      .WORD MENASC
      .WORD -1
      .WORD T#LOLIM
      .WORD T#HILIM
2588 10001$: BNCOMPLETE 1$ ;RETRY IF ERROR
      BCC 1$
2589 MOV MENRES,R0 ;GET THE OPERATOR'S REPLY
2590 CMP R0,R1 ;COMPARE TO MAXIMUM ALLOWED
2591 BLOS 5$ ;BRANCH IF OK
2592 PRINTF #MENERR ;DISPLAY ERROR MESSAGE
2593 MOV #MENERR, -(SP)
      MOV #1, -(SP)
      MOV SP,R0
      TRAP C#PNTF
      ADD #4, SP
      BR 1$ ;RETRY
2594 5$: RTS PC ;RETURN TO CALLER
2595 MENERR: .ASCIZ '#N#A *** Menu Selection Too Large ***'
2596 SELASC: .ASCIZ '#N#T'
2597 MENASC: .ASCIZ 'Enter Menu Selection: '
2598 .EVEN
2599 MENRES: .WORD 0
2600

```

```

2602 .SBTTL CHKMAN CHECK MANUAL INTERVENTION LEGALITY
2603 ;*
2604 ;
2605 ;ROUTINE TO TEST FOR MANUAL INTERVENTION LEGALITY.
2606 ;
2607 ;INPUT:
2608 ;
2609 ; NONE.
2610 ;
2611 ;OUTPUT:
2612 ;
2613 ; CARRY 0 MANUAL INTERVENTION NOT ALLOWED
2614 ; 1 MANUAL INTERVENTION IS OK
2615 ;
2616 ;SIDE EFFECTS:
2617 ;
2618 ; A MESSAGE IS DISPLAYED WARNING THAT TEST IS
2619 ; NOT EXECUTED IF MANUAL INTERVENTION IS NOT
2620 ; ALLOWED.
2621 ;
2622 ;
2623 ;
2624 020560 CHKMAN::
2625 020560 SAVREG ;SAVE THE REGISTERS
2626 020564 104450 MANUAL ;SEE IF MANUAL INTERVENTION OK
2627 020566 103411 TRAP C#MANI
2628 020570 012746 020614 BCOMPLETE 1# ;BRANCH IF ALLOWED
2629 020574 012746 000001 BCS 1#
2630 020600 010600 PRINTF #NOMAN ;PRINT THE WARNING MESSAGE
2631 020602 104417 MOV #NOMAN, -(SP)
2632 020604 062706 000004 MOV #1, -(SP)
2633 020610 000241 TRAP C#PNTF
2634 020612 000207 ADD #4, SP
2635 020614 045 116 045 NOMAN: .ASCIZ 'N/A *** Manual Intervention not Allowed - Test Aborted ***'
2636 .even
2637 CLC ;CLEAR CARRY FOR ERROR
2638 RTS PC ;RETURN

```

```

2635          .SBTTL  ENVIRN  - SETUP FREE DIAGNOSTIC SPACE
2636          ;
2637          ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
2638          ;
2639 020710     ENVIRN: MEMORY  R0
                TRAP      C$MEM
2640 020710     104431          MOV      R0,FREE      ; GET 1ST FREE ADDRESS...
2641 020712     010037 003116  ADD      #2,FREE
2642 020716     062737 000002 003116  MOV      (R0),FRESIZ ; ...AND WORD COUNT.
2643 020724     011037 003120          SUB      #4,FRESIZ
2644 020730     162737 000004 003120  MOV      L$UNIT,R2   ; GET NUMBER OF UNITS
2645 020736     013702 002012          SUB      #7,FRESIZ   ; TAKE AWAY 7 WORDS PFR UNIT
2646 020742     162737 000007 003120 10$: DEC      R2
2647 020750     005302          BNE     10$
2648 020752     001373          MOV      FREE,R0      ;GET FIRST FREE ADDRESS
2649 020754     013700 003116  MOV      FRESIZ,R0   ;POINT TO LAST FREE ADDRESS
2650 020760     063700 003120  ADD      #2,R0
2651 020764     162700 000002          SUB      #2,R0      ;BACKUP 1 WORD
2652 020770     010037 003122  MOV      R0,FREEHI   ;STORE LAST FREE ADDRESS
2653 020774     000240          NOP
2654 020776     012701 177520  MOV      #BDVPCR,R1  ;*****
2655 021002     010102          MOV      R1,R2      ;GET BDV11 PCR ADDRESS
2656 021004     062702 000002          ADD      #2,R2      ;COPY TO R2
2657 021010     004737 016456  JSR     PC,XNXM     ;SET THE RANGE
2658 021014     103001          BCC     15$        ;SEE IF WE HAVE ONE
2659 021016     000423          BR      40$        ;OK TO SET FLAGS
2660 021020     013701 177520 15$: MOV      BDVPCR,R1  ;RETURN WITH FLAGS CLEAR
2661 021024     062701 000001          ADD      #1,R1      ;SAVE PCR CONTENTS
2662 021030     012702 177520  MOV      #BDVPCR,R2  ;ADD ONE TO IT
2663 021034     005212          INC     (R2)        ;GET BDV11 PCR ADDRESS
2664 021036     013703 177520  MOV      BDVPCR,R3  ;TRY TO WRITE TO IT
2665 021042     020103          CMP     R1,R3      ;GET RESULTS
2666 021044     001006          BNE     20$        ;DID IT CHANGE?
2667 021046     005237 003136          INC     T23A       ;NO, MUST BE 11/238
2668 021052     042737 170000 002120 BIC     #170000,L$HIME ;SET THE FLAG
2669          ; NOP      ;SUPERVISOR COULD BE WRONG
2670 021060     000402          PRINTF #M8186     ;BR 40$ FOR RELEASE
2671 021062     005237 003140          BR      40$        ;TELL THE SYSTEM TYPE
2672          ; NOP      ;RETURN
2673          ; PRINTF #M8189 ;SET THE FLAG
2674 021066     4C$:          ;BR 40$ FOR RELEASE
2675 021066     000207          RTS     PC        ;TELL THE SYSTEM TYPE
                ; RETURN

```

TSV3 - GLOBAL AREAS MACRO M1113 14 JUN-84 16:41
 KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS

SEQ 0090

```

2677          .SBTTL  KTINIT  SETUP KT11 MEMORY MANAGEMENT REGISTERS
2678          ;*
2679          ;ROUTINE TO INIT KT 11
2680          ;
2681          ;
2682          ;
2683          ;
2684          ;KTINIT:
2685          ;CLP      KTFLG      ; INIT >28K MEMORY FLAG
2686          ;CLR      KTENABLE   ; INIT TEST >28K FLAG
2687          ;CMP      L#HIME,#1577 ; GOT ENOUGH MEMORY (>28K)?
2688          ;BLOS    9#          ; NO.
2689          ;MOV      @#ERRVEC,RO  ; SAVE OLD ERR VEC PTR.
2690          ;MOV      @2#,@#ERRVEC ; SET ERR VEC PTR.
2691          ;TST     @#SRO        ; GOT KT11?
2692          ;NOP                     ; (TRAP IF NO).
2693          ;MOV      L#HIME,KTFLG ; YES. SET KT FLAG.
2694          ;BIC     @177,KTFLG   ;
2695          ;MOV      RO,@#ERRVEC  ; RESTORE OLD ERR VEC PTR.
2696          ;CLR      RO          ; RO = AR DATA.
2697          ;MOV      @KIPAR0,R1   ; R1 = KI REGS PTR.
2698          ;MOV      @77406,-40(R1) ; SET DESCRIPTOR REG.
2699          ;MOV      RO,(R1)+    ; SET KIPAR REG.
2700          ;ADD     @200,RO       ; BUMP AR DATA BY "4K".
2701          ;CMP     RO,@2000     ; AT "I/O"?
2702          ;BNE    1#           ; NO.
2703          ;MOV     @177600,-(R1) ; YES. SET KTPAR7 FOR I/O.
2704          ;BR     9#
2705          ;
2706          ;2#:  MOV     @6#,(SP)  ; SET UP RETURN
2707          ;RTI                     ; RTI TO NEXT LOCATION
2708          ;
2709          ;6#:  MOV     RO,@#ERRVEC ; RESTORE OLD ERR VEC PTR.
2710          ;
2711          ;9#:  RTS     PC

```

```

2713 ;*
2714 ; SUBROUTINE TO SET EXTENDED FEATURES SWITCH
2715 ;
2716 ; Requires that SOFINIT and WATCHR have been done previous to call.
2717 ;
2718 ;
2719 ; INPUTS:
2720 ; R5 CURRENT UNIT NUMBER
2721 ; OUTPUTS:
2722 ; The Extended Features Switch is set.
2723 ;
2724 ;-
2725
2726 021222 INVERT::
2727
2728 021222 005737 002220 TST EXTFEA ; IS SWITCH SET?
2729 021226 001020 BNE 1$ ; YES,EXIT STAGE RIGHT!(or the next one outa town!)
2730 021230 012737 100206 021274 MOV #100206,CMDPKT ; WRT SUB-SYS MEM CMD
2731 021236 012737 021304 021276 MOV #WSMBK,CMDPKT+2 ; MSG BUF ADDR
2732 021244 012737 000006 021302 MOV #6,CMDPKT+6 ; BYTE COUNT
2733 021252 012737 100010 021304 MOV #100010,WSMBK ; INVERT THE SWITCH
2734 021260 012704 021274 MOV #CMDPKT,R4 ; SET CMDPKT INTO R4
2735 021264 004737 010742 JSR PC,WATCHR ; DO IT
2736 021270 000207 1$: RTS PC ; RETURN
2737
2738 ; COMMAND PACKET.
2739
2740 021274 = <..+3>E177774 ;MUST BE ON MOD 4 BOUNDRY.
2741
2742 021274 000000 CMDPKT:: 0 ;1ST WORD IS TS05 COMMAND.
2743 021276 000000 0 ;2ND WORD IS THE BUFFER LOW ADDRESS.
2744 021300 000000 0 ;3RD WORD IS THE BUFFER HIGH ADDRESS.
2745 021302 000000 0 ;4TH WORD IS THE BYTE/RECORD/FILE COUNT.
2746
2747 ; WRITE SUB-SYSTEM MEMORY CHARACTERISTIC BLOCK.
2748
2749 021304 000000 WSMBK:: 0 ;1ST WORD:: SEL 0
2750 021306 000000 0 ;2ND WORD:: SEL 2
2751 021310 000000 0 ;3RD WORD:: SEL 4
2752 .EVEN
2753 ;*
2754 ; SUBROUTINE TO CHECK WETHER OR NOT WE'LL TEST NXM
2755 ;
2756 ;
2757 ; INPUTS:
2758 ; OUTPUTS:
2759 ; The NXMFLG is set if we can test.
2760 ; The NXMLO and NXMHI addresses are setup.
2761 ;-
2762
2763 021312 MEMCK::
2764
2765 021312 SAVREG ;SAVE THE REGISTERS
2766 021316 005037 003130 CLR NXMFLG ;CLEAR THE FLAG
2767 021322 005037 003132 CLR NXMLO ;CLEAR THE TEST ADDRESS LO
2768 021326 005037 003134 CLR NXMHI ;CLEAR THE TEST ADDRESS HI
2769 021332 005737 003140 TST T23B ;IS IT A 11/23B?
  
```

```

2770 021336 001407          BEQ      14      ;NO
2771 021340 023727 002120 007777  CMP     L#HIME,#7777 ; GREATER THAN 128K
2772 021346 103406          BLO     24      ; NO
2773 021350 004737 021466  JSR     PC,NXMTST ;SETUP THE ADDRESS
2774 021354 000427          BR      134     ;SET THE FLAG AND EXIT
2775 021356 005737 003136 14:    TST     T23A    ;IS IT A 11/23A?
2776 021362 001413          BEQ     44      ;NO
2777 021364 023727 002120 005777 24:    CMP     L#HIME,#5777 ;GREATER THAN 96K
2778 021372 101023          BHI     144     ;YES,23A/23B WITH 128K MEMORY
2779 021374 023727 002120 003777  CMP     L#HIME,#3777 ;GREATER THAN 64K BUT LESS THAN 92K?
2780 021402 103403          BLO     44      ;NO, CHECK 24K
2781 021404 004737 021466  JSR     PC,NXMTST ;SETUP THE ADDRESS
2782 021410 000411          BR      134     ;SET THE FLAG AND EXIT
2783 021412 023727 002120 001577 44:    CMP     L#HIME,#1577 ;GREATER THAN 24K BUT LESS THAN 64K?
2784 021420 103410          BLO     144     ;NO, TELL THEM AND EXIT WITH FLAG CLEAR
2785 021422 004737 021466  JSR     PC,NXMTST ;SETUP THE ADDRESS
2786 021426 062737 000077 003134  ADD     #77,NXMHI ;FOOL THE 11/02 & 11/03
2787 021434 005237 003130 134:   INC     NXMFLG  ;SET THE FLAG
2788 021440 000411          BR      154     ;EXIT
2789 021442 000410 144:   BR      154     ;NOP FOR PRINTOUT
2790 021444          PRINTF  #NOMEM  ;TELL THEM & EXIT ***NO PRINT*****
      021444 012746 005454          MOV     #NOMEM,-(SP)
      021450 012746 000001          MOV     #1,-(SP)
      021454 010600          MOV     SP,R0
      021456 104417          TRAP   C#PNTF
      021460 062706 000004          ADD     #4,SP
2791 021464 000207 154:   RTS     PC      ;RETURN
2792
2793
2794
2795
2796
2797
2798
2799
2800 021466 013701 002120  NXMTST: MOV     L#HIME,R1      ;GET TOP OF MEMORY
2801 021472 062701 000200  ADD     #200,R1      ;MAKE IT I/O BLOCK OR OTHER NXM
2802 021476 042701 000177  BIC     #177,R1
2803 021502 010102          MOV     R1,R2      ;RESAVE RESULTS
2804          000006          .REPT   6
2805          ASL     R1      ;PUT IN PLACE FOR XFER
2806          .ENDR
2807 021520 010137 003132  MOV     R1,NXMLO    ;SAVE TEST ADDRESS LOW
2808          000012          .REPT   10
2809          ASR     R2      ;PUT IN PLACE FOR XFER
2810          .ENDR
2811 021550 042702 177700  BIC     #177700,R2  ;DON'T WANT ILA!
2812 021554 010237 003134  MOV     R2,NXMHI    ;SAVE TEST ADDRESS HIGH
2813 021560 000207          RTS     PC      ;RETURN
2814
2815 021562          ENDMOD

```

CH

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 14 JUN 84 16:41
KTINIT SETUP KT11 MEMORY MANAGEMENT REGISTERS

SFQ 0093

```

7          .TITLE  TSV4  MISCELLANEOUS SECTIONS
8
9 021562   BGNMOD  TSV4
10          TSV4::
11
12
13
14
15
16
17
18
19          .SBTTL  PROTECTION TABLE
20 021562   BGNPROT
21 021562   L$PROT::
22 021572   .WORD  -1.  1.  -1.  1          ;NO DEVICE PROTECTION REQUIRED.

```

```

24          .SBTTL INITIALIZE SECTION
25
26          ;**
27          ;THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
28          ;AT THE BEGINNING OF EACH PASS.
29
30          ;IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
31          ;IF "CONTINUE", NOTHING IS REQUIRED.
32
33          ;--
34          ;*
35          ;INSERT TEMPORARY JUMP TO ODT
36          ;
37          BGNINIT
38          L$INIT::
39          40$: CLR      EXTFEA
40          CLR      NXMFLG
41          MOV      @EPR1,EPR1SW          ;SET UP PRIMARY MESSAGE FOR REPLACEMENT
42          CLR      SIFLAG                ;CLEAR "SOFT INIT" FLAG
43          CLR      KTENABLE              ;CLEAR TEST ABOVE 28K FLAG
44          CLR      RAMSIZ                ;CLEAR WAM SIZE FOR RAMERR ROUTINE
45          READDEF @EF.CONTINUE
46          MOV      @EF.CONTINUE,RO
47          TRAP     C$REFG
48          BNCOMPLETE 1$
49          BCC      1$
50          CMP      UNITN,L$UNIT          ;UNIT IN RANGE?
51          BHIS     4$                    ;BR IF NO.
52          TST      DFLG                  ;DROPPED UNIT?
53          BMI      NXTU                  ;BR IF YES
54          MOV      UNITN,R1
55          ASL      R1
56          TST      ERTABL(R1)
57          BEQ      SETU
58          BIT      @BIT14,ERTABL(R1)    ;DROPPED?
59          BNE      NXTU
60          EXIT     INIT                  ;DO NOTHING IF "CONTINUE".
61          TRAP     C$EXIT
62          .WORD    L10030-.
63          1$: READDEF @EF.NEW
64          MOV      @EF.NEW,RO
65          TRAP     C$REFG
66          BNCOMPLETE NXTU                ;TAKE NEXT UNIT IF NOT NEW PASS.
67          BCC      NXTU
68          READDEF @EF.START
69          MOV      @EF.START,RO
70          TRAP     C$REFG
71          BCOMPLETE 2$
72          BCS      2$
73          READDEF @EF.RESTART
74          MOV      @EF.RESTART,RO
75          TRAP     C$REFG
76          BNCOMPLETE 31$
77          BCC      31$
78          2$: BRESET
79          TRAP     C$RESET
80
81          ;1ST PASS, BUS-INIT...
82          ;BUS RESET.

```



```

65 021734 005037 002206      CLR      TSTCNT      ;NUMBER OF TESTS RUN IN PASS
66 021740 005037 002214      CLR      FATFLG     ;CLEAR FATAL ERROR COUNT
67 021744 005037 003136      CLR      T23A      ;CLEAR 11/23A FLAG
68 021750 005037 003140      CLR      T23B      ;CLEAR 11/23B FLAG
69                               ;
70                               ;
71                               ;
72 021754 005037 003372      CLR      SKIPT      ;CLEAR THE SUBTEST "SKIPPER"
73 021760                               20$:
74 021760 012737 177777 002176  MOV      #-1,QVP    ;...QUICK VERIFY...
75 021766 004737 020710      JSR      PC,ENVIRN  ;SET ENVIRONMENT.
76 021772 004737 021070      JSR      PC,KTINIT  ;INITIALIZE KT MEMORY MANAGEMENT
77 021776 012700 003170      MOV      #ERTABL,RO
78 022002 005020      30$:      CLR      (RO)+      ;CLEAR THE ERROR TABLE
79 022004 020027 003370      CMP      RO,#ERTABE
80 022010 103774      BLO     30$
81 022012 000404      BR      4$
82 022014 005037 002176      31$:      CLR      QVP
83 022020 000137 022070      JMP      PASRPT    ;GO REPORT THE STATUS
84
85 022024                               4$:
86 022024 012737 177777 002174  NEWPAS: MOV      #-1,UNITN  ;INIT UNIT NUMBER...
87 022032 005037 002212      CLR      DEVCNT    ;CLEAR COUNT OF DEVICES RUNNING
88 022036                               NXTU:
89 022040 005237 002174      BREAK   C#BRK
90 022044 023737 002174 002012  INC      UNITN
91 022052 103423      TRAP   C#BRK
92 022054 012737 177777 003104  INC      UNITN,L#UNIT
93 022062 000401      CMP     SETU
94 022064                               MOV     #-1,DUFLG
95 022066 000240      BR      11$
96 022070                               DOCLN  C#DCLN
97 022070 023727 002012 000001  11$:      TRAP   C#DCLN
98 022076 101752      PASRPT: NOP
99 022100 005737 002212      CMP     L#UNIT,#1  ;HOW MANY UNITS SELECTED?
100 022104 001747      BLOS   NEWPAS      ;BR IF ONLY 1
101 022106      TST    DEVCNT    ;ARE ANY STILL RUNNING?
102 022110 032700 000100      BEQ   NEWPAS      ;BR IF NO
103 022114 001343      BEQ   NEWPAS
104
105 022116      RFLGS  RO
106 022120 000741      TRAP  C#RFLA
107 022122      BIT   #ISR,RO
108      BNE   NEWPAS  ;SHOULD WE PRINT STATISTICS
109 022122      DORPT TRAP  C#DRPT
110 022122 013700 002174      BR    NEWPAS
111 022126 104442      10$:
112 022130      SETU:  GPHARD UNITN,RO  ;GET UNIT N P-TABLE POINTER.
113 022132 005037 003104      MOV   UNITN,RO
114 022134 005237 002212      TRAP C#GPHRD
115 022136 012001      BNCOMPLETE NXTU  ;BR IF UNIT NOT AVAILABLE.
116 022138 010137 002200      BCC  NXTU
117      CLR  DUFLG    ;CLEAR "DROPPED" FLAG.
118      INC  DEVCNT
119      MOV  (RO)+,R1  ;GET 1ST REGISTER ADDRESS.
120      MOV  R1,CSRADDR ;ADDRESS OF REGISTERS OF UNIT UNDER TEST

```

```

115
116 022150 012001      MOV      (R0),R1      ;GET VECTOR ADDRESS.
117                  ;MOV      (R0),R2      ;GET INTERRUPT PRIORITY
118                  ;MOV      R2,IPRI    ;SET INTERRUPT PRIORITY.
119 022152 010137 002202 MOV      R1,IVEC     ;SET INTERRUPT VECTOR POINTER..
120 022156 012721 016276 MOV      @INTR,(R1). ;...VECTOR...
121 022162 013721 002204 MOV      IPRI,(R1).  ;...AND PRIORITY.
122
123 022166             1$:
124                   ;      TST      QVP      ;1ST PASS ??
125                   ;      BEQ      5$      ;NO, SKIP THE PASS 1 STUFF.
126
127                   ;
128                   ;1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
129                   ;THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
130                   ;
131 022166 013701 002174      MOV      UNITN,R1
132 022172 006301            ASL      R1
133 022174 052761 100000 003170 BIS      @BIT15,ERTABL(R1) ;SAY DEVICE RUNNING
134 022202 005037 005766      CLR      EXTA      ;CLEAR ERROR EXTENSION FLAG.
135 022206 023727 0020i2 000001 CMP      L#UNIT,#1    ;ARE WE TESTING MULTIPLE UNITS?
136 022214 101416            BLOS     10$      ;BR IF NO.
137 022216            RFLAGS  R0      ;YES -- GET OPERATOR FLAGS.
138 022220 032700 001000      TRAP     C#RFLA
139 022224 001412            BIT      @PNT,R0      ;SHOULD WE PRINT UNIT #?
140 022226            BEQ      10$      ;BR IF NOT.
141 022226 013746 002174      PRINTF  @PUNIT,UNITN ;PRINT THE UNIT #
142 022232 012746 022320      MOV      UNITN,-(SP)
143 022236 012746 000002      MOV      @PUNIT,-(SP)
144 022242 010600            MOV      #2,-(SP)
145 022244 104417            MOV      SP,R0
146 022246 062706 000006      TRAP     C#PNTF
147 022252            ADD      @6,SP
148 022252 005037 003106      10$:    CLR      NODEV
149 022256 013701 002200      MOV      CSRADDR,R1 ;ADDRESS OF FIRST REGISTER
150 022262 010102            MOV      R1,R2      ;START OF REGISTERS
151 022264 062702 000002      ADD      @TSSR,R2  ;ADDRESS OF TSSR REGISTER
152 022270 004737 016456      JSR      PC,XNXM    ;TEST BOTH CONTROLLER REGISTERS...
153 022274 103005            BCC     2$      ;...AND BR IF ALL OK.
154 022276 010137 003106      MOV      R1,NODEV  ;FLAG DEVICE AS NON-EXISTENT
155 022302 012737 177777 003104 MOV      #-1,DUFLG ;DROP THIS UNIT.
156 022310            2$:
157 022310            ;
158 022310            ;FINALLY, SET CPU PRIORITY AND WE'RE DONE.
159 022310            ;
160 022310 012700 000000      5$:    SETPRI  @PRI00      ;ENABLE INTERRUPTS.
161 022314 104441            MOV      @PRI00,R0
162 022316            TRAP     C#SPRI
163 022316            ENDINIT
164 022316 104411            L10030: TRAP     C#INIT
165 022320 045 116 045 PUNIT: .ASCIZ /#N#N#A***** TESTING UNIT #D2#A *****/
166 022320            .EVEN

```

.SBTTL ADD AND DROP UNITS SECTIONS

```

160
161
162
163
164
165
166
167 022366
    022366
168 022366 010001
169 022370 006301
170 022372 052761 100000 003170
171 022400 042761 040000 003170
172 022406
    022406 010046
    022410 012746 022434
    022414 012746 000002
    022420 010600
    022422 104417
    022424 062706 000006
173 022430
    022430 000167
    022432 000026
174 022434 045 116 045 14:
175
176
177 022462
    022462
    022462 104452
178
179
180
181
182
183
184
185
186
187
188
189 022464
    022464
190 022464 012737 177777 003104
191 022472 010001
192 022474 006301
193 022476 052761 140000 003170
194 022504 000240 000240 000240
195 022512
    022512 010046
    022514 012746 022540
    022520 012746 000002
    022524 010600
    022526 104417
    022530 062706 000006
196 022534
    022534 000167
    022536 000030

```

```

; **
; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
; OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
; ---
BGNAU
L$AU::
MOV      R0,R1          ; GET UNIT TO BE ADDED (R0)
ASL      R1             ; MAKE IT A WORD INDEX
BIS      #100000,ERTABL(R1) ; SET THE "ACTIVE" BIT
BIC      #40000,ERTABL(R1) ; CLEAR THE "DROPPED" BIT
PRINTF   #1$,R0
MOV      R0,-(SP)
MOV      #1$,-(SP)
MOV      #2,-(SP)
MOV      SP,R0
TRAP     C$PNTF
ADD      #6,SP
EXIT     AU
.WORD    J$JMP
.WORD    L10031-2-.
14:      .ASCIZ  /#N#A UNIT #D#A ADDED/
        .EVEN
ENDAU    ; UNUSED.
L10031:  TRAP     C$AU
; **
; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
; TO BE REMOVED FROM THE TEST LIST.
;
; SUPVSR DOES THE "DROPPING". THIS IS JUST TO TELL THE MAN.
; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
; COMMAND, OTHERWISE REMAIN INACTIVE. THE "DISPLAY" COMMAND
; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
; WHICH ARE STILL ACTIVE.
; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.
BGNDU
L$DU::
MOV      #-1,DUFLG
MOV      R0,R1
ASL      R1
BIS      #140000,ERTABL(R1) ; SAY DROPPED
        240,240,240 ; ??????????
PRINTF   #1$,R0
MOV      R0,-(SP)
MOV      #1$,-(SP)
MOV      #2,-(SP)
MOV      SP,R0
TRAP     C$PNTF
ADD      #6,SP
EXIT     DU
.WORD    J$JMP
.WORD    L10032-2-.

```

TSV4 - MISCELLANEOUS SECTIONS
ADD AND DROP UNITS SECTIONS

MACRO M1113 14-JUN-84 16:41

SEQ 0098

```

197 022540      045      116      045 1$: .ASCIZ /#N#A UNIT #D#A DROPPED/
198                                     .EVEN
199 022570                                     ENDDU
    022570      104453      L10032: TRAP C#DU
    022570
200                                     ;**
201                                     ; AUTO-DROP CODE SECTION.
202                                     ; -
203 022572                                     BGNAUTO
    022572      L#AUTO::
204 022572      013705      002200      MOV CSRADDR,R5 ;POINT TO DEVICE REGISTER
205 022576      012703      000550      MOV #360.,R3 ;ENOUGH TIME FOR 2400' REEL TO REWIND
206 022602      004737      016330      10$: JSR PC,WAITF ;WAIT FOR SSR TO SET
207 022606      103420      BCS 20$ ;LEAVE WHEN SSR IS SET
208 022610      DELAY 250. ;WAIT FOR .25 SECONDS
    022610      012727      000372      MOV #250.,(PC).
    022614      000000      .WORD 0
    022616      013727      002116      MOV L#DLY,(PC).
    022622      000000      .WORD 0
    022624      005367      177772      DEC -6(PC)
    022630      001375      BNE .-4
    022632      005367      177756      DEC -22(PC)
    022636      001367      BNE .-20
209 022640      005303      DEC R3 ;BUMP COUNTER DOWN
210 022642      001357      BNE 10$ ;KEEP GOING
211 022644      004737      017262      JSR PC,CKDROP ;TRY AND DROP UNIT
212 022650      20$: ENDAUTO ; UNUSED.
213 022650      L10033: TRAP C#AUTO
    022650      104461

```



```

023016 012746 000002      MOV      #2,(SP)
023022 010600      MOV      SP,R0
023024 104416      TRAP     C:PNTS
023026 062706 000006      ADD      #6,SP
254 023032 000431      BR       4:
255 023034 020227 160001  3:      CMP      R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
256 023040 001012      BNE     30:             ; BR IF NO.
257 023042      PRINTS  #DEVNRD,R3
      023042 010346      MOV      R3,-(SP)
      023044 012746 023331      MOV      #DEVNRD,-(SP)
      023050 012746 000002      MOV      #2,-(SP)
      023054 010600      MOV      SP,R0
      023056 104416      TRAP     C:PNTS
023060 062706 000006      ADD      #6,SP
258 023064 000414      BR       4:
259 023066 042702 170000  30:     BIC      #C7777,R2
260 023072      PRINTS  #DEVDR0,R3,R2
      023072 010246      MOV      R2,-(SP)
      023074 010346      MOV      R3,-(SP)
      023076 012746 023412      MOV      #DEVDR0,-(SP)
      023102 012746 000003      MOV      #3,-(SP)
      023106 010600      MOV      SP,R0
      023110 104416      TRAP     C:PNTS
      023112 062706 000010      ADD      #10,SP
261 023116 062704 000002  4:      ADD      #2,R4
262 023122 065203      INC      R3
263 023124 020427 003370      CMP      R4,#ERTABE
264 023130 103701      BLO     1:
265 023132 012604      MOV      (SP)+,R4
266 023134 012603      MOV      (SP)+,R3
267 023136 012602      MOV      (SP)+,R2
268 023140      ENDRPT      ; UNUSED.
      023140      L10035:
      023140 104425      TRAP     C:RPT
269
270 023142      045      116      045  DEVSUM: .ASCIZ /#N#ADEVICE STATUS SUMMARY:#N/
271 023177      045      101      040  DEVONL: .ASCIZ /#A UNIT #D3#A ONLINE, ERRORS = #D#N/
272 023247      045      101      040  DEVNXR: .ASCIZ /#A UNIT #D3#A DROPPED, NON-EXISTENT REGISTER#N/
273 023331      045      101      040  DEVNRD: .ASCIZ /#A UNIT #D3#A DROPPED, NOT READY AT STARTUP#N/
274 023412      045      101      040  DEVDR0: .ASCIZ /#A UNIT #D3#A DROPPED, ERRORS = #D#N/
275
276
277 023462      ENDMOD
278

```

TSV4 - MISCELLANEOUS SECTIONS MACRO M1113 14 JUN-84 16:41
CLEAN-UP AND REPORT CODING SECTIONS

SEQ 0101

1
2
3
10
11 023462
023462
17

.TITLE TEST 1 - HARDWARE TEST 1-8 TESTS

BGNMOD TSV7B
TSV7B::

TEST 1 HARDWARE TEST 1 B TEST MACRO M1113 14 JUN 84 16:41
 TEST 1: WRITE TAPE MARK RETRY

SEQ 0105

	024176	000154								.WORD	108
	024200	003646								.WORD	SFIERR
	024202	012114								.WORD	SFIMSG
177	024204	013737	002174	026270	20:	MOV	UNITN,T29DSW		;SET UP UNIT NUMBER		
178											
179	024212	012704	026250			MOV	@T29PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS		
180	024216	004737	010742			JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS		
181	024222	103407				BCS	25:		;BR, IF COMMAND ISSUED OK		
182	024224	005237	002214			INC	FATFLG		;ERROR COUNT		
186	024230	010001				MOV	R0,R1		;SAVE CONTENTS OF TSSR		
187	024232					ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED		
	024232	104456								TRAP	C1ERRRD
	024234	000155								.WORD	109
	024236	005052								.WORD	WRTMSG
	024240	012114								.WORD	SFIMSG
188	024242				25:	CKLOOP			;LOOP IF SELECTED		
	024242	104406								TRAP	C1CLP1
189	024244	001737	011074		26:	JSR	PC,REWIND		;CALL TAPE REWIND COMMAND		
190	024250	016501	000002			MOV	TSSR(R5),R1		;GET TSSR		
191	024254	012702	000200			MOV	@SSR,R2		;SET UP EXPECTED TSSR		
192	024260	103407				BCS	30:		;BR, IF NO PROBLEM		
193	024262	010004				MOV	R0,R4		;PACKET ADDRESS SET UP		
194	024264	005237	002214			INC	FATFLG		;ERROR COUNT		
198	024270					ERRHRD	ERRNO,T29RWN,PKTSSR		;REWIND NOT ACCEPTED		
	024270	104456								TRAP	C1ERRRD
	024272	000156								.WORD	110
	024274	030235								.WORD	T29RWN
	024276	012126								.WORD	PKTSSR
199	024300				30:	CKLOOP			;LOOP IF SELECTED		
	024300	104406								TRAP	C1CLP1
200	024302	013701	026300			MOV	T29BFR*6,R1		;PICK UP XSTO		
201	024306	010102				MOV	R1,R2		;SET UP EXPECTED		
202	024310	052702	000002			BIS	@BIT1,R2		;SET BOT BIT IN EXPECTED		
203	024314	020102				CMP	R1,R2		;DOES EXP = REC'D		
204	024316	001406				BEQ	40:		;BR, IF EQUAL (OK)		
205	024320	005237	002214			INC	FATFLG		;ERROR COUNT		
209	024324					ERRHRD	ERRNO,T29BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	024324	104456								TRAP	C1ERRRD
	024326	000157								.WORD	111
	024330	027726								.WORD	T29BOT
	024332	015554								.WORD	EXPREC
210	024334	012737	000001	025372	40:	MOV	@1,T29RB		;NUMBER OF RECORDS TO SPACE OVER		
211	024342	012737	000400	026376		MOV	@256.,T29SZ		;SET UP RECORD SIZE		
212	024350	012737	140005	026370		MOV	@140005,T29PK3		;WRITE FORWARD,CVC=1,ACK COMMAND		
213	024356	012704	026370			MOV	@T29PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
214	024362	010465	000000			MOV	R4,TSD8(R5)		;ISSUE COMMAND		
215	024366	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET		
216	024372	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
217	024376	012702	000200			MOV	@SSR,R2		;SET UP EXPECTED		
218	024402	020102				CMP	R1,R2		;ARE THEY EQUAL		
219	024404	001420				BEQ	75:		;BR, IF OK		
220	024406	013703	026300			MOV	T29BFR*6,R3		;PICK UP XT50		
221	024412	032703	000004			BIT	@4,R3		;IS UNIT WRITE-LOCKED?		
222	024416	001405				BEQ	41:		;NO,PROCEED WITH NORMAL ERROR		
223	024420					ERRDF	ERRNO,T29WLK,SFIMSG		;TAPE IS WRITE LOCKED		
	024420	104455								TRAP	C1ERDF
	024422	000157								.WORD	111

	024424	027574							.WORD	T29MLK
	024426	012114							.WORD	SFIMSG
224	024430				DOCLN					;DROP IT
	024430	104444							TRAP	C1DCLN
225	024432	005237	002214	414:	INC	FATFLG				;ERROR COUNT
229	024436				ERRHRD	ERRNO,T29WRT,PKTSSR				;TSSR INCORRECT AFTER WRITE DATA
	024436	104456							TRAP	C1ERRHRD
	024440	000160							.WORD	112
	024442	027661							.WORD	T29WRT
	024444	012126							.WORD	PKTSSR
230	024446			754:	CKLOOP					;LOOP IF SELECTED
	024446	104406							TRAP	C1CLP1
231	024450	012737	000001		MOV	#1,T29RB				;NUMBER OF RECORDS TO SPACE OVER
232	024456	012737	140410		MOV	#140410,T29PK3				;SET UP COMMAND IN PACKET
UP	SPACE REVERSE									
233	024464	012704	026370		MOV	#T29PK3,R4				;SET UP R4 WITH PACKET ADDRESS
234	024470	010465	000000		MOV	R4,TSDB(R5)				;ISSUE COMMAND
235	024474	004737	016330		JSR	PC,WAITF				;WAIT FOR SSR TO SET
236	024500	016501	000002		MOV	TSSR(R5),R1				;GET TSSR CONTENTS
237	024504	012702	000200		MOV	#SSR,R2				;SET UP EXPECTED
238	024510	020102			CMP	R1,R2				;ARE THEY EQUAL
239	024512	001406			BEQ	1754				;BR, IF OK
240	024514	005237	002214		INC	FATFLG				;ERROR COUNT
244	024520				ERRHRD	ERRNO,T29WDE,PKTSSR				;TSSR INCORRECT AFTER READ DATA
	024520	104456							TRAP	C1ERRHRD
	024522	000161							.WORD	113
	024524	027512							.WORD	T29WDE
	024526	012126							.WORD	PKTSSR
245	024530			1754:	CKLOOP					;LOOP IF SELECTED
	024530	104406							TRAP	C1CLP1
246	024532	013737	003116		MOV	FREE,T29RB				;ADDRESS OF BUFFER
247	024540	012737	141011		MOV	#141011,T29PK3				;WRITE TAPE MARK RETRY,ACK,CVC=1 CMD.
248	024546	012704	026370		MOV	#T29PK3,R4				;SET UP R4 WITH PACKET ADDRESS
249	024552	010465	000000		MOV	R4,TSDB(R5)				;ISSUE COMMAND
250	024556	004737	016330		JSR	PC,WAITF				;WAIT FOR SSR TO SET
251	024562	016501	000002		MOV	TSSR(R5),R1				;GET TSSR CONTENTS
252	024566	012702	100204		MOV	#SSR!SC!BIT2,R2				;SET UP EXPECTED
253	024572	020102			CMP	R1,R2				;ARE THEY EQUAL
254	024574	001406			BEQ	1804				;BR, IF OK
255	024576	005237	002214		INC	FATFLG				;ERROR COUNT
259	024602				ERRHRD	ERRNO,T29WDE,PKTSSR				;TSSR INCORRECT AFTER READ DATA
	024602	104456							TRAP	C1ERRHRD
	024604	000162							.WORD	114
	024606	027512							.WORD	T29WDE
	024610	012126							.WORD	PKTSSR
260	024612			1804:	CKLOOP					;LOOP IF SELECTED
	024612	104406							TRAP	C1CLP1
261	024614	013701	026306		MOV	T29RFR+14,R1				;GET XST3 STATUS WORD
262	024620	010102			MOV	R1,R2				;SET UP EXPECTED
263	024622	052702	000001		BIS	#BIT0,R2				;SET THE RIB BIT
264	024626	020102			CMP	R1,R2				;ARE THEY EQUAL
265	024630	001406			BEQ	1904				;BR, IF EQUAL (GOOD)
266	024632	005237	002214		INC	FATFLG				;ERROR COUNT
270	024636				ERRHRD	ERRNO,T29RIB,EXPREC				;NEF SHOULD BE SET
	024636	104456							TRAP	C1ERRHRD
	024640	000163							.WORD	115
	024642	031654							.WORD	T29RIB
	024644	015554							.WORD	EXPREC


