



The main body of the document is a dense grid of approximately 15 columns and 25 rows of technical data. Each cell in the grid contains a small, structured table or diagram. The data is organized into several distinct sections, likely representing different functional components of the DMR-11 system. The text within these cells is small and difficult to read, but it appears to include various parameters, values, and possibly flow diagrams. The overall layout is highly systematic and repetitive, characteristic of a technical manual or a data sheet for a complex electronic device.

DMR-11, DMP11 DMR-11 FCTNL DIAG
CZDMIBO

AH-F832B-MC
FICHE 2 OF 2

JUN 1980
COPYRIGHT © 1980
MADE IN USA



[Faded, illegible text columns on the left side of the page]



5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40

.NLIST TOC

.REM @

IDENTIFICATION

PRODUCT CODE: AC-F830B-MC
PRODUCT NAME: CZDMIBO DMR-11 FCTNL DIAG
PRODUCT DATE: APRIL 1980
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: MIKE O'CONNOR

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) 1980 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

CONTENTS

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

- 1.0 INTRODUCTION
- 2.0 HARDWARE REQUIREMENTS
- 3.0 PRELIMINARY PROGRAM REQUIREMENTS
- 4.0 GENERAL PROGRAM CONSIDERATIONS
 - 4.1 DIAGNOSTIC SUPERVISOR
 - 4.2 EXECUTION TIME
 - 4.3 XXDP+
 - 4.4 ACT/SLIDE
 - 4.5 APT
 - 4.6 MEMORY MANAGEMENT
 - 4.7 MEMORY PARITY OPTION
 - 4.8 ERROR LOGGING
- 5.0 PROGRAM LOAD MEDIA
- 6.0 OPERATING INSTRUCTIONS
 - 6.1 LOADING AND STARTING PROCEDURES
 - 6.1.1 LOADING PROCEDURES
 - 6.1.2 STARTING PROCEDURES
 - 6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION
 - 6.2 INITIAL DIALOGUE
 - 6.3 PROGRAM OPTIONS
 - 6.3.1 START COMMAND
 - 6.3.1.1 TESTS SWITCH
 - 6.3.1.2 PASS SWITCH
 - 6.3.1.3 FLAGS SWITCH
 - 6.3.1.4 END OF PASS SWITCH
 - 6.3.1.5 EFFECT OF START COMMAND
 - 6.3.2 RESTART COMMAND
 - 6.3.2.1 TESTS, PASS, AND FLAG SWITCHES
 - 6.3.2.2 UNITS SWITCH
 - 6.3.2.3 EFFECT OF RESTART COMMAND
 - 6.3.3 CONTINUE COMMAND
 - 6.3.3.1 PASS SWITCH
 - 6.3.3.2 FLAGS SWITCH
 - 6.3.3.3 EFFECT OF CONTINUE COMMAND
 - 6.3.4 PROCEED COMMAND
 - 6.3.4.1 FLAGS SWITCH
 - 6.3.4.2 EFFECT OF PROCEED COMMAND
 - 6.3.5 ADD COMMAND
 - 6.3.5.1 UNITS SWITCH
 - 6.3.5.2 EFFECT OF ADD COMMAND
 - 6.3.6 DROP COMMAND
 - 6.3.6.1 UNITS SWITCH
 - 6.3.6.2 EFFECT OF DROP COMMAND
 - 6.3.7 PRINT COMMAND
 - 6.3.7.1 EFFECT OF PRINT COMMAND

58	6.3.8 DISPLAY COMMAND
59	6.3.8.1 UNITS SWITCH
60	6.3.8.2 EFFECT OF DISPLAY COMMAND
61	6.3.9 FLAGS COMMAND
62	6.3.9.1 EFFECT OF FLAGS COMMAND
63	6.3.10 ZFLAGS COMMAND
64	6.3.10.1 EFFECT OF ZFLAGS COMMAND
65	6.3.11 CONTROL CHARACTERS
66	6.3.12 HARDWARE PARAMETERS
67	6.3.13 SOFTWARE PARAMETERS
68	6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE
69	
70	7.0 DEVICE INFORMATION TABLES
71	
72	8.0 TEST DESCRIPTIONS
73	
74	9.0 ERROR INFORMATION
75	9.1 ERROR REPORTING

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1.0 INTRODUCTION

THIS PROGRAM WILL BE IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR AND A STRUCTURED PROGRAMMING APPROACH. BECAUSE THE DESIGN WILL CONFORM TO THE SUPERVISOR (STANDALONE VERSION) THE PROGRAM WILL BE COMPATIBLE WITH ACT, APT, XXDP+, AND SLIDE.

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM WILL ALLOW MODIFICATION OF DEVICE PARAMETERS, SUCH AS UNIBUS ADDRESS, VECTOR ADDRESSES AND TEST CONFIGURATION. IN ADDITION, THE OPERATOR CAN SPECIFY PARTICULAR TESTS TO BE RUN AND A VARIETY OF LOOPING, RUNNING, AND REPORTING MODES.

DEVICE ERRORS WILL BE REPORTED AS THEY OCCUR. THE REPORT WILL INCLUDE A TEST NUMBER AND DESCRIPTION OF THE ERROR, GOOD AND BAD TEST DATA, AND APPLICABLE DEVICE REGISTER CONTENTS.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DMR-11 FUNCTIONAL DIAGNOSTIC TESTS:

PDP-11/04,05,10,20,30,34,35,40,45,50,60, OR 70
16K MEMORY
CONSOLE TERMINAL
DMR-11

3.0 PRELIMINARY PROGRAM REQUIREMENTS

IT IS ADVISED THAT THE STATIC DIAGNOSTICS BE RUN BEFORE THESE FUNCTIONAL DIAGNOSTICS. IT IS ASSUMED THAT THE PROCESSOR IS IN PROPER WORKING CONDITION.

ENSURE THAT THE SWITCH 1 AT LOCATION E-85 ON THE M8207 IS ON. IF THIS SWITCH IS OFF, THE MAINTENANCE BITS IN BSEL1 CAN'T BE USED AND CERTAIN TESTS WILL BE NOT BE CORRECTLY RUN.

WHEN CHOSING A CABLE TEST CONNECTION, ENSURE THAT THE SWITCH PACK E-39 ON THE M8203 IS PROPERLY SET UP FOR THE DESIRED INTERFACE. IF CHOSING TEST CONFIGURATION OPTIONS 1-4, IT IS NOT NECESSARY TO SELECT THE INTERFACE; HOWEVER THE BAUD RATE MUST BE CORRECT. FOR EXAMPLE IF IT IS DESIRED TO RUN CONFIGURATION 3 (H3255-EIA), IT IS NOT NECESSARY TO HAVE SWITCH 7 OF THE SWITCH PACK IN THE OFF POSITION. IT IS, HOWEVER, NECESSARY TO HAVE THE BAUD RATE SELETCTED TO BE

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114

WITHIN THE EIA RANGE.

NOTE THAT A MANUFACTURING-ONLY PATCH IS REQUIRED TO RUN WHEN USING THE SPECIAL MANUFACTURING TEST CONNECTORS. THIS PATCH WILL CHANGE THE FLAG WORD 'MANUF' TO A NON-ZERO VALUE. WHEN THE FLAG IS NON-ZERO, THE MAINTENANCE BIT IS SET BY A MODEM WRITE COMMAND IF THE V.35 OR EIA ONBOARD CONNECTORS ARE USED.

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

EXECUTION TIME IS DEPENDENT ON THE PROCESSOR SPEED AND THE DMR BAUD RATE. EXAMPLES OF EXECUTION TIME

11/70 WITH CACHE AND DMR AT 2.4K	4 AND 1/2 MINUTES
11/70 WITHOUT CACHE AND DMR AT 2.4K	5 AND 1/2 MINUTES
11/34 AND DMR AT 2.4K	10 MINUTES

4.3 XXDP+

THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

4.6 MEMORY MANAGEMENT

IF MEMORY MANAGEMENT IS AVAILABLE, IT IS USED BY CERTAIN TESTS IN THIS FUNCTIONAL DIAGNOSTIC.

4.7 MEMORY PARITY OPTION

115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171

IF PARITY MEMORY IS INSTALLED, MEMORY PARITY TRAPS ARE DISABLED BY THE PROGRAM.

4.8 ERROR LOGGING

AT THE END OF EACH PASS ON ALL UNITS, THE PROGRAM PRINTS OUT THE CUMULATIVE TOTAL NUMBER OF ERRORS SINCE THE LAST START OR RESTART COMMAND.

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP+ LOAD MEDIA. WHEN LOADED UNDER XXDP+, THE DIAGNOSTIC SUPERVISOR WILL BE LOADED AUTOMATICALLY.