```

313 025032 004737 011074      JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
314 025036 103411              BCS      304           ;BR, IF NO PROBLEM
315 025040 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR
316 025044 010004              MOV      R0,R4         ;SAVE PACKET ADDRESS
317 025046 005237 002214      INC      FATFLG        ;ERROR COUNT
321 025052              ERRHRD  ERRNO,T29RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C1ERHRD
                                .WORD    118
                                .WORD    T29RWN
                                .WORD    PKTSSR
322 025062              304:   CKLOOP      ;LOOP IF SELECTED
                                TRAP      C1CLP1
323 025064 013701 026300      MOV      T29BFR+6,R1   ;PICK UP XSTO
324 025070 010102              MOV      R1,R2         ;SET UP EXPECTED
325 025072 052702 000002      BIS      @BIT1,R2      ;SET BOT BIT IN EXPECTED
326 025076 020102              CMP      R1,R2         ;DOES EXP = REC'D
327 025100 001406              BEQ      404           ;BR, IF EQUAL (OK)
328 025102 005237 002214      INC      FATFLG        ;ERROR COUNT
332 025106              ERRHRD  ERRNO,T29BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C1ERHRD
                                .WORD    119
                                .WORD    T29BOT
                                .WORD    EXPREC
333 025116              404:   CKLOOP      ;LOOP IF SELECTED
                                TRAP      C1CLP1
334 025120 012737 140011 026370  MOV      @140011,T29PK3 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
335 025126 012704 026370      MOV      @T29PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
336 025132 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
337 025136 004737 016330      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
338 025142 016501 000002      MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
339 025146 012702 000200      MOV      @SSR,R2       ;SET UP EXPECTED
340 025152 020102              CMP      R1,R2         ;ARE THEY EQUAL
341 025154 001406              BEQ      704           ;BR, IF OK
342 025156 005237 002214      INC      FATFLG        ;ERROR COUNT
346 025162              ERRHRD  ERRNO,T29WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE TAPE MARK
                                TRAP      C1ERHRD
                                .WORD    120
                                .WORD    T29WDC
                                .WORD    PKTSSR
347 025172              704:   CKLOOP      ;LOOP IF SELECTED
                                TRAP      C1CLP1
348 025174 012703 000001      MOV      @1.,R3        ;NUMBER OF RECORDS TO WRITE TH
349 025200 012737 141011 026370  MOV      @141011,T29PK3 ;WRITE TAPE MARK RETRY,ACK,CVC=1 COMMAND
350 025206 012704 026370      MOV      @T29PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
351 025212 010465 000000      MOV      R4,TSDB(R5)   ;ISSUE COMMAND
352 025216 004737 016330      JSR      PC,WAITF      ;WAIT FOR SSR TO SET
353 025222 016501 000002      MOV      TSSR(R5),R1   ;PICK UP TSSR
354 025226 012702 000200      MOV      @SSR,R2       ;SET UP EXPECTED (SSR ONLY)
355 025232 020102              CMP      R1,R2         ;WAS STATUS GOOD
356 025234 001406              BEQ      1654          ;BR, IF TERMINATION WAS GOOD
357 025236 005237 002214      INC      FATFLG        ;ERROR COUNT
361 025242              ERRHRD  ERRNO,T29WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP      C1ERHRD
                                .WORD    121
                                .WORD    T29WDC
                                .WORD    PKTSSR
362 025252              1654:  CKLOOP      ;LOOP IF SELECTED

```


TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 1: WRITE TAPE MARK RETRY

SEQ 0111

453	025644	012737	140011	026370	MOV	#140011,T29PK3	;WRITE TAPE MARK,ACK,CVC=1 COMMAND
454	025652	012704	026370		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
455	025656	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND
456	025662	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
457	025666	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS
458	025672	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED
459	025676	020102			CMP	R1,R2	;ARE THEY EQUAL
460	025700	001406			BEQ	70#	;BR, IF OK
461	025702	005237	002214		INC	FATFLG	;ERROR COUNT
465	025706				ERRHRD	ERRNO,T29WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE TAPE MARK
	025706	104456					TRAP C#ERHRD
	025710	000200					.WORD 128
	025712	030627					.WORD T29WDC
	025714	012126					.WORD PKTSSR
466	025716			70#:	CKLOOP		;LOOP IF SELECTED
	025716	104406					TRAP C#CLP1
467	025720	012703	000012		150#:	MOV #10.,R3	;NUMBER OF RECORDS TO WRITE TM
468	025724	012737	000001	026372	MOV	#1,T29RB	;SET UP PACKET
469	025732	012737	141011	026370	MOV	#141011,T29PK3	;WRITE TAPE MARK RETRY,ACK,CVC=1 COMMAND
470	025740	012704	026370		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
471	025744	010465	000000		155#:	MOV R4,TSDB(R5)	;ISSUE COMMAND
472	025750	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
473	025754	016501	000002		MOV	TSSR(R5),R1	;PICK UP TSSR
474	025760	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED (SSR ONLY)
475	025764	020102			CMP	R1,R2	;WAS STATUS GOOD
476	025766	001406			BEQ	165#	;BR, IF TERMINATION WAS GOOD
477	025770	005237	002214		INC	FATFLG	;ERROR COUNT
481	025774				ERRHRD	ERRNO,T29WDC,PKTSSR	;TSSR NOT CORRECT AFTER WRT TAPE M.
	025774	104456					TRAP C#ERHRD
	025776	000201					.WORD 129
	026000	030627					.WORD T29WDC
	026002	012126					.WORD PKTSSR
482	026004			165#:	CKLOOP		;LOOP IF SELECTED
	026004	104406					TRAP C#CLP1
483	026006	005303			DEC	R3	;BUMP COUNTER DOWN
484	026010	001355			BNE	155#	;BR, IF LESS THAN 10 TAPE MARKS
485	026012	012737	140410	026370	MOV	#140410,T29PK3	;SPACE REVERSE,ACK,CVC=1, COMMAND
486	026020	012737	000001	026372	MOV	#1,T29RB	;NUMBER OF RECORDS TO SPACE BACK
487	026026	012704	026370		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
488	026032	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND
489	026036	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET
490	026042	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS
491	026046	012702	100204		MOV	#SSR!SC!BI;2,R2	;SET UP EXPECTED
492	026052	020102			CMP	R1,R2	;ARE THEY EQUAL
493	026054	001406			BEQ	222#	;BR, IF OK
494	026056	005237	002214		INC	FATFLG	;ERROR COUNT
498	026062				ERRHRD	ERRNO,T29WDE,PKTSSR	;TSSR INCORRECT AFTER SPACE CMD.
	026062	104456					TRAP C#ERHRD
	026064	000202					.WORD 130
	026066	027512					.WORD T29WDE
	026070	012126					.WORD PKTSSR
499	026072			222#:	CKLOOP		;LOOP IF SELECTED
	026072	104406					TRAP C#CLP1
500	026074	012737	100410	026370	MOV	#100410,T29PK3	;SPACE REVERSE,ACK, COMMAND
501	026102	012737	000005	026372	MOV	#5,T29RB	;NUMBER OF RECORDS TO SPACE BACK
502	026110	012704	026370		MOV	#T29PK3,R4	;SET UP R4 WITH PACKET ADDRESS
503	026114	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND


```

555 026270 000000 T29DSW: .WORD 0 ;SELECT DRIVE 0
556 026272 T29BFR: .BLKW 25. ;MESSAGE BUFFER
557 ;
558 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
559 ;
561 026360 T29PK2: .=<..*10>E177770
563 026360 T29PK2: .WORD 100006 ;WRITE SUB SYS MEM COMMAND, AND ACK
564 026360 100006 .WORD T29BF2 ;ADDRESS OF SELECT BLOCK DATA
565 026362 026400 .WORD 0
566 026364 000000 .WORD 6. ;SIZE OF DATA PACKET
567 026366 000006
568
572 026370 T29PK3:
573 026370 140005 .WORD 140005 ;WRITE TAPE MARK RETRY COMMAND, CVC=1 AND ACK
574 026372 T29RB:
575 026372 003116 T29WB: .WORD FREE ;ADDRESS OF WRITE BUFFER
576 026374 000000 .WORD 0
577 026376 000000 T29SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
578 .EVEN
579 ;
580 ;
581 ;
582 026400 T29BF2:
583 026400 010 T29BS0: .BYTE 10 ;BSELO AREA
584 026401 200 T29BS1: .BYTE 200 ;BSEL1 AREA
585 026402 000000 T29S2: .WORD 0 ;SEL 2 AREA
586 026404 000000 T29S3: .WORD 0 ;DATA AREA
587 ;
588 ;
589 .EVEN
590 ;TAPE MOTION PACKET COMMAND VALUES
591
592 026406 140001 T29RN: .WORD 140001 ;READ DATA
593 026410 140401 T29WR: .WORD 140401 ;READ DATA REVERSE
594 026412 141001 T29CON: .WORD 141001 ;READ PREVIOUS OPP=0
595 026414 161001 .WORD 161001 ;READ PREVIOUS OPP=1
596 026416 141401 .WORD 141401 ;WRITE TAPE MARK RETRY NEXT OPP=0
597 026420 161401 .WORD 161401 ;WRITE TAPE MARK RETRY NEXT OPP=1
598 026422 177777 .WORD 177777 ;END OF DATA
599
600 ;
601 026424 000000 T29CNT: .WORD 0 ;TAPE RECORD COUNTER STORAGE AREA
602
603 026426 000000 T29RSZ: .WORD 0 ;RECORD STORAGE SIZE AREA
604 026430 000000 T29DLY: .WORD ;DELAY COUNTER STORAGE AREA
605 ;
606 ;LOCAL TEXT MESSAGES FOR TEST
607 ;-
608
609 026432 104 162 151 T29OFL: .ASCIZ 'Drive is OFFLINE'
610 026453 124 141 160 T29WNG: .ASCIZ 'Tape Position Incorrect After WRITE TAPE MARK RETRY Previous (OPP=1)'
611 026560 127 122 111 T29NEF: .ASCIZ 'WRITE TAPE MARK RETRY, At BOT, Failed To Set NEF (XST0)'
612 026650 124 123 123 T29RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
613 026717 127 122 111 T29RRF: .ASCIZ 'WRITE TAPE MARK RETRY Previous (Space Reverse, Read Forward) Command Failed'
614 027033 127 122 111 T29RRG: .ASCIZ 'WRITE TAPE MARK RETRY Previous (Read Forward, Space Reverse) Command Failed'
615 027147 120 117 123 T29SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
616 027231 122 111 102 T29LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'

```

617	027301	124	123	123	T29WDF:	.ASCIZ	'TSSR Not Correct After Illegal Mode Bits Set'
618	027356	111	154	154	T29L00:	.ASCIZ	'Illegal Mode Bits, Failed To Set ILC Bit In XST0
619	027437	127	122	111	T29SR:	.ASCIZ	'WRITE TAPE MARK RETRY COMMAND Not Accepted'
620	027512	124	123	123	T29WDE:	.ASCIZ	'TSSR Not Correct After SPACE REVERSE DATA Command'
621	027574	052	052	052	T29WLK:	.ASCIZ	'*****TAPE IS WRITE-LOCKED AND WILL CAUSE ERRORS*****'
622	027661	124	123	123	T29WRT:	.ASCIZ	'TSSR Not Correct After WRITE Command'
623	027726	124	14	160	T29BOT:	.ASCIZ	'Tape Not At BOT After REWIND Command'
624	027773	104	14	164	T29DTA:	.ASCIZ	'Data Written To Tape Not Equal To Data Read From Tape'
625	030061	127	122	111	T29EOT:	.ASCIZ	'WRITE TAPE MARK RETRY DATA OVER EOT GAVE NO TAPE STATUS ALERT
626	030157	124	123	123	T29TH:	.ASCIZ	'TSSR Not Correct After SPACE REVERSE Into BOT'
627	030235	122	145	167	T29RWV:	.ASCIZ	'Rewind (POSITION) Command Not Accepted'
628	030304	122	101	115	T29RNC:	.ASCIZ	'RAM Error, Correct Data Pattern Not In Ram'
629	030357	124	123	123	T29AM3:	.ASCIZ	'TSSR Init. Failed After WRITE TAPE MARK RETRY COMMAND'
630	030445	104	162	151	T29OF7:	.ASCIZ	'Drive 7 Select Failed To Set "OFL" In TSSR'
631	030520	124	123	123	T29WDD:	.ASCIZ	'TSSR Not Correct After WRITE TAPE MARK RETRY DATA Command, SWB Bit Set
632	030627	124	123	123	T29WDC:	.ASCIZ	'TSSR Not Correct After WRITE TAPE MARK RETRY DATA Command'
633	030721	103	126	103	T29VCK:	.ASCIZ	'CVC Set, Didn't Reset VCK In Message Buffer'
634	030774	124	123	102	T29BA:	.ASCIZ	'TSBA Not Correct After WRITE TAPE MARK RETRY DATA Command'
635	031066	127	122	111	T29WSS:	.ASCIZ	'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
636	031155	122	145	141	T29LON:	.ASCIZ	'Reading Long Record Failed To Set RLL Bit In XST0'
637	031237	122	145	141	T29LOP:	.ASCIZ	'Reading Long Record Failed To Set RLS Bit In XST0'
638	031321	122	145	163	T29PBP:	.ASCIZ	'Residual Byte Count Incorrect After Short Record Read'
639	031407	122	145	141	T29TRL:	.ASCIZ	'Reading Long Record Failed To Give Tape Status Alert'
640	031475	104	141	164	T29NEQ:	.ASCIZ	'Data WRITE TAPE MARK RETRY From Tape Not Correct, After SWB=1'
641	031573	124	123	123	T29RDG:	.ASCIZ	'TSSR Incorrect After READ REVERSE Into Tape Mark'
642	031654	127	122	111	T29RIB:	.ASCIZ	'WRITE TAPE MARK RETRY At First Record, Failed To Set RIB (XST3)'
643	031754	124	115	113	T29RRN:	.ASCIZ	'TMK (XST0) Failed To Set After READ REVERSE Into Tape Mark'
644	032047	127	162	151	TST29ID:	.ASCIZ	'Write Tape Mark Retry'

```

645      .EVEN
646      ;*
647      ;
648      ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
649      ;WRITE SUBSYSTEM MEMORY COMMAND
650      ;
651      ;-
652

```

```

653 032076      T29REST:
654 032076      SAVREG      ;SAVE THE REGISTERS
655 032102 012701 026250      MOV      #T29PACKET,R1      ;START OF THE PACKET
656 032106 012721 140004      MOV      #140004,(R1)+      ;WRITE SUBSYSTEM MEM. WITH ACK, CVC=1
657 032112 012721 026260      MOV      #T29DATA,(R1)+      ;ADDRESS OF CHARAISTICS DATA BLOCK
658 032116 005021      CLR      (R1)+      ;EXTENDED ADDRESS
659 032120 012721 000012      MOV      #10,(R1)+      ;SIZE OF DATA BLOCK IN BYTES
660 032124 012721 026272      MOV      #T29BFR,(R1)+      ;ADDRESS OF MESSAGE BUFFER
661 032130 005021      CLR      (R1)+
662 032132 012721 000024      MOV      #20,(R1)+      ;LENGTH OF MESSAGE BUFFER
663 032136 005021      CLR      (R1)+
664 032140 012711 000000      MOV      #0,(R1)      ;SELECT DRIVE ZERO (0)
665 032144 012702 000030      MOV      #24,R2      ;NUMBER OF LOCATIONS TO BE CLEARED
666 032150 012762 177777 026272 64$:      MOV      #177777,T29BFR(R2) ;ALL ONES TO MESSAGE BUFFER
667 032156 005742      TST      -(R2)      ;NEXT LOCATION
668 032160 020227 000000      CMP      R2,#0      ;CHECK FOR END OF LOOP
669 032164 001371      BNE      64$      ;KEEP GOING UNTIL DONE
670 032166 000207      RTS      PC      ;RETURN
671

```

```

672 032170      T29RT2:
673 032170      SAVREG      ;SAVE THE REGISTERS

```

```

674 032174 012701 026360      MOV      #T29PK2,R1      ;START OF THE PACKET
675 032200 012721 140006      MOV      #140006,(R1)+  ;WRITE SUBSYSTEM MEM. WITH ACK,CVC-1.
676 032204 012721 026400      MOV      #T298F2,(R1)+ ;ADDRESS OF DATA BLOCK
677 032210 005021              CLR      (R1)+          ;EXTENDED ADDRESS
678 032212 012721 000006      MOV      #6.,(R1)+     ;SIZE OF DATA BLOCK IN BYTES
679 032216 005021              CLR      (R1)+
680 032220 012701 026400      MOV      #T298F2,R1    ;POINT TO DATA SEL AREA
681 032224 005021              CLR      (R1)+
682 032226 005011              CLR      (R1)+
683 032230 000207              RTS      PC              ;RETURN
684 032232
685 032232      T29RT3: SAVREG      ;SAVE THE REGISTERS
686 032236 012701 026370      MOV      #T29PK3,R1    ;START OF THE PACKET
687 032242 012721 000000      MOV      #0,(R1)+     ;WRITE SUBSYSTEM MEM. WITH ACK.
688 032246 012721 000000      MOV      #0,(R1)+     ;ADDRESS OF DATA BLOCK
689 032252 005021              CLR      (R1)+          ;EXTENDED ADDRESS
690 032254 012711 000000      MOV      #0,(R1)      ;SIZE OF DATA BLOCK IN BYTES
691 032260 000207              RTS      PC              ;RETURN
692 032262
        ENDTST

```

L10036: TRAP C\$ETST

693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731

.SBTTL TEST 2: SKIP TAPE MARKS

;
;
;THIS TEST VERIFIES PROPER OPERATION OF THE SKIP TAPE MARKS
;FORWARD AND SKIP TAPE MARKS REVERSE COMMANDS. PROPER OPERATION
;UNDER CONTROL OF ALL COMBINATIONS OF THE ENABLE SKIP TAPE MARKS
;STOP (ESS) AND ENABLE TAPE MARKS STOP OFF BOT (ENB) BITS SPECIFIED
;BY THE WRITE CHARACTERISTICS COMMAND. THE TEST CONSISTS OF THE
;FOLLOWING SUBTESTS (FOR EACH SUBTEST, THE TAPE IS FIRST WRITTEN
;WITH AN APPROPRIATE SERIES OF DATA RECORDS AND/OR TAPE MARKS
;AND/OR DOUBLE TAPE MARKS.

;THE TEST CONSISTS OF THE FOLLOWING 11 SUBTESTS

BGNTST

```

T2::
MOV      #EPRT1,EPTSW      ;PRIMARY ERROR MESSAGE
MOV      #TST30ID,R0      ;ASCII MESSAGE TO IDENTIFY TEST
JSR      PC,TSTSETUP      ;DO INITIAL TEST SETUP
MOV      #5,LOOPCNT       ;PERFORM 5 ITERATIONS

```

;TEST 2, SUBTEST 1

;VERIFIES THAT A SKIP TAPE MARKS FORWARD COMMAND WITH
;A TAPE MARK COUNT OF 1 OPERATES PROPERLY. THE TAPE
;IS FIRST REWOUND, THEN WRITTEN WITH SEVERAL "FILES";
;EACH FILE CONSISTS OF A NUMBER OF DATA RECORDS
;FOLLOWED BY A TAPE MARK. THE FINAL FILE IS
;TERMINATED BY A DOUBLE TAPE MARK. EACH DATA RECORD
;CONTAINS A FILE NUMBER AND THE RECORD NUMBER WITHIN

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 2: SKIP TAPE MARKS

SEQ 0117

```

781 032446 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
782 032450          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      032450 104456          TRAP      C$ERHRD
      032452 000312          .WORD    202
      032454 005052          .WORD    WRTMSG
      032456 012114          .WORD    SFIMSG
783 032460          23$:   CKLOOP          ;LOOP IF SELECTED
      032460 104406          TRAP      C$CLP1
784
785          ;*****
786          ;
787          ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
788          ;
789          ;*****
790
791 032462 004737 011074          JSR      PC,REWIND          ;CALL TAPE REWIND COMMAND
792 032466 103411          BCS     30$                ;BP, IF NO PROBLEM
793 032470 010004          MOV     R0,R4              ;GET PACKET ADDRESS
794 032472 016501 000002          MOV     TSSR(R5),R1        ;GET STATUS REGISTER
795 032476 005237 002214          INC     FATFLG              ;ERROR COUNT
799 032502          ERRHRD  ERRNO,T3ORWN,PKTSSR ;REWIND NOT ACCEPTED
      032502 104456          TRAP      C$ERHRD
      032504 000313          .WORD    203
      032506 040170          .WORD    T3ORWN
      032510 012126          .WORD    PKTSSR
800 032512          30$:   CKLOOP          ;LOOP IF SELECTED
      032512 104406          TRAP      C$CLP1
801
802          ;*****
803          ;
804          ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
805          ;
806          ;*****
807
808 032514 013701 036460          MOV     T30BFR+6,R1        ;PICK UP XSTO
809 032520 010102          MOV     R1,R2              ;SET UP EXPECTED
810 032522 052702 000002          BIS     @BIT1,R2           ;SET BOT BIT IN EXPECTED
811 032526 020102          CMP     R1,R2              ;DOES EXP = REC'D
812 032530 001406          BEQ     40$                ;BR, IF EQUAL (OK)
813 032532 005237 002214          INC     FATFLG              ;ERROR COUNT
817 032536          ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      032536 104456          TRAP      C$ERHRD
      032540 000314          .WORD    204
      032542 037771          .WORD    T30BOT
      032544 015554          .WORD    EXPREC
818 032546          40$:   CKLOOP          ;LOOP IF SELECTED
      032546 104406          TRAP      C$CLP1
819 032550 012737 000001 036604          MOV     @1.,T30FCN         ;SET "FILE" COUNTER AT 1 DECIMAL
820 032556 012703 000001          64$:   MOV     @1,R3         ;ONE RECORD PER "FILE"
821 032562 013737 003116 036552          65$:   MOV     FREE,T30WB    ;SET UP PACKETS'S WRITE BUFFER
822 032570 012737 003720 036556          MOV     @2000.,T30SZ      ;SET RECORD SIZE AT 2000 BYTES
823
824          ;*****
825          ;
826          ;WRITE DATA,ACK,CVC=1 COMMAND
827          ;
828          ;*****
    
```

```

829
830 032576 012737 140005 036550      MOV      #140005,T30PK3      ;WRITE DATA,ACK,CVC=1 COMMAND
831 032604 012704 036550              MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
832 032610 013702 036604              MOV      T30FCN,R2        ;GET FILE COUNTER
833 032614 000302                      SWAB     R2                ;MOVE TO UPPER BYTE
834 032616 010301                      MOV      R3,R1            ;GET RECORD COUNTER
835 032620 060201                      ADD      R2,R1            ;FILE COUNTER IN UPPER, RECORD # LOW
836 032622 010177 150270              MOV      R1,BFREE        ;MOV TO OUT PUT BUFFER
837 032626 010465 000000              MOV      R4,TSD8(R5)     ;ISSUE COMMAND
838 032632 004737 016330              JSR      PC,WAITF        ;WAIT FOR SSR TO SET
839 032636 016501 000002              MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
840 032642 012702 000200              MOV      #SSR,R2        ;SET UP EXPECTED
841 032646 020102                      CMP      R1,R2           ;ARE THEY EQUAL
842 032650 001406                      BEQ      70#             ;BR, IF OK
843 032652 005237 002214              INC      FATFLG          ;ERROR COUNT
847 032656                      ERRHRD  ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C#ERHRD
                                .WORD    205
                                .WORD    T30WDD
                                .WORD    PKTSSR
                                032656 104456
                                032660 000315
                                032662 037120
                                032664 012126
848 032666                      70# :  CKLOOP            ;LOOP IF SELECTED
                                TRAP      C#CLP1
849 032670 104406                      INC      R3              ;COUNT THE RECORD COUNTER DOWN
850 032672 005203                      CMP      R3,#21         ;AT 20 YET
851 032676 020327 000021              BNE     65#             ;BR, IF NOT AT 20 RECORDS WRITTEN
852
853 ;*****
854 ;
855 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
856 ;
857 ;*****
858
859 032700 012737 141011 036550      MOV      #141011,T30PK3  ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
860 032706 012704 036550              MOV      #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
861 032712 010465 000000              MOV      R4,TSD8(R5)    ;ISSUE COMMAND
862 032716 004737 016330              JSR      PC,WAITF        ;WAIT FOR SSR TO SET
863 032722 016501 000002              MOV      TSSR(R5),R1    ;PICK UP TSSR
864 032726 012702 000200              MOV      #SSR,R2        ;SET UP EXPECTED (SSR ONLY)
865 032732 020102                      CMP      R1,R2           ;WAS STATUS GOOD
866 032734 001406                      BEQ      160#           ;BR, IF TERMINATION WAS GOOD
867 032736 005237 002214              INC      FATFLG          ;ERROR COUNT
871 032742                      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP      C#ERHRD
                                .WORD    206
                                .WORD    T30WDC
                                .WORD    PKTSSR
                                032742 104456
                                032744 000316
                                032746 040312
                                032750 012126
872 032752                      160# : CKLOOP            ;LOOP IF SELECTED
                                TRAP      C#CLP1
873 032754 005237 036604              INC      T30FCN         ;COUNT THE "FILE" COUNTER DOWN
874 032760 023727 036604 000006      CMP      T30FCN,#6     ;WRITE 5 FILE TO TAPE
875 032766 001273                      BNE     64#             ;BR, IF NOT AT 5 FILES WRITTEN
876
877 ;*****
878 ;
879 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
880 ;
881 ;*****

```



```

882
883 032770 012737 141011 036550      MOV      #141011,T30PK3      ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
884 032776 012704 036550              MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
885 033002 010465 000000              MOV      R4,TSD8(R5)       ;ISSUE COMMAND
886 033006 004737 016330              JSR      PC,WAITF          ;WAIT FOR SSR TO SET
887 033012 016501 000002              MOV      TSSR(R5),R1      ;PICK UP TSSR
888 033016 012702 000200              MOV      #SSR,R2         ;SET UP EXPECTED (SSR ONLY)
889 033022 020102                      CMP      R1,R2            ;WAS STATUS GOOD
890 033024 001406                      BEQ      165#             ;BR, IF TERMINATION WAS GOOD
891 033026 005237 002214              INC      FATFLG           ;ERROR COUNT
895 033032                      ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP      C#ERRRD
                                .WORD    207
                                .WORD    T30WDC
                                .WORD    PKTSSR
                                033032 104456
                                033034 000317
                                033036 040312
                                033040 012126
896 033042                      165# :  CKLOOP           ;LOOP IF SELECIED
                                TRAP      C#CLP1
                                033042 104406
897
898 ;*****
899 ;
900 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
901 ;
902 ;*****
903
904 033044 004737 011074              JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
905 033050 103411                      BCS     170#             ;BR, IF NO PROBLEM
906 033052 010004                      MOV      R3,R4           ;GET PACKET ADDRESS
907 033054 016501 000002              MOV      TSSR(R5),R1      ;GET STATUS REGISTER
908 033060 005237 002214              INC      FATFLG           ;ERROR COUNT
912 033064                      ERRHRD  ERRNO,T30RWV,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C#ERRRD
                                .WORD    208
                                .WORD    T30RWV
                                .WORD    PKTSSR
                                033064 104456
                                033066 000320
                                033070 040170
                                033072 012126
913 033074                      170# :  CKLOOP           ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                033074 104406
914
915 ;*****
916 ;
917 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
918 ;
919 ;*****
920
921 033076 013701 036460              MOV      T30BFR+6,R1     ;PICK UP XSTO
922 033102 010102                      MOV      R1,R2           ;SET UP EXPECTED
923 033104 052702 000002              BIS      #BIT1,R2        ;SET BOT BIT IN EXPECTED
924 033110 020102                      CMP      R1,R2           ;DOES EXP = REC'D
925 033112 001406                      BEQ      180#             ;BR, IF EQUAL (OK)
926 033114 005237 002214              INC      FATFLG           ;ERROR COUNT
930 033120                      ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C#ERRRD
                                .WORD    209
                                .WORD    T30BOT
                                .WORD    EXPREC
                                033120 104456
                                033122 000321
                                033124 037771
                                033126 015554
931 033130                      180# :  CKLOOP           ;LOOP IF SELECTED
                                TRAP      C#CLP1
                                033130 104406
932 033132 012703 036566              MOV      #T30IMV,R3      ;SET UP POINTER TO COMMAND TABLE

```

```

933 033136 013737 002174 036450      MOV      UNITN,T300SW      ;SET UP UNIT NUMBER
934 033144 011337 036446      1824:   MOV      (R3),T30ETM  ;GET NEXT COMMAND
935 033150 012704 036430      MOV      @T30PACKET,R4   ;SUBROUTINE NEEDS PACKET ADDRESS
936
937      ;*****
938      ;
939      ;ISSUE WRITE CHARACTERISTICS COMMAND
940      ;
941      ;*****
942
943 033154 004737 010742      JSR      PC,WHATCHR      ;ISSUE WRITE CHARACTERISTICS
944 033160 103407      BCS      1884            ;BR, IF COMMAND ISSUED OK
945 033162 005237 002214      INC      FATFLG          ;ERROR COUNT
949 033166 010001      MOV      R0,R1          ;SAVE CONTENTS OF TSSR
950 033170      ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      033170 104456      TRAP     C#ERRHD
      033172 000322      .WORD   210
      033174 005052      .WORD   WRTMSG
      033176 012114      .WORD   SFIMSG
951 033200      1884:   CKLOOP          ;LOOP IF SELECTED
      033200 104406      TRAP     C#CLP1
952
953      ;*****
954      ;
955      ;SKIP TAPE MARK,ACK,CVC-1 COMMAND
956      ;
957      ;*****
958
959 033202 012737 141010 036550      MOV      @141010,T30PK3  ;SKIP TAPE MARK,ACK,CVC-1 COMMAND
960 033210 012737 000001 036552      MOV      @1,T30RB       ;SET UP NUMBER TO SKIP
961 033216 012704 036550      MOV      @T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
962 033222 010465 000000      1894:   MOV      R4,TSD8(R5)  ;ISSUE COMMAND
963 033226 012737 176750 036606      MOV      @65000.,T30DLY ;SET UP DELAY COUNTER
964 033234 004737 016330      1904:   JSR      PC,WAITF      ;WAIT FOR SSR TO SET
965 033240 016501 000002      MOV      TSSR(R5),R1    ;PICK UP TSSR
966 033244 032701 000200      BIT      @SSR,R1        ;IS SSR SET YET
967 033250 001017      BNE     1914            ;BR, IF SSR IS SET
968 033252      DELAY   250          ;CALL DELAY ROUTINE
      033252 012727 000250      MOV      @250,(PC)+
      033256 000000      .WORD   0
      033260 013727 002116      MOV      L#DLY,(PC)+
      033264 000000      .WORD   0
      033266 005367 177772      DEC     -6(PC)
      033272 001375      BNE     .4
      033274 005367 177756      DEC     -22(PC)
      033300 001367      BNE     .-20
969 033302 005337 036606      DEC     T30DLY          ;BUMP DELAY ROUTINE
970 033306 001352      BNE     1904            ;BR, IF MORE DELAY TO GO
971 033310 012702 000200      1914:   MOV      @SSR,R2        ;SET UP EXPECTED (SSR ONLY)
972 033314 020102      CMP     R1,R2          ;WAS STATUS GOOD
973 033316 001406      BEQ     1924            ;BR, IF TERMINATION WAS GOOD
974 033320 005237 002214      INC     FATFLG          ;ERROR COUNT
978 033324      ERRHRD  ERRNO,T30SKM,PKTSSR ;TSSR NOT CORRECT AFTER SKIP TAPE M.
      033324 104456      TRAP     C#EPHRD
      033326 000323      .WORD   211
      033330 037044      .WORD   T30SKM
      033332 012126      .WORD   PKTSSR
    
```

E10

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14-JUN-84 16:41
 TEST 2: SKIP TAPE MARKS

SEQ 0121

```

979 033334      192:  CKLOOP                ;LOOP IF SELECTED
      033334 104406                                TRAP  C#CLP1
980
981 ;*****
982 ;
983 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
984 ;
985 ;*****
986
987 033336 013701 036460      MOV      T30BFR+6,R1      ;PICK UP XSTO
988 033342 010102            MOV      R1,R2           ;SET UP EXPECTED
989 033344 052702 100000     BIS      @BIT15,R2       ;SET TMK BIT IN EXPECTED
990 033350 020102            CMP      R1,R2           ;DOES EXP = REC'D
991 033352 001406            BEQ      195:           ;BR, IF EQUAL (OK)
992 033354 005237 002214     INC      FATFLG          ;ERROR COUNT
996 033360            ERRHRD  ERRNO,T30TMK,EXPREC ;TMK NOT SET AFTER WRT TAPE MARK
      033360 104456                                TRAP  C#ERHRD
      033362 000324                                .WORD 212
      033364 040444                                .WORD T30TMK
      033366 015554                                .WORD EXPREC
997 033370      195:  CKLOOP                ;LOOP IF SELECTED
      033370 104406                                TRAP  C#CLP1
998 033372 012700 177777     MOV      @177777,R0      ;VALUE TO WRITTEN TO MEMORY
999 033376 004737 017502     JSR      PC,FILLMEM      ;FILL MEM WITH ALL ONES
1000 033402 013737 003116 036552  MOV      FREE,T30RB      ;STARTING READ BUFFER ADDRESS
1001
1002 ;*****
1003 ;
1004 ;READ FORWARD,ACK,CVC=1 COMMAND
1005 ;
1006 ;*****
1007
1008 033410 012737 140001 036550  MOV      @140001,T30PK3  ;READ FORWARD,ACK,CVC=1 COMMAND
1009 033416 012704 036550      MOV      @T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1010 033422 012737 003720 036556  MOV      @2000,T30SZ     ;SET UP RECORD SIZE IN PACKET
1011 033430 010465 000000      MOV      R4,T30B(R5)    ;ISSUE COMMAND
1012 033434 004737 016330      JSR      PC,WAIT        ;WAIT FOR SSR TO SET
1013 033440 016501 000002      MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
1014 033444 012702 000200      MOV      @SSR,R2        ;SET UP EXPECTED
1015 033450 020102            CMP      R1,R2          ;ARE THEY EQUAL
1016 033452 001406            BEQ      200:           ;BR, IF OK
1017 033454 005237 002214     INC      FATFLG          ;ERROR COUNT
1021 033460            ERRHRD  ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      033460 104456                                TRAP  C#ERHRD
      033462 000325                                .WORD 213
      033464 037343                                .WORD T30RDF
      033466 012126                                .WORD PKTSSR
1022 033470      200:  CKLOOP                ;LOOP IF SELECTED
      033470 104406                                TRAP  C#CLP1
1023 033472 017701 147420     MOV      @FREE,R1        ;FIRST LOC IN READ BUFFER
1024 033476 012702 177777     MOV      @177777,R2     ;EXPECTED IF NO DATA TRANS.
1025 033502 020102            CMP      R1,R2          ;DID ANY DATA GET TRANSFERRED
1026 033504 001006            BNE      220:           ;BR, IF NO DATA TRANS (GOOD)
1027 033506 005237 002214     INC      FATFLG          ;ERROR COUNT
1031 033512            ERRHRD  ERRNO,T30DTR,EXPREC ;DATA TRANSFERRED ON READ TAPE MARK
      033512 104456                                TRAP  C#ERHRD
      033514 000326                                .WORD 214
  
```

```

033516 041020 .WORD T3OCTR
033520 015554 .WORD EXPREC
1032 033522 220: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
033522 104406 ;SET UP RECORD NUMBER EXPECTED (FILE 2)
1033 033524 012702 001001 MOV #1001,R2 ;GET INFO FROM BUFFER
1034 033530 017701 147362 MOV #FREE,R1 ;ARE THEY EQUAL
1035 033534 020201 CMP R2,R1 ;BR, IF EQUAL (OK)
1036 033536 001406 BEQ 228: ;ERROR COUNT
1037 033540 005237 002214 INC FATFLG ;RECORD POSITION WAS NOT CORRECT
1041 033544 ERRHRD ERRNO,T3OPTB,EXPREC TRAP C$ERHRD
033544 104456 .WORD 215
033546 000327 .WORD T3OPTB
033550 037172 .WORD EXPREC
033552 015554
1042 033554 228: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
033554 104406
1043 ;*****
1044 ;
1045 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1046 ;
1047 ;*****
1048 ;
1049 JSR PC,REWIND ;CALL TAPE REWIND COMMAND
1050 033556 004737 011074 BCS 230: ;BR, IF NO PROBLEM
1051 033562 103411 MOV R0,R4 ;SAVE PACKET ADDRESS
1052 033564 010004 MOV TSSR(R5),R1 ;GET TSSR STATUS
1053 033566 016501 000002 INC FATFLG ;ERROR COUNT
1054 033572 005237 002214 ERRHRD ERRNO,T3ORWN,PKTSSR ;REWIND NOT ACCEPTED
1058 033576 104456 TRAP C$ERHRD
033600 000330 .WORD 216
033602 040170 .WORD T3ORWN
033604 012126 .WORD PKTSSR
1059 033606 230: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
033606 104406
1060 ;*****
1061 ;
1062 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1063 ;
1064 ;*****
1065 ;
1066 MOV T3OBF+6,R1 ;FICK UP XSTO
1067 033610 013701 036460 MOV R1,R2 ;SET UP EXPECTED
1068 033614 010102 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
1069 033616 052702 000002 CMP R1,R2 ;DOES EXP = REC'D
1070 033622 020102 BEQ 240: ;BR, IF EQUAL (OK)
1071 033624 001406 INC FATFLG ;ERROR COUNT
1072 033626 005237 002214 ERRHRD ERRNO,T3OBOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
1076 033632 104456 TRAP C$ERHRD
033632 000331 .WORD 217
033634 037771 .WORD T3OBOT
033636 015554 .WORD EXPREC
1077 033642 240: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
033642 104406
1078 033644 005723 TST (R3)+ ;POINT TO NEXT POSITION
1079 033646 011301 MOV (R3),R1 ;GET NEXT COMMAND ETC.
    