6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP+, WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

- A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
- B) RECEIVE DIAGNOSTIC SUPERVISOR IDENTIFICATION AND PROMPT (DRS-C>)
- C) ENTER STA<CR>
- D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
- E) GET END OF PASS MESSAGES OR ERROR MESSAGES
- F) TO END EXECUTION, ENTER CONTROL/C

6.2 INITIAL DIALOGUE

172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED, THE FOLLOWING IDENTIFICATION IS TYPED :

DRS LOADED
DIAG. RUN-TIME SERVICES

DR>

THE OPERATOR THEN PROCEEDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3. (FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
 <FLAG-LIST>/EOP:<INCR>

6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.2 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285

- HOE HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
- LOE LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
- IER INHIBIT ERROR REPORTING
- IBE INHIBIT BASIC ERROR REPORTS
- IXE INHIBIT EXTENDED ERROR REPORTS
- PRI DIRECT ALL MESSAGES TO A LINE PRINTER
- PNT PRINT NUMBER OF TEST BEING EXECUTED
- BOE BELL ON ERROR
- UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
- ISR INHIBIT STATISTICAL REPORTS
- IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
- LOT LOOP ON TEST

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.5 EFFECT OF START COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC TESTS THEMSELVES.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION '# UNITS?' TO WHICH THE OPERATOR REPLIES WITH A DECIMAL NUMBER N FROM 1 TO 16. THE TERM 'UNIT' REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE AFTER THE PARENTHESES.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK

286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342

VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

WHEN THE QUESTION '# UNITS?' IS ANSWERED, MEMORY STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE 'TOO MANY UNITS' IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

6.3.2 RESTART COMMAND

```
*****  
RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:  
  <FLAG-LIST>/UNITS:<UNIT-LIST>  
*****
```

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIALOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED

343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399

(OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

```
*****  
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>  
*****
```

6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.3.3 EFFECT OF CONTINUE COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

6.3.4 PROCEED COMMAND

```
*****  
PRO(CEED)/FLAGS:<FLAG-LIST>  
*****
```

6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.4.2 EFFECT OF PROCEED COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456

6.3.5 ADD COMMAND

ADD/UNITS:<UNIT-LIST>

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

6.3.6 DROP COMMAND

DRO(P)/UNITS:<UNIT-LIST>

6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.6.2 EFFECT OF DROP COMMAND

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

6.3.7 PRINT COMMAND

PRI(NT)

6.3.7.1 EFFECT OF PRINT COMMAND

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE !SR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513

6.3.8 DISPLAY COMMAND

DIS(PLAY)/UNITS:<UNIT-LIST>

6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR 'DROP' COMMAND ARE SO DESIGNATED.

6.3.9 FLAGS COMMAND

FLA(GS)

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

ZFL(AGS)

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR DIALOGUES- HARD CORE QUESTIONS (SEE 6.2), HARDWARE DIALOGUE (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC

514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570

CAUSES ALL TELETYPE OUTPUT TO BE SUPPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER 0 IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 3 QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

1. CSR ADDRESS: (0) 160070?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SEL0) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000-177776 (OCTAL), AND THE DEFAULT VALUE IS 160070.

2. VECTOR ADDRESS: (0) 300 ?

THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE DEFAULT VALUE IS 300.

3. TEST CONFIGURATION -

0 = INTERNAL (NO CONNECTOR)

1 = H3254 - V.35 (NOTE: MODE 1-4 ALLOWS

2 = H3254 - INTEGRAL PROGRAM INTERFACE SELECTION)

3 = H3255 - RS232C/423

4 = H3255 - RS422

5 = CABLE AND SW PACK INTERFACE SELECTED

(V.35-H3250, INTEGRAL-BC55A-10, RS232C-H325, RS423/422-H3251)

* SELECT THE FOLLOWING ONLY IF THE MODEM SUPPORTS LOOPBACK *

6 = LOCAL LOOP

7 = REMOTE LOOP

(0) 5 ?

THIS QUESTION WILL COVER ALL THE POSSIBLE TEST CONFIGURATIONS. THE DEFAULT IS FOR ACTUAL CABLE LOOPBACK (5). CONFIGURATION 0 WILL ENABLE LINE UNIT (TTL) LOOPBACK. IF THIS IS SELECTED NO CABLES OR CONNECTORS SHOULD BE CONNECTED. CONFIGURATIONS 1-4 WILL SELECT THE INTERFACE REGARDLESS OF THE SWITCH SETTING AS LONG AS THE PROPER BAUD RATE IS SELECTED (I.E. EIA - 2.4K-19.2K).

6.3.13 SOFTWARE PARAMETERS

THE ONLY SOFTWARE PARAMETER QUESTION ASKED BY THE DIAGNOSTIC CONCERNS A SOFTWARE TIMEOUT VARIABLE THAT IS USED TO PREVENT SOFTWARE 'HUNG' CONDITIONS. THIS VARIABLE IS A VALUE FROM 1-5.

SELECTABLE PROGRAM LOOP TIME-OUT VARIABLE
[REFER TO LISTING 6.3.13] (MAX=5; MIN=1) (0) 5 ?

THERE ARE TWO FACTORS THAT SHOULD BE CONSIDERED WHEN ANSWERING THIS QUESTION. THE FIRST IS PROCESSOR SPEED; THE FASTER THE PROCESSOR THE HIGHER THE VARIABLE SHOULD BE. THE SECOND IS

571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627

BAUD RATE; THE SLOWER THE DMR BAUD RATE THE HIGHER THE VARIABLE SHOULD BE. FOR EXAMPLE:

11/70 WITH CACHE AND DMR AT 1 MEG.: 4
11/34 AND DMR AT 56K: 2
11/40 AND DMR AT 2.4K: 3

THE DEFAULT IS 5. THIS WILL COVER THE WORST CASE (I.E. 11/70 WITH CACHE AND THE DMR AT 2.4K).

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION '# UNITS?' IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER

628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644

(0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE
VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE
THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR
THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

UNITS (D) ? 16
UNIT 0
<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

UNIT 7
<QUESTION 1> ?
<QUESTION 2> ? 7-11,,13-15
<QUESTION 3> ? 77

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,...,6 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 7 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM 'UNIT XX' AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS AN 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

7.0 DEVICE INFORMATION TABLES

SEE THE GLOBAL EQUATES SECTION FOR DEVICE CSR BIT DEFINITIONS

8.0 TEST DESCRIPTIONS

```
*****  
* TEST 1 - DMR-11  
* VERIFY THAT ADDRESSING THE 4 UNIBUS CSRS DOES NOT CAUSE A NON-  
* EXISTENT MEMORY TRAP.  
*  
* THE DMR IS AN NPR DEVICE RESIDING ON A UNIBUS. COMMUNICATION  
* BETWEEN THE MAIN CPU AND THE DMR IS ACCOMPLISHED THROUGH A  
* SET OF FOUR 16-BIT UNIBUS CONTROL AND STATUS REGISTERS (CSRS).  
* THE FOUR REGISTERS ARE ASSIGNED ADDRESSES IN THE I/O PAGE  
* FLOATING ADDRESS SPACE: 76XXX0 - 76XXX6  
*  
* NOTE: THIS TEST IS REDUNDANT IN THAT STATIC LOGIC TESTS SHOULD  
* HAVE BEEN RUN BEFORE THESE FREE-RUNNING TESTS WERE STARTED, AND  
* THEY SHOULD HAVE DETECTED ANY CSR ADDRESSING PROBLEMS.  
* BUT JUST IN CASE THOSE STATIC TESTS AREN'T RUN, WE'LL BE SAFE.  
*****
```

```
*****  
* TEST 2 - DMR-11  
* ROM CRC/CCITT - CHECK ROM POSITION AND CALCULATE CRC/CCITT. THE  
* LAST 4 BYTES CONTAIN INFORMATION ABOUT THE ROM TO CHECK. THE 1ST  
* OF THESE BYTES CONTAINS THE ASCII VERSION NUMBER. THE 2ND BYTE  
* CONTAINS THE ROM NUMBER. THE 3RD AND 4TH BYTES CONTAIN A NEGATIVE  
* CRC/CCITT WORD FOR THE ROM.  
*  
* CHIP ADDRESS RANGE  
* LOCATION CHIP NO. BYTE ADDRESS RANGE  
* E03 0 LOW 0000 - 1777  
* E02 1 HIGH 0000 - 1777  
* E04 2 LOW 2000 - 3777  
* E01 3 HIGH 2000 - 3777  
* E05 4 LOW 4000 - 5777  
* E14 5 HIGH 4000 - 5777  
*****
```

```
***** IMPORTANT !!!!!!!!!!!!! *****  
* FOR THIS TEST TO RUN CORRECTLY, ENSURE THAT SWITCH 1 AT LOCATION  
* E85 ON THE M8207 IS ON. IF THIS SWITCH IS OFF, BSEL1 WILL BE  
* LOCKED OUT AND THE MAINTENANCE FEATURES WILL NOT BE ENABLED.  
*****
```

```
* SUBTEST 1 - ON THE FIRST PASS PRINT THE VERSION # IN EACH ROM  
* SUBTEST 2 - GENERATE THE CRC-CCITT IN EACH ROM AND COMPARE IT  
* IT AGAINST THE CRC BLASTED IN THE ROM  
* SUBTEST 3 - COMPARE THE ROM # BLASTED IN THE ROM AGAINST THE  
* EXPECTED ROM #.  
*****
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

```

*****
*                                     TEST 3 - DMR-11
* MASTER CLEAR
* THIS TEST WILL ISSUE 2 MASTER CLEARS.  EACH CALL TO THE MASTER
* CLEAR ROUTINE WILL ENSURE THAT THE RUN BIT WILL BE SET.  ALSO
* THE MASTER CLEAR WILL CAUSE THE DIAGNOSTIC MICROTESTS TO BE
* RUN WHEN THE MICRODIAGNOSTIC BIT (BIT 13 IN SEL0) IS CORRECTLY
* SET OR CLEARED.  BECAUSE THE RUNNING OF MICROTESTS DEPENDS ON THE
* EXCLUSIVE OR OF THE HARDWARE SWITCH 10 ON E134 OF THE M8203 AND
* THE MICRODIAGNOSTIC BIT, WE CAN'T KNOW WHETHER THE SETTING OR
* CLEARING OF BIT 13 WILL RESULT IN THE RUNNING OF MICROTESTS.
* THEREFORE THE MASTER CLEAR SUBROUTINE WILL TOGGLE (I.E. SET
* BIT 13 ONLY ON EVERY OTHER MASTER CLEAR) THE SOFTWARE BIT.
* THIS WILL ENSURE THAT REGARDLESS OF THE POSITION OF THE
* HARDWARE SWITCH, MICROTESTS WILL BE RUN EVERY OTHER MASTER CLEAR.
* WHEN RUNNING THIS TEST, WE EXPECT TO ADD THE RESULTS OF BSEL3
* AFTER EACH MASTER CLEAR.
* BSEL3 = 100      - MICROTESTS DISABLED
* BSEL3 = 200      - MICROTESTS RUN SUCCESSFULLY
* IF THE RESULT OF THE 2 MASTER CLEARS IS NOT 300, AN ERROR IS
* REPORTED.
*
* ADDITIONALLY THIS ROUTINE WILL REPORT WHENEVER THE RESULT OF
* BSEL3 IS 0.  THIS WILL MEAN THAT THE DEVICE IS NOT A DMR
* (I.E. DMC)
*****

```