```



```

1123 ;*****
1124 ;
1125 ;ISSUE WRITE CHARACTERISTICS COMMAND
1126 ;
1127 ;*****
1128
1129 034026 004737 010742      JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
1130 034032 103407              BCS      23$           ;BR, IF COMMAND ISSUED OK
1131 034034 005237 002214      INC      FATFLG        ;ERROR COUNT
1135 034040 010001              MOV      R0,R1         ;SAVE CONTENTS OF TSSR
1136 034042              ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICS FAILED
                                TRAP      C$ERHRD
                                .WORD    219
                                .WORD    WRTMSG
                                .WORD    SFIMSG
1137 034052 23$: CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                034052 104406
1138 ;*****
1139 ;
1140 ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1141 ;
1142 ;*****
1143 ;
1144
1145 034054 004737 011074      JSR      PC,REWIND     ;CALL TAPE REWIND COMMAND
1146 034060 103411              BCS      30$           ;BR, IF NO PROBLEM
1147 034062 010004              MOV      R0,R4         ;GET PACKET ADDRESS
1148 034064 016501 000002      MOV      TSSR(R5),R1   ;GET STATUS REGISTER
1149 034070 005237 002214      INC      FATFLG        ;ERROR COUNT
1153 034074              ERRHRD  ERRNO,T3ORLN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    220
                                .WORD    T3ORLN
                                .WORD    PKTSSR
1154 034104 30$: CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                034104 104406
1155 ;*****
1156 ;
1157 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1158 ;
1159 ;*****
1160 ;
1161
1162 034106 013701 036460      MOV      T30BFR+6,R1   ;PICK UP XSTO
1163 034112 010102              MOV      R1,R2         ;SET UP EXPECTED
1164 034114 052702 000002      BIS      @BIT1,R2      ;SET BOT BIT IN EXPECTED
1165 034120 020102              CMP      R1,R2         ;DOES EXP = REC'D
1166 034122 001406              BEQ      40$           ;BR, IF EQUAL (OK)
1167 034124 005237 002214      INC      FATFLG        ;ERROR COUNT
1171 034130              ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    221
                                .WORD    T30BOT
                                .WORD    EXPREC
1172 034140 40$: CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                034140 104406
1173 034142 012737 000001 036604  MOV      @1.,T30FCN    ;SET "FILE" COUNTER AT 1 DECIMAL
    
```

```

1174 034150 012703 000001      64:  MOV    #1,R3          ;ONE RECORD PER "FILE"
1175 034154 013737 003116 036552 65:  MOV    FREE,T30WB       ;SET UP PACKETS'S WRITE BUFFER
1176 034162 012737 000024 036556      MOV    #20.,T30SZ      ;SET RECORD SIZE AT 2000 BYTES
1177
1178 ;*****
1179 ;
1180 ;WRITE DATA,ACK,CVC=1 COMMAND
1181 ;
1182 ;*****
1183
1184 034170 012737 140005 036550      MOV    #140005,T30PK3   ;WRITE DATA,ACK,CVC=1 COMMAND
1185 034176 012704 036550      MOV    #T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1186 034202 013702 036604      MOV    T30FCN,R2       ;GET FILE COUNTER
1187 034206 000302      SWAB   R2              ;MOVE TO UPPER BYTE
1188 034210 010301      MOV    R3,R1          ;GET RECORD COUNTER
1189 034212 060201      ADD    R2,R1          ;FILE COUNTER IN UPPER, RECORD # LOW
1190 034214 010177 146676      MOV    R1,#FREE        ;MOV TO OUT PUT BUFFER
1191 034220 010465 000000      MOV    R4,TSD8(R5)     ;ISSUE COMMAND
1192 034224 004737 016330      JSR   PC,WAITF        ;WAIT FOR SSR TO SET
1193 034230 016501 000002      MOV    TSSR(R5),R1     ;GET TSSR CONTENTS
1194 034234 012702 000200      MOV    #SSR,R2        ;SET UP EXPECTED
1195 034240 020102      CMP   R1,R2          ;ARE THEY EQUAL
1196 034242 001406      BEQ   70$            ;BR, IF OK
1197 034244 005237 002214      INC   FATFLG         ;ERROR COUNT
1201 034250      ERRHRD ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      034250 104456      TRAP  C$ERHRD
      034252 000336      .WORD 222
      034254 037120      .WORD T30WDD
      034256 012126      .WORD PKTSSR
1202 034260      70$:  CKLOOP          ;LOOP IF SELECTED
      034260 104406      TRAP  C$CLP1
1203 034262 005203      INC   R3            ;COUNT THE RECORD COUNTER DOWN
1204 034264 020327 000021      CMP   R3,#21        ;AT 20 YET
1205 034270 001331      BNE  65$            ;BR, IF NOT AT 20 RECORDS WRITTEN
1206
1207 ;*****
1208 ;
1209 ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1210 ;
1211 ;*****
1212
1213 034272 012737 141011 036550      MOV    #141011,T30PK3   ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1214 034300 012704 036550      MOV    #T30PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
1215 034304 010465 000000      MOV    R4,TSD8(R5)     ;ISSUE COMMAND
1216 034310 004737 016330      JSR   PC,WAITF        ;WAIT FOR SSR TO SET
1217 034314 016501 000002      MOV    TSSR(R5),R1     ;PICK UP TSSR
1218 034320 012702 000200      MOV    #SSR,R2        ;SET UP EXPECTED (SSR ONLY)
1219 034324 020102      CMP   R1,R2          ;WAS STATUS GOOD
1220 034326 001406      BEQ   160$          ;BR, IF TERMINATION WAS GOOD
1221 034330 005237 002214      INC   FATFLG         ;ERROR COUNT
1225 034334      ERRHRD ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
      034334 104456      TRAP  C$ERHRD
      034336 000337      .WORD 223
      034340 040312      .WORD T30WDC
      034342 012126      .WORD PKTSSR
1226 034344      160$: CKLOOP          ;LOOP IF SELECTED
      034344 104406      TRAP  C$CLP1
    
```

```

1227 034346 005237 036604          INC      T30FCN          ;COUNT THE "FILE" COUNTER DOWN
1228 034352 023727 036604 000031  CMP      T30FCN,#25.      ;WRITE 25 FILES TO TAPE
1229 034360 001273                BNE      64$              ;BR, IF NOT AT 25 FILES WRITTEN
1230
1231          ;*****
1232          ;
1233          ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1234          ;
1235          ;*****
1236
1237 034362 012737 141011 036550      MOV      #141011,T30PK3    ;WRITE TAPE MARK,ACK,CVC=1 COMMAND
1238 034370 012704 036550                MOV      #T30PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
1239 034374 010465 000000                MOV      R4,TSD8(R5)       ;ISSUE COMMAND
1240 034400 004737 016330                JSR      PC,WAITF          ;WAIT FOR SSR TO SET
1241 034404 016501 000002                MOV      TSSR(R5),R1       ;PICK UP TSSR
1242 034410 012702 000200                MOV      #SSR,R2          ;SET UP EXPECTED (SSR ONLY)
1243 034414 020102                CMP      R1,R2             ;WAS STATUS GOOD
1244 034416 001406                BEQ      165$              ;BR, IF TERMINATION WAS GOOD
1245 034420 005237 002214                INC      FATFLG            ;ERROR COUNT
1249 034424                ERRHRD  ERRNO,T30WDC,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
                                TRAP      C$ERHRD
                                .WORD    224
                                .WORD    T30WDC
                                .WORD    PKTSSR
1250 034434                165$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
1251
1252          ;*****
1253          ;
1254          ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1255          ;
1256          ;*****
1257
1258 034436 004737 011074                JSR      PC,REWIND         ;CALL TAPE REWIND COMMAND
1259 034442 103411                BCS      170$              ;BR, IF NO PROBLEM
1260 034444 010004                MOV      R0,R4             ;GET PACKET ADDRESS
1261 034446 016501 000002                MOV      TSSR(R5),R1       ;GET STATUS REGISTER
1262 034452 005237 002214                INC      FATFLG            ;ERROR COUNT
1266 034456                ERRHRD  ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    225
                                .WORD    T30RWN
                                .WORD    PKTSSR
1267 034466                170$:  CKLOOP              ;LOOP IF SELECTED
                                TRAP      C$CLP1
1268
1269          ;*****
1270          ;
1271          ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1272          ;
1273          ;*****
1274
1275 034470 013701 036460                MOV      T30BFR+6,R1       ;PICK UP XSTO
1276 034474 010102                MOV      R1,R2             ;SET UP EXPECTED
1277 034476 052702 000002                BIS      #BIT1,R2          ;SET BOT BIT IN EXPECTED
1278 034502 020102                CMP      R1,R2             ;DOES EXP = REC'D
1279 034504 001406                BEQ      180$              ;BR, IF EQUAL (OK)
    
```



```

1280 034506 005237 002214          INC      FATFLG          ;ERROR COUNT
1284 034512          ERRHRD  ERRNO,T30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      034512 104456          TRAP      C#ERHRD
      034514 000342          .WORD    226
      034516 037771          .WORD    T30BOT
      034520 015554          .WORD    EXPREC
1285 034522          180$:  CKLOOP          ;LOOP IF SELECTED
      034522 104406          TRAP      C#CLP1
1286 034524 012737 000002 036604  MOV      #2,T30FCN      ;SET TO NUMBER OF SKIP "FILES"
1287 034532 012703 036566          MOV      #T30IMV,R3    ;SET UP POINTER TO COMMAND TABLE
1288 034536 013737 002174 036450  MOV      UNITN,T30DSW  ;SET UP UNIT NUMBER
1289 034544 011337 036446 182$:  MOV      (R3),T30ETM ;GET NEXT COMMAND
1290 034550 012704 036430  MOV      #T30PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
1291
1292          ;*****
1293          ;
1294          ;ISSUE WRITE CHARACTERISTICS COMMAND
1295          ;
1296          ;*****
1297
1298 034554 004737 010742          JSR      PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
1299 034560 103407          BCS      188$          ;BR, IF COMMAND ISSUED OK
1300 034562 005237 002214          INC      FATFLG          ;ERROR COUNT
1304 034566 010001          MOV      R0,R1          ;SAVE CONTENTS OF TSSR
1305 034570          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
      034570 104456          TRAP      C#ERHRD
      034572 000343          .WORD    227
      034574 005052          .WORD    WRTMSG
      034576 012114          .WORD    SFIMSG
1306 034600          188$:  CKLOOP          ;LOOP IF SELECTED
      034600 104406          TRAP      C#CLP1
1307
1308          ;*****
1309          ;
1310          ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
1311          ;
1312          ;*****
1313
1314 034602 012737 141010 036550  MOV      #141010,T30PK3 ;SKIP TAPE MARK,ACK,CVC=1 COMMAND
1315 034610 013737 036604 036552  MOV      T30FCN,T30RB  ;SET UP NUMBER TO SKIP
1316 034616 012704 036550          MOV      #T30PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
1317 034622 010465 000000 189$:  MOV      R4,T30DB(R5) ;ISSUE COMMAND
1318 034626 012737 176750 036606  MOV      #65000.,T30DLY ;SET UP DELAY COUNTER
1319 034634 004737 016330 190$:  JSR      PC,WAITF    ;WAIT FOR SSR TO SET
1320 034640 016501 000002  MOV      TSSR(R5),R1   ;PICK UP TSSR
1321 034644 032701 000200  BIT      #SSR,R1       ;IS SSR SET YET
1322 034650 001017          BNE      191$          ;BR, IF SSR IS SET
1323 034652          DELAY  250          ;CALL DELAY ROUTINE
      034652 012727 000250          MOV      #250,(PC)+
      034656 000000          .WORD    0
      034660 013727 002116          MOV      L#DLY,(PC)+
      034664 000000          .WORD    0
      034666 005367 177772          DEC      -6(PC)
      034672 001375          BNE      .-4
      034674 005367 177756          DEC      -22(PC)
      034700 001367          BNE      . 20
1324 034702 005337 036606          DEC      T30DLY        ;BUMP DELAY ROUTINE

```

```

1325 034706 001352          BNE      190$          ;BR, IF MORE DELAY TO GO
1326 034710 012702 000200 191$:  MOV     @SSR,R2      ;SET UP EXPECTED (SSR ONLY)
1327 034714 020102          CMP     R1,R2         ;WAS STATUS GOOD
1328 034716 001406          BEQ     192$          ;BR, IF TERMINATION WAS GOOD
1329 034720 005237 002214          INC     FATFLG        ;ERROR COUNT
1333 034724          ERRHRD  ERRNO,T30SKM,PKTSSR ;TSSR NOT CORRECT AFTER SKIP TAPE M.
                                TRAP    C$ERHRD
                                .WORD   228
                                .WORD   T30SKM
                                .WORD   PKTSSR
1334 034734          192$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
1335
1336 ;*****
1337 ;
1338 ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1339 ;
1340 ;*****
1341
1342 034736 013701 036460          MOV     T30BFR+6,R1   ;PICK UP XSTO
1343 034742 010102          MOV     R1,R2         ;SET UP EXPECTED
1344 034744 052702 100000          BIS     @BIT15,R2     ;SET TMK BIT IN EXPECTED
1345 034750 020102          CMP     R1,R2         ;DOES EXP = REC'D
1346 034752 001406          BEQ     195$          ;BR, IF EQUAL (OK)
1347 034754 005237 002214          INC     FATFLG        ;ERROR COUNT
1351 034760          ERRHRD  ERRNO,T30TMK,EXPREC ;TMK NOT SET AFTER WRT TAPE MARK
                                TRAP    C$ERHRD
                                .WORD   229
                                .WORD   T30TMK
                                .WORD   EXPREC
1352 034770          195$:  CKLOOP          ;LOOP IF SELECTED
                                TRAP    C$CLP1
1353 034772 012700 177777          MOV     @177777,R0    ;VALUE TO WRITTEN TO MEMORY
1354 034776 004737 017502          JSR     PC,FILLMEM    ;FILL MEM WITH ALL ONES
1355 035002 013737 003116 036552          MOV     FREE,T30RB    ;STARTING READ BUFFER ADDRESS
1356
1357 ;*****
1358 ;
1359 ;READ FORWARD,ACK,CVC=1 COMMAND
1360 ;
1361 ;*****
1362
1363 035010 012737 140001 036550          MOV     @140001,T30PK3 ;READ FORWARD,ACK,CVC=1 COMMAND
1364 035016 012704 036550          MOV     @T30PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
1365 035022 012737 000024 036556          MOV     @20.,T30SZ    ;SET UP RECORD SIZE IN PACKET
1366 035030 010465 000000          MOV     R4,T30DB(R5)  ;ISSUE COMMAND
1367 035034 004737 016330          JSR     PC,WAITF      ;WAIT FOR SSR TO SET
1368 035040 016501 000002          MOV     TSSR(R5),R1   ;GET TSSR CONTENTS
1369 035044 012702 000200          MOV     @SSR,R2       ;SET UP EXPECTED
1370 035050 020102          CMP     R1,R2         ;ARE THEY EQUAL
1371 035052 001406          BEQ     200$          ;BR, IF OK
1372 035054 005237 002214          INC     FATFLG        ;ERROR COUNT
1376 035060          ERRHRD  ERRNO,T30RDF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP    C$ERHRD
                                .WORD   230
                                .WORD   T30RDF
                                .WORD   PKTSSR
1376 035060 104456
1376 035062 000346
1376 035064 037343
1376 035066 012126
    
```

```

1377 035070          2004:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      035070 104406
1378 035072 017701 146020      MOV      $FREE,R1          ;FIRST LOC IN READ BUFFER
1379 035076 012702 177777      MOV      #177777,R2       ;EXPECTED IF NO DATA TRANS.
1380 035102 020102          CMP      R1,R2           ;DID ANY DATA GET TRANSFERRED
1381 035104 001006          BNE     220$            ;BR. IF NO DATA TRANS (GOOD)
1382 035106 005237 002214      INC     FATFLG          ;ERROR COUNT
1386 035112          ERRHRD  ERRNO,T30DTR,EXPREC ;DATA TRANSFERRED ON READ TAPE MARK
      035112 104456          TRAP  C$ERHRD
      035114 000347          .WORD 231
      035116 041020          .WORD T30DTR
      035120 015554          .WORD EXPREC

1387 035122          2204:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      035122 104406
1388 035124 013702 036604      MOV     T30FCN,R2       ;GET NUMBER OF SKIPS
1389 035130 005202          INC     R2              ;SET TO CORRECT FILE VALUE
1390 035132 000302          SWAB   R2              ;SWAP BYTE HALVES
1391 035134 052702 000001      BIS    #BIT0,R2        ;SET FOR RECORD #1
1392 035140 017701 145752      MOV     $FREE,R1       ;GET INFO FROM BUFFER
1393 035144 020201          CMP    R2,R1           ;ARE THEY EQUAL
1394 035146 001406          BEQ   228$            ;BR. IF EQUAL (OK)
1395 035150 005237 002214      INC     FATFLG          ;ERROR COUNT
1399 035154          ERRHRD  ERRNO,T30PTB,EXPREC ;RECORD POSITION WAS NOT CORRECT
      035154 104456          TRAP  C$ERHRD
      035156 000350          .WORD 232
      035160 037172          .WORD T30PTB
      035162 015554          .WORD EXPREC

1400 035164          2284:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      035164 104406

1401
1402
1403
1404
1405
1406
1407
      ;*****
      ;
      ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
      ;
      ;*****
1408 035166 004737 011074      JSR    PC,REWIND       ;CALL TAPE REWIND COMMAND
1409 035172 103411          BCS    230$            ;BR. IF NO PROBLEM
1410 035174 010004          MOV    R0,R4          ;SAVE PACKET ADDRESS
1411 035176 016501 000002      MOV    TSSR(R5),R1     ;GET TSSR STATUS
1412 035202 005237 002214      INC    FATFLG          ;ERROR COUNT
1416 035206          ERRHRD  ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
      035206 104456          TRAP  C$ERHRD
      035210 000351          .WORD 233
      035212 040170          .WORD T30RWN
      035214 012126          .WORD PKTSSR

1417 035216          2304:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      035216 104406

1418
1419
1420
1421
1422
1423
1424
      ;*****
      ;
      ;GET EXTENDED STATUS REGISTER ZERO (XST0) FROM MESSAGE BUFFER
      ;
      ;*****
1425 035220 013701 036460      MOV    T30BFR+6,R1     ;PICK UP XST0
1426 035224 010102          MOV    R1,R2          ;SET UP EXPECTED
  
```


TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN-84 16:41
 TEST 2: SKIP TAPE MARKS

SEQ 0131

```

035372 013727 002116                                MOV     L8DLY,(PC).
035376 000000                                .WORD  0
035400 005367 177772                                DEC     6(PC)
035404 001375                                BNE     .4
035406 005367 177756                                DEC     22(PC)
035412 001367                                BNE     .20
1475 035414 005337 036606                DEC     T30DLY                ;BUMP COUNTER
1476 035420 001356                BNE     101                    ;BR. IF MORE COUNTING TO DO
1477 035422 005237 002114                INC     FATFLG                ;ERROR COUNT
1481 035426 010001                MOV     R0,R1                ;CONTENTS OF TSSR REGISTER
1482 035430                ERROF  ERRNO,SFIERR,SFIMSG    ;FATAL ERROR TSSR WAS NOT OK
                                TRAP   C1EROF
                                .WORD  235
                                .WORD  SFIERR
                                .WORD  SFIMSG
035430 104455
035432 000353
035434 003646
035436 012114
1483 035440
1484 035440 013737 002174 036450 201:    MOV     UNITN,T30DSW          ;SET UP UNIT NUMBER
1485 035446 012704 036430                MOV     @T30PACKET,R4        ;SUBROUTINE NEEDS PACKET ADDRESS
1486
1487                ;*****
1488                ;
1489                ;ISSUE WRITE CHARACTERISTICS COMMAND
1490                ;
1491                ;*****
1492
1493 035452 004737 010742                JSR     PC,WRTCHK            ;ISSUE WRITE CHARACTERISTICS
1494 035456 103407                BCS     231                    ;BR. IF COMMAND ISSUED OK
1495 035460 005237 002214                INC     FATFLG                ;ERROR COUNT
1499 035464 010001                MOV     R0,R1                ;SAVE CONTENTS OF TSSR
1500 035466                ERHRD  ERRNO,WRTMSG,SFIMSG    ;WRITE CHARACTERISTIC FAILED
                                TRAP   C1ERHRD
                                .WORD  236
                                .WORD  WRTMSG
                                .WORD  SFIMSG
035466 104456
035470 000354
035472 005052
035474 012114
1501 035476                231:    CKLOOP                    ;LOOP IF SELECTED
035476 104406                                TRAP   C1CLP1
1502
1503                ;*****
1504                ;
1505                ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1506                ;
1507                ;*****
1508
1509 035500 004737 011074                JSR     PC,REWIND            ;CALL TAPE REWIND COMMAND
1510 035504 103411                BCS     301                    ;BR. IF NO PROBLEM
1511 035506 010004                MOV     R0,R4                ;GET PACKET ADDRESS
1512 035510 016501 000002                MOV     TSSR(R5),R1          ;GET STATUS REGISTER
1513 035514 005237 002214                INC     FATFLG                ;ERROR COUNT
1517 035520                ERHRD  ERRNO,T30RWN,PKTSSR    ;REWIND NOT ACCEPTED
                                TRAP   C1ERHRD
                                .WORD  237
                                .WORD  T30RWN
                                .WORD  PKTSSR
035520 104456
035522 000355
035524 040170
035526 012126
1518 035530                301:    CKLOOP                    ;LOOP IF SELECTED
035530 104406                                TRAP   C1CLP1
1519
1520                ;*****
    
```


TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 14-JUN-84 16:41
 TEST 2: SKIP TAPE MARKS

SEQ 0134

```

1616 036034 013737 002174 036450      MOV      UNITN,T30DSW      ;SET UP UNIT NUMBER
1617 036042 012704 036430      MOV      @T30PACKET,R4    ;SUBROUTINE NEEDS PACKET ADDRESS
1618
1619      ;*****
1620      ;
1621      ;ISSUE WRITE CHARACTERISTICS COMMAND
1622      ;
1623      ;*****
1624
1625 036046 004737 010742      JSR      PC,WRTCHR        ;ISSUE WRITE CHARACTERISTICS
1626 036052 103407                BCS      23$              ;BR, IF COMMAND ISSUED OK
1627 036054 005237 002214      INC      FATFLG           ;ERROR COUNT
1631 036060 010001                MOV      R0,R1            ;SAVE CONTENTS OF TSSR
1632 036062                ERRHRD  ERRNO,WRTMSG,SFMSG ;WRITE CHARACTERISTIC FAILED
                                TRAP      C$ERHRD
                                .WORD    242
                                .WORD    WRTMSG
                                .WORD    SFMSG
                                TRAP      C$CLP1
1633 036072 23$: CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
1634
1635      ;*****
1636      ;
1637      ;ISSUE A REWIND TO TAPE DRIVE AND WAIT FOR SSR TO SET
1638      ;
1639      ;*****
1640
1641 036074 004737 011074      JSR      PC,REWIND        ;CALL TAPE REWIND COMMAND
1642 036100 103411                BCS      30$              ;BR, IF NO PROBLEM
1643 036102 010004                MOV      R0,R4            ;GET PACKET ADDRESS
1644 036104 016501 000002      MOV      TSSR(R5),R1      ;GET STATUS REGISTER
1645 036110 005237 002214      INC      FATFLG           ;ERROR COUNT
1649 036114                ERRHRD  ERRNO,T30RWN,PKTSSR ;REWIND NOT ACCEPTED
                                TRAP      C$ERHRD
                                .WORD    243
                                .WORD    T30RWN
                                .WORD    PKTSSR
                                TRAP      C$CLP1
1650 036124 30$: CKLOOP                ;LOOP IF SELECTED
                                TRAP      C$CLP1
                                .WORD    104406
1651
1652      ;*****
1653      ;
1654      ;GET EXTENDED STATUS REGISTER ZERO (XSTO) FROM MESSAGE BUFFER
1655      ;
1656      ;*****
1657
1658 036126 013701 036460      MOV      T30BFR+6,R1     ;PICK UP XSTO
1659 036132 010102                MOV      R1,R2            ;SET UP EXPECTED
1660 036134 052702 000002      BIS      @BIT1,R2        ;SET BOT BIT IN EXPECTED
1661 036140 020102                CMP      R1,R2            ;DOES EXP = REC'D
1662 036142 001406                BEQ      40$              ;BR, IF EQUAL (OK)
1663 036144 005237 002214      INC      FATFLG           ;ERROR COUNT
1667 036150                ERRHRD  ERRNO,;30BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
                                TRAP      C$ERHRD
                                .WORD    244
                                .WORD    T30BOT
                                .WORD    EXPREC
                                TRAP      C$CLP1
1668 036152 104456
1669 036154 000364
1670 036154 037771
1671 036156 015554

```



```

1668 036160          40$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      036160 104406
1669 036162 013737 003116 036552      MOV  FREE,T30WB      ;SET UP GOOD WRITE BUFFER
1670 036170 012737 000400 036556      MOV  #256.,T30SZ    ;SET UP SIZE
1671
1672      ;*****
1673      ;
1674      ;WRITE DATA,ACK,CVC=1 COMMAND
1675      ;
1676      ;*****
1677
1678 036176 012737 140005 036550      MOV  #140005,T30PK3 ;WRITE DATA,ACK,CVC=1 COMMAND
1679 036204 012704 036550      MOV  #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
1680 036210 010465 000000      MOV  R4,TSDB(R5)    ;ISSUE COMMAND
1681 036214 004737 016330      JSR  PC,WAITF       ;WAIT FOR SSR TO SET
1682 036220 016501 000002      MOV  TSSR(R5),R1    ;GET TSSR CONTENTS
1683 036224 012702 000200      MOV  #SSR,R2        ;SET UP EXPECTED
1684 036230 020102      CMP  R1,R2          ;ARE THEY EQUAL
1685 036232 001406      BEQ  70$            ;BR. IF OK
1686 036234 005237 002214      INC  FATFLG         ;ERROR COUNT
1690 036240      ERRHRD  ERRNO,T30WDD,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      036240 104456          TRAP  C$ERHRD
      036242 000365          .WORD 245
      036244 037120          .WORD T30WDD
      036246 012126          .WORD PKTSSR
1691 036250          70$:  CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      036250 104406
1692
1693      ;*****
1694      ;
1695      ;SKIP TAPE MARK REVERSE,ACK,CVC=1 COMMAND
1696      ;
1697      ;*****
1698
1699 036252 012737 000001 036552      MOV  #1,T30WB       ;# OF TM TO SKIP
1700 036260 012737 141410 036550      MOV  #141410,T30PK3 ;SKIP TAPE MARK REVERSE,ACK,CVC=1 CMD
1701 036266 012704 036550      MOV  #T30PK3,R4     ;SET UP R4 WITH PACKET ADDRESS
1702 036272 010465 000000      MOV  R4,TSDB(R5)    ;ISSUE COMMAND
1703 036276 004737 016330      JSR  PC,WAITF       ;WAIT FOR SSR TO SET
1704 036302 016501 000002      MOV  TSSR(R5),R1    ;PICK UP TSSR
1705 036306 012702 100204      MOV  #SSR!BIT2!SC,R2 ;SET UP EXPECTED (SSR AND SC ONLY)
1706 036312 020102      CMP  R1,R2          ;WAS STATUS GOOD
1707 036314 001406      BEQ  160$           ;BR. IF TERMINATION WAS GOOD
1708 036316 005237 002214      INC  FATFLG         ;ERROR COUNT
1712 036322      ERRHRD  ERRNO,T30IBU,PKTSSR ;TSSR NOT CORRECT AFTER WRT TAPE M.
      036322 104456          TRAP  C$ERHRD
      036324 000366          .WORD 246
      036326 036610          .WORD T30IBU
      036330 012126          .WORD PKTSSR
1713 036332          160$: CKLOOP          ;LOOP IF SELECTED          TRAP  C$CLP1
      036332 104406
1714
1715      ;*****
1716      ;
1717      ;GET EXTENDED STATUS REGISTER ZERO (XST3) FROM MESSAGE BUFFER
1718      ;
1719      ;*****
    
```


TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN-84 16:41
 TEST 2: SKIP TAPE MARKS

SEQ 0137

1778	036550	100205			.WORD	100205		;REREAD COMMAND, IE AND ACK
1779	036552				T30RB:			
1780	036552	003116			T30WB:	.WORD	FREE	;ADDRESS OF WRITE BUFFER
1781	036554	000000				.WORD	0	
1782	036556	000000			T30SZ:	.WORD	0	;SIZE OF BUFFER (EXTENT)
1783						.EVEN		
1784					:			
1785					:			
1786					:			
1787	036560				T30BF2:			
1788	036560	010			T30BS0:	.BYTE	10	;BSELO AREA
1789	036561	200			T30BS1:	.BYTE	200	;BSEL1 AREA
1790	036562	000000			T30S2:	.WORD	0	;SEL 2 AREA
1791	036564	000000			T30S3:	.WORD	0	;DATA AREA
1792					:			
1793					:			
1794					.EVEN			
1795					;TAPE MOTION PACKET COMMAND VALUES			
1796								
1797	036566				T30IMV:			
1798	036566				T30RN:			
1799	036566	000000				.WORD	000000	;NEITHER EWB NOR ESS
1800	036570	000100				.WORD	000100	;EWB SET
1801	036572	000200				.WORD	000200	;ESS SET
1802	036574	000300				.WORD	000300	;BOTH EWB AND ESS SET
1803	036576	177777				.WORD	177777	;END OF DATA
1804					:			
1805					:			
1806	036600	000000			T30CNT:	.WORD	0	;TAPE TIMER COUNTER STORAGE AREA
1807	036602	000000			T30CNU:	.WORD	0	;TAPE TIMER COUNTER STORAGE AREA
1808	036604	000000			T30FCN:	.WORD	0	;FILE NUMBER COUNTER
1809	036606	000000			T30DLY:	.WORD	0	;DELAY COUNTER STORAGE
1810					:			
1811					;LOCAL TEXT MESSAGES FOR TEST			
1812					;-			
1813					:			
1814	036610	124	123	123	T30IBU:	.ASCIZ		'TSSR Incorrect After SKIP TAPE MARK REVERSE Into BOT'
1815	036675	122	111	102	T30RIB:	.ASCIZ		'RIB Bit (XST3) Failed To Set After Reverse Into BOT'
1816	036761	124	123	123	T30IBT:	.ASCIZ		'TSSR Incorrect After SKIP TAPE MARK REVERSE At BOT'
1817	037044	124	123	123	T30SKM:	.ASCIZ		'TSSR Incorrect After SKIP TAPE MARK Command'
1818	037120	124	123	123	T30WDD:	.ASCIZ		'TSSR Not Correct After WRITE DATA Command'
1819	037172	124	141	160	T30PTB:	.ASCIZ		'Tape Not Positioned On Correct Record After READ REVERSE'
1820	037263	124	141	160	T30TPB:	.ASCIZ		'Tape Not Positioned On Second File First Record'
1821	037343	124	123	123	T30RDF:	.ASCIZ		'TSSR Incorrect After READ FORWARD Into "File"'
1822	037421	124	123	123	T30RDG:	.ASCIZ		'TSSR Incorrect After SPACE Command Into TAPE MARK'
1823	037503	124	123	123	T30WDF:	.ASCIZ		'TSSR Not Correct After Illegal Mode Bits Set'
1824	037560	111	154	154	T30LOQ:	.ASCIZ		'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
1825	037641	127	122	111	T30SSR:	.ASCIZ		'WRITE MISCELLANEOUS Command Not Accepted'
1826	037712	124	123	123	T30WDE:	.ASCIZ		'TSSR Not Correct After SKIP TAPE MARKS, At BOT'
1827	037771	124	141	160	T30BOT:	.ASCIZ		'Tape Not At BOT After REWIND Command'
1828	040036	124	123	123	T30TH:	.ASCIZ		'TSSR Not Correct After SPACE FORWARD Command'
1829	040113	124	123	123	T30TH2:	.ASCIZ		'TSSR Not Correct After SPACE REVERSE Command'
1830	040170	122	145	167	T30RWI:	.ASCIZ		'Rewind (POSITION) Command Not Accepted'
1831	040237	104	162	151	T30OFL:	.ASCIZ		'Drive 7 Select Failed To Set "OFL" In TSSR'
1832	040312	124	123	123	T30WDC:	.ASCIZ		'TSSR Not Correct After WRITE TAPE MARK Command'
1833	040371	103	126	103	T30VCK:	.ASCIZ		'CVC Set, Didn't Reset VCK In Message Buffer'
1834	040444	124	115	113	T30THK:	.ASCIZ		'THK Not Set After WRITE TAPE MARK (RETRY) Command'

1835	040526	123	113	111	T3ONEF: .ASCIZ	'SKIP TAPE MARKS, f OT, Failed To Set NEF Bit'
1836	040605	124	115	113	T3ORRM: .ASCIZ	'TMK Not Set After READ REVERSE Into TAPE MARK
1837	040663	124	115	113	T3ORRN: .ASCIZ	'TMK Not Set After SPACE REVERSE Into TAPE MARK'
1838	040742	124	115	113	T3ORRP: .ASCIZ	'TMK Not Set After READ FORWARD Into TAPE MARK'
1839	041020	116	117	040	T3ODTR: .ASCIZ	'NO Data Transferred On READ FORWARD'
1840	041064	104	141	164	T3ODTA: .ASCIZ	'Data Compare Error, Data Read From Tape Not Equal To Written'
1841	041161	123	153	151	TST3OID: .ASCIZ	'Skip Tape Marks'
1842					.EVEN	
1843					;	
1844					;	
1845					;	
1846					;	
1847					;	
1848					;	
1849					;	
1850	041202				T3OREST:	
1851	041202				SAVREG	;SAVE THE REGISTERS
1852	041206	012701	036430		MOV #T3OPACKET,R1	;START OF THE PACKET
1853	041212	012721	100004		MOV #100004,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
1854	041216	012721	036440		MOV #T30DATA,(R1)+	;ADDRESS OF CHARAISTICS DATA BLOCK
1855	041222	005021			CLR (R1)+	;EXTENDED ADDRESS
1856	041224	012721	000012		MOV #10.,(R1)+	;SIZE OF DATA BLOCK IN BYTES
1857	041230	012721	036452		MOV #T30BFR,(R1)+	;ADDRESS OF MESSAGE BUFFER
1858	041234	005021			CLR (R1)+	
1859	041236	012721	000024		MOV #20 ,(R1)+	;LENGTH OF MESSAGE BUFFER
1860	041242	005021			CLR (R1)+	
1861	041244	012711	000000		MOV #0,(R1)	;SELECT DRIVE ZERO
1862	041250	012702	000030		MOV #24.,R2	;NUMBER OF LOCATIONS TO BE CLEARED
1863	041254	012762	177777	036452 64#:	MOV #177777,T30BFR(R2)	;ALL ONES TO MESSAGE BUFFER
1864	041262	005742			TST -(R2)	;NEXT LOCATION
1865	041264	022702	000000		CMP #0.,R2	;CHECK R2 FOR DONE
1866	041270	001371			BNE 64#	;KEEP GOING UNTIL DONE
1867	041272	000207			RTS PC	;RETURN
1868						
1869	041274				T3ORT2:	
1870	041274				SAVREG	;SAVE THE REGISTERS
1871	041300	012701	036540		MOV #T30PK2,R1	;START OF THE PACKET
1872	041304	012721	100006		MOV #100006,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
1873	041310	012721	036560		MOV #T30BF2,(R1)+	;ADDRESS OF DATA BLOCK
1874	041314	005021			CLR (R1)+	;EXTENDED ADDRESS
1875	041316	012721	000006		MOV #6.,(R1)+	;SIZE OF DATA BLOCK IN BYTES
1876	041322	005021			CLR (R1)+	
1877	041324	012701	036560		MOV #T30BF2,R1	;POINT TO DATA SEL AREA
1878	041330	005021			CLR (R1)+	
1879	041332	005011			CLR (R1)	
1880	041334	000207			RTS PC	;RETURN
1881	041336				T3ORT3:	
1882	041336				SAVREG	;SAVE REGISTERS
1883	041342	012701	036550		MOV #T30PK3,R1	;SET UP POINTER ADDRESS
1884	041346	005021			CLR (R1)+	;COMMAND SPACE
1885	041350	005021			CLR (R1)+	;ADDRESS OF DATA BLOCK
1886	041352	005021			CLR (R1)+	;EXTENDED ADDRESS
1887	041354	005011			CLR (R1)	;SIZE OF DATA TRANSFER BLOCK
1888	041356	000207			RTS PC	;RETURN
1889	041360				ENDTST	
	041360					
	041360	104401				

L10043: TRAP C\$ETST

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

SEQ 0140

	041456	000000							.WORD	0
	041460	005367	177772						DEC	6(PC)
	041464	001375							BNE	4
	041466	005367	177756						DEC	22(PC)
	041472	001367							BNE	20
1945	041474	005337	043272		DEC	T31DLY				
1946	041500	001356			BNE	10\$				
1947	041502	005237	002214		INC	FATFLG				
1951	041506	010001			MOV	R0,R1				
1952	041510				ERRDF	ERRNO,SFIERR,SFIMSG				
	041510	104455							TRAP	C\$ERDF
	041512	000455							.WORD	301
	041514	003646							.WORD	SFIERR
	041516	012114							.WORD	SFIMSG
1953	041520	013737	002174	043140	20\$:	MOV	UNITN,T31DSW			
1954	041526	012704	043120			MOV	#T31PACKET,R4			
1955	041532	004737	010742			JSR	PC,WRTCHR			
1956	041536	103407				BCS	23\$			
1957	041540	005237	002214			INC	FATFLG			
1961	041544	010001				MOV	R0,R1			
1962	041546					ERRHRD	ERRNO,WRTMSG,SFIMSG			
	041546	104456							TRAP	C\$ERHRD
	041550	000456							.WORD	302
	041552	00505\$.WORD	WRTMSG
	041554	012114							.WORD	SFIMSG
1953	041556				23\$:	CKLOOP				
	041556	104406							TRAP	C\$CLP1
1964	041560	004737	011074			JSR	PC,REWIND			
1965	041564	103407				BCS	30\$			
1966	041566	010004				MOV	R0,R4			
1967	041570	005237	002214			INC	FATFLG			
1971	041574					ERRHRD	ERRNO,T31RWN,PKTSSR			
	041574	104456							TRAP	C\$ERHRD
	041576	000457							.WORD	303
	041600	044624							.WORD	T31RWN
	041602	012126							.WORD	PKTSSR
1972	041604				30\$:	CKLOOP				
	041604	104406							TRAP	C\$CLP1
1973	041606	013701	043150			MOV	T31BFR+6,R1			
1974	041612	010102				MOV	R1,R2			
1975	041614	052702	000002			BIS	#BIT1,R2			
1976	041620	020102				CMP	R1,R2			
1977	041622	001406				BEQ	40\$			
1978	041624	005237	002214			INC	FATFLG			
1982	041630					ERRHRD	ERRNO,T31BOT,EXPREC			
	041630	104456							TRAP	C\$ERHRD
	041632	000460							.WORD	304
	041634	044275							.WORD	T31BOT
	041636	015554							.WORD	EXPREC
1983	041640				40\$:	CKLOOP				
	041640	104406							TRAP	C\$CLP1
1984	041642	013737	003116	043242		MOV	FREE,T31WB			
1985	041650	012737	140005	043240	65\$:	MOV	#140005,T31PK3			
1986	041656	012704	043240			MOV	#T31PK3,R4			
1987	041662	012700	000144			MOV	#100.,R0			
1988	041666	004737	017502			JSR	PC,FILLMEM			
1989	041672	012737	000144	043246		MOV	#100.,T31SZ			

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 14-JUN 84 16:41
 TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

SEQ 0141

1990	041700	010465	000000	MOV	R4,TSDB(R5)	;ISSUE COMMAND		
1991	041704	004737	016330	JSR	PC,WAITF	;WAIT FOR SSR TO SET		
1992	041710	016501	000002	MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
1993	041714	012702	000200	MOV	#SSR,R2	;SET UP EXPECTED		
1994	041720	020102		CMP	R1,R2	;ARE THEY EQUAL		
1995	041722	001406		BEQ	80\$;BR, IF OK		
1996	041724	005237	002214	INC	FATFLG	;ERROR COUNT		
2000	041730			ERRHRD	ERRNO,T31WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA		
	041730	104456				TRAP	C#ERHRD	
	041732	000461				.WORD	305	
	041734	045160				.WORD	T31WDC	
	041736	012126				.WORD	PKTSSR	
2001	041740			80\$:	CKLOOP	;LOOP IF SELECTED		
	041740	104406				TRAP	C#CLP1	
2002	041742	004737	011074	JSR	PC,REWIND	;CALL TAPE REWIND COMMAND		
2003	041746	103407		BCS	230\$;BR, IF NO PROBLEM		
2004	041750	010001		MOV	R0,R1	;SAVE TSSR		
2005	041752	005237	002214	INC	FATFLG	;ERROR COUNT		
2009	041756			ERRHRD	ERRNO,T31RWN,EXPREC	;REWIND NOT ACCEPTED		
	041756	104456				TRAP	C#ERHRD	
	041760	000462				.WORD	306	
	041762	044624				.WORD	T31RWN	
	041764	015554				.WORD	EXPREC	
2010	041766			230\$:	CKLOOP	;LOOP IF SELECTED		
	041766	104406				TRAP	C#CLP1	
2011	041770	013701	043150	MOV	T31BFR+6,R1	;PICK UP XSTO		
2012	041774	010102		MOV	R1,R2	;SET UP EXPECTED		
2013	041776	052702	000002	BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
2014	042002	020102		CMP	R1,R2	;DOES EXP = REC'D		
2015	042004	001406		BEQ	240\$;BR, IF EQUAL (OK)		
2016	042006	005237	002214	INC	FATFLG	;ERROR COUNT		
2020	042012			ERRHRD	ERRNO,T31BOT,EXPREC	;TAPE NO AT BOT AFTER REWIND		
	042012	104456				TRAP	C#ERHRD	
	042014	000463				.WORD	307	
	042016	044275				.WORD	T31BOT	
	042020	015554				.WORD	EXPREC	
2021	042022			240\$:	CKLOOP	;LOOP IF SELECTED		
	042022	104406				TRAP	C#CLP1	
2022	042024	012737	041012	043240	265\$:	MOV	#041012,T31PK3	;NO-OP,CVC=1 COMMAND
2023	042032	012704	043240			MOV	#T31PK3,R4	;SET UP R4 WITH PACKET ADDRESS
2024	042036	010337	043246			MOV	R3,T31SZ	;SET UP RECORD SIZE IN PACKET
2025	042042	010465	000000			MOV	R4,TSDB(R5)	;ISSUE COMMAND
2026	042046	004737	016330			JSR	PC,WAITF	;WAIT FOR SSR TO SET
2027	042052	016501	000002			MOV	TSSR(R5),R1	;GET TSSR CONTENTS
2028	042056	012702	000200			MOV	#SSR,R2	;SET UP EXPECTED
2029	042062	020102				CMP	R1,R2	;ARE THEY EQUAL
2030	042064	001406				BEQ	280\$;BR, IF OK
2031	042066	005237	002214			INC	FATFLG	;ERROR COUNT
2035	042072					ERRHRD	ERRNO,T31RDF,PKTSSR	;TSSR INCORRECT AFTER READ DATA
	042072	104456				TRAP	C#ERHRD	
	042074	000464				.WORD	308	
	042076	043473				.WORD	T31RDF	
	042100	012126				.WORD	PKTSSR	
2036	042102			280\$:	CKLOOP	;LOOP IF SELECTED		
	042102	104406				TRAP	C#CLP1	
2037	042104	013701	043150			MOV	T31BFR+6,R1	;PICK UP XSTO
2038	042110	010102				MOV	R1,R2	;SET UP EXPECTED

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 3: NO OP ("CLEAN TAPE") AND INITIALIZE

SEQ 0146