```

*****
*                                     TEST 4 - DMR-11
* BASE IN COMMANDS
*
* SUBTEST 1 - ISSUE A BASE IN - DMR MODE.
* ENSURE THAT THE DMR MODE BIT (BIT 4) IS SET IN
* THE MICROCODE SCRATCH PAD 7 AND THAT THE DDCMP
* MESSAGE VARIABLES ARE PROPERLY INITIALIZED.
*
* SUBTEST 2 - ISSUE A BASE IN - DMC MODE.
* ENSURE THAT THE DMC MODE BIT (BIT 4) IS CLEAR IN
* THE MICROCODE SCRATCH PAD 7 AND THAT THE DDCMP
* MESSAGE VARIABLES ARE PROPERLY INITIALIZED.
*****

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```
*****  
* TEST 5 - DMR-11  
* DMR COMMANDS  
* SUBTEST 1 - ISSUE AN ENABLE EXTENDED ERROR COMMAND AND CHECK THAT  
* THE EXT. ENABLE BIT IS SET IN SCRATCH PAD 13. THEN  
* DISABLE EXTENDED ERROR AND CHECK THAT THE ENABLE BIT  
* IS CLEAR.  
* SUBTEST 2 - SET REP/SEL TIMER VALUE AND SET THE DMR THRESHOLD  
* VALUES. CHECK THAT THE VALUES ARE CORRECT IN  
* THE BASE TABLE AFTER HALTING THE DMR.  
*****
```

```
*****  
* TEST 6 - DMR-11  
* CONTROL IN COMMAND TEST -  
* SUBTEST 1 - CONTROL IN, FULL DUPLEX, DDUMP MODE. ENSURE THAT  
* THE HALF-DUPLEX BIT IS CLEAR IN THE MODEM STATUS WORD,  
* ALSO ENSURE THAT DDUMP MODE BIT IS SET IN SCRATCH PAD 7.  
* SUBTEST 2 - CONTROL IN, HALF DUPLEX. ENSURE THAT THE HALF DUPLEX  
* BIT IS SET.  
* SUBTEST 3 - CONTROL IN, MAINTENANCE MODE. ENSURE THAT MAINT. MODE  
* BIT IS SET IN SCRATCH PAD 7.  
* SUBTEST 4 - CONTROL IN USING SELECTED LOOPBACK. ISSUE A CONTROL IN  
* USING THE USER SELECTED LOOPBACK. IF THE LOOPBACK IS  
* NOT CORRECT, DMR RUN MODE ACKNOWLEDGE WILL NOT BE  
* RECEIVED.  
*****
```

```
*****  
* TEST 7 - DMR-11  
* MODEM WRITE COMMAND  
* SUBTEST 1 - WRITE DATA PATTERNS INTO THE MODEM WRITE REGISTER.  
* ENSURE THAT ON THE NEXT MODEM READ THAT THE  
* MICROCODE RETURNS THE PATTERN WRITTEN INTO BSEL6.  
* SUBTEST 2 - ATTEMPT TO WRITE BOTH THE HALF-DUPLEX BIT AND THE  
* RTS HOLD BIT. THE MICROCODE SHOULD NOT ALLOW THIS  
* TO HAPPEN. WHEN READING THE MODEM STATUS, ONLY  
* THE HALF-DUPLEX SHOULD BE SET.  
*****
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```
*****  
* TEST 8 - DMR-11  
* SUBTEST 1 - TRANSMIT A BUFFER THREE TIMES WIHOUT ASSIGNING A  
* RECEIVE BUFFER. BY ASSIGNING A NO BUFFER THRESHOLD  
* OF THREE, ENSURE THAT A NO BUFFER ERROR IS RECEIVED  
* AFTER THE THIRD THRANSMISSION.  
* SUBTEST 2 - TRANSMIT A BUFFER WITHOUT A RECEIVE BUFFER.  
* ASSIGN THE NAKS THRESHOLD OF 3 AND A NO BUFFER  
* THRESHOLD OF 7. CHECK THAT THE NAKS ERROR COUNT IS  
* THREE AFTER SHUTDOWN.  
*****
```

```
*****  
* TEST 9 - DMR-11  
* NON-EXISTENT MEMORY (NXM) ERROR CHECK  
* PERFORM DMR COMMANDS USING NXM ADDRESSES; VERIFY THAT NXM ERROR IS  
* REPORTED IN EACH OF THE FOLLOWING SUBTESTS:  
* SUBTEST 1 - BASE IN RESUME COMMAND - BASE TABLE ADDRESS IS NXM  
* SUBTEST 2 - BA/CC IN RECEIVE COMMAND - BA/CC IN ADDRESS IS NXM  
* SUBTEST 3 - BA/CC IN TRANSMIT COMMAND - BA/CC IN ADDRESS IS NXM  
*****
```

```
*****  
* TEST 10 - DMR-11  
* TIME OUT - FORCE A TIMEOUT AND VERIFY THAT THE ERROR IS REPORTED  
*****
```

```
*****  
* TEST 11 - DMR-11  
* MESSAGE TOO LONG - TRANSMIT A MESSAGE THAT IS TOO LONG FOR THE  
* RECEIVE BUFFER AND VERIFY THAT THE 'TOO LONG' ERROR IS RECEIVED.  
*****
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

```
*****  
* TEST 12 - DMR-11  
* PROCEDURE ERRORS -  
* THE FOLLOWING SHOULD CAUSE THE DMR-11 TO HALT AND RESPOND WITH  
* A PROCEDURE ERROR:  
* SUBTEST 1 - A SECOND BASE IN COMMAND  
* SUBTEST 2 - A CONTROL IN BEFORE A BASE IN  
* SUBTEST 3 - A BA/CC IN BEFORE A BASE IN  
* SUBTEST 4 - A BA/CC IN RCV WITH A BUFFER LENGTH OF 0  
* SUBTEST 5 - A BA/CC IN XMIT. WITH A BUFFER LENGTH OF 0  
*  
*****
```

```
*****  
* TEST 13 - DMR-11  
* FREE RUNNING FLAG MODE DATA TEST  
* TRANSMIT A MESSAGE AND VERIFY THE RECEIVED DATA IS CORRECT.  
* IN THIS TEST NO INTERRUPTS ARE USED AND THE LINE UNIT IS IN  
* INTERNAL (TTL) LOOPBACK. THIS TEST IS THE FIRST TEST IN WHICH  
* THE DMR IS USED IN A DATA TRANSMISSION MODE.  
*****
```

```
*****  
* TEST 14 - DMR-11  
* IN THIS TEST - SEE IF WE HAVE MEMORY MANAGEMENT, IF SO SEE IF WE  
* HAVE THE MEMORY TO CHECK BITS 16 & 17 IN SEL6. THIS WILL ALLOW  
* US TO TRANSFER DATA USING THOSE EXTENDED ADDRESSING BITS. AS IN  
* TEST 13 THE TEST IS NON-INTERRUPT AND INTERNAL (TTL) LOOPBACK IS  
* USED.  
*  
*****
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

*****
* TEST 15 - DMR-11
* RESUME BASE IN - DMC MODE
* ** WILL NOT RUN IF MODEM LOOPBACK IS SELECTED **
* IN THIS TEST THE DMR WILL TRANSMIT AND RECEIVE 7 BUFFERS. DURING THE
* TEST THE DMR WILL BE HALTED AND RESTARTED BY A BASE-IN RESUME IN THE
* FOLLOWING MANNER:
*   BASE IN
*   CONTROL IN
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   1 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*   2 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*   2 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*   1 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*
* ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
* THE RECEIVE/TRANSMIT TABLE.
*
* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
* SEVEN RECEIVE AND SEVEN TRANSMIT BUFFERS. THE ROUTINE WILL
* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
* HIERARCHY:
*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
*      THAN 2K BYTES, USE THAT MEMORY
*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
*****

```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

*****
* TEST 16 - DMR-11
* RESUME BASE IN - DMR MODE
* IN THIS TEST THE DMR WILL TRANSMIT AND RECEIVE 7 BUFFERS. DURING THE
* TEST THE DMR WILL BE HALTED AND RESTARTED BY A BASE-IN RESUME IN THE
* FOLLOWING MANNER:
*   BASE IN
*   CONTROL IN
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   1 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*   2 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*   2 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*   1 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*
* ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
* THE RECEIVE/TRANSMIT TABLE.
*
* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
* SEVEN RECEIVE AND SEVEN TRANSMIT BUFFERS. THE ROUTINE WILL
* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
* HIERARCHY:
*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
*      THAN 2K BYTES, USE THAT MEMORY
*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
*****

```

PROGRAM DOCUMENT

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

*****
* TEST 17 - DMR-11
* INTERRUPT DRIVEN EXERCISE
* IN THIS TEST 64 BUFFERS WILL BE TRANSMITTED AND RECEIVED
*
* ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
* THE RECEIVE/TRANSMIT TABLE.
*
* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
* 64 RECEIVE AND 64 TRANSMIT BUFFERS. THE ROUTINE WILL
* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
* HIERARCHY:
*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
*      THAN 2K BYTES, USE THAT MEMORY
*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
*****