```

043116 003600 .WORD L10050
2235 ;*
2236 ;LOCAL STORAGE FOR THIS TEST
2237 ;
2241 043120 T31PACKET: ;COMMAND PACKET FOR TEST
2242 043120 100004 .WORD 100004 ;WRITE CHARACTERISTICS COMMAND, WITH . ACK
2243 043122 043130 .WORD T31DATA ;ADDRESS OF CHARACTERISTICS BLOCK
2244 043124 000000 .WORD 0
2245 043126 000012 .WORD 10 ;STARTING VALUE OF BLOCK SIZE
2246 043130 T31DATA: ;CHARACTERISTICS DATA BLOCK
2247 043130 043142 .WORD T31BFR ;ADDRESS OF MESSAGE BUFFER
2248 043132 000000 .WORD 0
2249 043134 000024 .WORD 20. ;LENGTH OF MESSAGE BUFFER
2250 043136 000000 .WORD 0
2251 043140 000000 T31DSW: .WORD 0 ;SELECT DRIVE 0
2252 043142 T31BFR: .BLKW 25. ;MESSAGE BUFFER
2253 ;
2254 ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
2255 ;
2257 043230 .=<.10>&177770
2259 043230 T31PK2: ;WRITE SUB SYS MEM COMMAND, AND ACK
2260 043230 100006 .WORD 100006 ;ADDRESS OF SELECT BLOCK DATA
2261 043232 043250 .WORD T31BF2
2262 043234 000000 .WORD 0
2263 043236 000006 .WORD 6. ;SIZE OF DATA PACKET
2264
2268 043240 T31PK3: ;REREAD COMMAND, AND ACK
2269 043240 100005 .WORD 100005
2270 043242 T31RB: ;ADDRESS OF WRITE BUFFER
2271 043242 003116 T31WB: .WORD FREE
2272 043244 000000 .WORD 0
2273 043246 000000 T31SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
2274 .EVEN
2275 ;
2276 ;
2277 ;
2278 043250 T31BF2:
2279 043250 010 T31BS0: .BYTE 10 ;BSELO AREA
2280 043251 200 T31BS1: .BYTE 200 ;BSEL1 AREA
2281 043252 000000 T31S2: .WORD 0 ;SEL 2 AREA
2282 043254 000000 T31S3: .WORD 0 ;DATA AREA
2283 ;
2284 ;
2285 .EVEN
2286 ;TAPE MOTION PACKET COMMAND VALUES
2287
2288 043256 100205 T31RN: .WORD 100205 ;REREAD DATA (NEXT)
2289 043260 100605 T31WR: .WORD 100605 ;REREAD DATA RETRY
2290 043262 102205 T31CON: .WORD 102205 ;WRITE CONTINUOUS
2291 043264 177777 .WORD 177777 ;END OF DATA
2292
2293 ;
2294 043266 000000 T31CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
2295 043270 000000 T31CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
2296 043272 000000 T31DLY: .WORD 0 ;DELAY COUNTER
2297 ;*
2298 ;LOCAL TEXT MESSAGES FOR TEST

```

TEST 1 - HARDWARE TEST 1 & TEST MACRO M1113 14 JUN-84 16:41
 TEST 3: NO-OP ("CLEAN TAPE") AND INITIALIZE

SEQ 0147

```

2299
2300
2301 043274      124      123      123  T31RDE: .ASCIZ 'TSSR Not Correct After READ Command'
2302 043340      124      141      160  T31WNH: .ASCIZ 'Tape Position Incorrect After INITIALIZE Command'
2303 043421      124      141      160  T31WNG: .ASCIZ 'Tape Position Incorrect After NOP Command'
2304 043473      124      123      123  T31RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
2305 043542      122      105      122  T31RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
2306 043637      120      117      123  T31SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
2307 043721      122      111      102  T31LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
2308 043771      124      123      123  T31WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
2309 044046      111      154      154  T31LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XSTO'
2310 044127      122      105      122  T31SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
2311 044163      124      123      123  T31WDE: .ASCIZ 'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE Command, At BOT'
2312 044275      124      141      160  T31BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XSTO)'
2313 044370      116      117      055  T31TIM: .ASCIZ 'NO-OP ("CLEAN TAPE") AND INITIALIZE'S Erase Tape Not Long Enough'
2314 044470      122      105      122  T31EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
2315 044547      124      123      123  T31TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
2316 044624      122      145      167  T31RMN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
2317 044673      122      101      115  T31RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
2318 044746      124      123      123  T31AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
2319 045015      104      162      151  T31OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
2320 045070      124      123      123  T31WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
2321 045160      124      123      123  T31WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
2322 045233      103      126      103  T31VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
2323 045306      124      123      102  T31BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
2324 045361      127      122      111  T31WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
2325 045450      122      145      141  T31LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XSTO'
2326 045532      122      145      141  T31LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XSTO'
2327 045614      122      145      163  T31PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
2328 045702      122      145      141  T31TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
2329 045770      116      117      055  T31NEF: .ASCIZ 'NO-OP ("CLEAN TAPE") AND INITIALIZE, At First Record, Failed To Set RIB Bit
X
2330 046111      124      123      123  T31SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
2331 046166      124      123      123  T31TSA: .ASCIZ 'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE, Into BOT'
2332 046273      124      123      123  T31WRF: .ASCIZ 'TSSR Not Correct After NO-OP ("CLEAN TAPE") AND INITIALIZE Command'
2333 046376      104      141      164  T31DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
2334 046473      116      117      055  TST31ID: .ASCIZ 'NO-OP ("Clean Tape") And INITIALIZE'
2335
2336
2337
2338
2339
2340
2341
2342
2343 046540
2344 046540
2345 046544      012701      043120
2346 046550      012721      100004
2347 046554      012721      043130
2348 046560      005021
2349 046562      012721      000012
2350 046566      012721      043142
2351 046572      005021
2352 046574      012721      000024
2353 046600      005021
2354 046602      012711      000000
2355 046606      012702      000030

;
;
;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;WRITE SUBSYSTEM MEMORY COMMAND
;
;
T31REST:
      SAVREG
      MOV      #T31PACKET,R1
      MOV      #100004,(R1)
      MOV      #T31DATA,(R1)
      CLR      (R1)
      MOV      #10,(R1)
      MOV      #T31BFR,(R1)
      CLR      (R1)
      MOV      #20,(R1)
      CLR      (R1)
      MOV      #0,(R1)
      MOV      #24,,R2

;SAVE THE REGISTERS
;START OF THE PACKET
;WRITE SUBSYSTEM MEM. WITH ACK.
;ADDRESS OF CHARAISTICS DATA BLOCK
;EXTENDED ADDRESS
;SIZE OF DATA BLOCK IN BYTES
;ADDRESS OF MESSAGE BUFFER
;LENGTH OF MESSAGE BUFFER
;SELECT DRIVE ZERO
;NUMBER OF LOCATIONS TO BE CLEARED

```

```

2356 046612 012762 177777 043142 64: MOV @177777,T31BFR(R2) ;ALL ONES TO MESSAGE BUFFER
2357 046620 005742 TST -(R2) ;NEXT LOCATION
2358 046622 022702 000000 CMP #0,R2 ;AT END OF LOOP YET
2359 046626 001371 BNE 64: ;KEEP GOING UNTIL DONE
2360 046630 000207 RTS PC ;RETURN
2361
2362 046632 T31RT2:
2363 046632 SAVREG ;SAVE THE REGISTERS
2364 046636 012701 043230 MOV @T31PK2,R1 ;START OF THE PACKET
2365 046642 012721 100006 MOV #100006,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK.
2366 046646 012721 043250 MOV @T31BF2,(R1)+ ;ADDRESS OF DATA BLOCK
2367 046652 005021 CLR (R1)+ ;EXTENDED ADDRESS
2368 046654 012721 000006 MOV #6,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
2369 046660 005021 CLR (R1)+
2370 046662 012701 043250 MOV @T31BF2,R1 ;POINT TO DATA SEL AREA
2371 046666 005021 CLR (R1)+
2372 046670 005011 CLR (R1)+
2373 046672 000207 RTS PC ;RETURN
2374 046674 T31RT3:
2375 046674 SAVREG ;SAVE REGISTERS
2376 046700 012701 043240 MOV @T31PK3,R1 ;SET UP POINTER ADDRESS
2377 046704 005021 CLR (R1)+ ;COMMAND SPACE
2378 046706 005021 CLR (R1)+ ;ADDRESS OF DATA BLOCK
2379 046710 005021 CLR (R1)+ ;EXTENDED ADDRESS
2380 046712 005011 CLR (R1)+ ;SIZE OF DATA TRANSFER BLOCK
2381 046714 000207 RTS PC ;RETURN
2382 046716 ENDTST
046716 L10050: TRAP C#ETST
046716 104401

```

2383
 2384
 2385
 2386
 2387
 2388
 2389
 2390
 2391
 2392
 2393
 2394
 2395
 2396
 2397
 2398
 2399
 2400
 2401
 2402
 2403
 2404
 2405
 2406
 2407
 2408
 2409
 2410

.SBTTL TEST 4: Erase And Operation Incomplete

VERIFIES THAT AN ERASE COMMAND ISSUED WHEN THE TAPE IS POSITIONED AT BOT OPERATES PROPERLY AND ACTUALLY ERASES TAPE. THE FOLLOWING TEST SEQUENCE IS PERFORMED:

1. THE TAPE IS FIRST REWOUND, SEVERAL TEST RECORDS ARE WRITTEN, AND THE TAPE IS REWOUND AGAIN.
2. AN ERASE COMMAND IS ISSUED, WHICH SHOULD ERASE A NUMBER OF THE TEST RECORDS.
3. NORMAL TERMINATION IS VERIFIED AND STATUS IS CHECKED (BOT SHOULD BE 0).
4. A READ REVERSE COMMAND IS ISSUED. IT IS VERIFIED THAT THE COMMAND TERMINATES WITH TAPE STATUS ALERT, THAT THE REVERSE INTO BOT (RIB) STATUS BIT IS SET, AND THAT NO DATA IS TRANSFERRED. THIS DEMONSTRATES THAT NO DATA WAS ENCOUNTERED IN THE AREA ERASED BY THE ERASE COMMAND.

2660	050006	013737	003116	051402		MOV	FREE,T32WB		;STARTING WRITE BUFFER ADDRESS
2661	050014	012737	140005	051400	65#:	MOV	#140005,T32PK3		;WRITE DATA,CVC=1,ACK COMMAND
2662	050022	012704	051400			MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS
2663	050026	010300				MOV	R3,R0		;SET PATTERN IN CORRECT REGISTER
2664	050030	004737	017502			JSR	PC,FILLMEM		;FILL MEMORY WITH RECORD SIZE
2665	050034	010337	051406			MOV	R3,T32SZ		;SET UP RECORD SIZE IN PACKET
2666	050040	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND
2667	050044	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET
2668	050050	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2669	050054	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED
2670	050060	020102				CMP	R1,R2		;ARE THEY EQUAL
2671	050062	001406				BEQ	80#		;BR, IF OK
2672	050064	005237	002214			INC	FATFLG		;ERROR COUNT
2676	050070					ERRHRD	ERRNO,T32WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	050070	104456							TRAP C#ERHRD
	050072	000637							.WORD 415
	050074	052466							.WORD T32WDC
	050076	012126							.WORD PKTSSR
2677	050100				80#:	CKLOOP			;LOOP IF SELECTED
	050100	104406							TRAP C#CLP1
2678	050102	005723				TST	(R3)+		;BUMP RECORD SIZE COUNTER
2679	050104	020327	000156			CMP	R3,#110.		;AT 160 SIZE YET
2680	050110	001341				BNE	65#		;BR, IF MORE RECORDS TO WRITE
2681	050112	004737	011074			JSR	PC,REWIND		;CALL TAPE REWIND COMMAND
2682	050116	103407				BCS	230#		;BR, IF NO PROBLEM
2683	050120	010001				MOV	R0,R1		;SAVE TSSR
2684	050122	005237	002214			INC	FATFLG		;ERROR COUNT
2688	050126					ERRHRD	ERRNO,T32RWN,EXPREC		;REWIND NOT ACCEPTED
	050126	104456							TRAP C#ERHRD
	050130	000640							.WORD 416
	050132	051630							.WORD T32RWN
	050134	015554							.WORD EXPREC
2689	050136				230#:	CKLOOP			;LOOP IF SELECTED
	050136	104406							TRAP C#CLP1
2690	050140	013701	051310			MOV	T32BFR+6,R1		;PICK UP XSTO
2691	050144	010102				MOV	R1,R2		;SET UP EXPECTED
2692	050146	052702	000002			BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED
2693	050152	020102				CMP	R1,R2		;DOES EXP = REC'D
2694	050154	001406				BEQ	240#		;BR, IF EQUAL (OK)
2695	050156	005237	002214			INC	FATFLG		;ERROR COUNT
2699	050162					ERRHRD	ERRNO,T32BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND
	050162	104456							TRAP C#ERHRD
	050164	000641							.WORD 417
	050166	051446							.WORD T32BOT
	050170	015554							.WORD EXPREC
2700	050172				240#:	CKLOOP			;LOOP IF SELECTED
	050172	104406							TRAP C#CLP1
2701	050174	012703	000001			MOV	#1,R3		;SET UP FOR SPACE COMMAND
2702	050200	004737	010544			JSR	PC,SPACE		;ISSUE SPACE COMMAND 1 FORWARD
2703	050204	012737	140411	051400	265#:	MOV	#140411,T32PK3		;ERASE DATA,ACK COMMAND
2704	050212	012704	051400			MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS
2705	050216	010465	000000			MOV	R4,TSDB(R5)		;ISSUE COMMAND
2706	050222	004737	016330			JSR	PC,WAITF		;WAIT FOR SSR TO SET
2707	050226	016501	000002			MOV	TSSR(R5),R1		;GET TSSR CONTENTS
2708	050232	012702	000200			MOV	#SSR,R2		;SET UP EXPECTED
2709	050236	020102				CMP	R1,R2		;ARE THEY EQUAL
2710	050240	001406				BEQ	280#		;BR, IF OK

TEST 1 - HARDWARE TEST 1 B TEST MACRO M1113 14 JUN 84 16:41
 TEST 4: ERASE AND OPERATION INCOMPLETE

SEQ 0157

2806	050546	004737	010742		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS		
2807	050552	103407			BCS	231		;BR, IF COMMAND ISSUED OK		
2808	050554	005237	002214		INC	FATFLG		;ERROR COUNT		
2812	050560	010001			MOV	R0,R1		;SAVE CONTENTS OF TSSR		
2813	050562				ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTIC FAILED		
	050562	104456							TRAP	C1ERHRD
	050564	000646							.WORD	422
	050566	005052							.WORD	WRTMSG
	050570	012114							.WORD	SFIMSG
2814	050572			231:	CKLOOP			;LOOP IF SELECTED		
	050572	104406							TRAP	C1CLP1
2815	050574	004737	011074		JSR	PC,REWIND		;CALL TAPE REWIND COMMAND		
2816	050600	103411			BCS	301		;BR, IF NO PROBLEM		
2817	050602	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
2818	050606	010004			MOV	R0,R4		;GET PACKET ADDRESS		
2819	050610	005237	002214		INC	FATFLG		;ERROR COUNT		
2823	050614				ERRHRD	ERRNO,T32RWN,PKTSSR		;REWIND NOT ACCEPTED		
	050614	104456							TRAP	C1ERHRD
	050616	000647							.WORD	423
	050620	051630							.WORD	T32RWN
	050622	012126							.WORD	PKTSSR
2824	050624			301:	CKLOOP			;LOOP IF SELECTED		
	050624	104406							TRAP	C1CLP1
2825	050626	013701	051310		MOV	T32BFR+6,R1		;PICK UP XSTO		
2826	050632	010102			MOV	R1,R2		;SET UP EXPECTED		
2827	050634	052702	000002		BIS	#BIT1,R2		;SET BOT BIT IN EXPECTED		
2828	050640	020102			CMF	R1,R2		;DOES EXP = REC D		
2829	050642	001406			BEQ	401		;BR, IF EQUAL (OK)		
2830	050644	005237	002214		INC	FATFLG		;ERROR COUNT		
2834	050650				ERRHRD	ERRNO,T32BOT,EXPREC		;TAPE NOT AT BOT AFTER REWIND		
	050650	104456							TRAP	C1ERHRD
	050652	000650							.WORD	424
	050654	051446							.WORD	T32BOT
	050656	015554							.WORD	EXPREC
2835	050660			401:	CKLOOP			;LOOP IF SELECTED		
	050660	104406							TRAP	C1CLP1
2836	050662	012737	140411	051400	651:	MOV	#140411,T32PK3	;ERASE DATA,CVC=1,ACK COMMAND		
2837	050670	012704	051400		MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS		
2838	050674	010337	051406		MOV	R3,T32SZ		;SET UP RECORD SIZE IN PACKET		
2839	050700	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND		
2840	050704	004737	016330		JSR	PC,WAITF		;WAIT FOR SSR TO SET		
2841	050710	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS		
2842	050714	012702	000206		MOV	#SSR,R2		;SET UP EXPECTED		
2843	050720	020102			CMF	R1,R2		;ARE THEY EQUAL		
2844	050722	001757			BEQ	651		;BR, IF OK		
2845	050724	032701	000004		BIT	#BIT2,R1		;CHECK FOR TAPE STATUS ALERT		
2846	050730	001006			BNE	801		;BR, IF TAPE STATUS ALERT SET		
2847	050732	005237	002214		INC	FATFLG		;ERROR COUNT		
2851	050736				ERRHRD	ERRNO,T32WDC,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA		
	050736	104456							TRAP	C1ERHRD
	050740	000651							.WORD	425
	050742	052466							.WORD	T32WDC
	050744	012126							.WORD	PKTSSR
2852	050746			801:	CKLOOP			;LOOP IF SELECTED		
	050746	104406							TRAP	C1CLP1
2853	050750	013701	051310		MOV	T32BFR+6,R1		;PICK UP XSTO		
2854	050754	010102			MOV	R1,R2		;SET UP EXPECTED		

2855	050756	052702	000001		BIS	#BIT0,R2		;SET EOT BIT IN EXPECTED	
2856	050762	020102			CMP	R1,R2		;DOES EXP = REC'D	
2857	050764	001406			BEQ	240\$;BR, IF EQUAL (0F)	
2858	050766	005237	002214		INC	FATFLG		;ERROR COUNT	
2862	050772				ERRHRD	ERRNO,T32EOT,EXPREC		;TAPE NOT AT EOT AFTER ERASE COMMANDS	
	050772	104456						TRAP	C\$ERHRD
	050774	000652						.WORD	426
	050776	051541						.WORD	T32EOT
	051000	015554						.WORD	EXPREC
2863	051002			240\$:	CKLOOP			;LOOP IF SELECTED	
	051002	104406						TRAP	C\$CLP1
2864	051004	012703	051410		MOV	#T32CMD,R3		;STARTING RECORD SIZE	
2865	051010	013737	003116	051402	MOV	FREE,T32RB		;STARTING READ BUFFER ADDRESS	
2866	051016	011337	051400		265\$:	MOV	(R3),T32PK3	;READ DATA,ACK COMMAND	
2867	051022	012704	051400		MOV	#T32PK3,R4		;SET UP R4 WITH PACKET ADDRESS	
2868	051026	012700	177777		MOV	#177777,R0		;SET PATTERN IN CORRECT REGISTER	
2869	051032	004737	017502		JSR	PC,FILLMEM		;FILL MEMORY WITH ALL ONES	
2870	051036	012737	000144	051406	MOV	#100.,T32SZ		;SET UP RECORD SIZE IN PACKET	
2871	051044	010465	000000		MOV	R4,T32DB(R5)		;ISSUE COMMAND	
2872	051050	012737	000062	051444	MOV	#50.,T32DLY		;SET UP DELAY COUNTER	
2873	051056	004737	016330		270\$:	JSR	PC,WAITF	;WAIT FOR SSR TO SET	
2874	051062	016501	000002		MOV	T32SR(R5),R1		;GET T32SR CONTENTS	
2875	051066	012702	100214		MOV	#SSR!SC!BIT2!BIT3,R2		;SET UP EXPECTED	
2876	051072	020102			CMP	R1,R2		;ARE THEY EQUAL	
2877	051074	001425			BEQ	280\$;BR, IF OK	
2878	051076				DELAY	250		;DELAY FOR SSR TO BE SET	
	051076	012727	000250					MOV	#250,(PC).
	051102	000000						.WORD	0
	051104	013727	002116					MOV	L\$DLY,(PC).
	051110	000000						.WORD	0
	051112	005367	177772					DEC	-6(PC)
	051116	001375						BNE	. 4
	051120	005367	177756					DEC	-22(PC)
	051124	001367						BNE	. 20
2879	051126	005337	051444		DEC	T32DLY		;COUNT DELAY ROUTINE DOWN	
2880	051132	001351			BNE	270\$;BR, IF DELAY HAS NOT ENDED	
2881	051134	005237	002214		INC	FATFLG		;ERROR COUNT	
2885	051140				ERRHRD	ERRNO,T32ECF,PKTSSR		;T32SR INCORRECT AFTER READ DATA	
	051140	104456						TRAP	C\$ERHRD
	051142	000653						.WORD	427
	051144	052405						.WORD	T32ECF
	051146	012126						.WORD	PKTSSR
2886	051150			280\$:	CKLOOP			;LOOP IF SELECTED	
	051150	104406						TRAP	C\$CLP1
2887	051152	013701	051316		MOV	T32BFR+14,R1		;PICK UP XST3	
2888	051156	010102			MOV	R1,R2		;SET UP EXPECTED	
2889	051160	052702	000100		BIS	#BIT6,R2		;SET OPI BIT IN EXPECTED	
2890	051164	020102			CMP	R1,R2		;IS OPI BIT SET	
2891	051166	001406			BEQ	290\$;BR, IF BIT IS SET	
2892	051170	005237	002214		INC	FATFLG		;ERROR COUNT	
2896	051174				ERRHRD	ERRNO,T32OPI,EXPREC		;OPI BIT NOT SET	
	051174	104456						TRAP	C\$ERHRD
	051176	000654						.WORD	428
	051200	052533						.WORD	T32OPI
	051202	015554						.WORD	EXPREC
2897	051204			290\$:	CKLOOP			;LOOP IF SELECTED	
	051204	104406						TRAP	C\$CLP1

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN-84 16:41
 TEST 4: ERASE AND OPERATION INCOMPLETE

SEQ 0160

```

2958
2959
2960
2961 051410
2962 051410 140410
2963 051412 141410
2964 051414 140401
2965 051416 141001
2966 051420 161401
2967 051422 161001
2968 051424 141401
2969 051426 140001
2970 051430 141410
2971 051432 141010
2972 051434 141005
2973 051436 177777
2974
2975
2976 051440 000000
2977 051442 000000
2978 051444 000000
2979
2980
2981
2982
2983 051446 124 141 160
2984 051541 124 141 160
2985 051630 122 145 167
2986 051677 124 123 123
2987 051746 124 123 123
2988 052013 124 123 102
2989 052066 122 105 101
2990 052164 124 123 123
2991 052241 124 123 123
2992 052316 102 117 124
2993 052405 105 122 101
2994 052466 124 123 123
2995 052533 117 126 111
2996 052570 105 162 141
2997
2998
2999
3000
3001
3002
3003
3004
3005 052630
3006 052630
3007 052634 012701 051260
3008 052640 012721 100004
3009 052644 012721 051270
3010 052650 005021
3011 052652 012721 000012
3012 052656 012721 051302
3013 052662 005021
3014 052664 012721 000024

      .EVEN
;TAPE MOTION PACKET COMMAND VALUES

T32CMD:
      .WORD 140410 ;SPACE RECORDS REVERSE
      .WORD 141410 ;SKIP TAPE MARKS REVERSE
      .WORD 140401 ;READ REVERSE
      .WORD 141001 ;REREAD PREVIOUS (OPP=0)
      .WORD 161401 ;REREAD NEXT (OPP=1)
      .WORD 161001 ;REREAD PREVIOUS (OPP=1)
      .WORD 141401 ;REREAD NEXT (OPP=0)
      .WORD 140001 ;READ NEXT
      .WORD 141410 ;SKIP TAPE MARKS REVERSE
      .WORD 141010 ;SKIP RECORDS FORWARD
      .WORD 141005 ;WRITE DATA RETRY
      .WORD 177777 ;END OF DATA

;
T32CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T32CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
T32DLY: .WORD 0 ;DELAY COUNTER

;
;LOCAL TEXT MESSAGES FOR TEST
;-
T32BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
T32EOT: .ASCIZ 'Tape Status Alert During Erase To EOT, But EOT Not Set'
T32RMN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
T32AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
T32ERA: .ASCIZ 'TSSR Not Correct After ERASE Command'
T32BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
T32RIB: .ASCIZ 'READ REVERSE, After ERASE From BOT, Failed To Set RIB In XST3'
T32SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
T32TSA: .ASCIZ 'TSSR Not Correct After READ REVERSE Into BOT'
T32BOE: .ASCIZ 'BOT (XST0) Still Set After Erase From Tape's BOT Marker'
T32ECF: .ASCIZ 'ERASE Failed To Clear Tape (Erase) Tape Properly'
T32WDC: .ASCIZ 'TSSR Not Correct After ERASE Command'
T32OPI: .ASCIZ 'OPI Bit (XST3) Failed To Set'
T32ID: .ASCIZ 'Erase And Operation Incomplete'

      .EVEN

;
;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
;WRITE SUBSYSTEM MEMORY COMMAND
;-

T32REST:
      SAVREG ;SAVE THE REGISTERS
      MOV #T32PACKET,R1 ;START OF THE PACKET
      MOV #100004,(R1) ;WRITE SUBSYSTEM MEM. WITH ACK.
      MOV #T32DATA,(R1) ;ADDRESS OF CHARAISTICS DATA BLOCK
      CLR (R1) ;EXTENDED ADDRESS
      MOV #10,(R1) ;SIZE OF DATA BLOCK IN BYTES
      MOV #T32BFR,(R1) ;ADDRESS OF MESSAGE BUFFER
      CLR (R1)
      MOV #20,(R1) ;LENGTH OF MESSAGE BUFFER

```

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN-84 16:41
 TEST 4: ERASE AND OPERATION INCOMPLETE

SEQ 0161

```

3015 052670 005021          CLR      (R1).
3016 052672 012711 000000  MOV      #0,(R1)          ;SELECT DRIVE ZERO
3017 052676 012702 000030  MOV      #24.,R2         ;NUMBER OF LOCATIONS TO BE CLEARED
3018 052702 012762 177777 051302 64$: MOV      #177777,T32BFR(R2) ;ALL ONES TO MESSAGE BUFFER
3019 052710 005742          TST      -(R2)           ;NEXT LOCATION
3020 052712 022702 000000  CMP      #0,R2          ;AT END OF LOOP YET
3021 052716 001371          BNE      64$            ;KEEP GOING UNTIL DONE
3022 052720 000207          RTS      PC              ;RETURN
3023

```

```

3024 052722          T32RT2:
3025 052722          SAVREG          ;SAVE THE REGISTERS
3026 052726 012701 051370  MOV      #T32PK2,R1     ;START OF THE PACKET
3027 052732 012721 100006  MOV      #100006,(R1). ;WRITE SUBSYSTEM MEM. WITH ACK.
3028 052736 005021          CLR      (R1).         ;ADDRESS OF DATA BLOCK
3029 052740 005021          CLR      (R1).         ;EXTENDED ADDRESS
3030 052742 012721 000006  MOV      #6.,(R1).     ;SIZE OF DATA BLOCK IN BYTES
3031 052746 005021          CLR      (R1).
3032 052750 000207          RTS      PC              ;RETURN
3033

```

```

3034 052752          T32RT3:
3035 052756 012701 051400  SAVREG          ;SAVE REGISTERS
3036 052762 005021          MOV      #T32PK3,R1     ;SET UP POINTER ADDRESS
3037 052764 005021          CLR      (R1).         ;COMMAND SPACE
3038 052766 005021          CLR      (R1).         ;ADDRESS OF DATA BLOCK
3039 052770 005011          CLR      (R1).         ;EXTENDED ADDRESS
3040 052772 000207          CLR      (R1)         ;SIZE OF DATA TRANSFER BLOCK
3041 052774          RTS      PC              ;RETURN
3041 052774          ENDTST

```

L10053: TRAP C#ETST

3042 .SBTTL TEST 5: DATA PARITY TEST

```

3043 ;
3044 ;
3045 ;
3046 ;
3047 ;
3048 ;
3049 ;TEST 5 -- Data Parity Test
3050 ;
3051 ;
3052 ;This test verifies that the data parity circuitry in both the controller and the
3053 ;transport is operating properly by forcing data records with wrong parity to be
3054 ;written onto tape and checking the results obtained when the data is read. The
3055 ;following test sequence is performed:
3056 ;
3057 ;
3058 ; 1. A Write Characteristics command is issued and the resulting status is
3059 ; examined to determine the states of the Extended Features and Buffering
3060 ; Enable switches on the controller module. If buffering is disabled, no
3061 ; further actions need be taken in this step and the program proceeds to
3062 ; the next step. If buffering is enabled, it is disabled via the Buffer
3063 ; Control field in the extended characteristics data word supplied by a
3064 ; Write Characteristics command. (The module must be in Extended mode,
3065 ; so if it is not already, a Write Subsystem Memory command is issued to
3066 ; change the logical sense of the Extended Features switch.)
3067 ;
3068 ; 2. The Write Subsystem Memory command is used to set the Force Wrong
3069 ; Parity control flip flop.

```


TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN-84 16:41
 TEST 5: DATA PARITY TEST

SEQ 0164

3171	053274	104406	002220	42:	TST	EXTFEA			TRAP	C#CLP1
3172	053302	001025			BNE	55:				
3173	053304	112737	000200	054631	MOVB	#200, T33BS1				
3174	053312	112737	000010	054630	MOVB	#10, T33BS0				
3175	053320	012704	054610		MOV	#T33PK2, R4				
3176	053324	010465	000000		MOV	R4, TSD8(R5)				
3177	053330	004737	016416		JSR	PC, CHKTSSR				
3178	053334	103407			BCS	50:				
3179	053336	010001			MOV	R0, R1				
3180	053340	005237	002214		INC	FATFLG				
3184	053344				ERRHRD	ERRNO, T33SSR, PKTSSR				
	053344	104456							TRAP	C#ERHRD
	053346	000771							.WORD	505
	053350	055171							.WORD	T33SSR
	053352	012126							.WORD	PKTSSR
3185	053354			50:	CKLOOP					
	053354	104406							TRAP	C#CLP1
3186	053356	005737	002222	55:	TST	BENBSW				
3187	053362	001426			BEQ	70:				
3188	053364	013737	002174	054520	MOV	UNITN, T33DSW				
3189	053372	042737	000020	054520	BIC	#BIT4, T33DSW				
3190	053400	052737	000010	054520	BIS	#BIT3, T33DSW				
3191	053406	012704	054500		MOV	#T33PACKET, R4				
3192	053412	004737	010742		JSR	PC, WRTCHR				
3193	053416	103407			BCS	60:				
3194	053420	005237	002214		INC	FATFLG				
3198	053424	010001			MOV	R0, R1				
3199	053426				ERRHRD	ERRNO, WRTMSG, SFIMSG				
	053426	104456							TRAP	C#ERHRD
	053430	000772							.WORD	506
	053432	005052							.WORD	WRTMSG
	053434	012114							.WORD	SFIMSG
3200	053436			60:	CKLOOP					
	053436	104406							TRAP	C#CLP1
3201	053440			70:						
3202	053440	112737	000100	054631	MOVB	#100, T33BS1				
3203	053446	112737	000011	054630	MOVB	#11, T33BS0				
3204	053454	012704	054610		MOV	#T33PK2, R4				
3205	053460	010465	000000		MOV	R4, TSD8(R5)				
3206	053464	004737	016416		JSR	PC, CHKTSSR				
3207	053470	103407			BCS	80:				
3208	053472	010001			MOV	R0, R1				
3209	053474	005237	002214		INC	FATFLG				
3213	053500				ERRHRD	ERRNO, T33SSR, PKTSSR				
	053500	104456							TRAP	C#ERHRD
	053502	000773							.WORD	507
	053504	055171							.WORD	T33SSR
	053506	012126							.WORD	PKTSSR
3214	053510			80:	CKLOOP					
	053510	104406							TRAP	C#CLP1
3215	053512	012703	000026		MOV	#22, R3				
3216	053516	013737	003116	054622	MOV	FREE, T33WB				
3217	053524	005037	054650		CLR	T33CNU				
3218	053530	012737	140005	054620	110:	MOV	#140005, T33PK3			
3219	053536	012704	054620		MOV	#T33PK3, R4				
3220	053542	012737	000024	054626	MOV	#20, T33SZ				

3221	053550	013777	054650	127340	MOV	T33CNU,0FREE	;MEMORY FILLED WITH DATA IN RECORD		
3222	053556	005237	054650		INC	T33CNU	;READY FOR NEXT RECORD		
3223	053562	010465	000000		MOV	R4,TSD8(R5)	;ISSUE COMMAND		
3224	053566	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
3225	053572	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
3226	053576	012702	100210		MOV	#SSR!SC!BIT3,R2	;SET UP EXPECTED		
3227	053602	020102			CMP	R1,R2	;ARE THEY EQUAL		
3228	053604	001406			BEQ	120\$;BR, IF OK		
3229	053606	005237	002214		INC	FATFLG	;ERROR COUNT		
3233	053612				ERRHRD	ERRNO,T33WPW,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA		
	053612	104456					TRAP	C\$ERHRD	
	053614	000774					.WORD	508	
	053616	054732					.WORD	T33WPW	
	053620	012126					.WORD	PKTSSR	
3234	053622			120\$:	CKLOOP		;LOOP IF SELECTED		
	053622	104406					TRAP	C\$CLP1	
3235	053624	013701	054532		MOV	T338FR+10,R1	;PICK UP XST1		
3236	053630	010102			MOV	R1,R2	;SET UP EXPECTED		
3237	053632	052702	000002		BIS	#BIT1,R2	;SET UNC BIT IN EXPECTED		
3238	053636	020102			CMP	R1,R2	;DOES EXP = REC'D		
3239	053640	001406			BEQ	130\$;BR, IF EQUAL (OK)		
3240	053642	005237	002214		INC	FATFLG	;ERROR COUNT		
3244	053646				ERRHRD	ERRNO,T33UNC,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	053646	104456					TRAP	C\$ERHRD	
	053650	000775					.WORD	509	
	053652	055012					.WORD	T33UNC	
	053654	015554					.WORD	EXPREC	
3245	053656			130\$:	CKLOOP		;LOOP IF SELECTED		
	053656	104406					TRAP	C\$CLP1	
3246	053660	005303			DEC	R3	;DEC RECORD COUNTER		
3247	053662	001322			BNE	110\$;BR, IF MORE RECORDS TO WRITE		
3248	053664	004737	011074		JSR	PC,REWIND	;CALL TAPE REWIND COMMAND		
3249	053670	103411			BCS	140\$;BR, IF NO PROBLEM		
3250	053672	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
3251	053676	010004			MOV	R0,R4	;GET PACKET ADDRESS		
3252	053700	005237	002214		INC	FATFLG	;ERROR COUNT		
3256	053704				ERRHRD	ERRNO,T33RWN,PKTSSR	;REWIND NOT ACCEPTED		
	053704	104456					TRAP	C\$ERHRD	
	053706	000776					.WORD	510	
	053710	055350					.WORD	T33RWN	
	053712	012126					.WORD	PKTSSR	
3257	053714			140\$:	CKLOOP		;LOOP IF SELECTED		
	053714	104406					TRAP	C\$CLP1	
3258	053716	013701	054530		MOV	T338FR+6,R1	;PICK UP XST0		
3259	053722	010102			MOV	R1,R2	;SET UP EXPECTED		
3260	053724	052702	000002		BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED		
3261	053730	020102			CMP	R1,R2	;DOES EXP = REC'D		
3262	053732	001406			BEQ	150\$;BR, IF EQUAL (OK)		
3263	053734	005237	002214		INC	FATFLG	;ERROR COUNT		
3267	053740				ERRHRD	ERRNO,T338OT,EXPREC	;TAPE NOT AT BOT AFTER REWIND		
	053740	104456					TRAP	C\$ERHRD	
	053742	000777					.WORD	511	
	053744	055255					.WORD	T338OT	
	053746	015554					.WORD	EXPREC	
3268	053750			150\$:	CKLOOP		;LOOP IF SELECTED		
	053750	104406					TRAP	C\$CLP1	
3269	053752	005037	054650		CLR	T33CNU	;CLEAR DATA VALUE IN RECORD		

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 5: DATA PARITY TEST

SEQ 0166

3270	053756	012703	000024		MOV	#20.,R3		;RECORD SIZE
3271	053762	013737	003116	054622	1554:	MOV	FREE,T33RB	;STARTING WRITE BUFFER ADDRESS
3272	053770	012737	140001	054620		MOV	#140001,T33PK3	;READ DATA,CVC=1,ACK COMMAND
3273	053776	012704	054620			MOV	#T33PK3,R4	;SET UP R4 WITH PACKET ADDRESS
3274	054002	012737	000024	054626		MOV	#20.,T33SZ	;SET UP RECORD SIZE IN PACKET
3275	054010	010465	000000			MOV	R4,TSDB(R5)	;ISSUE COMMAND
3276	054014	004737	016330			JSR	PC,WAITF	;WAIT FOR SSR TO SET
3277	054020	016501	000002			MOV	TSSR(R5),R1	;GET TSSP CONTENTS
3278	054024	012702	100210			MOV	#SSR!SC!BIT3,R2	;SET UP EXPECTED
3279	054030	020102				CMP	R1,R2	;ARE THEY EQUAL
3280	054032	001406				BEQ	1604	;BR, IF OK
3281	054034	005237	002214			INC	FATFLG	;ERROR COUNT
3285	054040					ERRHRD	ERRNO,T33WDC,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA
	054040	104456						TRAP C#ERHRD
	054042	001000						.WORD 512
	054044	055417						.WORD T33WDC
	054046	012126						.WORD PKTSSR
3286	054050				1604:	CKLOOP		;LOOP IF SELECTED
	054050	104406						TRAP C#CLP1
3287	054052	013701	054532			MOV	T338FR+10,R1	;PICK UP XST1
3288	054056	010102				MOV	R1,R2	;SET UP EXPECTED
3289	054060	052702	000002			BIS	#BIT1,R2	;SET UNC BIT IN EXPECTED
	054064	020102				CMP	R1,R2	;DOES EXP = REC'D
3291	054066	001406				BEQ	1704	;BR, IF EQUAL (OK)
3292	054070	005237	002214			INC	FATFLG	;ERROR COUNT
3296	054074					ERRHRD	ERRNO,T33UND,EXPREC	;UNC BIT NOT SET AFTER READ CMD.
	054074	104456						TRAP C#ERHRD
	054076	001001						.WORD 513
	054100	055102						.WORD T33UND
	054102	015554						.WORD EXPREC
3297	054104				1704:	CKLOOP		;LOOP IF SELECTED
	054104	104406						TRAP C#CLP1
3298	054106	013701	054532			MOV	T338FR+10,R1	;PICK UP XST1
3299	054112	010102				MOV	R1,R2	;SET UP EXPECTED
3300	054114	052702	000400			BIS	#BIT8,R2	;SET RBP BIT IN EXPECTED
3301	054120	020102				CMP	R1,R2	;DOES EXP = REC'D
3302	054122	001406				BEQ	1804	;BR, IF EQUAL (OK)
3303	054124	005237	002214			INC	FATFLG	;ERROR COUNT
3307	054130					ERRHRD	ERRNO,T33RBP,EXPREC	;READ BUS PARITY ERROR BIT NOT SET
	054130	104456						TRAP C#ERHRD
	054132	001002						.WORD 514
	054134	054654						.WORD T33RBP
	054136	015554						.WORD EXPREC
3308	054140				1804:	CKLOOP		;LOOP IF SELECTED
	054140	104406						TRAP C#CLP1
3309	054142	017701	126750			MOV	#FREE,R1	;GET DATA READ
3310	054146	013702	054650			MOV	T33CNU,R2	;GET PATTERN
3311	054152	020102				CMP	R1,R2	;ARE THEY EQUAL
3312	054154	001406				BEQ	1824	;BR, IF OK
3313	054156	005237	002214			INC	FATFLG	;ERROR COUNT
3317	054162					ERRHRD	ERRNO,T33DTA,EXPREC	;DATA NOT CORRECT
	054162	104456						TRAP C#ERHRD
	054164	001003						.WORD 515
	054166	055500						.WORD T33DTA
	054170	015554						.WORD EXPREC
3318	054172				1824:	CKLOOP		;LOOP IF SELECTED
	054172	104406						TRAP C#CLP1

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 5: DATA PARITY TEST

SEQ 0169

```

3428 ;
3429 ;
3430 ;
3431 054630 T33BF2:
3432 054630 010 T33BS0: .BYTE 10 ;BSELO AREA
3433 054631 200 T33BS1: .BYTE 200 ;BSEL1 AREA
3434 054632 000000 T33S2: .WORD 0 ;SEL 2 AREA
3435 054634 000000 T33S3: .WORD 0 ;DATA AREA
3436 ;
3437 ;
3438 ; .EVEN
3439 ;TAPE MOTION PACKET COMMAND VALUES
3440
3441 054636 100205 T33RN: .WORD 100205 ;REREAD DATA (NEXT)
3442 054640 100605 T33WR: .WORD 100605 ;REREAD DATA RETRY
3443 054642 102205 T33CON: .WORD 102205 ;WRITE CONTINUOUS
3444 054644 177777 .WORD 177777 ;END OF DATA
3445
3446 ;
3447 054646 000000 T33CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
3448 054650 000000 T33CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
3449 054652 000000 T33DLY: .WORD 0 ;DELAY COUNTER
3450
3451 ;*
3452 ;LOCAL TEXT MESSAGES FOR TEST
3453 ;-
3454 054654 122 145 141 T33RBP: .ASCIZ 'Read Bus Parity Bit Not Set (XST1), Should r'
3455 054732 124 123 123 T33WPW: .ASCIZ 'TSSR Incorrect After Wrong Parity Write Command'
3456 055012 125 116 103 T33UNC: .ASCIZ 'UNC Bit (XST1) Not Set After Wrong Parity WRITE Command'
3457 055102 125 116 103 T33UND: .ASCIZ 'UNC Bit (XST1) Not Set After Wrong Parity READ Command'
3458 055171 127 122 111 T33SSR: .ASCIZ 'WRITE MISCELLANEOUS CONT/READ COMMAND Not Accepted'
3459 055255 124 141 160 T33BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
3460 055350 122 145 167 T33RMN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
3461 055417 124 123 123 T33MDC: .ASCIZ 'TSSR Not Correct After READ Wrong Parity Command'
3462 055500 104 141 164 T33DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
3463 055575 104 141 164 TST33ID: .ASCIZ 'Data Parity'
3464 ; .EVEN
3465 ;*
3466 ;
3467 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
3468 ;WRITE SUBSYSTEM MEMORY COMMAND
3469 ;
3470 ;-
3471
3472 055612 T33REST:
3473 055612 SAVREG ;SAVE THE REGISTERS
3474 055616 012701 054500 MOV #T33PACKET,R1 ;START OF THE PACKET
3475 055622 012721 100004 MOV #100004,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK,
3476 055626 012721 054510 MOV #T33DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
3477 055632 005021 CLR (R1)+ ;EXTENDED ADDRESS
3478 055634 012721 000012 MOV #10.,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
3479 055640 012721 054522 MOV #T33BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
3480 055644 005021 CLR (R1)+
3481 055646 012721 000024 MOV #20.,(R1)+ ;LENGTH OF MESSAGE BUFFER
3482 055652 005021 CLR (R1)+
3483 055654 012711 000000 MOV #0,(R1) ;SELECT DRIVE ZERO
3484 055660 012702 000030 MOV #24.,R2 ;NUMBER OF LOCATIONS TO BE CLEARED

```



```

3543 :
3544 :
3545 :
3546 :
3547 :
3548 :
3549 :
3550 :
3551 :
3552 :
3553 :
3554 :
3555 :
3556 :
3557 :
3558 :
3559 :
3560 :
3561 :
3562 :
3563 :
3564 :
3565 :
3566 :
3567 :
3568 :
3569 :
3570 :
3571 :
3572 :
3573 :
3574 :
3575 :
3576 :
3577 :
3578 :
3579 :
3580 :
3581 :
3582 :
3583 :
3584 :
3585 :
3586 :
3587 :
3588 :
3589 :
3590 :
3591 :
3592 :
3593 :
3594 :
3595 :
3596 :
3597 056022 T34LOOP:
3598 :
3599 :

```

1. THE TAPE IS REWOUND.
2. WRITE DATA COMMANDS ARE REPEATEDLY ISSUED UNTIL TAPE STATUS ALERT TERMINATION IS SEEN WITH EOT=1. ERRORS OTHER THAN OCCASIONAL CORRECTABLE OR UNCORRECTABLE DATA ERRORS CAUSE A FATAL ERROR REPORT. RECORDS WITH DATA ERRORS ARE RETRIED, SO THE TAPE ENDS UP WITH GOOD DATA.
3. ANOTHER WRITE DATA COMMAND IS ISSUED, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1.
4. A WRITE TAPE MARK COMMAND IS ISSUED, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1.
5. A SKIP TAPE MARKS REVERSE COMMAND IS ISSUED, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1 AND TMK=1.
6. A SPACE RECORDS REVERSE COMMAND, WITH A RECORD COUNT OF 1, IS ISSUED, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=1 AND TMK=1.
7. A SPACE RECORDS REVERSE COMMAND, WITH A RECORD COUNT OF 1, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
8. A SPACE RECORDS FORWARD COMMAND, WITH A RECORD COUNT OF 1, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
9. A READ REVERSE COMMAND IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
10. A READ FORWARD COMMAND IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
11. A SPACE RECORDS REVERSE COMMAND, WITH A RECORD COUNT OF 3, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=0.
12. A SPACE RECORDS FORWARD COMMAND, WITH A RECORD COUNT OF 3, IS ISSUED, AND IT IS CHECKED THAT NORMAL TERMINATION OCCURS, WITH EOT=1.
13. A SKIP FILE MARKS REVERSE COMMAND IS ISSUED, WHICH SHOULD SKIP ALL THE WAY TO BOT, AND IT IS CHECKED THAT TAPE STATUS ALERT TERMINATION OCCURS, WITH EOT=0, BOT=1, AND RIB=1.

3696	056430	004737	016330		JSR	PC, WAITF		; WAIT FOR SSR TO SET
3697	056434	016501	000002		MOV	TSSR(R5), R1		; GET TSSR CONTENTS
3698	056440	012702	100204		MOV	#SC!SSR!BIT2, R2		; SET UP EXPECTED
3699	056444	020102			CMP	R1, R2		; ARE THEY EQUAL
3700	056446	001406			BEQ	90#		; BR, IF THEY ARE OK
3701	056450	005237	002214		INC	FATFLG		; ERROR COUNT
3705	056454				ERRHRD	ERRNO, T34ET2, PKTSSR		; WRITE TAPE AT EOT FAILED TO SET TSA
	056454	104456					TRAP	C#ERHRD
	056456	001135					.WORD	605
	056460	061317					.WORD	T34ET2
	056462	012126					.WORD	PKTSSR
3706	056464			90#:	CKLOOP			; LOOP IF SELECTED
	056464	104406					TRAP	C#CLP1
3707	056466	013701	060520		MOV	T34BFR+6, R1		; PICK UP XSTO
3708	056472	010102			MOV	R1, R2		; SET UP EXPECTED
3709	056474	052702	000001		BIS	#BIT0, R2		; SET THE EOT BIT ON IN EXPECTED
3710	056500	020102			CMP	R1, R2		; WAS THE BIT ON
3711	056502	001406			BEQ	100#		; BR, IF EOT WAS FOUND
3712	056504	005237	002214		INC	FATFLG		; ERROR COUNT
3716	056510				ERRHRD	ERRNO, T34ETN, EXPREC		; EOT BIT (XSTO) NOT SET
	056510	104456					TRAP	C#ERHRD
	056512	001136					.WORD	606
	056514	061401					.WORD	T34ETN
	056516	015554					.WORD	EXPREC
3717	056520			100#:	CKLOOP			; LOOP IF SELECTED
	056520	104406					TRAP	C#CLP1
3718	056522	012737	140011	060610	MOV	#140011, T34PK3		; WRITE TAPE MARK, ACK, CVC=1 COMMAND
3719	056530	012704	060610		MOV	#T34PK3, R4		; R4 = POINTER TO PACKET
3720	056534	010465	000000		MOV	R4, TSD8(R5)		; ISSUE COMMAND
3721	056540	004737	016330		JSR	PC, WAITF		; WAIT FOR SSR TO SET
3722	056544	016501	000002		MOV	TSSR(R5), R1		; GET TSSR CONTENTS
3723	056550	012702	100204		MOV	#SC!SSR!BIT2, R2		; SET UP EXPECTED
3724	056554	020102			CMP	R1, R2		; ARE THEY EQUAL
3725	056556	001406			BEQ	110#		; BR, IF STATUS IS GOOD (OK)
3726	056560	005237	002214		INC	FATFLG		; ERROR COUNT
3730	056564				ERRHRD	ERRNO, T34WTH, PKTSSR		; EOT NOT FOUND (USE SHORTER TAPE?)
	056564	104456					TRAP	C#ERHRD
	056566	001137					.WORD	607
	056570	061230					.WORD	T34WTH
	056572	012126					.WORD	PKTSSR
3731	056574			110#:	CKLOOP			; LOOP IF SELECTED
	056574	104406					TRAP	C#CLP1
3732	056576	013701	060520		MOV	T34BFR+6, R1		; PICK UP XSTO
3733	056602	010102			MOV	R1, R2		; SET UP EXPECTED
3734	056604	052702	000001		BIS	#BIT0, R2		; SET THE EOT BIT ON IN EXPECTED
3735	056610	020102			CMP	R1, R2		; WAS THE BIT ON
3736	056612	001406			BEQ	120#		; BR, IF EOT WAS FOUND
3737	056614	005237	002214		JNC	FATFLG		; ERROR COUNT
3741	056620				ERRHRD	ERRNO, T34ETO, EXPREC		; EOT BIT (XSTO) NOT SET
	056620	104456					TRAP	C#ERHRD
	056622	001140					.WORD	608
	056624	060732					.WORD	T34ETO
	056626	015554					.WORD	EXPREC
3742	056630			120#:	CKLOOP			; LOOP IF SELECTED
	056630	104406					TRAP	C#CLP1
3743	056632	012737	141410	060610	MOV	#141410, T34PK3		; SKIP TAPE MARK REVERSE ACK, CVC=1 COMMAND
3744	056640	012737	000001	060612	MOV	#1, T34WB		; SET NUMBER (1) OF TMS TO SKIP

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 6: OPERATIONS AT EOT

SEQ 0177

```

3844 057314 012702 000200      MOV      #SSR,R2      ;SET UP EXPECTED
3845 057320 020102      CMP      R1,R2      ;ARE THEY EQUAL
3846 057322 001406      BEQ      190$      ;BR, IT MIGHT BE END OF TAPE
3847 057324 005237 002214      INC      FATFLG      ;ERROR COUNT
3851 057330      ERRHRD  ERRNO,T34POS,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
      057330 104456      TRAP      C$ERHRD
      057332 001150      .WORD     616
      057334 060644      .WORD     T34POS
      057336 012126      .WORD     PKTSSR
3852 057340      190$:  CKLOOP      ;LOOP IF SELECTED
      057340 104406      TRAP      C$CLP1
3853 057342 013701 060520      MOV      T34BFR+6,R1 ;PICK UP XSTO
3854 057346 010102      MOV      R1,R2      ;SET UP EXPECTED
3855 057350 052702 000001      BIS      @BIT0,R2    ;SET THE EOT BIT ON IN EXPECTED
3856 057354 020102      CMP      R1,R2      ;WAS THE BIT ON
3857 057356 001406      BEQ      200$      ;BR, IF EOT WAS FOUND
3858 057360 005237 002214      INC      FATFLG      ;ERROR COUNT
3862 057364      ERRHRD  ERRNO,T34ETS,EXPREC ;EOT BIT (XSTO) NOT SET
      057364 104456      TRAP      C$ERHRD
      057366 001151      .WORD     617
      057370 061460      .WORD     T34ETS
      057372 015554      .WORD     EXPREC
3863 057374      200$:  CKLOOP      ;LOOP IF SELECTED
      057374 104406      TRAP      C$CLP1
3864 057376 012737 140401 060610      MOV      #140401,T34PK3 ;READ REVERSE, ACK, CVC=1
3865 057404 013737 003116 060612      MOV      FREE,T34RB   ;SET UP WRITE BUFFER ADDRESS
3866 057412 012704 060610      MOV      #T34PK3,R4   ;R4 = POINTER TO PACKET
3867 057416 010465 000000      MOV      R4,TSDB(R5) ;ISSUE COMMAND
3868 057422 004737 016330      JSR      PC,WAITF     ;WAIT FOR SSR TO SET
3869 057426 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
3870 057432 012702 000200      MOV      #SSR,R2     ;SET UP EXPECTED
3871 057436 020102      CMP      R1,R2      ;ARE THEY EQUAL
3872 057440 001406      BEQ      205$      ;BR, ONLY SSR IS SET
3873 057442 005237 002214      INC      FATFLG      ;ERROR COUNT
3877 057446      ERRHRD  ERRNO,T34RRE,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
      057446 104456      TRAP      C$ERHRD
      057450 001152      .WORD     618
      057452 061016      .WORD     T34RRE
      057454 012126      .WORD     PKTSSR
3878 057456      205$:  CKLOOP      ;LOOP IF SELECTED
      057456 104406      TRAP      C$CLP1
3879 057460 012737 140401 060610      MOV      #140401,T34PK3 ;READ REVERSE, ACK, CVC=1
3880 057466 013737 003116 060612      MOV      FREE,T34RB   ;SET UP WRITE BUFFER ADDRESS
3881 057474 012704 060610      MOV      #T34PK3,R4   ;R4 = POINTER TO PACKET
3882 057500 010465 000000      MOV      R4,TSDB(R5) ;ISSUE COMMAND
3883 057504 004737 016330      JSR      PC,WAITF     ;WAIT FOR SSR TO SET
3884 057510 016501 000002      MOV      TSSR(R5),R1 ;GET TSSR CONTENTS
3885 057514 012702 000200      MOV      #SSR,R2     ;SET UP EXPECTED
3886 057520 020102      CMP      R1,R2      ;ARE THEY EQUAL
3887 057522 001406      BEQ      210$      ;BR, IT MIGHT BE END OF TAPE
3888 057524 005237 002214      INC      FATFLG      ;ERROR COUNT
3892 057530      ERRHRD  ERRNO,T34RRE,PKTSSR ;EOT NOT FOUND (USE SHORTER TAPE?)
      057530 104456      TRAP      C$ERHRD
      057532 001153      .WORD     619
      057534 061016      .WORD     T34RRE
      057536 012126      .WORD     PKTSSR
3893 057540      210$:  CKLOOP      ;LOOP IF SELECTED

```

3894	057540	012737	140001	060610	MOV	#140001,T34PK3	;READ DATA, ACK, CVC=1	TRAP	C#CLP1
3895	057550	013737	003116	060612	MOV	FREE,T34RB	;SET UP WRITE BUFFER ADDRESS		
3896	057556	012737	006654	060616	MOV	#3500.,T34SZ	;SET UP BUFFER SIZE (4K BYTES)		
3897	057564	012704	060610		MOV	#T34PK3,R4	;R4 = POINTER TO PACKET		
3898	057570	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND		
3899	057574	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
3900	057600	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
3901	057604	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED		
3902	057610	020102			CMP	R1,R2	;ARE THEY EQUAL		
3903	057612	001406			BEQ	230#	;BR, IT MIGHT BE END OF TAPE		
3904	057614	005237	002214		INC	FATFLG	;ERROR COUNT		
3908	057620				ERRHRD	ERRNO,T34RRE,PKTSSR	;EOT NOT FOUND (USE SHORTER TAPE?)		
	057620	104456						TRAP	C#ERHRD
	057622	001154						.WORD	620
	057624	061016						.WORD	T34RRE
	057626	012126						.WORD	PKTSSR
3909	057630		230#:		CKLOOP		;LOOP IF SELECTED		
	057630	104406						TRAP	C#CLP1
3910	057632	012737	140001	060610	MOV	#140001,T34PK3	;READ DATA, ACK, CVC=1		
3911	057640	013737	003116	060612	MOV	FREE,T34RB	;SET UP WRITE BUFFER ADDRESS		
3912	057646	012737	006654	060616	MOV	#3500.,T34SZ	;SET UP BUFFER SIZE (4K BYTES)		
3913	057654	012704	060610		MOV	#T34PK3,R4	;R4 = POINTER TO PACKET		
3914	057660	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND		
3915	057664	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
3916	057670	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
3917	057674	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED		
3918	057700	020102			CMP	R1,R2	;ARE THEY EQUAL		
3919	057702	001406			BEQ	235#	;BR, IT MIGHT BE END OF TAPE		
3920	057704	005237	002214		INC	FATFLG	;ERROR COUNT		
3924	057710				ERRHRD	ERRNO,T34RRE,PKTSSR	;EOT NOT FOUND (USE SHORTER TAPE?)		
	057710	104456						TRAP	C#ERHRD
	057712	001155						.WORD	621
	057714	061016						.WORD	T34RRE
	057716	012126						.WORD	PKTSSR
3925	057720		235#:		CKLOOP		;LOOP IF SELECTED		
	057720	104406						TRAP	C#CLP1
3926	057722	013701	060520		MOV	T34BFR+6,R1	;PICK UP XSTO		
3927	057726	010102			MOV	R1,R2	;SET UP EXPECTED		
3928	057730	052702	000001		BIS	#BIT0,R2	;SET THE EOT BIT ON IN EXPECTED		
3929	057734	020102			CMP	R1,R2	;WAS THE BIT ON		
3930	057736	001406			BEQ	240#	;BR, IF EOT WAS FOUND		
3931	057740	005237	002214		INC	FATFLG	;ERROR COUNT		
3935	057744				ERRHRD	ERRNO,T34ETZ,EXPREC	;EOT BIT (XSTO) NOT SET		
	057744	104456						TRAP	C#ERHRD
	057746	001156						.WORD	622
	057750	061552						.WORD	T34ETZ
	057752	015554						.WORD	EXPREC
3936	057754		240#:		CKLOOP		;LOOP IF SELECTED		
	057754	104406						TRAP	C#CLP1
3937	057756	012737	140410	060610	MOV	#140410,T34PK3	;SPACE RECORDS REVERSE, ACK, CVC=1 CMD.		
3938	057764	012737	000005	060612	MOV	#5,T34RB	;NUMBER OF RECORDS TO SPACE		
3939	057772	012704	060610		MOV	#T34PK3,R4	;R4 = POINTER TO PACKET		
3940	057776	010465	000000		MOV	R4,TSDB(R5)	;ISSUE COMMAND		
3941	060002	004737	016330		JSR	PC,WAITF	;WAIT FOR SSR TO SET		
3942	060006	016501	000002		MOV	TSSR(R5),R1	;GET TSSR CONTENTS		
3943	060012	012702	000200		MOV	#SSR,R2	;SET UP EXPECTED		

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 6: OPERATIONS AT EOT

SEQ 0181

```

4036 060452 000137 056022          JMP      T34LOOP      ;EXECUTE AGAIN
4037 060456          1638:  EXIT      TST          ;ALL DONE THIS TEST
      060456 104432
      060460 002662
4038                                ;*
4039                                ;LOCAL STORAGE FOR THIS TEST
4040                                ;
4042                                ;=<. *10>E177770
4044 060470          T34PACKET:          ;COMMAND PACKET FOR TEST
      060470 100004          .WORD      100004          ;WRITE CHARACTERISTICS COMMAND, WITH ACK
4046 060472 060500          .WORD      T34DATA          ;ADDRESS OF CHARACTERISTICS BLOCK
4047 060474 000000          .WORD      0
4048 060476 000010          .WORD      8.
4049 060500          T34DATA:          ;STARTING VALUE OF BLOCK SIZE
      060500 060512          .WORD      T34BFR          ;CHARACTERISTICS DATA BLOCK
4051 060502 000000          .WORD      0          ;ADDRESS OF MESSAGE BUFFER
4052 060504 000012          .WORD      10.
4053 060506 000000          .WORD      0          ;LENGTH OF MESSAGE BUFFER
4054 060510 000000          T34DSW: .WORD      0          ;SELECT DRIVE 0
4055 060512          T34BFR: .BLKW     25.          ;MESSAGE BUFFER
4056                                ;
4057                                ;WRITE SUBSYSTEM MEMORY COMMAND PACKET
4058                                ;
4060                                ;=<. *10>E177770
4062 060600          T34PK2:          ;WRITE SUB SYS MEM COMMAND, AND ACK
      060600 100006          .WORD      100006          ;ADDRESS OF SELECT BLOCK DATA
4064 060602 060626          .WORD      T34BF2
4065 060604 000000          .WORD      0
4066 060606 000006          .WORD      6.          ;SIZE OF DATA PACKET
4067                                ;
4071 060610          T34PK3:          ;WRITE COMMAND, AND ACK
      060610 100005          .WORD      100005
4073 060612          T34RB:          ;ADDRESS OF WRITE/READ BUFFER
      060612 000000          .WORD      0
4074 060614 000000          .WORD      0
4075 060614 000000          T34SZ: .WORD      0          ;SIZE OF BUFFER (EXTENT)
4076 060616 000000          .EVEN
4077                                ;
4078                                ;
4079 060620 000000          T34RSZ: .WORD      0          ;LARGEST TAPE RECORD IN BYTES
4080 060622 000000          T34CNT: .WORD      0          ;TAPE RECORD COUNTER
4081 060624 000000          T34DLY: .WORD      0          ;DELAY COUNTER
4082                                ;
4083                                ;
4084 060626          T34BF2:          ;BSEL0 AREA
      060626 010          T34BS0: .BYTE      10          ;BSEL1 AREA
4086 060627 0200          T34BS1: .BYTE      200          ;SEL 2 AREA
4087 060630 000000          T34S2: .WORD      0          ;DATA AREA
4088 060632 000000          T34S3: .WORD      0
4089                                ;
4090                                ;
4091                                ;EVEN
4092                                ;TAPE MOTION PACKET COMMAND VALUES
4093                                ;
4094 060634 100005          T34WD: .WORD      100005          ;WRITE DATA (NEXT)
4095 060636 100405          T34WDR: .WORD      100405          ;WRITE DATA RETRY
4096 060640 102005          T34CON: .WORD      102005          ;WRITE CONTINUOUS
4097 060642 177777          .WORD      177777          ;END OF DATA

```

```

4098
4099 ;*
4100 ;LOCAL TEXT MESSAGES FOR TEST
4101 ;
4102 060644 124 123 123 T34POS: .ASCIZ 'TSSR Incorrect After Position (SPACE RECORDS) Command'
4103 060732 127 122 111 T34ETO: .ASCIZ 'WRITE TAPE MARK At EOT Failed To Set EOT Bit (XSTO)'
4104 061016 122 105 101 T34RRE: .ASCIZ 'READ Command At EOT Didn't Give Normal Termination (TSSR)'
4105 061107 125 156 141 T34ETC: .ASCIZ 'Unable To Clear EOT Indication, (XSTO) Bit 0'
4106 061164 122 105 127 T34BOT: .ASCIZ 'REWIND Failed To Set BOT (XSTO) Bit'
4107 061230 127 122 111 T34WTM: .ASCIZ 'WRITE TAPE MARK At EOT Failed To Set Tape Status Alert'
4108 061317 127 122 111 T34ET2: .ASCIZ 'WRITE DATA At EOT Failed To Set Tape Status Alert'
4109 061401 127 122 111 T34ETN: .ASCIZ 'WRITE DATA At EOT Failed To Set EOT Bit (XSTO)'
4110 061460 123 120 101 T34ETS: .ASCIZ 'SPACE RECORDS FORWARD At EOT Failed To Set EOT Bit (XSTO)'
4111 061552 122 105 101 T34ETZ: .ASCIZ 'READ DATA At EOT Failed To Set EOT Bit (XSTO)'
4112 061630 124 123 123 T34STM: .ASCIZ 'TSSR Incorrect After SKIP TAPE MARK REVERSE At EOT'
4113 061713 120 117 123 T34TMK: .ASCIZ 'POSITION Command At EOT Onto Tape Mark Failed To Set TMK (XSTO)'
4114 062013 127 122 111 T34SSR: .ASCIZ 'WRITE Command Not Accepted'
4115 062046 105 117 124 T34ET: .ASCIZ 'EOT Not Found In 65000 3.5K Writes, (Use Shorter Tape)'
4116 062135 127 122 111 T34EOY: .ASCIZ 'WRITE DATA OVER EOT GAVE NO TAPE STATUS ALERT'
4117 062213 124 123 123 T34TM: .ASCIZ 'TSSR Not Correct After WRITE Command Reject'
4118 062267 122 145 167 T34RMN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
4119 062336 122 101 115 T34RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
4120 062411 124 123 123 T34AM3: .ASCIZ 'TSSR Init. Failed After WRITE Command'
4121 062457 104 162 151 T34OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
4122 062532 124 123 123 T34WDD: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, SWB Bit Set'
4123 062621 124 123 123 T34WDC: .ASCIZ 'TSSR Not Correct After WRITE DATA Command, Check For Tape Offline'
4124 062723 103 126 103 T34VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
4125 062776 124 123 102 T34BA: .ASCIZ 'TSBA Not Correct After WRITE DATA Command'
4126 063050 127 122 111 T34WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
4127 063137 117 160 145 TST34ID: .ASCIZ 'Operations At EOT'
4128 .EVEN
4129 ;*
4130 ;
4131 ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
4132 ;WRITE SUBSYSTEM MEMORY COMMAND
4133 ;
4134 ;-
4135
4136 063162 T34REST:
4137 063162 SAVREG ;SAVE THE REGISTERS
4138 063166 012701 060470 MOV #T34PACKET,R1 ;START OF THE PACKET
4139 063172 012721 100004 MOV #100004,(R1)+ ;WRITE SUBSYSTEM MEM. WITH ACK
4140 063176 012721 060500 MOV #T34DATA,(R1)+ ;ADDRESS OF CHARAISTICS DATA BLOCK
4141 063202 005021 CLR (R1)+ ;EXTENDED ADDRESS
4142 063204 012721 000012 MOV #10,(R1)+ ;SIZE OF DATA BLOCK IN BYTES
4143 063210 012721 060512 MOV #T34BFR,(R1)+ ;ADDRESS OF MESSAGE BUFFER
4144 063214 005021 CLR (R1)+
4145 063216 012721 000024 MOV #20,(R1)+ ;LENGTH OF MESSAGE BUFFER
4146 063222 005021 CLR (R1)+
4147 063224 012711 000000 MOV #0,(R1) ;SELECT DRIVE ZERO
4148 063230 012702 000030 MOV #24,R2 ;NUMBER OF LOCATIONS TO BE CLEARED
4149 063234 012762 177777 060512 64$: MOV #177777,T34BFR(R2) ;ALL ONES TO MESSAGE BUFFER
4150 C 3242 005742 TST -(R2) ;BUMP DOWN TO NEXT LOCATION
4151 063244 020227 000000 CMP R2,#0 ;R2 AT ZERO YET
4152 063250 001371 BNE 64$ ;KEEP GOING UNTIL DONE
4153 063252 000207 RTS PC ;RETURN
4154
    
```



```

4260 063572          ERRHRD  ERRNO,T35RWN,PKTSSR      ;REWIND NOT ACCEPTED
      063572 104456
      063574 001277
      063576 070574
      063600 012126
      TRAP      C#ERRHRD
      .WORD     703
      .WORD     T35RWN
      .WORD     PKTSSR
4261 063602          30#:  CKLOOP                    ;LOOP IF SELECTED
      063602 104406
      TRAP      C#CLP1
4262 063604 013701 067350      MOV      T35RFR*6,R1      ;PICK UP XSTO
4263 063610 010102          MOV      R1,R2           ;SET UP EXPECTED
4264 063612 052702 000002      BIS      @BIT1,R2       ;SET BOT BIT IN EXPECTED
4265 063616 020102          CMP      R1,R2           ;DOES EXP = REC'D
4266 063620 001406          BEQ      40#            ;BR, IF EQUAL (OK)
4267 063622 005237 002214      INC      FATFLG         ;ERROR COUNT
4271 063626          ERRHRD  ERRNO,T35BOT,EXPREC      ;TAPE NOT AT BOT AFTER REWIND
      063626 104456
      TRAP      C#ERRHRD
      063630 001300
      .WORD     704
      063632 070270
      .WORD     T35BOT
      063634 015554
      .WORD     EXPREC
4272 063636          40#:  CKLOOP                    ;LOOP IF SELECTED
      063636 104406
      TRAP      C#CLP1
4273 063640 012703 000024      MOV      @20.,R3        ;NUMBER OF RECORDS
4274 063644 012737 000400 067446  MOV      @256.,T35SZ     ;SET UP RECORD SIZE
4275 063652 013737 003116 067442  MOV      FREE,T35WB      ;ADDRESS OF WRITE BUFFER
4276
4277
4278
4279
4280
4281
4282
      ;*****
      ;WRITE DATA,ACK,CVC=1 COMMAND
      ;*****
4283 063660 012737 140005 067440      MOV      @140005,T35PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
4284 063666 012704 067440          MOV      @T35PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4285 063672 010465 000000          50#:  MOV      R4,T35DB(R5) ;ISSUE COMMAND
4286 063676 004737 016330          JSR      PC,WAITF        ;WAIT FOR SSR TO SET
4287 063702 016501 000002          MOV      T35R(R5),R1     ;GET T35R CONTENTS
4288 063706 012702 000200          MOV      @SSR,R2         ;SET UP EXPECTED
4289 063712 020102          CMP      R1,R2           ;ARE THEY EQUAL
4290 063714 001406          BEQ      60#            ;BR, IF OK
4291 063716 005237 002214      INC      FATFLG         ;ERROR COUNT
4295 063722          ERRHRD  ERRNO,T35WDE,PKTSSR      ;T35R INCORRECT AFTER WRITE DATA
      063722 104456
      TRAP      C#ERRHRD
      063724 001301
      .WORD     705
      063726 070216
      .WORD     T35WDE
      063730 012126
      .WORD     PKTSSR
4296 063732          60#:  CKLOOP                    ;LOOP IF SELECTED
      063732 104406
      TRAP      C#CLP1
4297 063734 005303          DEC      R3              ;BUMP RECORD COUNTER
4298 063736 001355          BNE     50#              ;BR, IF MORE RECORDS TO COUNT
4299
4300
4301
4302
4303
4304
4305
      ;*****
      ;WAIT FOR TAPE TO STOP ALL MOTION
      ;*****
4306 063740 012737 000012 067472      MOV      @10.,T35DLY     ;SET UP DELAY COUNTER
4307 063746          70#:  DELAY      250      ;WAIT ABOUT .25 SEC

```

TEST 1 HARDWARE TEST 1-8 TEST MACRO M1113 14-JUN-84 16:41
 TEST 7: EXTENDED MODE FEATURES

SEQ 0186

063746	012727	000250				MOV	#250,(PC),	
063752	000000					.WORD	0	
063754	013727	002116				MOV	L#DLY,(PC).	
063760	000000					.WORD	0	
063762	005367	177772				DEC	6(PC)	
063766	001375					BNE	. 4	
063770	005367	177756				DEC	22(PC)	
063774	001367					BNE	. 20	
4308	063776	005337	067472			DEC	T35DLY	;BUMP COUNTER DOWN
4309	064002	001361				BNE	70#	;BR, IF MORE TO DELAY
4310	064004	005737	002220			TST	EXTFEA	;CHECK FOR EXTENDED FEATURES SW SWITCH
4311	064010	001042				BNE	110#	;BR IF SWITCH IS ON
4312	064012	112737	000200	067451		MOVR	#200,T358S1	;WRITE MISCELLANEOUS CONT/READ STATUS
4313	064020	112737	000010	067450		MOVB	#10,T358S0	;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4314	064026	012704	067430			MOV	#T35PK2,R4	;WRITE SUBSYS MEM PACKET
4315	064032	010465	000000			MOV	R4,TSD8(R5)	;ISSUE COMMAND
4316	064036	004737	016416			JSR	PC,CHKTSSR	;WAIT FOR SSR
4317	064042	103407				BCS	90#	;BR, IF NO ERROR
4318	064044	010001				MOV	R0,R1	;ERROR, SAVE TSSR
4319	064046	005237	002214			INC	FATFLG	;ERROR COUNT
4323	064052					ERRHRD	ERRNO,T35SSR,PKTSSR	;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
	064052	104456				TRAP	C#ERHRD	
	064054	001302				.WORD	706	
	064056	072352				.WORD	T35SSR	
	064060	012126				.WORD	PKTSSR	
4324	064062				90#:	CKLOOP		;LOOP IF SELECTED
	064062	104406						TRAP C#CLP1
4325	064064	012704	067320			MOV	#T35PACKET,R4	;SUBROUTINE NEEDS PACKET ADDRESS
4326	064070	004737	010742			JSR	PC,WRTCHR	;ISSUE WRITE CHARACTERISTICS
4327	064074	103407				BCS	100#	;BR, IF COMMAND ISSUED OK
4328	064076	005237	002214			INC	FATFLG	;ERROR COUNT
4332	064102	010001				MOV	R0,R1	;SAVE CONTENTS OF TSSR
4333	064104					ERRHRD	ERRNO,WRTMSG,SFIMSG	;WRITE CHARACTERISTIC SC FAILED
	064104	104456				TRAP	C#ERHRD	
	064106	001303				.WORD	707	
	064110	005052				.WORD	WRTMSG	
	064112	012114				.WORD	SFIMSG	
4334	064114				100#:	CKLOOP		;SCOPE LOOP
	064114	104406						TRAP C#CLP1
4335	064116	012737	176750	067472	110#:	MOV	#65000.,T35DLY	;SET UP DELAY COUNTER
4336	064124	005037	067466			CLR	T35CNT	;DELAY COUNTER
4337								
4338								
4339								
4340								
4341								
4342								
4343								
4344	064130	012737	142012	067440		MOV	#142012,T35PK3	;REWIND IMED. INTERRUPT,ACK,CVC=1,IE=0 COMMAND
4345	064136	012704	067440			MOV	#T35PK3,R4	;SET UP R4 WITH PACKET ADDRESS
4346	064142	010465	000000			MOV	R4,TSD8(R5)	;ISSUE COMMAND
4347	064146	016501	000002		120#:	MOV	TSSR(R5),R1	;GET TSSR CONTENTS
4348	064152	032701	000200			BIT	#SSR,R1	;CHECK FOR SSR SET
4349	064156	001021				BNE	130#	;BR, WHEN SSR IS SET
4350	064160	005237	067466			INC	T35CNT	;BUMP THE CYCLE COUNTER
4351	064164					DELAY	1	;DELAY TO KEEP COUNTER DOWN
	064164	012727	000001					MOV #1,(PC).

064170	000000					.WORD	0
064172	013727	002116				MOV	L#DLY,(PC).
064176	000000					.WORD	0
064200	005367	177772				DEC	6(PC)
064204	001375					BNE	. 4
064206	005367	177756				DEC	-22(PC)
064212	001367					BNE	. 20
4352	064214	005337	067472		DEC	T35DLY	;DROP DEAD TIMER BUMP DOWN
4353	064220	001352			BNE	120\$;BR, IF MORE TIME TO GO
4354	064222	012702	000200	130\$:	MOV	#SSR,R2	;SET UP EXPECTED
4355	064226	020102			CMP	R1,R2	;ARE THEY EQUAL
4356	064230	001406			BEQ	140\$;BR, IF OK
4357	064232	005237	002214		INC	FATFLG	;ERROR COUNT
4361	064236				ERRHRD	ERRNO,T35RWE,PKTSSR	;TSSR INCORRECT AFTER WRITE DATA
	064236	104456				TRAP	C#ERHRD
	064240	001304				.WORD	708
	064242	072720				.WORD	T35RWE
	064244	012126				.WORD	PKTSSR
4362	064246			140\$:	CKLOOP		;LOOP IF SELECTED
	064246	104406				TRAP	C#CLP1
4363	064250	005737	002216		TST	INTRECV	;CHECK FOR INTERRUPTS
4364	064254	001410			BEQ	150\$;BR, IF NO INTERRUPTS DETECTED
4365	064256	016501	000002		MOV	TSSR(R5),R1	;GET TSSR STATUS FOR PRINTOUT
4366	064262	005237	002214		INC	FATFLG	;ERROR COUNT
4370	064266				ERRHRD	ERRNO,T35INT,PKTSSR	;INTERRUPT RECEIVED (BAD)
	064266	104456				TRAP	C#ERHRD
	064270	001305				.WORD	709
	064272	072531				.WORD	T35INT
	064274	012126				.WORD	PKTSSR
4371	064276			150\$:	CKLOOP		;LOOP IF SELECTED
	064276	104406				TRAP	C#CLP1
4372							
4373							
4374							
4375							
4376							
4377							
4378							
4379	064300	013701	067350		MOV	T35BFR+6,R1	;PICK UP XST0
4380	064304	010102			MOV	R1,R2	;SET UP EXPECTED
4381	064306	052702	000200		BIS	#BIT7,R2	;SET MOT BIT IN EXPECTED
4382	064312	020102			CMP	R1,R2	;DOES EXP = REC'D
4383	064314	001406			BEQ	160\$;BR, IF EQUAL (OK)
4384	064316	005237	002214		INC	FATFLG	;ERROR COUNT
4388	064322				ERRHRD	ERRNO,T35MOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND
	064322	104456				TRAP	C#ERHRD
	064324	001306				.WORD	710
	064326	072433				.WORD	T35MOT
	064330	015554				.WORD	EXPREC
4389	064332			160\$:	CKLOOP		;LOOP IF SELECTED
	064332	104406				TRAP	C#CLP1
4390	064334	013701	067354		MOV	T35BFR+12,R1	;PICK UP XST2
4391	064340	010102			MOV	R1,R2	;SET UP EXPECTED
4392	064342	052702	100000		BIS	#BIT15,R2	;SET OPM BIT IN EXPECTED
4393	064346	020102			CMP	R1,R2	;DOES EXP = REC'D
4394	064350	001406			BEQ	170\$;BR, IF EQUAL (OK)
4395	064352	005237	002214		INC	FATFLG	;ERROR COUNT

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 7: EXTENDED MODE FEATURES

SEQ 0189

064522	012727	000250				MOV	#250,(PC).	
064526	000000					.WORD	0	
064530	013727	002116				MOV	L#DLY,(PC).	
064534	000000					.WORD	0	
064536	005367	177772				DEC	-6(PC)	
064542	001375					BNE	. 4	
064544	005367	177756				DEC	-22(PC)	
064550	001367					BNE	.-20	
4437	064552	005337	067472		DEC	T35DLY		;BUMP COUNTER
4438	064556	001356			BNE	10#		;BR, IF COUNTER NOT DONE
4439	064560	005237	002214		INC	FATFLG		;ERROR COUNT
4443	064564	010001			MOV	R0,R1		;CONTENTS OF TSSR REGISTER
4444	064566				ERRDF	ERRNO,SFIERR,SFIMSG		;FATAL ERROR TSSR WAS NOT OK
	064566	104455						TRAP
	064570	001310						.WORD
	064572	003646						.WORD
	064574	012114						.WORD
4445	064576	013737	002174	067340	20#:	MOV	UNITN,T35DSW	;SET UP DRIVE NUMBER
4446	064604	012704	067320			MOV	#T35PACKET,R4	;SUBROUTINE NEEDS PACKET ADDRESS
4447	064610	004737	010742			JSR	PC,WRTCHR	;ISSUE WRITE CHARACTERISTICS
4448	064614	103407				BCS	25#	;BR, IF COMMAND ISSUED OK
4449	064616	005237	002214			INC	FATFLG	;ERROR COUNT
4453	064622	010001				MOV	R0,R1	;SAVE CONTENTS OF TSSR
4454	064624					ERRHRD	ERRNO,WRTMSG,SFIMSG	;WRITE CHARACTERISTIC FAILED
	064624	104456						TRAP
	064626	001311						.WORD
	064630	005052						.WORD
	064632	012114						.WORD
4455	064634				25#:	CKLOOP		;LOOP IF SELECTED
	064634	104406						TRAP
4456	064636	004737	011074			JSR	PC,REWIND	;CALL TAPE REWIND COMMAND
4457	064642	103411				BCS	30#	;BR, IF NO PROBLEM
4458	064644	010004				MOV	R0,R4	;SET UP REWIND PACKET ADDRESS
4459	064646	016501	000002			MOV	TSSR(R5),R1	;GET TSSR CONTENTS
4460	064652	005237	002214			INC	FATFLG	;ERROR COUNT
4464	064656					ERRHRD	ERRNO,T35RWN,PKTSSR	;REWIND NOT ACCEPTED
	064656	104456						TRAP
	064660	001312						.WORD
	064662	070574						.WORD
	064664	012126						.WORD
4465	064666				30#:	CKLOOP		;LOOP IF SELECTED
	064666	104406						TRAP
4466	064670	013701	067350			MOV	T35BFR+6,R1	;PICK UP XSTO
4467	064674	010102				MOV	R1,R2	;SET UP EXPECTED
4468	064676	052702	000002			BIS	#BIT1,R2	;SET BOT BIT IN EXPECTED
4469	064702	020102				CMP	R1,R2	;DOES EXP = REC'D
4470	064704	001406				BEQ	40#	;BR, IF EQUAL (OK)
4471	064706	005237	002214			INC	FATFLG	;ERROR COUNT
4475	064712					ERRHRD	ERRNO,T35BOT,EXPREC	;TAPE NOT AT BOT AFTER REWIND
	064712	104456						TRAP
	064714	001313						.WORD
	064716	070270						.WORD
	064720	015554						.WORD
4476	064722				40#:	CKLOOP		;LOOP IF SELECTED
	064722	104406						TRAP
4477	064724	012703	000024			MOV	#20.,R3	;NUMBER OF RECORDS
4478	064730	012737	000400	067446		MOV	#256.,T35SZ	;SET UP RECORD SIZE