```

```

*****
* TEST 18 - DMR-11
* LARGE MESSAGE
* IN THIS MODE TRANSMIT AND RECEIVE 1 LARGE BUFFER
*
* THE BA/CC OUT RECEIVE AND TRANSMIT WILL BE ACCOUNTED FOR AND
* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
* THE RECEIVE/TRANSMIT TABLE.
*
* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
* ONE RECEIVE AND ONE TRANSMIT BUFFER. THE ROUTINE WILL
* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
* HIERARCHY:
*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
*      THAN 2K BYTES, USE THAT MEMORY
*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
*****

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

*****
* TEST 19 - DMR-11
* MAINTENANCE MODE OPERATION
*
* THE BA/CC OUT RECEIVE AND TRANSMIT WILL BE ACCOUNTED FOR AND
* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
* THE RECEIVE/TRANSMIT TABLE.
*
* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
* ONE RECEIVE AND ONE TRANSMIT BUFFER. THE ROUTINE WILL
* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
* HIERARCHY:
*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
*      THAN 2K BYTES, USE THAT MEMORY
*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
*****

```

9.0 ERROR INFORMATION

9.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH DESCRIBES AN "IRDY NOT SET" ERROR, AND PROVIDES THE PC OF THE ERROR CALL AND THE PC OF THE CALL TO THE SUBROUTINE REPORTING IT, THE FAILING REGISTER NAME, AND DEVICE REGISTER CONTENTS :

```

CZDMR DVC FTL ERR 0002 ON UNIT 00 TST 006 SUB 000 PC: 016210
TIME OUT
ERROR IN SUBROUTINE CALLED AT PC: 036174
STATUS OF BUFFERS
NUMBER OF BUFFERS: 7
BUFFER SIZE: 2048
IN - RCV ASSIGNED: 7      XMIT ASSIGNED: 7
OUT - RCV RETURNED: 0    XMIT RETURNED: 0
DMR RUN ACKNOWLEDGMENT NOT RECEIVED
(CHECK INTERFACE, BAUD AND TURNAROUND)

```

ALL THE MESSAGES IN THE DIAGNOSTIC USE BASIC MESSAGE CALLS. THEREFORE THE INHIBIT EXTENDED ERROR FLAG WILL HAVE NO EFFECT ON THE MESSAGE OUTPUT. THE INHIBIT BASIC MESSAGES WILL INHIBIT THE ERROR MESSAGES.

58
59
60
61
62
63
64
65

@

```
1          .TITLE CZDMIBO DMR-11 FUNCTIONAL TESTS
10         002000          .=2000
11
12
13
14
15         .MCALL  SVC
16 002000          SVC          ; INITIALIZE SUPERVISOR MACROS
17
18
19 002000          BGNMOD
20
21
22         000001          $LSTIN= 1      ; LIST INSTRUCTIONS
23         000001          $LSTTAG= 1
24         000001          SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
25         000001          SVCTST= 1      ; LIST TEST TAGS, SHIFTED RIGHT
26         000001          SVCSUB= 1      ; LIST SUBTEST TAGS, SHIFTED RIGHT
27         000001          SVCGBL= 1      ; LIST GLOBAL TAGS, SHIFTED RIGHT
28         000001          SVCTAG= 1      ; LIST OTHER TAGS, SHIFTED RIGHT
29
30         : CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
31         : TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
32         : SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
33         : CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
34
35 002000          POINTER BGNSW,BGNDU,BGNSFT
36
44
45
46
```


PROGRAM HEADER

002052
 002052 000000
 002054 000000
 002056
 002056 000000
 002060
 002060 010234
 002062
 002062 000000
 002064
 002064 000000
 002066
 002066 000000
 002070
 002070 000000
 002072
 002072 023724
 002074
 002074 000000
 002076
 002076 010242
 002100
 002100 104035
 002102
 002102 000000
 002104
 002104 020552
 002106
 002106 022156
 002110
 002110 022066
 002112
 002112 020544
 002114
 002114 000000
 002116
 002116 000000
 002120
 002120 000000

12
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28

.EVEN

L\$EF::
 .WORD 0
 .WORD 0
 L\$SPC::
 .WORD 0
 L\$DEVP::
 .WORD L\$DVTYP
 L\$REPP::
 .WORD 0
 L\$EXP4::
 .WORD 0
 L\$EXP5::
 .WORD 0
 L\$AUT::
 .WORD 0
 L\$DUT::
 .WORD L\$DU
 L\$LUN::
 .WORD 0
 L\$DESP::
 .WORD L\$DESC
 L\$LOAD::
 EMT E\$LOAD
 L\$ETP::
 .WORD 0
 L\$ICP::
 .WORD L\$INIT
 L\$CCP::
 .WORD L\$CLEAN
 L\$ACP::
 .WORD L\$AUTO
 L\$PRT::
 .WORD L\$PROT
 L\$TEST::
 .WORD 0
 L\$DLY::
 .WORD 0
 L\$HIME::
 .WORD 0

1
2
3
4
5
6
7
8
9
16
17
18
19
20

.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 19

002122
002122 000023
002124
002124 024004
002126 024322
002130 025636
002132 025770
002134 027046
002136 030260
002140 031040
002142 031462
002144 032164
002146 032676
002150 033100
002152 033260
002154 034164
002156 034714
002160 036460
002162 036564
002164 036632
002166 036702
002170 036752

.WORD 19
LSDISPATCH: :
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
.WORD T10
.WORD T11
.WORD T12
.WORD T13
.WORD T14
.WORD T15
.WORD T16
.WORD T17
.WORD T18
.WORD T19

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

.SBTTL DEFAULT SOFTWARE P-TABLE

:/ THE SOFTWARE P-TABLE CONTAINS THE VALUE OF THE PROGRAM
:/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.

002222 000001 BGNSW SFPTBL
002224 L\$SW: .WORD L10001-L\$SW/2
002224 SFPTBL:
002224
002224
002224
002224 C00005 SPEED: .WORD 5 ;PROCESSOR SPEED VARIABLE USED
002226 ENDSW ;TO ALTER THE WAIT VARIABLES.
002226 L10001:


```

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
              :*****
              :*****
              :SWITCH REGISTER OPTIONS
              :
100000      SW15= 100000
040000      SW14=  40000
020000      SW13= 20000
010000      SW12= 10000
004000      SW11=  4000
002000      SW10=  2000
001000      SW09=  1000
000400      SW08=   400
000200      SW07=   200
000100      SW06=   100
000040      SW05=    40
000020      SW04=    20
000010      SW03=    10
000004      SW02=     4
000002      SW01=     2
000001      SW00=     1
              :*****
              :CSR AND STAU WORD DEFINITIONS
              :SELO (CSR) - BSELO/BSEL1
100000      RUN=  BIT15      ;SET IF RUNNING
040000      MCLR= BIT14      ;MASTER CLEAR OF PROCESSOR AND LINE UNIT
020000      MDIAG= BIT13     ;CSR MAINTENANCE - ENABLE MICRODIAGNOSTICS
010000      STLU=  BIT12     ;CSR MAINTENANCE - STEP LINE UNIT
004000      LPLU=  BIT11     ;CSR MAINTENANCE - LINE UNIT LOOP
002000      ROMO=  BIT10     ;CSR MAINTENANCE
001000      ROMI=  BIT9      ;CSR MAINTENANCE
000400      STUP=  BIT8      ;CSR MAINTENANCE - USED WITH LOOP LU
              :WHEN ASSERTED, XMITTER SHIFTS; CLEAR, REC. SHIFTS
000200      RDI=   BIT7      ;CSR - DMR11 READY RESPONSE
000100      IESET= BIT6      ;CSR - INTERRUPT ENABLE INPUT - DMR11 INTERRUPTS
              :CPU WHEN RDI SET IN RESPONSE TO RQI BEING SET.
000040      RQI=   BIT5      ;CSR - REQUEST IN

```

9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

47      000020      IECLR= BIT4      ;CSR - INTERRUPT ENABLE INPUT - DMR11 INTERRUPTS
48      ;CPU WHEN RDI CLEARS IN RESPONSE TO RDI BEING CLEAR.
49      ;(DMR RUN MODE ONLY)
50      000004      RCV= BIT2      ;CSR - IF 0, TRANSMIT & IF 1, RECEIVE
51
52      ;SEL2 - BSEL2/BSEL3
53      000200      RDO= BIT7      ;SEL2 - DMR11 SETS TO INDICATE DATA READY FOR OUTPUT
54      000100      IEO= BIT6      ;SEL2 - SET TO ENABLE DMR11 TO INTERRUPT WHEN RDO
55
56      ;SEL6 - BSEL6/BSEL7
57      020000      BASEUP= BIT13 ;SEL6 - CONTROL OUT - RESPONSE TO DMR MODE BASE
58      ;TABLE UPDATE COMMAND.
59      C10000      RES= BIT12 ;SEL6 - BASE IN -- WHEN SET CAUSES
60      ;RESUMPTION OF OPERATION
61      010000      CTS= BIT12 ;SEL6 - CONTROL OUT - CTS FAILED
62      004000      SECN= BIT11 ;SEL6 - CONTROL IN -- START TIME (3 SEC IF SET
63      ;1 SEC IF CLEAR)
64      002000      HDX= BIT10 ;SEL6 - HALF-DUPLEX & CLEAR FOR FULL-DUPLEX
65      002000      CD= BIT10 ;SEL6 - CONTROL OUT - CD GLITCHED
66      001000      HALTC= BIT9 ;SEL6 - EXTENDED CONTROL OUT - HALT COMPLETED
67      000400      MAINT= BIT8 ;SEL6 - DDCMP MAINTENANCE DURING CONTROL IN
68      000522      DMR= BIT8!122 ;SEL6 - BASE IN -- SET FOR DMR11 MODE
69      ;122 IS THE DMR PASSWORD FOR BSEL6 AND
70      ;BIT8 SETS THE DMR MODE BIT IN BSEL7
71      000400      NXM= BIT8 ;SEL6 - CONTROL OUT - NON EXISTENT MEMORY
72      000200      STREC= BIT7 ;SEL6 - CONTROL OUT - START RECEIVED
73      000100      DISCON= BIT6 ;SEL6 - CONTROL OUT - DISCONNECT
74      000100      DTR= BIT6 ;SEL6 - MODEM WRITE - DATA TERMINAL READY
75      000040      DMRRUN= BIT5 ;SEL6 - CONTROL OUT - DMR RUN MODE
76      000020      TOLONG= BIT4 ;SEL6 - CONTROL OUT - MESSAGE TOO LONG
77      000010      MAINT1= BIT3 ;SEL6 - MODEM WRITE - LOCAL MODEM LOOPBACK
78      000010      MNTREC= BIT3 ;SEL6 - CONTROL OUT - MAINTENANCE MSG. RECEIVED
79      000004      NOBFR= BIT2 ;SEL6 - CONTROL OUT - NO BUFFER
80      000004      MAINT2= BIT2 ;SEL6 - MODEM WRITE - REMOTE MODEM LOOPBACK
81      000002      TOUT= BIT1 ;SEL6 - CONTROL OUT - TIME OUT
82      000001      NAKS= BIT0 ;SEL6 - CONTROL OUT - NAKS THRESHOLD EXCEEDED
83
84
85      ;*****
86      ;DDCMP COMMANDS - BITS 0 & 1 IN SEL0 AND SEL2
87
88      ;INPUT (SEL0)
89      000000      BACCT= 0 ;BUF ADDRESS AND CHARACTER COUNT TRANSMIT
90      000001      CNTRL= 1 ;CONTROL COMMAND (IN OR OUT)
91      000002      HLT= 2 ;HALT COMMAND
92      000003      BASEI= 3 ;BASE IN COMMAND
93      000004      BACCR= 4 ;BUF ADDRESS AND CHARACTER COUNT RECEIVE
94      000005      WMODEM= 5 ;WRITE MODEM STATUS REGISTER
95      000006      EXERR= 6 ;ENABLE EXTENDED ERROR NOTIFICATION
96      000007      DXERR= 7 ;DISABLE EXTENDED ERROR NOTIFICATION
97      000010      DDMC= 10 ;DESELECT DMC LINE MODE
98      000011      UPDATE= 11 ;REQUEST BASE TABLE UPDATE
99      000012      TIMER= 12 ;SET REP/SELECT TIMER VALUE
100     000013      THRESH= 13 ;SET THE FOLLOWING THRESHOLDS:
101     ;NAKS RECVD
102     ;NAKS SENT
103     ;REP/SEL

```