```

4479 064736 013737 003116 067442      MOV      FREE,T35WB      ;ADDRESS OF WRITE BUFFER
4480
4481      ;*****
4482      ;
4483      ;WRITE DATA,ACK,CVC=1 COMMAND
4484      ;
4485      ;*****
4486
4487 064744 012737 140005 067440      MOV      #140005,T35PK3  ;WRITE DATA,ACK,CVC=1 COMMAND
4488 064752 012704 067440      MOV      #T35PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4489 064756 010465 000000      50$:    MOV      R4,TSDB(R5)     ;ISSUE COMMAND
4490 064762 004737 016330      JSR      PC,WAITF        ;WAIT FOR SSR TO SET
4491 064766 016501 000002      MOV      TSSR(R5),R1     ;GET TSSR CONTENTS
4492 064772 012702 000200      MOV      #SSR,R2        ;SET UP EXPECTED
4493 064776 020102      CMP      R1,R2          ;ARE THEY EQUAL
4494 065000 001406      BEQ      60$            ;BR, IF OK
4495 065002 005237 002214      INC      FATFLG          ;ERROR COUNT
4499 065006      ERRHRD  ERRNO,T35WDE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
                                TRAP      C$ERHRD
                                .WORD    716
                                .WORD    T35WDE
                                .WORD    PKTSSR
                                TRAP      C$CLP1
065006 104456
065010 001314
065012 070216
065014 012126
4500 065016      60$:    CKLOOP          ;LOOP IF SELECTED
065016 104406
4501
4502      ;*****
4503      ;
4504      ;WAIT FOR TAPE TO STOP ALL MOTION
4505      ;
4506      ;*****
4507
4508 065020 012737 000012 067472      70$:    MOV      #10.,T35DLY  ;SET UP DELAY COUNTER
4509 065026      DELAY  250            ;WAIT ABOUT .25 SEC
                                MOV      #250,(PC)+
                                .WORD    0
                                MOV      L$DLY,(PC)+
                                .WORD    0
                                DEC      6(PC)
                                BNE     . 4
                                DEC      22(PC)
                                BNE     .-20
065026 012727 000250
065032 000000
065034 013727 002116
065040 000000
065042 005367 177772
065046 001375
065050 005367 177756
065054 001367
4510 065056 005337 067472      DEC      T35DLY          ;BUMP COUNTER DOWN
4511 065062 001361      BNE     70$            ;BR, IF MORE TO DELAY
4512 065064 005737 002220      TST     EXTFEA          ;CHECK FOR EXTENDED FEATURES SW SWITCH
4513 065070 001042      BNE     110$           ;BR IF SWITCH IS ON
4514 065072 112737 000200 067451      MOVB    #200,T35B51     ;WRITE MISCELLANEOUS CONT/READ STATUS
4515 065100 112737 000010 067450      MOVB    #10,T35B50     ;FUNCTION SELECTION BIT (TURN ON EXTFEA HW SWITCH)
4516 065106 012704 067430      MOV     #T35PK2,R4     ;WRITE SUBSYS MEM PACKET
4517 065112 010465 000000      MOV     R4,TSDB(R5)    ;ISSUE COMMAND
4518 065116 004737 016416      JSR     PC,CHKTSSR     ;WAIT FOR SSR
4519 065122 103407      BCS     90$           ;BR, IF NO ERROR
4520 065124 010001      MOV     R0,R1          ;ERROR, SAVE TSSR
4521 065126 005237 002214      INC     FATFLG          ;ERROR COUNT
4525 065132      ERRHRD  ERRNO,T35SSR,PKTSSR ;TSSR NOT CORRECT AFTER WRT. MISCELLANEOUS
                                TRAP      C$ERHRD
                                .WORD    717
                                .WORD    T35SSR
065132 104456
065134 001315
065136 072352
    
```



```

065140 012126
4526 065142          90$:  CKLOOP          ;LOOP IF SELECTED          .WORD  PKTSSR
      065142 104406          TRAP  C$CLP1
4527 065144 012704 067320  MOV    #T3SPACKET,R4      ;SUBROUTINE NEEDS PACKET ADDRESS
4528 065150 004737 010742  JSR    PC,WRTCHR         ;ISSUE WRITE CHARACTERISTICS
4529 065154 103407          BCS    100$              ;BR, IF COMMAND ISSUED OK
4530 065156 005237 002214  INC    FATFLG            ;ERROR COUNT
4534 065162 010001          MOV    R0,R1             ;SAVE CONTENTS OF TSSR
4535 065164          ERRHRD  ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTICSC FAILED
      065164 104456          TRAP  C$ERHRD
      065166 001316          .WORD  718
      065170 005052          .WORD  WRTMSG
      065172 012114          .WORD  SFIMSG
4536 065174          100$:  CKLOOP          ;SCOPE LOOP
      065174 104406          TRAP  C$CLP1
4537 065176 012737 176750 067472 110$:  MOV    #65000.,T35DLY  ;SET UP DELAY COUNTER
4538 065204 005037 067466  CLR    T35CNT           ;DELAY COUNTER
4539
4540          ;*****
4541          ;
4542          ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
4543          ;
4544          ;*****
4545
4546 065210 012737 142212 067440  MOV    #142212,T35PK3    ;REWIND IMMEDIATE,ACK,CVC=1,IE=1 COMMAND
4547 065216 012704 067440  MOV    #T35PK3,R4        ;SET UP R4 WITH PACKET ADDRESS
4548 065222 010465 000000  MOV    R4,TSD8(R5)       ;ISSUE COMMAND
4549 065226 016501 000002 120$:  MOV    TSSR(R5),R1     ;GET TSSR CONTENTS
4550 065232 032701 000200  BIT    #SSR,R1           ;CHECK FOR SSR SET
4551 065236 001021          BNE    130$              ;BR, WHEN SSR IS SET
4552 065240 005237 067466  INC    T35CNT           ;BUMP THE CYCLE COUNTER
4553 065244          DELAY  1                 ;DELAY TO KEEP COUNTER DOWN
      065244 012727 000001          MOV    #1.(PC),
      065250 000000          .WORD  0
      065252 013727 002116          MOV    L$DLY,(PC),
      065256 000000          .WORD  0
      065260 005367 177772          DEC    -6(PC)
      065264 001375          BNE    . 4
      065266 005367 177756          DEC    -22(PC)
      065272 001367          BNE    . 20
4554 065274 005337 067472  DEC    T35DLY           ;DROP DEAD TIMER BUMP DOWN
4555 065300 001352          BNE    120$              ;BR, IF MORE TIME TO GO
4556 065302 012702 000200 130$:  MOV    #SSR,R2         ;SET UP EXPECTED
4557 065306 020102          CMP    R1,R2             ;ARE THEY EQUAL
4558 065310 001406          BEQ    140$              ;BR, IF OK
4559 065312 005237 002214  INC    FATFLG            ;ERROR COUNT
4563 065316          ERRHRD  ERRNO,T35RWE,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
      065316 104456          TRAP  C$ERHRD
      065320 001317          .WORD  719
      065322 072720          .WORD  T35RWE
      065324 012126          .WORD  PKTSSR
4564 065326          140$:  CKLOOP          ;LOOP IF SELECTED
      065326 104406          TRAP  C$CLP1
4565 065330 005737 002216  TST    INTRECV           ;CHECK FOR INTERRUPTS
4566 065334 001010          BNE    150$              ;BR, IF INTERRUPTS DETECTED
4567 065336 016501 000002  MOV    TSSR(R5),R1     ;GET TSSR STATUS FOR PRINTOUT
4568 065342 005237 002214  INC    FATFLG            ;ERROR COUNT
  
```


M11

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
TEST 7: EXTENDED MODE FEATURES

SEQ 0194

```

065656 001325 .WORD 725
065660 070574 .WORD T35RWN
065662 012126 .WORD PKTSSR
4664 065664 30$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
065664 104406
4665 065666 013701 067350 MOV T35BFR+6,R1 ;PICK UP XSTO
4666 065672 010102 MOV R1,R2 ;SET UP EXPECTED
4667 065674 052702 000002 BIS #BIT1,R2 ;SET BOT BIT IN EXPECTED
4668 065700 020102 CMP R1,R2 ;DOES EXP = REC'D
4669 065702 001406 BEQ 40$ ;BR, IF EQUAL (OK)
4670 065704 005237 002214 INC FATFLG ;ERROR COUNT
4674 065710 ERRHRD ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
065710 104456 TRAP C$ERHRD
065712 001326 .WORD 726
065714 070270 .WORD T35BOT
065716 015554 .WORD EXPREC
4675 065720 40$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
065720 104406
4676 065722 012703 000024 MOV #20.,R3 ;STARTING RECORD SIZE
4677 065726 013737 003116 067442 MOV FREE,T35WB ;STARTING WRITE BUFFER ADDRESS
4678
4679 ;*****
4680 ;
4681 ;WRITE DATA,CVC=1,ACK COMMAND
4682 ;
4683 ;*****
4684
4685 065734 012737 140005 067440 65$: MOV #140005,T35PK3 ;WRITE DATA,CVC=1,ACK COMMAND
4686 065742 012704 067440 MOV #T35PK3,R4 ;SET UP R4 WITH PACKET ADDRESS
4687 065746 010300 MOV R3,R0 ;SET PATTERN IN CORRECT REGISTER
4688 065750 004737 017502 JSR PC,FILLMEM ;FILL MEMORY WITH RECORD SIZE
4689 065754 010337 067446 MOV R3,T35SZ ;SET UP RECORD SIZE IN PACKET
4690 065760 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
4691 065764 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
4692 065770 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS
4693 065774 012702 000200 MOV #SSR,R2 ;SET UP EXPECTED
4694 066000 020102 CMP R1,R2 ;ARE THEY EQUAL
4695 066002 001406 BEQ 80$ ;BR, IF OK
4696 066004 005237 002214 INC FATFLG ;ERROR COUNT
4700 066010 ERRHRD ERRNO,T35WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
066010 104456 TRAP C$ERHRD
066012 001327 .WORD 727
066014 071130 .WORD T35WDC
066016 012126 .WORD PKTSSR
4701 066020 80$: CKLOOP ;LOOP IF SELECTED TRAP C$CLP1
066020 104406
4702
4703 ;*****
4704 ;
4705 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
4706 ;
4707 ;*****
4708
4709 066022 012737 141005 067440 MOV #141005,T35PK3 ;WRITE DATA RETRY,CVC=1,ACK COMMAND
4710 066030 010465 000000 MOV R4,TSDB(R5) ;ISSUE COMMAND
4711 066034 004737 016330 JSR PC,WAITF ;WAIT FOR SSR TO SET
4712 066040 016501 000002 MOV TSSR(R5),R1 ;GET TSSR CONTENTS

```

```

4713 066044 012702 000200      MOV     #SSR,R2      ;SET UP EXPECTED
4714 066050 020102      CMP     R1,R2       ;ARE THEY EQUAL
4715 066052 001406      BEQ     90$         ;BR, IF OK
4716 066054 005237 002214      INC     FATFLG      ;ERROR COUNT
4720 066060      ERRHRD  ERRNO,T35WRF,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA RETRY
      066060 104456      TRAP   C$ERHRD
      066062 001330      .WORD  728
      066064 072175      .WORD  T35WRF
      066066 012126      .WORD  PKTSSR
4721 066070      90$:  CKLOOP      ;LOOP IF SELECTED
      066070 104406      TRAP   C$CLP1
4722 066072 005723      TST     (R3)+       ;BUMP RECORD SIZE COUNTER
4723 066074 020327 000052      CMP     R3,#42     ;AT 42 SIZE YET
4724 066100 001315      BNE     65$         ;BR, IF MORE RECORDS TO WRITE
4725 066102 004737 011074      JSR     PC,REWIND   ;CALL TAPE REWIND COMMAND
4726 066106 103411      BCS     230$        ;BR, IF NO PROBLEM
4727 066110 010001      MOV     R0,R1       ;SAVE TSSR
4728 066112 016501 000002      MOV     TSSR(R5),R1 ;GET TSSR CONTENTS
4729 066116 005237 002214      INC     FATFLG      ;ERROR COUNT
4733 066122      ERRHRD  ERRNO,T35RWN,EXPREC ;REWIND NOT ACCEPTED
      066122 104456      TRAP   C$ERHRD
      066124 001331      .WORD  729
      066126 070574      .WORD  T35RWN
      066130 015554      .WORD  EXPREC
4734 066132      230$: CKLOOP      ;LOOP IF SELECTED
      066132 104406      TRAP   C$CLP1
4735 066134 013701 067350      MOV     T35BFR+6,R1 ;PICK UP XSTO
4736 066140 010102      MOV     R1,R2       ;SET UP EXPECTED
4737 066142 052702 000002      BIS     #BIT1,R2    ;SET BOT BIT IN EXPECTED
4738 066146 020102      CMP     R1,R2       ;DOES EXP = REC'D
4739 066150 001406      BEQ     240$        ;BR, IF EQUAL (OK)
4740 066152 005237 002214      INC     FATFLG      ;ERROR COUNT
4741 066156      ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      066156 104456      TRAP   C$ERHRD
      066160 001332      .WORD  730
      066162 070270      .WORD  T35BOT
      066164 015554      .WORD  EXPREC
4745 066166      240$: CKLOOP      ;LOOP IF SELECTED
      066166 104406      TRAP   C$CLP1
4746 066170 012703 000024      MOV     #20.,R3     ;STARTING RECORD SIZE
4747 066174 013737 003116 067442      MOV     FREE,T35RB  ;STARTING READ BUFFER ADDRESS
4748
4749      ;*****
4750      ;
4751      ;READ DATA,ACK COMMAND
4752      ;
4753      ;*****
4754
4755 066202 012737 100001 067440 265$: MOV     #100001,T35PK3 ;READ DATA,ACK COMMAND
4756 066210 012704 067440      MOV     #T35PK3,R4  ;SET UP R4 WITH PACKET ADDRESS
4757 066214 012700 177777      MOV     #177777,R0  ;SET PATTERN IN CORRECT REGISTER
4758 066220 004737 017502      JSR     PC,FILLMEM  ;FILL MEMORY WITH RECORD SIZE
4759 066224 010337 067446      MOV     R3,T35SZ    ;SET UP RECORD SIZE IN PACKET
4760 066230 010465 000000      MOV     R4,TSDB(R5) ;ISSUE COMMAND
4761 066234 004737 016330      JSR     PC,WAITF    ;WAIT FOR SSR TO SET
4762 066240 016501 000002      MOV     TSSR(R5),R1 ;GET TSSR CONTENTS
4763 066244 012702 000200      MOV     #SSR,R2     ;SET UP EXPECTED
    
```



```

066616 104456                                TRAP  C1ERHRD
066620 001340                                .WORD 736
066622 070270                                .WORD T35BOT
066624 015554                                .WORD EXPREC
4859 066626 401:  CKLOOP                      ;LOOP IF SELECTED
066626 104406                                TRAP  C1CLP1
4860 066630 012703 000024                    MOV   #20.,R3          ;STARTING RECORD SIZE
4861 066634 013737 003116 067442            MOV   FREE,T35WB      ;STARTING WRITE BUFFER ADDRESS
4862
4863 ;*****
4864 ;
4865 ;WRITE DATA,CVC=1,ACK COMMAND
4866 ;
4867 ;*****
4868
4869 066642 012737 140005 067440 651:  MOV   #140005,T35PK3    ;WRITE DATA,CVC=1,ACK COMMAND
4870 066650 012704 067440                    MOV   #T35PK3,R4      ;SET UP R4 WITH PACKET ADDRESS
4871 066654 010300                            MOV   R3,R0           ;SET PATTERN IN CORRECT REGISTER
4872 066656 004737 017502                    JSR   PC,FILLMEM      ;FILL MEMORY WITH RECORD SIZE
4873 066662 010337 067446                    MOV   R3,T35SZ        ;SET UP RECORD SIZE IN PACKET
4874 066666 010465 000000                    MOV   R4,TSDB(R5)     ;ISSUE COMMAND
4875 066672 004737 016330                    JSR   PC,WAITF        ;WAIT FOR SSR TO SET
4876 066676 016501 000002                    MOV   TSSR(R5),R1     ;GET TSSR CONTENTS
4877 066702 012702 000200                    MOV   #SSR,R2        ;SET UP EXPECTED
4878 066706 020102                            CMP   R1,R2           ;ARE THEY EQUAL
4879 066710 001406                            BEQ   801             ;BR, IF OK
4880 066712 005237 002214                    INC   FATFLG          ;ERROR COUNT
4884 066716                                ERRHRD  ERRNO,T35WDC,PKTSSR ;TSSR INCORRECT AFTER WRITE DATA
066716 104456                                TRAP  C1ERHRD
066720 001341                                .WORD 737
066722 071130                                .WORD T35WDC
066724 012126                                .WORD PKTSSR
4885 066726 801:  CKLOOP                      ;LOOP IF SELECTED
066726 104406                                TRAP  C1CLP1
4886
4887 ;*****
4888 ;
4889 ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
4890 ;
4891 ;*****
4892
4893 066730 012737 111005 067440            MOV   #111005,T35PK3  ;WRITE DATA RETRY,ACK,SWB=1 COMMAND
4894 066736 010465 000000                    MOV   R4,TSDB(R5)     ;ISSUE COMMAND
4895 066742 004737 016330                    JSR   PC,WAITF        ;WAIT FOR SSR TO SET
4896 066746 016501 000002                    MOV   TSSR(R5),R1     ;GET TSSR CONTENTS
4897 066752 012702 000200                    MOV   #SSR,R2        ;SET UP EXPECTED
4898 066756 020102                            CMP   R1,R2           ;ARE THEY EQUAL
4899 066760 001406                            BEQ   901             ;BR, IF OK
4900 066762 005237 002214                    INC   FATFLG          ;ERROR COUNT
4904 066766                                ERRHRD  ERRNO,T35WRF,EXPREC ;TSSR INCORRECT AFTER WRITE DATA RETRY
066766 104456                                TRAP  C1ERHRD
066770 001342                                .WORD 738
066772 072175                                .WORD T35WRF
066774 015554                                .WORD EXPREC
4905 066776 901:  CKLOOP                      ;LOOP IF SELECTED
066776 104406                                TRAP  C1CLP1
4906 067000 005723                            TST   (R3).          ;BUMP RECORD SIZE COUNTER

```


TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 7: EXTENDED MODE FEATURES

SEQ 0199

```

4907 067002 020327 000052          CMP      R3,#42.          ;AT 42 SIZE YET
4908 067006 001315          BNE      65$             ;BR, IF MORE RECORDS TO WRITE
4909 067010 004737 011074          JSR      PC,REWIND      ;CALL TAPE REWIND COMMAND
4910 067014 103411          BCS      230$           ;BR, IF NO PROBLEM
4911 067016 016501 000002          MOV      TSSR(R5),R1    ;GET TSSR CONTENTS
4912 067022 010004          MOV      R0,R4         ;GET PACKET ADDRESS
4913 067024 005237 002214          INC      FATFLG        ;ERROR COUNT
4917 067030          ERRHRD  ERRNO,T35RWN,PKTSSR ;REWIND NOT ACCEPTED
      067030 104456          TRAP                                C$ERHRD
      067032 001343          .WORD                                739
      067034 070574          .WORD                                T35RWN
      067036 012126          .WORD                                PKTSSR
4918 067040          230$:  CKLOOP          ;LOOP IF SELECTED          TRAP                                C$CLP1
      067040 104406
4919 067042 013701 067350          MOV      T35BFR+6,R1    ;PICK UP XSTO
4920 067046 010102          MOV      R1,R2         ;SET UP EXPECTED
4921 067050 052702 000002          BIS      #BIT1,R2      ;SET BOT BIT IN EXPECTED
4922 067054 020102          CMP      R1,R2         ;DOES EXP = REC'D
4923 067056 001406          BEQ      240$           ;BR, IF EQUAL (OK)
4924 067060 005237 002214          INC      FATFLG        ;ERROR COUNT
4928 067064          ERRHRD  ERRNO,T35BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      067064 104456          TRAP                                C$ERHRD
      067066 001344          .WORD                                740
      067070 070270          .WORD                                T35BOT
      067072 015554          .WORD                                EXPREC
4929 067074          240$:  CKLOOP          ;LOOP IF SELECTED          TRAP                                C$CLP1
      067074 104406
4930 067076 012703 000024          MOV      #20.,R3       ;STARTING RECORD SIZE
4931 067102 013737 003116 067442          MOV      FREE,T35RB    ;STARTING READ BUFFER ADDRESS
4932
4933          ;*****
4934          ;
4935          ;READ DATA,ACK COMMAND
4936          ;
4937          ;*****
4938
4939 067110 012737 100001 067440 265$:  MOV      #100001,T35PK3 ;READ DATA,ACK COMMAND
4940 067116 012704 067440          MOV      #T35PK3,R4    ;SET UP R4 WITH PACKET ADDRESS
4941 067122 010337 067446          MOV      R3,T35SZ      ;SET UP RECORD SIZE IN PACKET
4942 067126 010465 000000          MOV      R4,T35DB(R5)  ;ISSUE COMMAND
4943 067132 004737 016330          JSR      PC,WAITF      ;WAIT FOR SSR TO SET
4944 067136 016501 000002          MOV      TSSR(R5),R1   ;GET TSSR CONTENTS
4945 067142 012702 000200          MOV      #SSR,R2      ;SET UP EXPECTED
4946 067146 020102          CMP      R1,R2         ;ARE THEY EQUAL
4947 067150 001406          BEQ      280$           ;BR, IF OK
4948 067152 005237 002214          INC      FATFLG        ;ERROR COUNT
4952 067156          ERRHRD  ERRNO,T35RDF,PKTSSR ;TSSR INCORRECT AFTER READ DATA
      067156 104456          TRAP                                C$ERHRD
      067160 001345          .WORD                                741
      067162 067562          .WORD                                T35RDF
      067164 012126          .WORD                                PKTSSR
4953 067166          280$:  CKLOOP          ;LOOP IF SELECTED          TRAP                                C$CLP1
      067166 104406
4954 067170 013702 003116          MOV      FREE,R2       ;GET BUFFER ADDRESS
4955 067174 010304          MOV      R3,R4         ;GET RECORD SIZE
4956 067176 162704 000024          SUB      #20.,R4       ;POINT BACK TO 1ST RECORD
4957 067202 060204          285$:  ADD      R2,R4    ;POINT TO 1ST LOC IN BUFFER

```


(5)14

```

5010 ;
5012 067430 T35PK2: .WORD 100006 ;WRITE SUB SYS MEM COMMAND, AND ACK
5014 067430 .WORD T35BF2 ;ADDRESS OF SELECT BLOCK DATA
5015 067430 100006 .WORD 0
5016 067432 067450 .WORD 6. ;SIZE OF DATA PACKET
5017 067434 000000
5018 067436 000006
5019 ;
5023 067440 T35PK3: .WORD 100005 ;REREAD COMMAND, AND ACK
5024 067440 100005
5025 067442 T35RB: .WORD FREE ;ADDRESS OF WRITE BUFFER
5026 067442 003116 T35WB: .WORD 0
5027 067444 000000 .WORD 0
5028 067446 000000 T35SZ: .WORD 0 ;SIZE OF BUFFER (EXTENT)
5029 .EVEN
5030 ;
5031 ;
5032 ;
5033 067450 T35BF2:
5034 067450 010 T35BS0: .BYTE 10 ;BSELO AREA
5035 067451 200 T35BS1: .BYTE 200 ;BSEL1 AREA
5036 067452 000000 T35S2: .WORD 0 ;SEL 2 AREA
5037 067454 000000 T35S3: .WORD 0 ;DATA AREA
5038 ;
5039 ;
5040 .EVEN
5041 ;TAPE MOTION PACKET COMMAND VALUES
5042 ;
5043 067456 100205 T35RN: .WORD 100205 ;REREAD DATA (NEXT)
5044 067460 100605 T35WR: .WORD 100605 ;REREAD DATA RETRY
5045 067462 102205 T35CON: .WORD 102205 ;WRITE CONTINOUS
5046 067464 177777 .WORD 177777 ;END OF DATA
5047 ;
5048 ;
5049 067466 000000 T35CNT: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
5050 067470 000000 T35CNU: .WORD 0 ;TAPE TIMER COUNTER STORAGE AREA
5051 067472 000000 T35DLY: .WORD 0 ;DELAY COUNTER
5052 ;
5053 ;LOCAL TEXT MESSAGES FOR TEST
5054 ;-
5055 ;
5056 067474 124 141 160 T35WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
5057 067562 124 123 123 T35RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
5058 067631 122 105 122 T35RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
5059 067726 120 117 123 T35SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
5060 070010 122 111 102 T35LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
5061 070060 124 123 123 T35WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
5062 070135 111 154 154 T35LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
5063 070216 124 123 123 T35WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
5064 070270 124 141 160 T35BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
5065 070363 127 122 111 T35TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
5066 070440 122 105 122 T35EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
5067 070517 124 123 123 T35TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
5068 070574 122 145 167 T35RWN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5069 070643 122 101 115 T35RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
5070 070716 124 123 123 T35AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
5071 070765 104 162 151 T35OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'

```

HIF

5072	071040	124	123	123	T35WDD:	.ASCIZ	'TSSR Not Correct After REREAD DATA Command, SWB Bit Set
5073	071130	124	123	123	T35WDC:	.ASCIZ	'TSSR Not Correct After REREAD DATA Command'
5074	071203	103	126	103	T35VCK:	.ASCIZ	'CVC Set, Didn't Reset VCK In Message Buffer'
5075	071256	124	123	102	T35BA:	.ASCIZ	'TSBA Not Correct After REREAD DATA Command'
5076	071331	127	122	111	T35WSS:	.ASCIZ	'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
5077	071420	122	145	141	T35LON:	.ASCIZ	'Reading Long Record Failed To Set RLL Bit In XST0'
5078	071502	122	145	141	T35LOP:	.ASCIZ	'Reading Long Record Failed To Set RLS Bit In XST0'
5079	071564	122	145	163	T35PBP:	.ASCIZ	'Residual Byte Count Incorrect After Short Record Read'
5080	071652	122	145	141	T35TRL:	.ASCIZ	'Reading Long Record Failed To Give Tape Status Alert'
5081	071740	127	122	111	T35NEF:	.ASCIZ	'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3
5082	072036	124	123	123	T35SCF:	.ASCIZ	'TSSR Not Correct After SPACE RECORDS Command'
5083	072113	124	123	123	T35TSA:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
5084	072175	124	123	123	T35WRF:	.ASCIZ	'TSSR Not Correct After WRITE DATA RETRY Command'
5085	072255	104	141	164	T35DTA:	.ASCIZ	'Data Compare Error, Data Read From Tape Not Equal To Written'
5086	072352	124	123	123	T35SSR:	.ASCIZ	'TSSR Incorrect After WRITE MISCELLANEOUS Command'
5087	072433	115	117	124	T35MOT:	.ASCIZ	'MOT Bit (XST0) Not Set During Rewind (Extended Features Mode)
5088	072531	111	156	164	T35INT:	.ASCIZ	'Interrupt Received After REWIND Command (IE Bit Not Set)
5089	072622	117	120	115	T35OPM:	.ASCIZ	'OPM Bit (XST2) Not Set During Rewind (Extended Features Mode)
5090	072720	124	123	123	T35RWE:	.ASCIZ	'TSSR Incorrect After Extended Features REWIND Command'
5091	073006	116	157	040	T35NIN:	.ASCIZ	'No Interrupt Detected After REWIND IMMEDIATE'
5092	073063	105	170	164	TST35ID:	.ASCIZ	'Extended Mode Functions
5093						.EVEN	
5094							
5095							
5096							;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5097							;WRITE SUBSYSTEM MEMORY COMMAND
5098							
5099							
5100							
5101	073114				T35REST:		
5102	073114				SAVREG		;SAVE THE REGISTERS
5103	073120	012701	067320		MOV	#T35PACKET,R1	;START OF THE PACKET
5104	073124	012721	100004		MOV	#100004,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
5105	073130	012721	067330		MOV	#T35DATA,(R1)+	;ADDRESS OF CHARAISTICS DATA BLOCK
5106	073134	005021			CLR	(R1)+	;EXTENDED ADDRESS
5107	073136	012721	000012		MOV	#10.,(R1)+	;SIZE OF DATA BLOCK IN BYTES
5108	073142	012721	067342		MOV	#T35BFR,(R1)+	;ADDRESS OF MESSAGE BUFFER
5109	073146	005021			CLR	(R1)+	
5110	073150	012721	000024		MOV	#20.,(R1)+	;LENGTH OF MESSAGE BUFFER
5111	073154	005021			CLR	(R1)+	
5112	073156	012711	000000		MOV	#0,(R1)	;SELECT DRIVE ZERO
5113	073162	012702	000030		MOV	#24.,R2	;NUMBER OF LOCATIONS TO BE CLEARED
5114	073166	012762	177777	067342 64#:	MOV	#177777,T35BFR(R2)	;ALL ONES TO MESSAGE BUFFER
5115	073174	005742			TST	-(R2)	;NEXT LOCATION
5116	073176	022702	000000		CMP	#0,R2	;AT END OF LOOP YET
5117	073202	001371			BNE	64#	;KEEP GOING UNTIL DONE
5118	073204	000207			RTS	PC	;RETURN
5119							
5120	073206				T35RT2:		
5121	073206				SAVREG		;SAVE THE REGISTERS
5122	073212	012701	067430		MOV	#T35PK2,R1	;START OF THE PACKET
5123	073216	012721	100006		MOV	#100006,(R1)+	;WRITE SUBSYSTEM MEM. WITH ACK,
5124	073222	012721	067450		MOV	#T35BF2,(R1)+	;ADDRESS OF DATA BLOCK
5125	073226	005021			CLR	(R1)+	;EXTENDED ADDRESS
5126	073230	012721	000006		MOV	#6.,(R1)+	;SIZE OF DATA BLOCK IN BYTES
5127	073234	005021			CLR	(R1)+	
5128	073236	012701	067450		MOV	#T35BF2,R1	;POINT TO DATA SEL AREA

5129 073242 005021
 5130 073244 005011
 5131 073246 000207
 5132 073250
 5133 073250
 5134 073254 012701 067440
 5135 073260 005021
 5136 073262 005021
 5137 073264 005021
 5138 073266 005011
 5139 073270 000207
 5140 073272
 073272
 073272 104401

T35RT3:
 CLR (R1).
 CLR (R1)
 RTS PC ;RETURN
 SAVREG ;SAVE REGISTERS
 MOV #T35PK3,R1 ;SET UP POINTER ADDRESS
 CLR (R1). ;COMMAND SPACE
 CLR (R1). ;ADDRESS OF DATA BLOCK
 CLR (R1). ;EXTENDED ADDRESS
 CLR (R1) ;SIZE OF DATA TRANSFER BLOCK
 RTS PC ;RETURN
 ENDTST

L10063: TRAP C1ETST

.SBTTL TEST 8: RECORD BUFFERING

5141
 5142
 5143
 5144
 5145
 5146
 5147
 5148
 5149
 5150
 5151
 5152
 5153
 5154
 5155
 5156
 5157
 5158
 5159
 5160
 5161
 5162
 5163
 5164
 5165
 5166
 5167
 5168
 5169
 5170
 5171
 5172
 5173
 5174
 5175
 5176
 5177
 5178
 5179
 5180
 5181
 5182
 5183

THIS TEST VERIFIES THAT RECORD BUFFERING, USED FOR WRITE DATA AND READ NEXT COMMANDS, OPERATES PROPERLY AND IS PROPERLY CONTROLLED BY THE EXTENDED CHARACTERISTICS DATA WORD. IF THE M7196 CONTROLLER MODULE IS NOT ALREADY IN EXTENDED FEATURES MODE (AS CONTROLLED BY THE DIP SWITCH ON THE MODULE), IT IS PLACED INTO THAT MODE BY INVERTING THE SENSE OF THE SWITCH USING THE WRITE SUBSYSTEM MEMORY COMMAND. NOTE THAT RECORD BUFFERING HAS BEEN ENABLED IN PREVIOUS TESTS OF READ AND WRITE AND SO HAS BEEN PARTIALLY TESTED ALREADY. THIS TEST VERIFIES THAT BUFFERING IS ACTUALLY OPERATING. THE FOLLOWING SUBTESTS ARE PERFORMED:

VERIFIES THAT NORMAL BUFFERING ON WRITE DATA COMMANDS OPERATES PROPERLY AT LOW TAPE SPEED. THE FOLLOWING SEQUENCE IS PERFORMED:

1. THE TAPE IS REWOUND.
2. BUFFERING IS DISABLED AND LOW TAPE SPEED IS SELECTED (VIA WRITE CHARACTERISTICS COMMAND).
3. AN INITIAL RECORD IS WRITTEN ONTO THE TAPE IN ORDER TO MOVE THE TAPE OFF BOT.
4. THE PROGRAM DELAYS FOR A TIME SUFFICIENT TO ALLOW THE TAPE TO REPOSITION AND COME TO REST.
5. A WRITE DATA COMMAND, WITH A BYTE COUNT LESS THAN 3.5K, IS ISSUED, AND THE PROGRAM COUNTS, IN A WAIT LOOP, THE TIME IT TAKES TO RECEIVE COMMAND TERMINATION. THIS SHOULD BE A RELATIVELY LONG TIME SINCE BUFFERING IS DISABLED.
6. BUFFERING IS ENABLED.
7. THE WRITE DATA COMMAND IS AGAIN ISSUED, WITH THE SAME BYTE COUNT AS THAT USED PREVIOUSLY. THE TIME TO COMPLETION IS AGAIN MEASURED.
8. THE COMPLETION TIMES MEASURED FOR THE NON-BUFFERED AND


```

5234 073424 005237 002214          INC    FATFLG          ;ERROR COUNT
5238 073430 010001          MOV    R0,R1          ;CONTENTS OF TSSR REGISTER
5239 073432          ERRDF  ERRNO,SFIERR,SFIMSG ;FATAL ERROR TSSR WAS NOT OK
      073432 104455          TRAP  C$ERDF
      073434 001441          .WORD 801
      073436 003646          .WORD SFIERR
      073440 012114          .WORD SFIMSG
5240 073442 013737 002174 075530 20$: MOV    UNITN,T36DSW    ;SET UP DRIVE NUMBER
5241 073450 012704 075510          MOV    @T36PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5242 073454 004737 010742          JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
5243 073460 103407          BCS    25$           ;BR, IF COMMAND ISSUED OK
5244 073462 005237 002214          INC    FATFLG          ;ERROR COUNT
5248 073466 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
5249 073470          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      073470 104456          TRAP  C$ERHRD
      073472 001442          .WORD 802
      073474 005052          .WORD WRTMSG
      073476 012114          .WORD SFIMSG
5250 073500          25$: CKLOOP          ;LOOP IF SELECTED
      073500 104406          TRAP  C$CLP1
5251 073502 004737 011074          JSR    PC,REWIND      ;CALL TAPE REWIND COMMAND
5252 073506 103407          BCS    30$           ;BR, IF NO PROBLEM
5253 073510 010004          MOV    R0,R4          ;SET UP REWIND PACKET ADDRESS
5254 073512 005237 002214          INC    FATFLG          ;ERROR COUNT
5258 073516          ERRHRD ERRNO,T36RWN,PKTSSR ;REWIND NOT ACCEPTED
      073516 104456          TRAP  C$ERHRD
      073520 001443          .WORD 803
      073522 077071          .WORD T36RWN
      073524 012126          .WORD PKTSSR
5259 073526          30$: CKLOOP          ;LOOP IF SELECTED
      073526 104406          TRAP  C$CLP1
5260 073530 013701 075540          MOV    T36BFR+6,R1   ;PICK UP XSTO
5261 073534 010102          MOV    R1,R2          ;SET UP EXPECTED
5262 073536 052702 000002          BIS    @BIT1,R2       ;SET BOT BIT IN EXPECTED
5263 073542 020102          CMP    R1,R2          ;DOES EXP = REC'D
5264 073544 001406          BEQ    40$           ;BR, IF EQUAL (OK)
5265 073546 005237 002214          INC    FATFLG          ;ERROR COUNT
5269 073552          ERRHRD ERRNO,T36BOT,EXPREC ;TAPE NOT AT BOT AFTER REWIND
      073552 104456          TRAP  C$ERHRD
      073554 001444          .WORD 804
      073556 076565          .WORD T36BOT
      073560 015554          .WORD EXPREC
5270 073562          40$: CKLOOP          ;LOOP IF SELECTED
      073562 104406          TRAP  C$CLP1
5271 073564 013737 002174 075530          MOV    UNITN,T36DSW    ;SET UP DRIVE NUMBER
5272 073572 052737 000036 075530          BIS    @BIT3!BIT4,T36DSW ;25-APR-83 REV B - TURN ON THE BUFFERING
5273 073600 012704 075510          MOV    @T36PACKET,R4 ;SUBROUTINE NEEDS PACKET ADDRESS
5274 073604 004737 010742          JSR    PC,WRTCHR      ;ISSUE WRITE CHARACTERISTICS
5275 073610 103407          BCS    50$           ;BR, IF COMMAND ISSUED OK
5276 073612 005237 002214          INC    FATFLG          ;ERROR COUNT
5280 073616 010001          MOV    R0,R1          ;SAVE CONTENTS OF TSSR
5281 073620          ERRHRD ERRNO,WRTMSG,SFIMSG ;WRITE CHARACTERISTIC FAILED
      073620 104456          TRAP  C$ERHRD
      073622 001445          .WORD 805
      073624 005052          .WORD WRTMSG
      073626 012114          .WORD SFIMSG
5282 073630          50$: CKLOOP          ;LOOP IF SELECTED

```

TEST 1 - HARDWARE TEST 1-8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 8: RECORD BUFFERING

SEQ 0206

	073630	104406							TRAP	C#CLP1
5283	073632	012737	003720	075636	MOV	#2000.,T36SZ				;SET UP RECORD SIZE
5284	073640	013737	003116	075632	MOV	FREE,T36WB				;ADDRESS OF WRITE BUFFER
5285	073646	012737	140005	075530	MOV	#140005,T36PK3				;WRITE DATA,ACK,CVC=1 COMMAND
5286	073654	012704	075630		MOV	#T36PK3,R4				;SET UP R4 WITH PACKET ADDRESS
5287	073660	010465	000000		MOV	R4,TSDB(R5)				;ISSUE COMMAND
5288	073664	004737	016330		JSR	PC,WAITF				;WAIT FOR SSR TO SET
5289	073670	016501	000002		MOV	TSSR(R5),R1				;GET TSSR CONTENTS
5290	073674	012702	000200		MOV	#SSR,R2				;SET UP EXPECTED
5291	073700	020102			CMP	R1,R2				;ARE THEY EQUAL
5292	073702	001406			BEQ	60\$;BR, IF OK
5293	073704	005237	002214		INC	FATFLG				;ERROR COUNT
5297	073710				ERRHRD	ERRNO,WRTErr,PKTSSR				;TSSR INCORRECT AFTER READ DATA
	073710	104456							TRAP	C#ERRHRD
	073712	001446							.WORD	806
	073714	005107							.WORD	WRTErr
	073716	012126							.WORD	PKTSSR
5298	073720			60\$:	CKLOOP					;LOOP IF SELECTED
	073720	104406							TRAP	C#CLP1
5299	073722	012737	000005	075662	MOV	#05.,T36DLY				;25-APR-83 REV B - DELAY FOR TAPE TO STOP
5300	073730			70\$:	DELAY	1				;25-APR-83 REV B - DELAY ROUTINE CALL
	073730	012727	000001						MOV	#1,(PC)+
	073734	000000							.WORD	0
	073736	013727	002116						MOV	L#DLY,(PC)+
	073742	000000							.WORD	0
	073744	005367	177772						DEC	-6(PC)
	073750	001375							BNE	.-4
	073752	005367	177756						DEC	-22(PC)
	073756	001367							BNE	.-20
5301	073760	005337	075662		DEC	T36DLY				;BUMP COUNTER DOWN
5302	073764	001361			BNE	70\$;BR, IF MORE DELAY TO GO
5303	073766	012737	006642	075636	MOV	#3490.,T36SZ				;SET SIZE OF TRANSFER
5304	073774	012737	140005	075630	MOV	#140005,T36PK3				;WRITE DATA,ACK,CVC=1 COMMAND
5305	074002	012704	075630		MOV	#T36PK3,R4				;SET UP R4 WITH PACKET ADDRESS
5306	074006	005037	075656		CLR	T36CNT				;CLEAR COUNTER
5307	074012	012737	001750	075662	MOV	#1000.,T36DLY				;SET DROP DEAD COUNTER VALUE
5308	074020	010465	000000		MOV	R4,TSDB(R5)				;ISSUE COMMAND
5309	074024	016501	000002		MOV	TSSR(R5),R1				;GET TSSR CONTENTS
5310	074030	032701	000200	80\$:	MOV	#SSR,R1				;CHECK FOR SSR SET
5311	074034	001021			BNE	90\$;BR, IF SSR IS SET
5312	074036	005237	075656		INC	T36CNT				;BUMP CYCLE COUNTER
5313	074042				DELAY	1				;CUT NUMBER OF LOOPS DOWN
	074042	012727	000001						MOV	#1,(PC)+
	074046	000000							.WORD	0
	074050	013727	002116						MOV	L#DLY,(PC)+
	074054	000000							.WORD	0
	074056	005367	177772						DEC	-6(PC)
	074062	001375							BNE	.-4
	074064	005367	177756						DEC	22(PC)
	074070	001367							BNE	. 20
5314	074072	005337	075662		DEC	T36DLY				;BUMP DROP DEAD COUNTER
5315	074076	001352			BNE	80\$;BR, IF THERE IS STILL TIME
5316	074100	012702	000200	90\$:	MOV	#SSR,R2				;SET UP EXPECTED
5317	074104	020102			CMP	R1,R2				;ARE THEY EQUAL
5318	074106	001406			BEQ	100\$;BR, IF OK
5319	074110	005237	002214		INC	FATFLG				;ERROR COUNT
5323	074114				ERRHRD	ERRNO,T36WDE,PKTSSR				;TSSR INCORRECT AFTER READ DATA

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14 JUN 84 16:41
 TEST 8: RECORD BUFFERING

SEQ 0207

	074114	104456					TRAP	C#ERHRD
	074116	001447					.WORD	807
	074120	076513					.WORD	T36WDE
	074122	012126					.WORD	PKTSSR
5324	074124		100\$:	CKLOOP				;LOOP IF SELECTED
	074124	104406					TRAP	C#CLP1
5325	074126	013737	002174	075530	MOV	UNITN,T36DSW		;SET UP DRIVE NUMBER
5326	074134	052737	000010	075530	BIS	#BIT3,T36DSW		;25-APR-83 REV B - TURN OFF BUFFERING
5327	074142	012704	075510		MOV	#T36PACKET,R4		;SUBROUTINE NEEDS PACKET ADDRESS
5328	074146	004737	010742		JSR	PC,WRTCHR		;ISSUE WRITE CHARACTERISTICS
5329	074152	103407			BCS	110\$;BR, IF COMMAND ISSUED OK
5330	074154	005237	002214		INC	FATFLG		;ERROR COUNT
5334	074160	010001			MOV	RO,R1		;SAVE CONTENTS OF TSSR
5335	074162				ERRHRD	ERRNO,WRTMSG,SFIMSG		;WRITE CHARACTERISTICSC FAILED
	074162	104456					TRAP	C#ERHRD
	074164	001450					.WORD	808
	074166	005052					.WORD	WRTMSG
	074170	012114					.WORD	SFIMSG
5336	074172		110\$:	CKLOOP				;LOOP IF SELECTED
	074172	104406					TRAP	C#CLP1
5337	074174	012737	006642	075636	MOV	#3490.,T36SZ		;SET SIZE OF TRANSFER
5338	074202	012737	140005	075630	MOV	#140005,T36PK3		;WRITE DATA,ACK,CVC=1 COMMAND
5339	074210	012704	075630		MOV	#T36PK3,R4		;SET UP R4 WITH PACKET ADDRESS
5340	074214	005037	075660		CLR	T36CNU		;CLEAR COUNTER
5341	074220	012737	001750	075662	MOV	#1000.,T36DLY		;SET DROP DEAD COUNTER VALUE
5342	074226	010465	000000		MOV	R4,TSDB(R5)		;ISSUE COMMAND
5343	074232	016501	000002		MOV	TSSR(R5),R1		;GET TSSR CONTENTS
5344	074236	032701	000200	120\$:	BIT	#SSR,R1		;CHECK FOR SSR SET
5345	074242	001021			BNE	130\$;BR, IF SSR IS SET
5346	074244	005237	075660		INC	T36CNU		;BUMP CYCLE COUNTER
5347	074250				DELAY	1		;CUT NUMBER OF LOOPS DOWN
	074250	012727	000001				MOV	#1,(PC)+
	074254	000000					.WORD	0
	074256	013727	002116				MOV	L#DLY,(PC)+
	074262	000000					.WORD	0
	074264	005367	177772				DEC	-6(PC)
	074270	001375					BNE	. 4
	074272	005367	177756				DEC	22(PC)
	074276	001367					BNE	. 20
5348	074300	005337	075662		DEC	T36DLY		;BUMP DROP DEAD COUNTER
5349	074304	001352			BNE	120\$;BR, IF THERE IS STILL TIME
5350	074306	012702	000200	130\$:	MOV	#SSR,R2		;SET UP EXPECTED
5351	074312	020102			CMP	R1,R2		;ARE THEY EQUAL
5352	074314	001406			BEQ	140\$;BR, IF OK
5353	074316	005237	002214		INC	FATFLG		;ERROR COUNT
5357	074322				ERRHRD	ERRNO,WRTERR,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA
	074322	104456					TRAP	C#ERHRD
	074324	001451					.WORD	809
	074326	005107					.WORD	WRTERR
	074330	012126					.WORD	PKTSSR
5358	074332		140\$:	CKLOOP				;LOOP IF SELECTED
	074332	104406					TRAP	C#CLP1
5359	074334	013701	075656		MOV	T36CNT,R1		;GET FIRST COUNTER
5360	074340	013702	075660		MOV	T36CNU,R2		;GET SECOND COUNTER
5361	074344	020102			CMP	R1,R2		;25-APR-83 REV B - COMPARE EM
5362	074346	003406			BLE	300\$;BR, IF VALUES ARE CORRECT (OK)
5363	074350	005237	002214		INC	FATFLG		;ERROR COUNT

TEST 1 HARDWARE TEST 1 8 TEST MACRO M1113 14-JUN-84 16:41
TEST 8: RECORD BUFFERING

SEQ 0208

```

5367 074354          ERRMRD  ERRNO,T36NAS,EXPREC      ;TAPE NOT AT CORRECT SPEED
      074354 104456          TRAP                  CIERMRD
      074356 001452          .WORD                  810
      074360 075664          .WORD                  T36NAS
      074362 015554          .WORD                  EXPREC
5368 074364          3001:  CKLOOP                    ;LOOP IF SELECTED
      074364 104406          TRAP                  C1CLP1
5369 074366          ENDSUB
      074366          L10071:
      074366 104403          TRAP                  C1ESUB
5370 074370 023727 002214 000017  CMP          FATFLG,#15.      ;IS ERROR COUNT AT 25
5371 074376 103402          BLO          9991          ;BR. IF LESS THAN 25
5372 074400 004737 017262  JSR          PC,CKDROP      ;TRY TO DROP THE UNIT
5373 074404          9991:

```

```

5374 :
5375 :
5376 :
5377 :
5378 :
5379 :
5380 :
5381 :
5382 :
5383 :
5384 :
5385 :
5386 :
5387 :
5388 :
5389 :
5390 :
5391 :
5392 :
5393 :
5394 :
5395 :
5396 :
5397 :
5398 :
5399 :
5400 :
5401 :
5402 :
5403 :
5404 :
5405 :
5406 :
5407 :
5408 :
5409 :
5410 :
5411 :
5412 :
5413 :
5414 :
5415 :
5416 :

```

TEST 8, SUBTEST 2

THIS TEST VERIFIES THAT RECORD BUFFERING, USED FOR WRITE DATA AND READ NEXT COMMANDS, OPERATES PROPERLY AND IS PROPERLY CONTROLLED BY THE EXTENDED CHARACTERISTICS DATA WORD. IF THE M7196 CONTROLLER MODULE IS NOT ALREADY IN EXTENDED FEATURES MODE (AS CONTROLLED BY THE DIP SWITCH ON THE MODULE), IT IS PLACED INTO THAT MODE BY INVERTING THE SENSE OF THE SWITCH USING THE WRITE SUBSYSTEM MEMORY COMMAND. NOTE THAT RECORD BUFFERING HAS BEEN ENABLED IN PREVIOUS TESTS OF READ AND WRITE AND SO HAS BEEN PARTIALLY TESTED ALREADY. THIS TEST VERIFIES THAT BUFFERING IS ACTUALLY OPERATING. THE FOLLOWING SUBTESTS ARE PERFORMED:

VERIFIES THAT NORMAL BUFFERING ON WRITE DATA COMMANDS OPERATES PROPERLY AT LOW TAPE SPEED. THE FOLLOWING SEQUENCE IS PERFORMED:

1. THE TAPE IS REWOUND.
2. BUFFERING IS DISABLED AND LOW TAPE SPEED IS SELECTED (VIA WRITE CHARACTERISTICS COMMAND).
3. AN INITIAL RECORD IS WRITTEN ONTO THE TAPE IN ORDER TO MOVE THE TAPE OFF BOT.
4. THE PROGRAM DELAYS FOR A TIME SUFFICIENT TO ALLOW THE TAPE TO REPOSITION AND COME TO REST.
5. A WRITE DATA COMMAND, WITH A BYTE COUNT LESS THAN 3.5K, IS ISSUED, AND THE PROGRAM COUNTS, IN A WAIT LOOP, THE TIME IT TAKES TO RECEIVE COMMAND TERMINATION. THIS SHOULD BE A RELATIVELY LONG TIME SINCE BUFFERING IS DISABLED.
6. BUFFERING IS ENABLED.
7. THE WRITE DATA COMMAND IS AGAIN ISSUED, WITH THE SAME BYTE COUNT AS THAT USED PREVIOUSLY. THE TIME TO COMPLETION IS AGAIN MEASURED.

TEST 1 - HARDWARE TEST 1 8 TEST MACRO M1113 14-JUN-84 16:41
 TEST 8: RECORD BUFFERING

SEQ 0211

075020	013727	002116			MOV	L#DLY,(PC)+
075024	000000				.WORD	0
075026	005367	177772			DEC	6(PC)
075032	001375				BNE	. 4
075034	005367	177756			DEC	22(PC)
075040	001367				BNE	. 20
5508	075042	005337	075662		DEC	T36DLY
5509	075046	001361			BNE	70#
5510	075050	012737	006642	075636	MOV	#3490.,T36SZ
5511	075056	012737	140005	075630	MOV	#140005,T36PK3
5512	075064	012704	075630		MOV	#T36PK3,R4
5513	075070	005037	075656		CLR	T36CNT
5514	075074	012737	001750	075662	MOV	#1000.,T36DLY
5515	075102	010465	000000		MOV	R4,T36D(R5)
5516	075106	016501	000002	80#:	MOV	TSSR(R5),R1
5517	075112	032701	000200		BIT	#SSR,R1
5518	075116	001021			BNE	90#
5519	075120	005237	075656		INC	T36CNT
5520	075124				DELAY	1
	075124	012727	000001			
	075130	000000				
	075132	013727	002116			
	075136	000000				
	075140	005367	177772			
	075144	001375				
	075146	005367	177756			
	075152	001367				
5521	075154	005337	075662		DEC	T36DLY
5522	075160	001352			BNE	80#
5523	075162	012702	000200	90#:	MOV	#SSR,R2
5524	075166	020102			CMP	R1,R2
5525	075170	001406			BEQ	100#
5526	075172	005237	002214		INC	FATFLG
5530	075176				ERRHRD	ERRNO,T36WDE,PKTSSR
	075176	104456				
	075200	001461				
	075202	076513				
	075204	012126				
5531	075206			100#:	CKLOOP	
	075206	104406				
5532	075210	013737	002174	075530	MOV	UNITN,T36DSW
5533	075216	052737	000010	075530	BIS	#BIT3,T36DSW
5534	075224	012704	075510		MOV	#T36PACKET,R4
5535	075230	004737	010742		JSR	PC,WRTCHR
5536	075234	103407			BCS	110#
5537	075236	005237	002214		INC	FATFLG
5541	075242	010001			MOV	RO,R1
5542	075244				ERRHRD	ERRNO,WRTMSG,SFIMSG
	075244	104456				
	075246	001462				
	075250	005052				
	075252	012114				
5543	075254			110#:	CKLOOP	
	075254	104406				
5544	075256	012737	006642	075636	MOV	#3490.,T36SZ
5545	075264	012737	140005	075630	MOV	#140005,T36PK3
5546	075272	012704	075630		MOV	#T36PK3,R4

5547	075276	005037	075660		CLR	T36CNU		;CLEAR COUNTER	
5548	075302	012737	001750	075662	MOV	#1000.,T36DLY		;SET DROP DEAD COUNTER VALUE	
5549	075310	010465	000000		MOV	R4,TSD8(R5)		;ISSUE COMMAND	
5550	075314	016501	000002	120#:	MOV	TSSR(R5),R1		;GET TSSR CONTENTS	
5551	075320	032701	000200		BIT	#SSR,R1		;CHECK FOR SSR SET	
5552	075324	001021			BNE	130#		;BR, IF SSR IS SET	
5553	075326	005237	075660		INC	T36CNU		;BUMP CYCLE COUNTER	
5554	075332				DELAY	1		;CUT NUMBER OF LOOPS DOWN	
	075332	012727	000001					MOV	#1,(PC).
	075336	000000						.WORD	0
	075340	013727	002116					MOV	L8DLY,(PC).
	075344	000000						.WORD	0
	075346	005367	177772					DEC	6(PC)
	075352	001375						BNE	.4
	075354	005367	177756					DEC	22(PC)
	075360	001367						BNE	.-20
5555	075362	005337	075662		DEC	T36DLY		;BUMP DROP DEAD COUNTER	
5556	075366	001352			BNE	120#		;BR, IF THERE IS STILL TIME	
5557	075370	012702	000200	130#:	MOV	#SSR,R2		;SET UP EXPECTED	
5558	075374	020102			CMF	R1,R2		;ARE THEY EQUAL	
5559	075376	001406			BEQ	140#		;BR, IF OK	
5560	075400	005237	002214		INC	FATFLG		;ERROR COUNT	
5564	075404				ERRHRD	ERRNO,WRERR,PKTSSR		;TSSR INCORRECT AFTER WRITE DATA	
	075404	104456						TRAP	C\$ERHRD
	075406	001463						.WORD	819
	075410	005107						.WORD	WRERR
	075412	012126						.WORD	PKTSSR
5565	075414			140#:	CKLOOP			;LOOP IF SELECTED	
	075414	104406						TRAP	C\$CLP1
5566	075416	013701	075656		MOV	T36CNT,R1		;GET FIRST COUNTER	
5567	075422	013702	075660		MOV	T36CNU,R2		;GET SECOND COUNTER	
5568	075426	020102			CMF	R1,R2		;25-APR-83 REV B - COMPARE EM	
5569	075430	003406			BLE	300#		;BR, IF VALUES ARE CORRECT (OK)	
5570	075432	005237	002214		INC	FATFLG		;ERROR COUNT	
5574	075436				ERRHRD	ERRNO,T36NAS,EXPREC		;TAPE NOT AT CORRECT SPEED	
	075436	104456						TRAP	C\$ERHRD
	075440	001464						.WORD	820
	075442	075664						.WORD	T36NAS
	075444	015554						.WORD	EXPREC
5575	075446			300#:	CKLOOP			;LOOP IF SELECTED	
	075446	104406						TRAP	C\$CLP1
5576	075450				ENDSUB				
	075450								
	075450	104403							
5577	075452	023727	002214	000017	CMF	FATFLG,#15.		;IS ERROR COUNT AT 25	
5578	075460	103402			NLO	999#		;BR, IF LESS THAN 25	
5579	075462	004737	017262		JSR	PC,CKDROP		;TRY TO DROP THE UNIT	
5580	075466			999#:					
5581				:					
5582				:					
5583				:					
5584	075466	004737	016536		JSR	PC,TSTLOOP		;DO WE NEED TO ITERATE TEST?	
5585	075472	103002			BCC	163#		;BR, IF NO LOOP REQUIRED	
5586	075474	000137	073330		JMP	T36LOOP		;EXECUTE AGAIN	
5587	075500			163#:					
5588	075500				EXIT	TST		;ALL DONE THIS TEST	
	075500	104432						TRAP	C\$EXIT

.WORD 110070 .

075502 003344
 5589
 5590
 5591
 5593 075510
 5595 075510
 5596 075510 100004
 5597 075512 075520
 5598 075514 000000
 5599 075516 000012
 5600 075520
 5601 075520 075532
 5602 075522 000000
 5603 075524 000024
 5604 075526 000000
 5605 075530 000000
 5606 075532
 5607
 5608
 5609
 5611 075620
 5613 075620
 5614 075620 100006
 5615 075622 075640
 5616 075624 000000
 5617 075626 000006
 5618
 5622 075630
 5623 075630 100005
 5624 075632
 5625 075632 003116
 5626 075634 000000
 5627 075636 000000
 5628
 5629
 5630
 5631
 5632 075640
 5633 075640 010
 5634 075641 200
 5635 075642 000000
 5636 075644 000000
 5637
 5638
 5639
 5640
 5641
 5642 075646 100205
 5643 075650 100605
 5644 075652 102205
 5645 075654 177777
 5646
 5647
 5648 075656 000000
 5649 075660 000000
 5650 075662 000000
 5651

```

; *
; LOCAL STORAGE FOR THIS TEST
;
; .<<..10>&177770
T36PACKET:
; .WORD 100004
; .WORD T36DATA
; .WORD 0
; .WORD 10.
T36DATA:
; .WORD T36BFR
; .WORD 0
; .WORD 20.
; .WORD 0
T36DSW: .WORD 0
T36BFR: .BLKW 25.
;
; WRITE SUBSYSTEM MEMORY COMMAND PACKET
;
; .<<..10>&177770
T36PK2:
; .WORD 100006
; .WORD T36BF2
; .WORD 0
; .WORD 6.
T36PK3:
; .WORD 100005
T36RB:
T36WB: .WORD FREE
; .WORD 0
T36SZ: .WORD 0
; .EVEN
;
;
;
T36BF2:
T36BS0: .BYTE 10
T36BS1: .BYTE 200
T36S2: .WORD 0
T36S3: .WORD 0
;
;
; .EVEN
; TAPE MOTION PACKET COMMAND VALUES
T36RN: .WORD 100205
T36WR: .WORD 100605
T36CON: .WORD 102205
; .WORD 177777
;
;
T36CNT: .WORD 0
T36CNU: .WORD 0
T36DLY: .WORD 0
; *

```

```

; COMMAND PACKET FOR TEST
; WRITE CHARACTERISTICS COMMAND, WITH . ACK
; ADDRESS OF CHARACTERISTICS BLOCK

; STARTING VALUE OF BLOCK SIZE
; CHARACTERISTICS DATA BLOCK
; ADDRESS OF MESSAGE BUFFER

; LENGTH OF MESSAGE BUFFER

; SELECT DRIVE 0
; MESSAGE BUFFER

; WRITE SUB SYS MEM COMMAND, AND ACK
; ADDRESS OF SELECT BLOCK DATA

; SIZE OF DATA PACKET

; REREAD COMMAND, AND ACK

; ADDRESS OF WRITE BUFFER

; SIZE OF BUFFER (EXTENT)

; BSELO AREA
; BSEL1 AREA
; SEL 2 AREA
; DATA AREA

; REREAD DATA (NEXT)
; REREAD DATA RETRY
; WRITE CONTINOUS
; END OF DATA

; TAPE TIMER COUNTER STORAGE AREA
; TAPE TIMER COUNTER STORAGE AREA
; DELAY COUNTER

```

```

5652          ;LOCAL TEXT MESSAGES FOR TEST
5653          ;-
5654
5655 075664    111    155    160 T36NAS: .ASCIZ 'Improper Tape Controller Buffering Speed'
5656 075735    124    141    160 T36WNG: .ASCIZ 'Tape Position Incorrect After REREAD Previous (OPP=1)'
5657 076023    124    123    123 T36RDF: .ASCIZ 'TSSR Incorrect After READ DATA Command'
5658 076072    122    105    122 T36RRF: .ASCIZ 'REREAD Previous (Space Reverse, Read Forward) Command Failed'
5659 076167    120    117    123 T36SC: .ASCIZ 'POSITION (Space Command) Failed, TSSR Not Correct'
5660 076251    122    111    102 T36LOR: .ASCIZ 'RIB NOT SET AFTER READ REVERSE INTO BOT'
5661 076321    124    123    123 T36WDF: .ASCIZ 'TSSR Not Correct After Illegal Mode Bits Set'
5662 076376    111    154    154 T36LOQ: .ASCIZ 'Illegal Mode Bits, Failed To Set ILC Bit In XST0'
5663 076457    122    105    122 T36SSR: .ASCIZ 'REREAD COMMAND Not Accepted'
5664 076513    124    123    123 T36WDE: .ASCIZ 'TSSR Not Correct After WRITE DATA Command'
5665 076565    124    141    160 T36BOT: .ASCIZ 'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'
5666 076660    127    122    111 T36TIM: .ASCIZ 'WRITE DATA RETRY'S Erase Tape Not Long Enough'
5667 076735    122    105    122 T36EOT: .ASCIZ 'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'
5668 077014    124    123    123 T36TM: .ASCIZ 'TSSR Not Correct After REREAD COMMAND Reject'
5669 077071    122    145    167 T36RMN: .ASCIZ 'Rewind (POSITION) Command Not Accepted'
5670 077140    122    101    115 T36RNC: .ASCIZ 'RAM Error, Correct Data Pattern Not In Ram'
5671 077213    124    123    123 T36AM3: .ASCIZ 'TSSR Init. Failed After REREAD COMMAND'
5672 077262    104    162    151 T36OFL: .ASCIZ 'Drive 7 Select Failed To Set "OFL" In TSSR'
5673 077335    124    123    123 T36WDD: .ASCIZ 'TSSR Not Correct After REREAD DATA Command, SWB Bit Set'
5674 077425    124    123    123 T36WDC: .ASCIZ 'TSSR Not Correct After REREAD DATA Command'
5675 077500    103    126    103 T36VCK: .ASCIZ 'CVC Set, Didn't Reset VCK In Message Buffer'
5676 077553    124    123    102 T36BA: .ASCIZ 'TSBA Not Correct After REREAD DATA Command'
5677 077626    127    122    111 T36WSS: .ASCIZ 'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'
5678 077715    122    145    141 T36LON: .ASCIZ 'Reading Long Record Failed To Set RLL Bit In XST0'
5679 077777    122    145    141 T36LOP: .ASCIZ 'Reading Long Record Failed To Set RLS Bit In XST0'
5680 100061    122    145    163 T36PBP: .ASCIZ 'Residual Byte Count Incorrect After Short Record Read'
5681 100147    122    145    141 T36TRL: .ASCIZ 'Reading Long Record Failed To Give Tape Status Alert'
5682 100235    127    122    111 T36NEF: .ASCIZ 'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3'
5683 100333    124    123    123 T36SCF: .ASCIZ 'TSSR Not Correct After SPACE RECORDS Command'
5684 100410    124    123    123 T36TSA: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY, Into BOT'
5685 100472    124    123    123 T36WRF: .ASCIZ 'TSSR Not Correct After WRITE DATA RETRY Command'
5686 100552    104    141    164 T36DTA: .ASCIZ 'Data Compare Error, Data Read From Tape Not Equal To Written'
5687 100647    122    145    143 TST36ID: .ASCIZ 'Record Buffering'
5688          .EVEN
5689          ;*
5690          ;
5691          ;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES
5692          ;WRITE SUBSYSTEM MEMORY COMMAND
5693          ;
5694          ;-
5695
5696 100670    T36REST:
5697 100670          SAVREG
5698 100674    012701 075510          MOV      #T36PACKET,R1          ;SAVE THE REGISTERS
5699 100700    012721 100004          MOV      #100004,(R1)+         ;START OF THE PACKET
5700 100704    012721 075520          MOV      #T36DATA,(R1)+       ;WRITE SUBSYSTEM MEM. WITH ACK.
5701 100710    005021                   CLR      (R1)+                 ;ADDRESS OF CHARAISTICS DATA BLOCK
5702 100712    012721 000012          MOV      #10.,(R1)+           ;EXTENDED ADDRESS
5703 100716    012721 075532          MOV      #T368FR,(R1)+       ;SIZE OF DATA BLOCK IN BYTES
5704 100722    005021                   CLR      (R1)+                 ;ADDRESS OF MESSAGE BUFFER
5705 100724    012721 000024          MOV      #20.,(R1)+           ;LENGTH OF MESSAGE BUFFER
5706 100730    005021                   CLR      (R1)+
5707 100732    012711 000000          MOV      #0,(R1)              ;SELECT DRIVE ZERO
5708 100736    012702 000030          MOV      #24.,R2              ;NUMBER OF LOCATIONS TO BE CLEARED

```



```

5709 100742 012762 177777 075532 64: MOV #177777,T36BFR(R2) ;ALL ONES TO MESSAGE BUFFER
5710 100750 005742 TST -(R2) ;NEXT LOCATION
5711 100752 022702 000000 CMP #0,R2 ;AT END OF LOOP YET
5712 100756 001371 BNE 64: ;KEEP GOING UNTIL DONE
5713 100760 000207 RTS PC ;RETURN
5714
5715 100762 T36RT2:
5716 100762 SAVREG ;SAVE THE REGISTERS
5717 100766 012701 075620 MOV #T36PK2,R1 ;START OF THE PACKET
5718 100772 012721 100006 MOV #100006,(R1). ;WRITE SUBSYSTEM MEM. WITH ACK.
5719 100776 012721 075640 MOV #T36BF2,(R1). ;ADDRESS OF DATA BLOCK
5720 101002 005021 CLR (R1). ;EXTENDED ADDRESS
5721 101004 012721 000006 MOV #6,(R1). ;SIZE OF DATA BLOCK IN BYTES
5722 101010 005021 CLR (R1).
5723 101012 012701 075640 MOV #T36BF2,R1 ;POINT TO DATA SEL AREA
5724 101016 005021 CLR (R1).
5725 101020 005011 CLR (R1)
5726 101022 000207 RTS PC ;RETURN
5727 101024 T36RT3:
5728 101024 SAVREG ;SAVE REGISTERS
5729 101030 012701 075630 MOV #T36PK3,R1 ;SET UP POINTER ADDRESS
5730 101034 005021 CLR (R1). ;COMMAND SPACE
5731 101036 005021 CLR (R1). ;ADDRESS OF DATA BLOCK
5732 101040 005021 CLR (R1). ;EXTENDED ADDRESS
5733 101042 005011 CLR (R1) ;SIZE OF DATA TRANSFER BLOCK
5734 101044 000207 RTS PC ;RETURN
5735 101046
101046 L10070: TRAP C#ETST
101046 104401
5736 .SBTTL TEST 9: FUNCTION TIMING
5737 ;*
5738 ;
5739 ;THIS TEST VERIFIES THAT THE TAPE TRANSPORT SEEMS TO BE WRITING
5740 ;RECORDS, GAPS, AND EXTENDED GAPS OF THE PROPER LENGTH. BOTH LOW
5741 ;AND HIGH SPEED MODES ARE TESTED. IT IS ALSO VERIFIED THAT A
5742 ;SPACE RECORDS COMMAND WITH A RECORD COUNT OF 80 OR MORE, AND A
5743 ;SKIP TAPE MARKS COMMAND WITH A COUNT OF 2 OF MORE, OPERATE THE
5744 ;TAPE IN HIGH-SPEED MODE. THIS TEST CAN ONLY BE RUN IF A
5745 ;REAL-TIME CLOCK IS AVAILIABLE ON THE SYSTEM. THE TEST OPERATES BY
5746 ;TIMING VARIOUS TAPE-MOTION OPERATIONS. USING A NUMBER OF
5747 ;DIFFERENT TEST RECORD LENGTHS.
5748 ;
5749 ;
5750 ;
5751 101050 BGNTST
101050 T9::
5752 101050 012737 006354 002172 MOV #EPRT1,EPRTSW ;PRIMARY ERROR MESSAGE
5753 101056 004737 017354 JSR PC,KTOFF ;TURN KT OFF
5758 101062 012700 105273 MOV #TST37ID,R0 ;ASCII MESSAGE TO IDENTIFY TEST
5759 101066 004737 016570 JSR PC,TSTSETUP ;DO INITIAL TEST SETUP
5760 101072 012737 000005 002210 MOV #5,LOOPCNT ;PERFORM 5 ITERATIONS
5761 101100 005037 102336 CLR T37CNT ;CLEAR TAPE RECORD COUNTER
5762 ;*
5763 ;
5764 ;TEST 9, SUBTEST 1
5765 ;
5766 ;

```


Line	Address	Label	Code	Comment	Instruction	Register	Value	Trap	Word
	101312	103515							T37RWN
	101314	012126							PKTSSR
5812	101316		30:	CKLOOP					
	101316	104406						TRAP	C#CLP1
5813	101320	013701	102220		MOV	T37BFR+6,R1			
5814	101324	010102			MOV	R1,R2			
5815	101326	052702	000002		BIS	#BIT1,R2			
5816	101332	020102			CMP	R1,R2			
5817	101334	001406			BEQ	40:			
5818	101336	005237	002214		INC	FATFLG			
5822	101342				ERRHRD	ERRNO,T37BOT,EXPREC			
	101342	104456						TRAP	C#ERHRD
	101344	001610						.WORD	904
	101346	103211						.WORD	T37BOT
	101350	015554						.WORD	EXPREC
5823	101352		40:	CKLOOP					
	101352	104406						TRAP	C#CLP1
5824	101354	012703	000144		MOV	#100.,R3			
5825	101360	013737	003116	102312	MOV	FREE,T37WB			
5826	101366	012737	140005	102310	65:	MOV	#140005,T37PK3		
5827	101374	012704	102310		MOV	#T37PK3,R4			
5828	101400	012737	001130	102316	MOV	#600.,T37SZ			
5829	101406	010465	000000		MOV	R4,TSDB(R5)			
5830	101412	004737	016330		JSR	PC,WAITF			
5831	101416	016501	000002		MOV	TSSR(R5),R1			
5832	101422	012702	000200		MOV	#SSR,R2			
5833	101426	020102			CMP	R1,R2			
5834	101430	001406			BEQ	70:			
5835	101432	005237	002214		INC	FATFLG			
5839	101436				ERRHRD	ERRNC,T37WDC,PKTSSR			
	101436	104456						TRAP	C#ERHRD
	101440	001611						.WORD	905
	101442	104051						.WORD	T37WDC
	101444	012126						.WORD	PKTSSR
5840	101446		70:	CKLOOP					
	101446	104406						TRAP	C#CLP1
5841	101450	005303			DEC	R3			
5842	101452	001345			BNE	65:			
5843	101454	004737	011074		JSR	PC,REWIND			
5844	101460	103411			BCS	130:			
5845	101462	016501	000002		MOV	TSSR(R5),R1			
5846	101466	010004			MOV	R0,R4			
5847	101470	005237	002214		INC	FATFLG			
5851	101474				ERRHRD	ERRNO,T37RWN,PKTSSR			
	101474	104456						TRAP	C#ERHRD
	101476	001612						.WORD	906
	101500	103515						.WORD	T37RWN
	101502	012126						.WORD	PKTSSR
5852	101504		130:	CKLOOP					
	101504	104406						TRAP	C#CLP1
5853	101506	013701	102220		MOV	T37BFR+6,R1			
5854	101512	010102			MOV	R1,R2			
5855	101514	052702	000002		BIS	#BIT1,R2			
5856	101520	020102			CMP	R1,R2			
5857	101522	001406			BEQ	140:			
5858	101524	005237	002214		INC	FATFLG			
5862	101530				ERRHRD	ERRNO,T37BOT,EXPREC			

	101530	104456					TRAP	C1ERMRO
	101532	001613					.WORD	907
	101534	103211					.WORD	T37BOT
	101536	015554					.WORD	EXPREC
5863	101540		1404:	CKLOOP				; LOOP IF SELECTED
	101540	104406					TRAP	C1CLP1
5864	101542	012704	102310	MOV	#T37PK3,R4			; SET UP PACKET ADDRESS
5865	101546	012737	000037	MOV	#31.,T37RB			; SET UP RECORDS TO SPACE OVER
5866	101554	012737	140010	MOV	#140010,T37PK3			; ACK,CVC=1,SPACE FORWARD COMMAND
5867	101562	010465	000000	1504:	MOV	R4,TSD8(R5)		; ISSUE COMMAND
5868	101566	005237	102336	1574:	INC	T37CNT		; BUMP TIMER
5869	101572			DELAY	1			; DELAY ABOUT 100US
	101572	012727	000001				MOV	#1,(PC).
	101576	000000					.WORD	0
	101600	013727	002116				MOV	LSDLY,(PC).
	101604	000000					.WORD	0
	101606	005367	177772				DEC	-6(PC)
	101612	001375					BNE	. 4
	101614	005367	177756				DEC	22(PC)
	101620	001367					BNE	. 20
5870	101622	016501	000002	MOV	TSSR(R5),R1			; GET TSSR
5871	101626	032701	000200	BIT	#SSR,R1			; CHECK FOR TSSR'S SSR SET
5872	101632	001755		BEQ	1524			; KEEP COUNTING UNTIL SET
5873	101634	012702	000200	MOV	#SSR,R2			; SET UP EXPECTED
5874	101640	020201		CMP	R2,R1			; WAS EVERYTHING OK
5875	101642	001406		BEQ	1604			; BR, IF ALL IS WELL
5876	101644	005237	002214	INC	FATFLG			; ERROR COUNT
5880	101650			ERRMRO	ERRNO,T37SCF,PKTSSR			; SPACE FORWARD DIDN T WORK OUT
	101650	104456					TRAP	C1ERMRO
	101652	001614					.WORD	908
	101654	104757					.WORD	T37SCF
	101656	012126					.WORD	PKTSSR
5881	101660		1604:	CKLOOP				; LOOP IF SELECTED
	101660	104406					TRAP	C1CLP1
5882	101662	004737	011074	JSR	PC,REWIND			; CALL TAPE REWIND COMMAND
5883	101666	103411		BCS	1704			; BR, IF NO PROBLEM
5884	101670	010004		MOV	R0,R4			; GET PACKET ADDRESS
5885	101672	016501	000002	MOV	TSSR(R5),R1			; GET STATUS FROM TSSR
5886	101676	005237	002214	INC	FATFLG			; ERROR COUNT
5890	101702			ERRMRO	ERRNO,T37RWN,PKTSSR			; REWIND NOT ACCEPTED
	101702	104456					TRAP	C1ERMRO
	101704	001615					.WORD	909
	101706	103515					.WORD	T37RWN
	101710	012126					.WORD	PKTSSR
5891	101712		1704:	CKLOOP				; LOOP IF SELECTED
	101712	104406					TRAP	C1CLP1
5892	101714	013701	102220	MOV	T37BFR.6,R1			; PICK UP XSTO
5893	101720	010102		MOV	R1,R2			; SET UP EXPECTED
5894	101722	052702	000002	BIS	#BIT1,R2			; SET BOT BIT IN EXPECTED
5895	101726	020102		CMP	R1,R2			; DOES EXP = REC 0
5896	101730	001406		BEQ	1754			; BR, IF EQUAL (OK)
5897	101732	005237	002214	INC	FATFLG			; ERROR COUNT
5901	101736			ERRMRO	ERRNO,T37BOT,EXPREC			; TAPE NOT AT BOT AFTER REWIND
	101736	104456					TRAP	C1ERMRO
	101740	001616					.WORD	910
	101742	103211					.WORD	T37BOT
	101744	015554					.WORD	EXPREC

TEST 1 HARDWARE TEST 1 B TEST MACRO M1113 14 JUN 84 16:41
TEST 9: FUNCTION TIMING

SEQ 0219

5902	101746			1758	CKLOOP		:LOOP IF SELECTED		
	101746	104406						TRAP	C1CLP1
5903	101750	012704	102310		MOV	@T37PK3,R4	:SET UP PACKET ADDRESS		
5904	101754	012737	000121	102312	MOV	@B1..T37RB	:SET UP RECORDS TO SPACE OVER		
5905	101762	012737	140010	102310	MOV	@140010,T37PK3	:ACK,CVC=1,SPACE FORWARD COMMAND		
5906	101770	010465	000000		MOV	R4.TSDB(R5)	:ISSUE COMMAND		
5907	101774	005237	102340	2528:	INC	T37CNT	:BUMP TIMER		
5908	102000			2528:	DELAY	1	:DELAY ABOUT 100US		
	102000	012727	000001					MOV	@1.(PC)
	102004	000000						WORD	C
	102006	013727	002116					MOV	L10L*(PC)
	102012	000000						WORD	0
	102014	005367	17772					DEC	6(PC)
	102020	001375						BNE	4
	102022	005367	177756					DEC	22(PC)
	102026	001367						BNE	20
5909	102030	016501	000002		MOV	TSSR(R5),R1	:GET TSSR		
5910	102034	032701	000200		BIT	@SSR,R1	:CHECK FOR TSSR S SSR SET		
5911	102040	001755			BEQ	2528	:KEEP COUNTING UNTIL SET		
5912	102042	012702	000200		MOV	@SSR,R2	:SET UP EXPECTED		
5913	102046	020201			CMP	R2,R1	:WAS EVERYTHING OK		
5914	102050	001406			BEQ	2608	:BR. IF ALL IS WELL		
5915	102052	005237	002214		INC	FATFLG	:ERROR COUNT		
5916	102056				ERRAND	ERRAND.T37SCF,PXTSSR	:SPACE FORWARD DIDN'T WORK OUT		
	102056	104456						TRAP	C1ERRND
	102060	001617						WORD	911
	102062	104757						WORD	T37SCF
	102064	012126						WORD	PXTSSR
5920	102066			2608	CKLOOP		:LOOP IF SELECTED		
	102066	104406						TRAP	C1CLP1
5921	102070	013701	102336		MOV	T37CNT,R1	:TIME FOR WRITE SPACING		
5922	102074	013702	102340		MOV	T37CNT,R2	:TIME FOR WRITE RETRY SPACING		
5923	102100	042701	000077		BIC	@000077,R1	:SETTING UP CONSTANTS		
5924	102104	042702	000077		BIC	@000077,R2	:SETTING UP CONSTANTS		
5925	102110	020102			CMP	R1,R2	:CHECK FOR DIFFERENCE		
5926	102112	003406			BLE	3008	:BR. IF GOOD CHECK 2000		
5927	102114	005237	002214		INC	FATFLG	:ERROR COUNT		
5931	102120				ERRAND	ERRAND.T37TIM,EXPREC	:TIME WAS NOT DIFFERENT ENOUGH		
	102120	104456						TRAP	C1ERRND
	102122	001620						WORD	912
	102124	103304						WORD	T37TIM
	102126	015554						WORD	EXPREC
5932	102130			3008:	CKLOOP		:LOOP IF SELECTED		
	102130	104406						TRAP	C1CLP1
5933	102132				ENDSUB		:..... END SUBTEST		
	102132							L10074:	
	102132	104403						TRAP	C1ESUB
5934	102134	023727	002214	000017	CMP	FATFLG.#15.	:IS ERROR COUNT AT 25		
5935	102142	103402			BLO	9998	:BR. IF LESS THAN 25		
5936	102144	004737	017262		JSR	PC,CXDROP	:TRY TO DROP THE UNIT		
5937	102150			9998:					
5938				:					
5939				:					
5940				:					
5941	102150	004737	016536		JSR	PC,TSTLOOP	:DO WE NEED TO ITERATE TEST		
5942	102154	103002			BCC	1638	:BR. IF NO LOOP REQUIRED		
5943	102156	000137	101104		JMP	T37LOOP	:EXECUTE AGAIN		

5944	102162		1634:			
5945	102162			EXIT	TST	; ALL DONE THIS TEST
	102162	104432				TRAP
	102164	003306				.WORD
						C0EXIT
5946						L10073
5947						
5948						
5950		102170				
5952	102170					
5953	102170	100004				
5954	102172	102200				
5955	102174	000000				
5956	102176	000012				
5957	102200					
5958	102200	102212				
5959	102202	000000				
5960	102204	000024				
5961	102206	000000				
5962	102210	000000				
5963	102212					
5964						
5965						
5966						
5968		102300				
5970	102300					
5971	102300	100006				
5972	102302	102320				
5973	102304	000000				
5974	102306	000006				
5975						
5979	102310					
5980	102310	100005				
5981	102312					
5982	102312	003116				
5983	102314	000000				
5984	102316	000000				
5985						
5986						
5987						
5988						
5989	102320					
5990	102320	010				
5991	102321	200				
5992	102322	000000				
5993	102324	000000				
5994						
5995						
5996						
5997						
5998						
5999	102326	100205				
6000	102330	100605				
6001	102332	102205				
6002	102334	177777				
6003						
6004						
6005	102336	000000				

6006	102340	000000				T37CNU: .WORD 0		;TAPE TIMER COUNTER STORAGE AREA
6007	102342	000000				T37DLY: .WORD 0		;DELAY COUNTER
6008						;		
6009						;LOCAL TEXT MESSAGES FOR TEST		
6010						;		
6011								
6012	102344	124	141	160	T37WNG: .ASCIZ	'Tape Position Incorrect After REREAD Previous (OPP=1)		
6013	102432	124	123	123	T37RDF: .ASCIZ	'TSSR Incorrect After READ DATA Command'		
6014	102501	122	105	122	T37RRF: .ASCIZ	'REREAD Previous (Space Reverse, Read Forward) Command Failed		
6015	102576	120	117	123	T37SC: .ASCIZ	'POSITION (Space Command) Failed, TSSR Not Correct'		
6016	102660	122	111	102	T37LOR: .ASCIZ	'RIB NOT SET AFTER READ REVERSE INTO BOT'		
6017	102730	124	123	123	T37WDF: .ASCIZ	'TSSR Not Correct After Illegal Mode Bits Set		
6018	103005	111	154	154	T37LOQ: .ASCIZ	'Illegal Mode Bits, Failed To Set ILC Bit In XST0'		
6019	103066	122	105	122	T37SSR: .ASCIZ	'REREAD COMMAND Not Accepted'		
6020	103122	124	123	123	T37WDE: .ASCIZ	'TSSR Not Correct After WRITE DATA RETRY Command, At BOT'		
6021	103211	124	141	160	T37BOT: .ASCIZ	'Tape Not At BOT After REWIND Command (BOT Not Set In XST0)'		
6022	103304	127	122	111	T37TIM: .ASCIZ	'WRITE DATA RETRY'S Erase Tape Not Long Enough'		
6023	103361	122	105	122	T37EOT: .ASCIZ	'REREAD DATA OVER EOT GAVE NO TAPE STATUS ALERT'		
6024	103440	124	123	123	T37TM: .ASCIZ	'TSSR Not Correct After REREAD COMMAND Reject		
6025	103515	122	145	167	T37RMN: .ASCIZ	'Rewind (POSITION) Command Not Accepted'		
6026	103564	122	101	115	T37RMC: .ASCIZ	'RAM Error, Current Data Pattern Not In Ram'		
6027	103637	124	123	123	T37AM3: .ASCIZ	'TSSR Init. Failed After REREAD COMMAND'		
6028	103706	104	162	151	T37OFL: .ASCIZ	'Drive 7 Select Failed To Set "OFL" In TSSR'		
6029	103761	124	123	123	T37MDO: .ASCIZ	'TSSR Not Correct After REREAD DATA Command, SMB Bit Set'		
6030	104051	124	123	123	T37MDC: .ASCIZ	'TSSR Not Correct After REREAD DATA Command'		
6031	104124	103	126	103	T37VCK: .ASCIZ	'CVC Set, Didn't Reset VCK In Message Buffer'		
6032	104177	124	123	102	T37BA: .ASCIZ	'TSBA Not Correct After REREAD DATA Command'		
6033	104252	127	122	111	T37MSS: .ASCIZ	'WRITE SUBSYSTEM MEMORY Command Not Accepted (RAM Read)'		
6034	104341	122	145	141	T37LON: .ASCIZ	'Reading Long Record Failed To Set RLL Bit In XST0'		
6035	104423	122	145	141	T37LOP: .ASCIZ	'Reading Long Record Failed To Set RLS Bit In XST0'		
6036	104505	122	145	163	T37PBP: .ASCIZ	'Residual Byte Count Incorrect After Short Record Read'		
6037	104573	122	145	141	T37TRL: .ASCIZ	'Reading Long Record Failed To Give Tape Status Alert'		
6038	104661	127	122	111	T37NEF: .ASCIZ	'WRITE DATA RETRY, At First Record, Failed To Set RIB Bit XST3		
6039	104757	124	123	123	T37SCF: .ASCIZ	'TSSR Not Correct After SPACE RECORDS Command'		
6040	105034	124	123	123	T37TSA: .ASCIZ	'TSSR Not Correct After WRITE DATA RETRY, Into BOT'		
6041	105116	124	123	123	T37WRF: .ASCIZ	'TSSR Not Correct After WRITE DATA RETRY Command'		
6042	105176	104	141	164	T37DTA: .ASCIZ	'Data Compare Error, Data Read From Tape Not Equal To Written		
6043	105273	106	165	156	T37ID: .ASCIZ	'Function Timing'		
6044						.EVEN		
6045					;			
6046					;			
6047					;ROUTINE TO RESTORE COMMAND PACKET TO START-UP (DEFAULT) VALUES			
6048					;WRITE SUBSYSTEM MEMORY COMMAND			
6049					;			
6050					;			
6051								
6052	105314				T37REST:			
6053	105314				SAVREG			;SAVE THE REGISTERS
6054	105320	012701	102170		MOV	@T37PACKET,R1		;START OF THE PACKET
6055	105324	012721	100004		MOV	@100004,(R1).		;WRITE SUBSYSTEM MEM. WITH ACK.
6056	105330	012721	102200		MOV	@T37DATA,(R1).		;ADDRESS OF CHARACTERISTICS DATA BLOCK
6057	105334	005021			CLR	(R1).		;EXTENDED ADDRESS
6058	105336	012721	000012		MOV	@10.,(R1).		;SIZE OF DATA BLOCK IN BYTES
6059	105342	012721	102212		MOV	@T37BFR,(R1).		;ADDRESS OF MESSAGE BUFFER
6060	105346	005021			CLR	(R1).		
6061	105350	012721	000024		MOV	@20.,(R1).		;LENGTH OF MESSAGE BUFFER
6062	105354	005021			CLR	(R1).		