```
104                                     ;NO BUFFER
105      000014      RRAM= 14           ;READ M8207 RAM (0-377)
106      000015      INTER= 15         ;WRITE INTERFACE IN AX3-15
107      000017      RMODEM= 17        ;READ MODEM STATUS (=NOP)
108
109                                     ;OUTPUT (SEL2)      NOTE: CNTRL IS USED FOR SEL2
110      000007      CMD= 7             ; ** MASK USED TO CLEAR COMMAND BITS 0-2 **
111
112      ;*****
113      ;:BASE TABLE OFFSETS
114
115                                     ;NOTE: THE OFFSETS FOR BASE+3.-BASE+10 WERE
116                                     ;INTENTIONALLY NOT LABELLED, BECAUSE THOSE LOCATIONS
117                                     ;MUST NOT BE CHANGED IN ORDER TO BE DMC COMPATIBLE.
118                                     ;THE LABELS BELOW CORRESPOND WITH THOSE USED IN THE
119                                     ;DMR MICROCODE.
119      000042      R= 42                ;#R - MESSAGE RECEIVED
120      000043      N= 43                ;#N - MESSAGE TRANSMITTED
121      000044      A= 44                ;#A - MESSAGE ACKNOWLEDGED
122      000045      T= 45                ;#T - NEXT MESSAGE TO BE TRANSMITTED
123      000046      X= 46                ;#X - LAST COMPLETED TRANSMISSION
124      000055      PRETIM= 55           ;PROGRAMMABLE REP/SEL TIMER VALUE.
125      000060      TH1L= 60            ;THRESHOLD LEVEL - NAKS RECEIVED .
126      000062      TH2L= 62            ;THRESHOLD LEVEL - NAKS SENT.
127      000064      TH3L= 64            ;THRESHOLD LEVEL - REP SENT.
128      000066      TH4L= 66            ;THRESHOLD LEVEL - NO BUFFER AVAILABLE.
129      000072      ISP7= 72            ;IMAGE OF SCRATCH PAD 7
130      000076      ISP13= 76           ;IMAGE OF SCRATCH PAD 13
131
132      ;*****
133      ;:INSTRUCTION DEFINITIONS
134
135      000207      RETURN=207           ;RETURN FROM SUB.      [= JSR PC]
136
137
138      ;*****
139      ; MISC. EQUATES
140
141      000006      LLOOP= 6             ;LOCAL MODEM LOOPBACK
142      000007      RLOOP= 7            ;REMOTE MODEM LOOPBACK.
143      000015      CR= 15              ;ASCII CARRIAGE RETURN
144      000012      LF= 12              ;ASCII LINE FEED
145
146
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

.SBTTL GLOBAL DATA SECTION

:/ THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
:/ IN MORE THAN ONE TEST.

:DMR11 VECTOR AND REGISTER INDIRECT POINTERS

DMRVEC: .WORD 0 ;DMR11 RECEIVER INTERRUPT VECTOR
DMTVEC: .WORD 0 ;DMR11 TRANSMITTER INT. VECTOR
CSR: .WORD 0 ;POINTER TO DMR11 CONTROL STATUS REGISTER
SEL2: .WORD 0 ;POINTER TO DMR11 CONTROL OUT REGISTER (SEL 2)
SEL4: .WORD 0 ;POINTER TO DMR11 PORT REGISTER (SEL 4)
SEL6: .WORD 0 ;POINTER TO DMR11 PORT REGISTER (SEL 6)
SELO= CSR ;CSR IS SELO
BSELO= CSR ;LOW BYTE OF CSR
BSEL1: .WORD 0 ;POINTER TO DMR11 CSR HIGH BYTE
BSEL2= SEL2 ;LOW BYTE OF SEL2
BSEL3: .WORD 0 ;POINTER TO SEL2 HIGH BYTE
BSEL4= SEL4 ;LOW BYTE OF SEL4
BSEL5: .WORD 0 ;POINTER TO SEL4 HIGH BYTE
BSEL6= SEL6 ;LOW BYTE OF SEL6
BSEL7: .WORD 0 ;POINTER TO SEL6 HIGH BYTE

:OTHER HARDWARE PARAMETERS

WTYPE: .WORD 0 ;MICROPROCESSOR TYPE
DMTURN: .WORD 0 ;TURN AROUND TYPE (0-7)
MICRO: .WORD 0 ;MICRODIAGNOSTICS (IF 1(YES) - ENABLED)

:PROGRAM CONTROL PARAMETERS

DMRFLG: .WORD 0 ;FLAG SET WHEN DMR MODE IS REQUESTED IN
;THE BASE IN COMMAND. USED TO FLAG THAT
;A DMR MODE ACKNOWLEDGE IS EXPECTED.
INFACE: .WORD 0 ;FLAG TO ALLOW CHANGE OF INTERFACE TYPE
;BY WRITING AX3-15. FLAG SET/CLEARED IN INIT.
FRSTIM: .WORD 0 ;FLAG=0 IF PROGRAM JUST LOADED
FRSPAS: .WORD 0 ;FLAG=0 IF FIRST PASS AFTER LOAD
STARES: .WORD 0 ;FLAG=0 IF 1ST TIME THRU AFTER STA OR RES

:FOLLOWING PARAMETERS ARE USED IN THE
:INTERRUPT TESTS (TESTS 15-19):
START: .WORD 0 ;FLAG SET WHEN A CONTROL IN HAS BEEN ISSUED.
RESUME: .WORD 0 ;FLAG SET WHEN A BASE IN WITH RESUME DESIRED.
DMCMDE: .WORD 0 ;FLAG SET WHEN A BASE IN WITH DMC MODE DESIRED
MNTMDE: .WORD 0 ;FLAG SET WHEN MAINTENANCE MODE IS DESIRED.
MMANAG: .WORD 0 ;FLAG RETURNED IN THE SUBROUTINE \$BUFFS
;MMANAG=1, MEMORY MANAGED BUFFERS USED