```

6063 105356 012711 000000      MOV      #0,(R1)          ;SELECT DRIVE ZERO
6064 105362 012702 000030      MOV      #24.,R2         ;NUMBER OF LOCATIONS TO BE CLEARED
6065 105366 012762 177777 102212 644:  MOV      #177777,T37BFR(R2) ;ALL ONES TO MESSAGE BUFFER
6066 105374 005742              TST      -(R2)           ;NEXT LOCATION
6067 105376 022702 000000      CMP      #0,R2          ;AT END OF LOOP YET
6068 105402 001371              BNE      644            ;KEEP GOING UNTIL DONE
6069 105404 000207              RTS      PC              ;RETURN
6070
6071 105406              T37RT2:
6072 105406              SAVREG
6073 105412 012701 102300      MOV      #T37PK2,R1     ;SAVE THE REGISTERS
6074 105416 012721 100006      MOV      #100006,(R1).  ;START OF THE PACKET
6075 105422 012721 102320      MOV      #T37BF2,(R1). ;WRITE SUBSYSTEM MEM. WITH ACK,
6076 105426 005021              CLR      (R1).          ;ADDRESS OF DATA BLOCK
6077 105430 012721 000006      MOV      #6.,(R1).      ;EXTENDED ADDRESS
6078 105434 005021              CLR      (R1).          ;SIZE OF DATA BLOCK IN BYTES
6079 105436 012701 102320      MOV      #T37BF2,R1     ;POINT TO DATA SEL AREA
6080 105442 005021              CLR      (R1).
6081 105444 005011              CLR      (R1)
6082 105446 000207              RTS      PC              ;RETURN
6083 105450              T37RT3:
6084 105450              SAVREG
6085 105454 012701 102310      MOV      #T37PK3,R1     ;SAVE REGISTERS
6086 105460 005021              CLR      (R1).          ;SET UP POINTER ADDRESS
6087 105462 005021              CLR      (R1).          ;COMMAND SPACE
6088 105464 005021              CLR      (R1).          ;ADDRESS OF DATA BLOCK
6089 105466 005011              CLR      (R1).          ;EXTENDED ADDRESS
6090 105470 000207              RTS      PC              ;SIZE OF DATA TRANSFER BLOCK
6091 105472              ENDTST
        105472
        105472 104401
6092 105474              ENDMOD
        L10073: TRAP C1ETST
  
```



```

:
:
12
18
19 105474
105474
20
21
22
23
24 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
25 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
26 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
27 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
28 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
29 ; WITH THE OPERATOR.
30
31 105474
105474 000010
105476
32
33 105476
105476 000031
105500 105516
105502 160010
105504 177776
34 105506
105506 001031
105510 105552
105512 000000
105514 000776
35
36 105516
105516
105516
37 105516 104 105 126
38 105552 111 116 124
39 105576 111 116 124
40

          .TITLE TSV6 PARAMETER CODING

          BGNMOD TSV6
TSV6::

          .SBTTL HARDWARE PARAMETER CODING SECTION

          ;**
          ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
          ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
          ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
          ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
          ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
          ; WITH THE OPERATOR.
          ;
          ;
          BGNHRD
          .WORD L10075-L#HARD/2
L#HARD::

          GPRMA HPM1,0,0,160010,177776,YES ;GET TSBA/TSDB REGISTER ADDRESS.
          .WORD T#CODE
          .WORD HPM1
          .WORD T#LOLIM
          .WORD T#HILIM
          GPRMA HPM2,2,0,0,776,YES ;GET VECTOR ADDRESS.
          .WORD T#CODE
          .WORD HPM2
          .WORD T#LOLIM
          .WORD T#HILIM
          ;GPRMD HPM3,4,0,340,0,7,YES ;GET INTERRUPT PRIORITY.
          ENDRD
          .EVEN

          L10075:
          HPM1: .ASCIZ 'DEVICE ADDRESS (TSBA/TSDB) '
          HPM2: .ASCIZ 'INTERRUPT VECTOR '
          HPM3: .ASCIZ 'INTERRUPT PRIORITY '
          .EVEN
  
```

```

42                                     .SBTTL SOFTWARE PARAMETER CODING SECTION
43
44                                     ;**
45                                     ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
46                                     ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
47                                     ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
48                                     ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
49                                     ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
50                                     ; WITH THE OPERATOR.
51                                     ;--
52 105626                                BGNSFT
53 105626 000003                          .WORD L10076-L#SOFT/2
54 105630                                L#SOFT::
55 105630 001130                          ; GPRML SPM1,0,-1,YES ; GET TRANSPORT TEST FLAG
56 105632 105666                          ; GPRML SPM4,2,-1,YES ; GET ITERATION CONTROL.
57 105634 177777                          .WORD T#CODE
58                                     .WORD SPM4
59                                     .WORD -1
60                                     ; GPRMD SPM6,4,D,7777,0,7777,YES ; GET LOCAL ERROR LIMIT
61                                     ; GPRMD SPM7,6,D,7777,0,7777,YES ; GET GLOBAL ERROR LIMIT
62                                     ENDSFT
63                                     .EVEN
64
65                                     L10076:
66 105636
67 105636 105 116 101 SPM1: .ASCIZ 'ENABLE TRANSPORT TESTS '
68 105666 111 116 110 SPM4: .ASCIZ 'INHIBIT ITERATIONS '
69 105716 120 105 122 SPM6: .ASCIZ 'PER TEST ERROR LIMIT '
70 105746 120 105 122 SPM7: .ASCIZ 'PER UNIT ERROR LIMIT '
71                                     .SBTTL PATCH AREA
72
73                                     ;
74                                     ; FINALLY A GENEROUS PATCH AREA.
75                                     ;
76                                     ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
77                                     ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
78                                     ;
79
80 PATCH::
81                                     .BLKW 32.
82                                     ., !377+1
83 106400 106400 LASTAD ;SET LAST USED ADDRESS.
84 106400 000000 .EVEN
85 106402 000000 .WORD 0
86 106404 .WORD 0
87
88 L#LAST::
89 106404 ENDMOD
90 106404 .END
  
```

ADSSR	012206	G	C#AU	=	000052	DEVDR0	023412	FRESIZ	003120	G	INTFLA	016225			
ADR	=	000020	C#AUTO	=	000061	DEVNRD	023331	FUSI	004113		INTMAS	016224			
AMBTSS	006713		C#BRK	=	000022	DEVNXR	023247	F#AU	=	000015	INTR	016276	G		
ASSEMB	=	000010	C#BSEG	=	000004	DEVONL	023177	F#AUTO	=	000020	INTREC	002216	G		
A1716	=	000003	C#BSUB	=	000002	DEVSUM	023142	F#BGN	=	000040	INTVEC	016226			
BADDAT	003150	G	C#CEFG	=	000045	DFPTBL	002150	F#CLEA	=	000007	INTX	004274			
BADSSR	015760	G	C#CLCK	=	000062	DIAGMC	=	000000	F#DU	=	000016	INVERT	021222	G	
BDVPCR	=	177520	C#CLEA	=	000012	DICED	=	000001	F#END	=	000041	IOKCKI	=	000200	
BENBSW	002222	G	C#CLOS	=	000035	DSBINT	016264	F#HARD	=	000004	IOKSTP	=	000001		
BIE	=	040000	C#CLP1	=	000006	DUAD12	004637	F#M	=	000013	IPRI	002204	G		
BITC	=	000001	C#CVEC	=	000036	DUFLG	003104	F#INIT	=	000006	ISR	=	000100	G	
BIT00	=	000001	C#DCLN	=	000044	DUMMY	003054	F#JMP	=	000050	IVEC	002202	G		
BIT01	=	000002	C#DODU	=	000051	EF.CON	=	000036	F#MOD	=	000000	IXE	=	004000	G
BIT02	=	000004	C#DRPT	=	000024	EF.MEW	=	000035	F#MSG	=	000011	I#AU	=	000041	
BIT03	=	000010	C#DU	=	000053	EF.PWR	=	000034	F#PROT	=	000021	I#AUTO	=	000041	
BIT04	=	000020	C#EDIT	=	000003	EF.RES	=	000037	F#PWR	=	000017	I#CLN	=	000041	
BIT05	=	000040	C#ERDF	=	000055	EF.STA	=	000040	F#RPT	=	000012	I#DU	=	000041	
BIT06	=	000100	C#ERRR	=	000056	EMAXDU	017057	F#SEG	=	000003	I#HRD	=	000041		
BIT07	=	000200	C#ERRO	=	000060	EN	=	000000	F#SOFT	=	000005	I#INIT	=	000041	
BIT08	=	000400	C#ERSF	=	000054	ENAIN	016232	F#SRV	=	000010	I#MOD	=	000041		
BIT09	=	001000	C#ERSO	=	000057	ENVIRN	020710	F#SUB	=	000002	I#MSG	=	000041		
BIT1	=	000002	C#ESCA	=	000010	EPRTSW	002172	F#SW	=	000014	I#PROT	=	000040		
BIT10	=	002000	C#ESEG	=	000005	EPRT1	006354	F#TEST	=	000001	I#PTAB	=	000041		
BIT11	=	004000	C#ESUB	=	000003	EPRT2	006413	GDDAT	003152	G	I#PWR	=	000041		
BIT12	=	010000	C#ETST	=	000001	ERRCM	012013	GERRMA	002166	G	I#RPT	=	000041		
BIT13	=	020000	C#EXIT	=	000032	ERRMI	002230	GETPAT	020254	G	I#SEG	=	000041		
BIT14	=	040000	C#GETB	=	000026	ERRK	017036	GETSEL	020336	G	I#SETU	=	000041		
BIT15	=	100000	C#GETW	=	000027	ERRLO	002232	G#CNT0	=	000200	I#SFT	=	000041		
BIT2	=	000004	C#GMAN	=	000043	ERRNO	=	001620	G#DELM	=	000372	I#SRV	=	000041	
BIT3	=	000010	C#GPHR	=	000042	ERRVEC	=	000004	G#DISP	=	000003	I#SUB	=	000041	
BIT4	=	000020	C#GPLO	=	000030	ERTABE	003370	G#EXCP	=	000400	I#TST	=	000041		
BIT5	=	000040	C#GPRI	=	000040	ERTABL	003170	G#HILI	=	000002	J#JMP	=	000167		
BIT6	=	000100	C#INIT	=	000011	ESUM	017040	G#LOLI	=	000001	KIPAR0	=	172340		
BIT7	=	000200	C#INLP	=	000020	EVL	=	000004	G#NO	=	000000	KIPAR1	=	172342	
BIT8	=	000400	C#MANI	=	000050	EXBCNT	=	000010	G#OFFS	=	000400	KIPAR2	=	172344	
BIT9	=	001000	C#MEM	=	000031	EXPBRE	015562	G#OSI	=	000376	KIPAR3	=	172346		
BOE	=	000400	C#MSG	=	000023	EXPD	002224	G#PRMA	=	000001	KIPAR4	=	172350		
BRINIT	004453		C#OPEN	=	000034	EXPGT	004527	G#PRMD	=	000002	KIPAR5	=	172352		
BSELO	=	000000	C#PNTB	=	000014	EXPGT2	004563	G#PRML	=	000000	KIPAR6	=	172354		
BSEL1	=	000001	C#PNTF	=	000017	EXPMSG	002314	G#RADA	=	000140	KIPAR7	=	172356		
CHKAMB	016124		C#PNTS	=	000016	EXPREC	015554	G#RADB	=	000000	KIPDR0	=	172300		
CHKMAN	020560	G	C#PNTX	=	000015	EXTA	005766	G#RADD	=	000040	KIPDR1	=	172302		
CHKTSJ	016416		C#QIO	=	000377	EXTEND	005764	G#RADL	=	000120	KIPDR2	=	172304		
CKDROP	017262		C#RDBU	=	000007	EXTFEA	002220	G#RADO	=	000020	KIPDR3	=	172306		
CKEMAX	017162		C#REFG	=	000047	E#END	=	002100	G#XFER	=	000004	KIPDR4	=	172310	
CKMSG	011440	G	C#RESE	=	000033	E#LOAD	=	000035	G#YES	=	000010	KIPDR5	=	172312	
CKMSG2	011560	G	C#REVI	=	000003	FATERR	=	000060	HIADDR	=	001400	KIPDR6	=	172314	
CKRAM	011174	G	C#RFLA	=	000021	FATFLG	002214	HOE	=	100000	G	KIPDR7	=	172316	
CKRAM2	011304	G	C#RPT	=	000025	FERCM	012002	HPM1	105516		KTENAB	003126	G		
CHDPKT	021274	G	C#SEFG	=	000046	FIFEXP	012250	HPM2	105552		KTFLG	003124	G		
CHPMEM	017740		C#SPRI	=	000041	FIF1MS	012322	HPM3	105576		KTINIT	021070			
CONFIG	017330		L#SVEC	=	000037	FIF2MS	012371	I#E	=	010000	G	KTOFF	017354		
COUNT	0023	G	C#TPRI	=	000013	FILLME	017502	IDU	=	000040	G	KTON	017336		
CSRADD	0022	G	DATA	002304	G	FNOINT	004211	IER	=	020000	G	LERRMA	002164	G	
CTAB	00315	G	DATASC	020312		FORCER	002170	IFALT	004252		LISTAL	=	000001		
CTABE	00317	G	DEBUGM	011712		FREE	003116	INCERK	017124		LOE	=	040000	G	
CTABM	00315	G	DEVcnt	002212	G	FREEHI	003122	INTCPC	016230		LOOPCN	002210	G		

LOOPCO	013206	L10001	002170	L10073	105472	O1ERRT	000000	PST32W	003144 G
LOOPFL	003154 G	L10002	005762	L10074	102132	O1GNSW	000001	PUNIT	022320
LOT	000010 G	L10003	012124	L10075	105516	O1POIN	000001	PW.D11	000021
L\$ACP	002110 G	L10004	012142	L10076	105636	O1SETU	000000	PW.D13	000022
L\$APT	002036 G	L10005	012160	MEMADD	014034 G	PASRPT	022070	PW.D22	000020
L\$AU	022366 G	L10006	012166	MEMCK	021312 G	PATCH	105776 G	PW.NOP	000000
L\$AUT	002070 G	L10007	012204	MENASC	020527	PATDAT	020310	PW.NO1	000023
L\$AUTO	022572 G	L10010	012222	MENERR	020454	PC.ERA	002400	PW.RDE	000024
L\$CCP	002106 G	L10011	012246	MENRES	020556	PC.IER	002000	PW.RDR	000001
L\$CLEA	022652 G	L10012	012320	MMVEC	000250	PC.MOO	001000	PW.RDS	000005
L\$CO	002032 G	L10013	012470	MSA.FR	000006	PC.REI	000000	PW.RFI	000003
L\$DEPO	002011 G	L10014	013204	MSA.NO	000000	PC.REW	000400	PW.WCT	000006
L\$DESC	003402 G	L10015	014032	MSA.NR	000004	PKBCNT	000006	PW.WFI	000004
L\$DESP	002076 G	L10016	014054	MSA.VO	000002	PKHI	000004	PW.WFM	000007
L\$DEVP	002060 G	L10017	015560	MSGEXP	012224 G	PKLOW	000002	PW.WMI	000010
L\$DISP	002124 G	L10020	015566	MSGLOO	013144 G	PKTADC	007632	PW.WNP	000011
L\$DLY	002116 G	L10021	015574	MSGSTA	012430 G	PKTFRM	007574	PW.WTR	000002
L\$DTP	002040 G	L10022	015606	MSGSUB	014022 G	PKTGET	012144 G	P.ACK	100000
L\$DTYP	002034 G	L10023	015630	MS.ATT	000006	PKTMES	012170 G	P.CMD	000037
L\$DU	022464 G	L10024	015656	MS.EXT	000200	PKTRAM	004741 G	P.CONT	000012
L\$DUT	002072 G	L10025	016016	MS.RSD	000001	PKTSSR	012126 G	P.CVC	040000
L\$DVTY	003374 G	L10026	016326	MS.RSF	000020	PNT	001000 G	P.FMT	000140
L\$EF	002052 G	L10030	022316	MS.RST	000010	PRAMPK	014056	P.FORM	000011
L\$ENVI	002044 G	L10031	022462	M8186	005550	PRASC	014603	P.GETS	000017
L\$ETP	002102 G	L10032	022570	M8189	005641	PRBEXP	015550	P.IE	000200
L\$EXP1	002046 G	L10033	022650	NBA	002000	PRBMSG	015416	P.INIT	000013
L\$EXP4	002064 G	L10034	022676	NEWPAS	022024	PRBREC	015552	P.MODE	007400
L\$EXP5	002066 G	L10035	023140	NODEV	003106 G	PRBTOT	015503	P.OPP	020000
L\$HARD	105476 G	L10036	032262	NOINIT	004331	PRBYTE	015202 G	P.POSI	000010
L\$HIME	002120 G	L10037	024124	NOINTR	004215	PRI	002000 G	P.READ	000001
L\$HPCP	002016 G	L10040	024646	NOITS	002162 G	PRIADD	010236	P.SWB	010000
L\$HPTP	002022 G	L10041	025372	NOMAN	020614	PRIAO	010306	P.WRIT	000005
L\$HW	002150 G	L10042	026214	NOMEM	005454	PRI BXO	007670 G	P.WRTC	000004
L\$ICP	002104 G	L10043	041360	NP.IR	000200	PRIEQU	010136	P.WRTS	000006
L\$INIT	021572 G	L10044	033664	NP.LOO	000040	PRIPKT	007446 G	QVP	002176 G
L\$LADP	002026 G	L10045	035310	NP.OUT	000100	PRIRAM	010144	RAMASC	014236
L\$LAST	106404 G	L10046	035704	NP.WRP	000020	PRITAD	010352	RAMDAT	002234 G
L\$LOAD	002100 G	L10047	036370	NSI	004146	PRITSS	006020	RAMERR	015570 G
L\$LUN	002074 G	L10050	046716	NSINIT	004403	PRITO	010434	RAMEXP	0'5610 G
L\$MREV	002050 G	L10051	042252	NUL	004523	PRITI	010477	RAMFOR	0.0174
L\$NAME	002000 G	L10052	043064	NULCR	004524	PRIXOR	010020 G	RAMSIZ	002274 G
L\$PRIO	002042 G	L10053	052774	NXM	004000	PRI00	000000 G	RAMTAD	015576 G
L\$PROT	021562 G	L10054	047572	NXMFLG	003130 G	PRI01	000040 G	RCVHIA	002276 G
L\$PRT	002112 G	L10055	050402	NXMI	003134 G	PRI02	000100 G	RCVLOA	002300 G
L\$REPP	002062 G	L10056	051216	NXMLO	003132 G	PRI03	000140 G	RDERR	005202
L\$REV	002010 G	L10057	055770	NXMTST	021466	PRI04	000200 G	RECHSG	002460 G
L\$RPT	022700 G	L10060	054436	NXR	003734	PRI05	000240 G	RECV	002226 G
L\$SOFT	105630 G	L10061	063342	NXRERR	005732 G	PRI06	000300 G	REGSAV	020220
L\$SPC	002056 G	L10062	060426	NXRX	003773	PRI07	000340 G	RETERR	005366
L\$SPCP	002020 G	L10063	073272	NXTU	022036	PRMESS	014322	REWIND	011074 G
L\$SPTP	002024 G	L10064	064434	OFL	000100	PRMNO	002312 G	RMCHBE	000167
L\$STA	002030 G	L10065	065514	ONEFIL	000000	PRMSGE	014632 G	RMCHEN	000200
L\$SW	002160 G	L10066	066356	O1APTS	000000	PRMSG0	015012	RMMSGB	000215
L\$TEST	002114 G	L10067	067260	O1AU	000001	PRMSG1	015057	RMMSGE	000234
L\$TIML	002014 G	L10070	101046	O1BGNR	000001	PRMSG2	015115	RMPKTB	000201
L\$UNIT	002012 G	L10071	074366	O1BGNS	000001	PROASC	014500	RMPKTE	000210
L10000	002156	L10072	075450	O1DU	000001	PR1ASC	014545	RMR	010000

RMPACK	011170	S2.INR=	000020	T#EXCP=	000000	T29CON	026412	T30BOT	037771
SC	= 100000	S2.OUT=	000040	T#FLAG=	000040	T29DAT	026260	T30BS0	036560
SCE	= 020000	S2.UND=	000003	T#GMAN=	000000	T29DLY	026430	T30BS1	036561
SCHERR	005274	1BLEND=	003054 G	T#HILI=	000776	T29DSW	026270	T30CNT	036600
SCME	005007	TCOASC	006554	T#LAST=	000001	T29DTA	027773	T30CNU	036602
SDELAY	010740	TCOCOD	006754	T#LOLI=	000000	T29EOT	030061	T30DAT	036440
SELASC	020522	TEMP1	003110 G	T#LSYM=	010000	T29LON	031155	T30DLY	036606
SELDAI=	000004	TEMP2	003112 G	T#LTNO=	000011	T29LOO	023512	T30DSW	036450
SEL2	= 000002	TERCLS=	000016	T#NEST=	177777	T29LOP	031237	T30DTA	041064
SETMAP	017376	TESTNO=	000011	T#NS0 =	000000	T29LOQ	027356	T30DTR	041020
SETU	022122	TEXASC	006513	T#NS1 =	000005	T29LOR	027231	T30ETH	036446
SFFMSG	012162 G	TFCASC	006615	T#NS2 =	000002	T29NEF	026560	T30FCN	036604
SFHERR	003701	TIMEXP	015632 G	T#PTNU=	000000	T29NEQ	031475	T30IBT	036761
SFIERR	003646	TIMSGO	015660	T#SAVL=	177777	T29OFL	026432	T30IBU	036610
SFIMSG	012114 G	TINERR	012101	T#SEGL=	177777	T29OF7	030445	T30IMV	036566
SFPTBL	002160 G	TMPBFR	002624 G	T#SUBN=	000001	T29PAC	026250	T3OLOO	032310
SIFLAG	003146 G	TNAM	016764	T#TAGL=	177777	T29PBP	031321	T3OLOQ	037560
SIMSG	012046	TRANST	002160 G	T#TAGN=	010077	T29PK2	026360	T3ONEF	040526
SKIPT	003372	TSBA =	000000 G	T#TEMP=	000000	T29PK3	026370	T3OFL	040237
SOFINI	016054 G	TSBAH =	000001 G	T#TEST=	000011	T29RB	026372	T3OPAC	036430
SPACE	010544 G	TSDB =	000000 G	T#TSTM=	177777	T29RDF	026650	T3OPK2	036540
SPM1	105636	TSDBH =	000001 G	T#TSTS=	000001	T29RDG	031573	T3OPK3	036550
SPM4	105666	TSFCOD	007314	T##AU =	010031	T29RES	032076	T3OPTB	037172
SPM6	105716	TSREJ =	000006	T##AUT=	010033	T29RIB	031654	T3ORB	036552
SPM7	105746	TSSDEF	006664	T##CLE=	010034	T29RN	026406	T3ORDF	037343
SRO	= 177572	TSSR =	000002 G	T##DU =	010032	T29RNC	030304	T3ORDG	037421
SR1	= 177574	TSSRBI	003476 G	T##HAR=	010075	T29RRF	026717	T3ORES	041202
SR2	= 177576	TSSRFO	006473	T##HW =	010000	T29RRG	027033	T3ORIB	036675
SR3	= 172516	TSSRH =	000003 G	T##INI=	010030	T29RRN	031754	T3ORN	036566
SSR	= 000200	TSSX	004014	T##MSG=	010025	T29RSZ	026426	T3ORAM	040605
STATCO	012472	TSTBLK	002744 G	T##PRO=	010027	T29RT2	032170	T3ORAN	040663
SVCLBL =	000000	TSTCNT	002206 G	T##RPT=	010035	T29RT3	032232	T3ORRP	040742
SVCINS=	000000	TSTEND	017000	T##SOF=	010076	T29RWN	030235	T3ORT2	041274
SVCSUB=	000001	TSTFLA	002306 G	T##SRV=	010026	T29SC	027147	T3ORT3	041336
SVCTAG=	000000	TSTLOO	016536 G	T##SUB=	010074	T29SSR	027437	T3ORWN	040170
SVCTST=	000001	TSTPTR	002310 G	T##SW =	010001	T29SZ	026376	T3OSKM	037044
S#LSYM=	010000	TSTSET	016570 G	T##TES=	010073	T29S2	026402	T3OSSR	037641
SO.IDB=	000010	TST29I	032047	T1	023462 G	T29S3	026404	T3OSZ	036556
SO.IFB=	000002	TST30I	041161	T1.1	023512	T29TM	030157	T3OS2	036562
SO.IFP=	000001	TST31I	046473	T1.2	024142	T29TRL	031407	T3OS3	036564
SO.ILD=	000020	TST32I	052570	T1.3	024664	T29VCK	030721	T30TM	040036
SO.ION=	000040	TST33I	055575	T1.4	025410	T29WB	026372	T30THK	040444
SO.IRD=	000100	TST34I	063137	T2	032264 G	T29WDC	030627	T30TM2	040113
SO.IRW=	000004	TST35I	073063	T2.1	032310	T29WDD	030520	T30TPB	037263
SO.ISP=	000200	TST36I	100647	T2.2	033702	T29WDE	027512	T30VCK	040371
S1.ICE=	002000	TST37I	105273	T2.3	035326	T29WDF	027301	T30WB	036552
S1.IEO=	010000	TSV2	002000 G	T2.4	035722	T29WDR	026410	T30WDC	040312
S1.IFM=	001000	TSV3	002170 G	T23A	003136 G	T29WLK	027574	T30WDD	037120
S1.IHE=	000400	TSV4	021562 G	T23B	003140 G	T29WNG	026453	T30WDE	037712
S1.IID=	004000	TSV6	105474 G	T29AM3	030357	T29WRT	027661	T30WDF	037503
S1.I1R=	020000	TSV7B	023462 G	T29BA	030774	T29WSS	031066	T31AM3	044746
S1.I2R=	040000	TTIBFR=	177562 G	T29BF1	026272	T3	041362 G	T31BA	045306
S1.PAR=	100000	TTICSR=	177560 G	T29BF2	026400	T3BFLG	003142 G	T31BFR	043142
S2.ATI=	000010	TTIVFC=	000060 G	T29BOT	027726	T3.1	041412	T31BF2	043250
S2.BTI=	000004	T#ARGC=	000003	T29BS0	026400	T3.2	042270	T31BOT	044275
S2.DIM=	000200	T#CODE=	001130	T29BS1	026401	T30BFR	036452	T31BS0	043250
S2.ILW=	000100	T#ERRN=	001620	T29CNT	026424	T30BF2	036560	T31BS1	043251

T31CNT	043266	T32CNU	051442	T34BA	062776	T35CON	067462	T36BS1	075641
T31CNU	043270	T32DAT	051270	T34BFR	060512	T35DAT	067330	T36CNT	075656
T31CON	043262	T32DLY	051444	T34BF2	060626	T35DLY	067472	T36CNU	075660
T31DAT	043130	T32DSW	051300	T34BOT	061164	T35DSW	067340	T36CON	075652
T31DLY	043272	T32ECF	052405	T34BS0	060626	T35DTA	072255	T36DAT	075520
T31DSW	043140	T32EOT	051541	T34BS1	060627	T35EOT	070440	T36DLY	075662
T31DTA	046376	T32ERA	051746	T34CNT	060622	T35INT	072531	T36DSW	075530
T31EOT	044470	T32L00	046750	T34CON	060640	T35LON	071420	T36DTA	100552
T31LON	045450	T32OPI	052533	T34DAT	060500	T35L00	063374	T36EOT	076735
T31L00	041412	T32PAC	051260	T34DLY	060624	T35L0P	071502	T36LON	077715
T31L0P	045532	T32PK2	051370	T34DSW	060510	T35L0Q	070135	T36L00	073330
T31L0Q	044046	T32PK3	051400	T34EOT	062135	T35LOR	070010	T36L0P	077777
T31LOR	043721	T32RB	051402	T34ET	062046	T35MOT	072433	T36L0Q	076376
T31NEF	045770	T32RES	052630	T34ETC	061107	T35NEF	071740	T36LOR	076251
T31OFL	045015	T32RIB	052066	T34ETN	061401	T35NIN	073006	T36NAS	075664
T31PAC	043120	T32RT2	052722	T34ETO	060732	T35OFL	070765	T36NEF	100235
T31PBP	045614	T32RT3	052752	T34ETS	061460	T35OPM	072622	T36OFL	077262
T31PK2	043230	T32RMN	051630	T34ETZ	061552	T35PAC	067320	T36PAC	075510
T31PK3	043240	T32SCF	052164	T34ET2	061317	T35PBP	071564	T36PBP	100061
T31RB	043242	T32SZ	051406	T34L00	056022	T35PK2	067430	T36PK2	075620
T31RDE	043274	T32TSA	052241	T34OFL	062457	T35PK3	067440	T36PK3	075630
T31RDF	043473	T32WB	051402	T34PAC	060470	T35RB	067442	T36RB	075632
T31RES	046540	T32WDC	052466	T34PK2	060600	T35RDF	067562	T36RDF	076023
T31RN	043256	T33BFR	054522	T34PK3	060610	T35RES	073114	T36RES	100670
T31RNC	044673	T33BF2	054630	T34POS	060644	T35RN	067456	T36RN	075646
T31RRF	043542	T33BOT	055255	T34RB	060612	T35RNC	070643	T36RNC	077140
T31RT2	046632	T33BS0	054630	T34RES	063162	T35RRF	076631	T36RRF	076072
T31RT3	046674	T33BS1	054631	T34RNC	062336	T35RT2	073206	T36RT2	100762
T31RMN	044624	T33CNT	054646	T34RRE	061016	T35RT3	073250	T36RT3	101024
T31SC	043637	T33CNU	054650	T34RSZ	060620	T35RWE	072720	T36RMN	077071
T31SCF	046111	T33CON	054642	T34RT2	063254	T35RWN	070574	T36SC	076167
T31SSR	044127	T33DAT	054510	T34RT3	063316	T35SC	067726	T36SCF	100333
T31SZ	043246	T33DLY	054652	T34RWN	062267	T35SCF	072036	T36SSR	076457
T31S2	043252	T33DSW	054520	T34SSR	062013	T35SSR	072352	T36SZ	075636
T31S3	043254	T33DTA	055500	T34STM	061630	T35SZ	067446	T36S2	075642
T31TIM	044370	T33L00	053026	T34SZ	060616	T35S2	067452	T36S3	075644
T31TM	044547	T33PAC	054500	T34S2	060630	T35S3	067454	T36TIM	076660
T31TRL	045702	T33PK2	054610	T34S3	060632	T35TIM	070363	T36TM	077014
T31TSA	046166	T33PK3	054620	T34TH	062213	T35TM	070517	T36TRL	100147
T31VCK	045233	T33RB	054622	T34TMK	061713	T35TRL	071652	T36TSA	100410
T31WB	043242	T33RBP	054654	T34VCK	062723	T35TSA	072113	T36VCK	077500
T31WDC	045160	T33RES	055612	T34WB	060612	T35VCK	071203	T36WB	075632
T31WDD	045070	T33RN	054636	T34WD	060634	T35WB	067442	T36WDC	077425
T31WDE	044163	T33RT2	055704	T34WDC	062621	T35WDC	071130	T36WDD	077335
T31WDF	043771	T33RT3	055746	T34WDD	062532	T35WDD	071040	T36WDE	076513
T31WDR	043260	T33RWN	055350	T34WDR	060636	T35WDE	070216	T36WDF	076321
T31WNG	043421	T33SSR	055171	T34WSS	063050	T35WDF	070060	T36WDR	075650
T31WNH	043340	T33SZ	054626	T34WTH	061230	T35WDR	067460	T36WNG	075735
T31WRF	046273	T33S2	054632	T35AM3	070716	T35WNG	067474	T36WRF	100472
T31WSS	045361	T33S3	054634	T35BA	071256	T35WRF	072175	T36WSS	077626
T32AM3	051677	T33UNC	055012	T35BFR	067342	T35WSS	071331	T37AM3	103637
T32BA	052013	T33UND	055102	T35BF2	067450	T36AM3	077213	T37BA	104177
T32BFR	051302	T33WB	054622	T35BOT	070270	T36BA	077553	T37BFR	102212
T32BOE	052316	T33WDC	055417	T35BS0	067450	T36BFR	075532	T37BF2	102320
T32BOT	051446	T33WDR	054640	T35BS1	067451	T36BF2	075640	T37BOT	103211
T32CMD	051410	T33WPW	054732	T35CNT	067466	T36BOT	076565	T37BS0	102320
T32CNT	051440	T34AM3	062411	T35CNU	067470	T36BS0	075640	T37BS1	102321

T37CNT	102336	T37SSR	103066	T7.4	066374	WSMBK	021304	G	X\$OFFS=	000400	
T37CNU	102340	T37SZ	102316	T8	073274	XFERAS	016020		X\$TRUE=	000020	
T37CON	102332	T37S2	102322	T8.1	073330	XNXM	016456		X1.COR=	020000	
T37DAT	102200	T37S3	102324	T8.2	074404	XORBF0	007752		X1.DLT=	100000	
T37DLY	102342	T37TIM	103304	T9	101050	XORFOR	010070		X1.MBZ=	017375	
T37DSW	102210	T37TM	103440	T9.1	101104	XST0	= 000006	G	X1.RBP=	000400	
T37DTA	105176	T37TRL	104573	UAM	= 000200	XST1	= 000010	G	X1.SPA=	040000	
T37EOT	103361	T37TSA	105034	UNITN	002174	XST2	= 000012	G	X1.UNC=	000002	
T37LON	104341	T37VCK	104124	UNREC	= 000006	XST3	= 000014	G	X2.BUF=	000100	
T37LOO	101104	T37WB	102312	USI	004117	XST4	= 000016	G	X2.EXT=	000200	
T37LOP	104423	T37WDC	104051	WAITF	016330	XS0BOT=	000002		X2.OPM=	100000	
T37LOQ	103005	T37WDO	103761	WC.IFA=	000200	XS0EOT=	000001		X2.RCE=	040000	
T37LOR	102660	T37WDE	103122	WC.IFE=	000002	XS0IE =	000040		X2.REV=	000077	
T37NEF	104661	T37WDF	102730	WC.IGO=	000001	XS0ILA=	000400		X2.SPA=	035400	
T37OFL	103706	T37WDR	102330	WC.IRE=	000010	XS0ILC=	001000		X2.UNI=	000007	
T37PAC	102170	T37WNG	102344	WC.IRW=	000004	XSOLET=	020000		X2.WCF=	002000	
T37PBP	104505	T37WRF	105116	WC.IOT=	000100	XSONOT=	000200		X3.DCK=	000010	
T37PK2	102300	T37WSS	104252	WC.I1T=	000040	XSONEF=	002000		X3.MBZ=	000006	
T37PK3	102310	T4	046720	G	WC.I5R=	000020	XS0ONL=	000100	X3.MDE=	177400	
T37RB	102312	T4.1	046750		WF.IED=	000010	XS0PED=	000010	X3.OPI=	000100	
T37RDF	102432	T4.2	047610		WF.IER=	000004	XS0RLL=	010000	X3.REV=	000040	
T37RES	105314	T4.3	050420		WF.IHI=	000200	XS0RLS=	040000	X3.RIB=	000001	
T37RN	102326	T5	052776	G	WF.IRE=	000040	XS0TMK=	100000	X3.SPA=	000200	
T37RNC	103564	T5.1	053026		WF.IWF=	000020	XS0VCK=	000020	X3.TRF=	000020	
T37RRF	102501	T6	055772	G	WF.IWR=	000100	XS0WLE=	004000	X4.HSP=	100000	
T37RT2	105406	T6.1	056022		WF.I3R=	000002	XS0WLK=	000004	X4.MBZ=	017400	
T37RT3	105450	T7	063344	G	WF.I4R=	000001	XXCOMM	003114	G	X4.RCE=	040000
T37RMN	103515	T7.1	063374		WRTCHR	010742	X\$ALWA=	000000		X4.TSM=	020000
T37SC	102576	T7.2	064452		WRTERR	005107	X\$FALS=	000040		X4.WRC=	000377
T37SCF	104757	T7.3	065532		WRTMSG	005052					

. ABS. 106404 000
000000 001
ERRORS DETECTED 0

VIRTUAL MEMORY USED: 30328 WORDS (119 PAGES)
DYNAMIC MEMORY: 20614 WORDS (79 PAGES)
ELAPSED TIME: 00:38:07
CVTSDC,CVTSDC/-SP=SV C/ML,TSV1D,TSV22D,TSV3B,TSV4,TSV7B,TSV6