GLOBAL DATA SECTION

```

58
59 002304 000000      AX3:      .WORD  0      ;BIT PATTERN TO WRITE INTO AX3-15, WHEN
60                                     ;IT IS REQUESTED TO ALLOW INTERFACE
61                                     ;SELECTION. (TEST CONFIGURATION 1-4)
62                                     ;BIT0 = TEST BIT (MUST BE SET TO ALLOW SELECT)
63                                     ;BIT3 = INTEGRAL MODEM
64                                     ;BIT4 = V.35
65                                     ;BIT6 = EIA
66                                     ;BIT7 = RS422
67 002306 000000      WMAINT: .WORD  0      ;FLAG SET WHEN IT IS NECESSARY TO WRITE
68                                     ;MODEM MAINTENANCE BITS (MAINTENANCE 1 & 2)
69                                     ;THIS FLAG IS SET OR CLEARED IN THE INIT CODE.
70 002310 000000      MANUF:   .WORD  0      ;***** MANUFACTURING USE ONLY *****
71                                     ;THIS WORD MAY BE PATCHED TO A NON ZERO WHEN
72                                     ;MANUFACTURING SPECIAL TEST CONNECTORS ARE
73                                     ;USED. THIS WILL ALLOW MAINTENANCE BITS
74                                     ;TO BE SET.
75
76
77      ;:*****
78      ;PROGRAM VARIABLES
79
80                                     ;WORD1-WORD3 VALUES DETERMINED IN INIT
81 002312 000000      WAIT1:   .WORD  0      ;CODE DEPENDING ON THE BAUD RATE.
82                                     ;VALUE FOR TIMEOUT COUNTER
83 002314 000000      WAIT2:   .WORD  0      ;USED IN $WAIT SUBROUTINE
84                                     ;VALUE FOR TIMEOUT COUNTER USED IN $MSCLR
85 002316 000000      WAIT3:   .WORD  0      ;AND $CLRQI SUBROUTINES.
86 002320 000000      WAIT4:   .WORD  0      ;VALUE FOR TIMEOUT COUNTER USED IN $INOUT.
87 002322 000000      BUFSIZ:  .WORD  0      ;WORD USED AS OUTER LOOP COUNTER IN $INOUT.
88 002324 000000      BUFNUM:  .WORD  0      ;CALCULATED BUFFER SIZE IN BYTES.
89                                     ;# OF RECEIVE & TRANSMIT BUFFERS. THIS
90                                     ;VARIABLE IS USED IN THE SUBROUTINE $BUFFS
91 002326 000000      INRCV:   .WORD  0      ;COUNTER FOR # OF BA/CC IN RECEIVES.
92 002330 000000      INXMIT:  .WORD  0      ;COUNTER FOR # OF BA/CC IN TRANSMITS.
93 002332 000000      OUTRCV:  .WORD  0      ;COUNTER FOR # OF BA/CC OUT RECEIVES.
94 002334 000000      OUTXMT:  .WORD  0      ;COUNTER FOR # OF BA/CC OUT TRANSMITS.
95
96      ;:*****
97      ;* MISCELLANEOUS STORAGE
98 002336 000000      TEMP:    .WORD  0      ;SCRATCH WORD USED FOR MISC. STORAGE IN SUB.
99 002340 000000      SAVE:    .WORD  0      ;SCRATCH WORD USED FOR MISC. STORAGE IN SUB.
100 002342 000000      FLAG:    .WORD  0      ;SCRATCH WORD USED FOR MISC. FLAG IN SUB.
101 002344 000000      SFLAG:   .WORD  0      ;FLAG USED IN TEST 15 FOR LOOP CONTROL.
102 002346 000000      SKIP:    .WORD  0      ;FLAG USED IN TEST 7 TO MARK WHETHER TO SKIP
103                                     ;A PORTION OF THE TEST.
104 002350 000000      NXMFLG: .WORD  0      ;FLAG USED TO MARK THAT THE DMR ADDRESS IS NXM
105
106 002352 000000      INFLAG:  .WORD  0      ;FLAG USED IN INISR TO FLAG WHEN ALL THE
107                                     ;BA/CC INS HAVE BEEN DONE.
108
109 002354 000000      OUTFLG:  .WORD  0      ;FLAG USED IN OUTISR TO FLAG WHEN ALL THE
110                                     ;BA/CC OUTS HAVE BEEN DONE.
111 002356 000000      RESFLG:  .WORD  0      ;FLAG USED IN IN ISR TO FLAG THAT THE RESUME
112                                     ;COMMAND HAS JUST BEEN ISSUED.
113 002360 000000      ERRFLG:  .WORD  0      ;FLAC USED IN THE WAIT SUBROUTINES ($WAIT
114                                     ; & $CLRQI) TO RETURN ERROR CONDITON (SEC)

```


GLOBAL DATA SECTION

```

115
116
117 002362 000000 LAST: .WORD 0 ;WORD USED TO STORE LAST COMMAND PROCESSED IN
118 ;THE INPUT INTERRUPT ROUTINE.
119 002364 000000 ERROR: .WORD 0 ;ERROR STORAGE
120 002366 000000 LOGDEV: .WORD 0 ;LOGICAL DEVICE NUMBER
121 002370 000000 PSTACK: .WORD 0 ;CONTAINS BASE LEVEL PROGRAM SP
122 002372 000000 SUBRPC: .WORD 0 ;PC OF SUBR CALL FOR ERROR REPORTS
123 002374 000000 NESTPC: .WORD 0 ;FLAG TO NOTIFY WHEN A SUBR IS NESTED
124 ;IN ANOTHER SUBROUTINE (WHEN SET)
125 002376 000000 CLRNO: .WORD 0 ;THIS WORD IS INCREMENTED DURING EACH MASTER
126 ;CLEAR. THIS WILL ALLOW EVERY OTHER MASTER
127 ;CLEAR TO RUN THE MICRO TESTS.
128
129 ;ROM CHECK VARIABLES
130 002400 000000 LOCRC: .WORD 0 ;CRC STORAGE FOR LOW BYTE CHIP
131 002402 000000 HICRC: .WORD 0 ;CRC STORAGE FOR HIGH BYTE CHIP
132 002404 000000 LOWORD: .WORD 0 ;TEMP. WORD CONTAINING 2 CONSECUTIVE LOW BYTES
133 002406 000000 HIWORD: .WORD 0 ;TEMP. WORD CONTAINING 2 CONSECUTIVE HI BYTES
134 002410 000000 ROMADR: .WORD 0 ;POINTER TO ROM ADDRESS.
135 002412 000000 CHIPNO: .WORD 0 ;CHIP NUMBER BEING CHECKED.
136 .EVEN
137
138 ;:*****
139 ;:*****
140 ;:BUFFER AREA
141 ;
142 ; ** CCITT PSUEDO-RANDOM TEST PATTERN **
143 ; THE FOLLOWING 32 WORDS TRANSLATE INTO A 512 BIT PATTERN
144 ; THAT WAS GENERATED ACCORDING TO CCITT RECOMMENDATION V.52. THIS
145 ; PATTERN WAS GENERATED BY A 9 BIT SHIFT REGISTER (INITIALIZED
146 ; AS 1S) WHOSE 5TH AND 9TH BITS ARE XORED. THIS XOR RESULT IS SHIFTED
147 ; INTO THE 1ST BIT OF THE REGISTER AS THE REGISTER IS SHIFTED RIGHT.
148 ; THE 9TH BIT (OR BIT SHIFTED OUT) IS SHIFTED INTO THE BIT PATTERN.
149 ; NOTE: CCITT RECOMMENDED 511 BITS, I'VE EXTENDED THIS BY 1 BIT TO END
150 ; ON A WORD BOUNDARY.
151 002414 177603 157427 031011 $CCITT: .WORD 177603,157427,031011
152 002422 047321 163715 105221 .WORD 047321,163715,105221
153 002430 143325 142304 040041 .WORD 143325,142304,040041
154 002436 014116 052606 172334 .WORD 014116,052606,172334
155 002444 105025 123754 111337 .WORD 105025,123754,111337
156 002452 111523 030030 145064 .WORD 111523,030030,145064
157 002460 137642 143531 063617 .WORD 137642,143531,063617
158 002466 135015 066730 026575 .WORD 135015,066730,026575
159 002474 052012 053627 070071 .WORD 052012,053627,070071
160 002502 151172 165044 031605 .WORD 151172,165044,031605
161 002510 166632 016741 .WORD 166632,016741
162
163 ;:*****
164 ;: TRANSMIT BUFFER (SMALL)
165
166 002514 000000 TFLAG: .WORD 0 ;FLAG FOR STATUS OF TRANSMIT BUFFER
167 000044 TCOUNT= 36. ;CHARACTER COUNT OF TBUF
168 002516 101 102 103 TBUF: .ASCII /ABCDEFGH IJKLMNOPQRSTUVWXYZ0123456789/
002521 104 105 106
002524 107 110 111
002527 112 113 114

```

002532	115	116	117	
002535	120	121	122	
002540	123	124	125	
002543	126	127	130	
002546	131	132	060	
002551	061	062	063	
002554	064	065	066	
002557	067	070	071	
002562	000			

169				.EVEN
170				
171				
172				::*****
173				:: RECEIVE BUFFER (SMALL)
174				
175	002564	000000		RFLAG: .WORD 0 ;FLAG FOR STATUS OF RECEIVE BUFFER
176		000044		RCOUNT= 36. ;CHARACTER COUNT OF RBUF
177	002566			RBUF: .BLKB 38. ;36. BYTE BUFFER + 2 BYTES USED
178				;TO MARK THE END OF THE RECEIVE BUFFER
179				.EVEN
180				
181				::*****
182				:: BASE TABLE
183				
184	002634			BASE: .BLKB 256. ;MICROPROCESSOR MEMORY ALLOCATION
185				
186				::*****
187				:: TRANSMIT AND RECEIVE BUFFER POINTERS
188				
189	003234			XMTBUF: .BLKW 128. ;POINTERS TO TRANSMIT BUFFERS (UP TO 64)
190				;1 WORD FOR ADDRESS AND 1 WORD FOR CHAR. COUNT
191	003634			RCVBUF: .BLKW 128. ;POINTERS TO RECEIVE BUFFERS (UP TO 64).
192				
193				::*****
194				:: BUFFER AREA (LARGE)
195				
196	004234			BIGBUF: .BLKB 4000 ;MAX BUFFER (2K BYTES)

1
2
3

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
32
33
34
35
36

.SBTTL GLOBAL TEXT SECTION

:XX
: THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
: MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
: MORE THAN ONE TEST.
:XX

:*****
:* NAMES OF DEVICES SUPPORTED BY PROGRAM
:*****
: DEVTYP <DMR11>

010234
010234 104 115 122
010234 061 061 000
010237

LSDVTYP::
.ASCIZ /DMR11/
.EVEN

:*****
:* TITLE OF PROGRAM
:*****
: DESCRIPT <DMR-11 FUNCTIONAL TESTS>

010242
010242
010242 104 115 122
010245 055 061 061
010250 040 106 125
010253 116 103 124
010256 111 117 116
010261 101 114 040
010264 124 105 123
010267 124 123 000

L\$DESC::
.ASCIZ /DMR-11 FUNCTIONAL T

.E /EN

:
: FORMAT STATEMENTS USED IN PRINT CALLS
:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```
.SBTTL GLOBAL SUBROUTINES

://////
:/ THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
://////

:*****
:
:   MACROS - THERE ARE 2 BASIC TYPES OF MACROS USED
:           1. NORMAL MACROS -
:           2. DMR11 FUNCTIONAL MACROS - THESE MACROS MAY
:              BE NOTHING MORE THAN A CALL TO A SUBROUTINE,
:              BUT THEY ARE DISTINCT DMR FUNCTIONS WHICH CAN
:              DISTINGUISHED BY THE IN-LINE MACRO NAME.
:*****

:*****
: CALL MACRO - CALL ROUTINE = JSR PC, ROUTINE
:              (NOTE: RETURN IS EQUATED TO A RTS PC)
:*****

:   .MACRO CALL ROUTIN
:   .IF B, ROUTIN
:   .ERROR ROUTINE: ## MISSING ROUTINE-EXPANSION ABORT ##
:   .MEXIT
:   .ENDC
:   JSR PC,ROUTIN
:   .ENDM

:*****
: WAIT $FLAG MACRO - THIS MACRO INTERPUTS THE $FLAG AS RDI, RQI OR RDO.
:                   IF RDI OR RDO, THE SUBROUTINE CALLED WILL WAIT UNTIL
:                   THE RESPECTIVE BIT IS SET. IF RQI, THE SUBROUTINE
:                   CALLED WILL CLEAR RQI AND WAIT UNTIL RDI IS CLEARED.
:*****

:   .MACRO WAIT $FLAG
:   .NLIST
:   .LIST ME
:   .LIST
:
:           ;**** MACRO EXPANSION ****
:   .IF B, $FLAG
:   .ERROR FLAG ;## MISSING FLAG FOR WAIT - EXPANSION ABORT ##
:   .MEXIT
:   .ENDC
:   .IF IDN $FLAG,RQI
:   JSR PC, $CLRQI ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
:   .ENDC
:   .IF IDN $FLAG,RDI
:   JSR PC, $WAIT ;CALL WAIT ROUTINE
:   .WORD 0 ;FLAG THAT WE'RE WAITING FOR RDI
:   .ENDC
:   .IF IDN $FLAG,RDO
:   JSR PC, $WAIT ;CALL WAIT ROUTINE
:   .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
```

58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77

```
.ENDC
;*****
.NLIST ME
.ENDM

:*****
: CLEAR MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE
: SMSCLR SUBROUTINE
:*****
.MACRO CLEAR
.NLIST
.LIST ME
.LIST
JSR PC, SMSCLR ;***** MACRO EXPANSION *****
;ISSUE A DMR MASTER CLEAR
;*****
.NLIST ME
.ENDM
```

GLOBAL SUBROUTINES

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

```

:*****
: BASEIN MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE
: $BASEIN SUBROUTINE (WITH DEFAULT ARGUMENTS
: IF ARGUMENTS NOT GIVEN)
:*****
.MACRO BASEIN $A,$B,$C
.NLIST
.LIST ME
.LIST

:***** MACRO EXPANSION *****
.IF B $A
JSR PC,$BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
.WORD LPLU ;SET LINE UNIT LOOP
.WORD BASE ;BASE TABLE ADDRESS
.WORD DMR ;DMR-11 MODE

.IFF
JSR PC,$BASEI ;CALL BASE IN ROUTINE
.WORD $A ;MAINTENANCE MODE BITS TO SET IN BSEL1
.WORD $B ;BASE TABLE ADDRESS
.WORD $C ;MODE

.ENDC

:***** *****
.NLIST ME
.ENDM

:*****
: CNTRIN MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE
: $CNTIN SUBROUTINE (WITH DEFAULT ARGUMENTS
: IF ARGUMENTS NOT GIVEN)
:*****
.MACRO CNTRIN $A
.NLIST
.LIST ME
.LIST

:***** MACRO EXPANSION *****
.IF B $A
JSR PC,$CNTIN ;CALL CONTROL IN ROUTINE WITH DEFAULT
.WORD 0 ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.

.IFF
JSR PC,$CNTIN ;CALL CONTROL IN ROUTINE
.WORD $A ;SEL6 - (DUPLEX, MODE)

.ENDC

:***** *****
.NLIST ME
.ENDM

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```
*****  
: DMRIN MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE  
: $DMRIN SUBROUTINE  
*****  
: .MACRO DMRIN SA,$B,$C  
: .NLIST  
: .LIST ME  
: .LIST  
: .IF B SA ;**** MACRO EXPANSION ****  
: .ERROR DMRIN; ## MISSING ARGUMENTS-EXPANSION ABORT ##  
: .MEXIT  
: .ENDC  
: JSR PC, $DMRIN ;CALL DMR MODE INPUT ROUTINE  
: .WORD SA ;INPUT COMMAND  
: .IF B $B  
: .WORD 0 ;NO SEL4  
: .IFF .WORD $B ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)  
: .ENDC  
: .IF B $C  
: .WORD 0 ;NO SEL6  
: .IFF .WORD $C ;SEL6 VALUE (OR BITS TO SET IN BSEL6)  
: .ENDC  
: .NLIST ME ;****  
: .ENDM *****  
  
*****  
: SHUTDN MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE  
: $HALT SUBROUTINE  
*****  
: .MACRO SHUTDN  
: .NLIST  
: .LIST ME  
: .LIST  
: JSR PC, $HALT ;**** MACRO EXPANSION ****  
: ;DMR HALT ROUTINE.  
: ;****  
: .NLIST ME  
: .ENDM
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```
*****
: BACCIR MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE
: SBACC SUBROUTINE (WITH DEFAULT ARGUMENTS
: IF ARGUMENTS NOT GIVEN)
*****
```

```
.MACRO BACCIR $A,$B
.NLIST
.LIST ME
.LIST
;**** MACRO EXPANSION ****
.IF B $A
JSR PC,$BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
.WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND
.WORD RBUF ;RECEIVE BUFFER
.WORD RCOUNT ;RECEIVE CHARACTER COUNT
.IFF
JSR PC,$BACC ;CALL BA/CC IN ROUTINE
.WORD RQI!BACCR ;BA/CC IN RECEIVE COMMAND
.WORD $A ;BUFFER ADDRESS BITS 0-15
.WORD $B ;BA BITS 16/17 AND CHAR. COUNT
.ENDC
;****
.NLIST ME
.ENDM
```

```
*****
: BACCIT MACRO - THIS IS A DMR FUNCTIONAL MACRO WHICH CALLS THE
: SBACC SUBROUTINE (WITH DEFAULT ARGUMENTS
: IF ARGUMENTS NOT GIVEN)
*****
```

```
.MACRO BACCIT $A,$B
.NLIST
.LIST ME
.LIST
;**** MACRO EXPANSION ****
.IF B $A
JSR PC,$BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
.WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
.WORD TBUF ;TRANSMIT BUFFER ADDRESS
.WORD TCOUNT ;TRANSMIT CHARACTER COUNT
.IFF
JSR PC,$BACC ;CALL BA/CC IN ROUTINE
.WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
.WORD $A ;BUFFER ADDRESS BITS 0-15
.WORD $B ;BA BITS 16 & 17 AND CHAR. COUNT
.ENDC
;****
.NLIST ME
.ENDM
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

*****
*****
SUBROUTINE $WAIT
FUNCTION - TO WAIT FOR RDI TO BE SET IN SEL0
OR RDO TO BE SET IN SEL2

CALLING FORMAT:      JSR    PC,    $WAIT
                     .WORD  FLAG
                     (MACRO CALL -- WAIT RDI)

NESTING LEVEL - MAY BE CALLED FROM ANOTHER SUBROUTINE

ENTRY CONDITIONS - FLAG = 1 - WAIT FOR RDO
                     = 0 - WAIT FOR RDI
                     WAIT1 = DELAY COUNTER (DETERMINED IN INIT.)
                     NESTPC = 1 - ROUTINE NESTED WITHIN ANOTHER
                               SUBROUTINE.
                     = 0 - ROUTINE NOT NESTED.

EXIT CONDITIONS - EITHER RDI OR RDO BIT SET AS EXPECTED
OR (ERROR CONDITONS):
1. RDI OR RDO SET, BUT NOT THE EXPECTED ONE
   THE USER WILL BE INFORMED. HOWEVER,
   THIS WILL NOT NECESSARILY BE AN ERROR.
2. BIT NOT SET BEFORE DELAY EXPIRED.
   THIS WILL RESULT IN A HARD ERROR MESSAGE
   AND THE CARRY BIT WILL BE SET. THE CARRY
   BIT SET FLAG THE ERROR CONDITION.
    
```

REGISTERS DESTROYED - RESTORED

```

*****
*****
$WAIT:
CLR    ERRFLG      ;CLEAR ERROR FLAG
TST    NESTPC      ;IS THIS NESTED IN ANOTHER SUBROUTINE?
BNE    10$         ;YES - USE THE SUBRPC ALREADY CALCULATED.
MOV    (SP),SUBRPC ;SAVE PC AFTER THE CALL TO $WAIT.
SUB    #4,SUBRPC   ;BACKUP TO THE PC OF THE ACTUAL CALL

10$:
MOV    @(SP),TEMP  ;GET THE FLAG FOR RDI OR RDO
ADD    #2,(SP)     ;INC THE PC LEFT ON THE STACK TO POINT
                     ;PAST THE FLAG ARGUMENT
MOV    R0,-(SP)    ;SAVE R0
MOV    R1,-(SP)    ;SAVE R1
MOV    WAIT1,R1    ;DELAY COUNTER DETERMINED BY BAUD RATE
                     ;(DETERMINED IN INIT ROUTINE).

30$:
CLR    R0          ;INNER LOOP COUNT OF DELAY COUNTER

40$:
BIT    #RDO,@SEL2 ;IS THE RDO BIT SET IN SEL2?
BNE    60$         ;YES - EXIT BIT CHECK LOOP.
BIT    #RDI,@SEL0 ;IS THE RDI BIT SET IN SEL0?
BNE    70$         ;YES - EXIT
    
```

```

010272
010272 005037 002360
010276 005737 002374
010302 001005
010304 011637 002372
010310 162737 000004 002372
010316
010316 017637 000000 002336
010324 062716 000002
010330 010046
010332 010146
010334 013701 002312
010340
010340 005000
010342
010342 032777 000200 171664
010350 001034
010352 032777 000200 171652
010360 001062
    
```

```

58 010362          BREAK          ;CALL SUPERVISOR - ALLOW CONSOLE INTERRUPT.
    010362 104422          TRAP      C$BRK
59 010364 005300          ;LOOP UNTIL R0 RETURNS TO 0
60 010366 001365          DEC      R0
    010370          BNE      40$
61 010370          DELAY     1      ;DELAY 100 MICROSECONDS
    010370 012727 000001          MOV      #1,(PC)+
    010374 000000          .WORD   0
    010376 013727 002116          MOV      L$DLY,(PC)+
    010402 000000          .WORD   0
    010404 005367 177772          DEC      -6(PC)
    010410 001375          BNE      -4
    010412 005367 177756          DEC      -22(PC)
    010416 001367          BNE      -20
62          ;BETWEEN LOOPS.
63 010420 005301          DEC      R1
64 010422 001346          BNE      30$
65 010424          ERRDF   1,EMG1,ERRG2 ;REPEAT UNTIL MAXIMUM LOOP SATISFIED.
    010424 104455          ;TIME OUT ERROR
    010426 000001          TRAP      C$ERDF
    010430 020052          .WORD   1
    010432 015126          .WORD   EMG1
66 010434 005237 002360          .WORD   ERRG2
67 010440 000445          INC      ERRFLG
68 010442          BR      100$          ;SET ERROR FLAG
69 010442 005737 002336          ;BRANCH TO COMMON EXIT.
70 010446 001042          TST     TEMP
71 010450 022737 000001 002364          BNE     100$          ;WERE WE WAITING FOR THE RDO FLAG?
72 010456 001436          CMP     #CNTRL,ERROR ;YES - OK, EXIT.
73 010460          BEQ     100$          ;IS THIS CONTROL OUT ERROR EXPECTED?
    010460 012746 010604          PRINTB #FMS1 ;IF YES, DON'T REPORT THE FOLLOWING ERRORS.
    010464 012746 000001          ;RECEIVED AN RDO, WHEN WAITING FOR RDI
    010470 010600          MOV     #FMS1,-(SP)
    010472 104414          MOV     #1,-(SP)
    010474 062706 000004          MOV     SP,R0
74 010500 032777 000001 171526          TRAP   C$PNTB
75 010506 001422          ADD     #4,SP
76 010510          BIT     #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
    010510 104455          BEQ     100$          ;NO NEED TO CHECK ERROR CODES.
    010512 000011          ERRDF  9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT.
    010514 020174          TRAP   C$ERDF
    010516 015126          .WORD  9
77 010520 005237 002360          .WORD  EMG9
78 010524 000413          .WORD  ERRG2
79 010526          INC      ERRFLG
80 010526 005737 002336          ;SET ERROR FLAG.
81 010532 001410          BR      100$
82 010534          TST     TEMP
    010534 012746 010645          BEQ     100$          ;WERE WE WAITING FOR THE RDI FLAG?
    010540 012746 000001          PRINTB #FMS2 ;YES - OK, EXIT
    010544 010600          ;RECEIVED AN RDI, WHEN WAITING FOR RDO
    010546 104414          MOV     #FMS2,-(SP)
    010550 062706 000004          MOV     #1,-(SP)
    010554          MOV     SP,R0
83 010554          TRAP   C$PNTB
84 010554 005737 002374          ADD     #4,SP
85 010560 001002          TST     NESTPC
86 010562 005037 002372          BNE     105$          ;WAS THIS NESTED IN ANOTHER SUBROUTINE?
87 010566          CLR     SUBRPC ;IF YES - LEAVE THE SUBROUTINE PC ALONE
    ;CLEAR THE PC

```

```

GLOBAL SUBROUTINES

88 010566 012601
89 010570 012600
90 010572 005737 002360
91 010576 001401
92 010600 000261
93 010602
94 010602 000207
95
96 010604 045 116 045 FMS1: .ASCIZ /%N%ARDO SET WHEN EXPECTING RDI%N/
   010607 101 122 104
   010612 117 040 123
   010615 105 124 040
   010620 127 110 105
   010623 116 040 105
   010626 130 120 105
   010631 103 124 111
   010634 116 107 040
   010637 122 104 111
   010642 045 116 000
97 010645 045 116 045 FMS2: .ASCIZ /%N%ARDI SET WHEN EXPECTING RDO%N/
   010650 101 122 104
   010653 111 040 123
   010656 105 124 040
   010661 127 110 105
   010664 116 040 105
   010667 130 120 105
   010672 103 124 111
   010675 116 107 040
   010700 122 104 117
   010703 045 116 000

98
99
100 .EVEN

```

```

MOV (SP)+,R1 :RESTORE R1
MOV (SP)+,R0 :RESTORE R0
TST ERRFLG :WAS THERE AN ERROR (CARRY CLEARED ON TST)
BEQ 110$ :IF NOT, RETURN WITH CARRY CLEAR
SEC :SET CARRY.

```

```

110$:
RETURN

```

GLOBAL SUBROUTINES

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
010770
010774
010776
011002
011004
011010
011012

SUBROUTINE \$CLRQI

FUNCTION - TO CLEAR RQI AND WAIT FOR RDI TO BE CLEARED

CALLING FORMAT: JSR PC, \$CLRQI
(MACRO CALL -- WAIT RQI)

NESTING LEVEL - MAY BE NESTED WITHIN ANOTHER SUBROUTINE

ENTRY CONDITIONS - WAIT2 = DELAY COUNTER (DETERMINED IN INIT. ROUTINE)
NESTPC= 1 - ROUTINE NESTED WITHIN ANOTHER SUBROUTINE.
= 0 - ROUTINE NOT NESTED.

EXIT CONDITIONS - 1. NON ERROR, DMR READY TO RECEIVE THE NEXT COMMAND
2. ERROR IF RDI DOES NOT CLEAR BEFORE THE DELAY ROUTINE EXPIRES. AN ERROR MESSAGE WILL OCCUR. ALSO A CARRY BIT WILL BE SET TO FLAG THE ERROR FOR THE USER.

REGISTERS DESTROYED - RESTORED


```

$CLRQI:
CLR      ERRFLG      ;CLEAR ERROR FLAG
BIC      #RQI,@SELO ;REQUEST INPUT CLEAR
TST      NESTPC      ;IS THIS NESTED IN ANOTHER SUBROUTINE?
BNE      10$         ;YES - USE SUBRPC CALCULATED
MOV      (SP),SUBRPC ;SAVE THE PC AFTER THE CALL TO $WAIT.
SUB      #4,SUBRPC   ;BACKUP TO THE PC OF THE ACTUAL CALL.
10$:
MOV      R0,-(SP)    ;SAVE R0
MOV      R1,-(SP)    ;SAVE R1
MOV      WAIT2,R1    ;GET THE DELAY COUNTER (DETERMINED BY
                    ;BAUD RATE IN INIT ROUTINE)
12$:
CLR      R0          ;INNER LOOP COUNT
20$:
BIT      #RDI,@SELO ;IS THE RDI BIT CLEAR IN SELO?
BEQ      30$         ;YES - EXIT
BREAK    ;CALL SUPERVISOR - ALLOW CONSOLE INTERRUPT.
TRAP    CSBRK
DEC      R0          ;LOOP UNTIL R0 RETURNS TO 0
BNE      20$
DELAY   1           ;DELAY 100 MICROSECONDS

MOV      #1,(PC)+
.WORD    0
MOV      LSDLY,(PC)+
.WORD    0
DEC      -6(PC)
BNE      -4
DEC      -22(PC)
    
```

010706 005037 002360
010712 042777 000040 171312
010720 005737 002374
010724 001005
010726 011637 002372
010732 162737 000004 002372
010740
010740 010046
010742 010146
010744 013701 002314
010750
010750 005000
010752
010752 032777 000200 171252
010760 001427
010762
010762 104422
010764 005300
010766 001371
010770
010770 012727 000001
010774 000000
010776 013727 002116
011002 000000
011004 005367 177772
011010 001375
011012 005367 177756

GLOBAL SUBROUTINES

```

011016 001367
50 011020 005301
51 011022 001352
52 011024 001352
    011024 104455
    011026 000001
    011030 020052
    011032 015126
53 011034 005237 002360
54 011040 005737 002374
55 011040 005737 002374
56 011044 001002
57 011046 005037 002372
58 011052 012601
59 011052 012600
60 011054 012600
61 011056 005737 002360
62 011062 001401
63 011064 000261
64 011066 000207
65 011066 000207
66
67

```

BNE .-20

```

    DEC R1 ;REPEAT UNTIL MAXIMUM LOOP SATISFIED.
    BNE 12$
    ERRDF 1,EMG1,ERRG2 ;TIME OUT ERROR
                                TRAP C$ERDF
                                .WORD 1
                                .WORD EMG1
                                .WORD ERRG2
30$: INC ERRFLG ;SET ERROR FLAG
    TST NESTPC ;WAS THIS A NESTED ROUTINE?
    BNE 40$ ;IF YES - LEAVE THE SUBRPC ALONE
    CLR SUBRPC ;CLEAR THE PC
40$: MOV (SP)+,R1 ;RESTORE R1
    MOV (SP)+,R0 ;RESTORE R0
    TST ERRFLG ;WAS THERE AN ERROR? (CARRY CLEARED ON TST)
    BEQ 50$ ;IF NOT - RETURN WITH CARRY CLEAR
    SEC ;SET CARRY.
50$: RETURN

```

GLOBAL SUBROUTINES

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

*****
*****
SUBROUTINE $MSCLR
FUNCTION - TO PERFORM A MASTER CLEAR FOR THE DMR11
CALLING FORMAT:      JSR    PC,    $MSCLR
                      (MACRO CALL -- CLEAR)
NESTING LEVEL - MAY ONLY BE CALLED FROM IN-LINE CODE (TEST,
SUBTEST OR TEST SEGMENT)
ENTRY CONDITIONS - WAIT2 = DELAY COUNTER (DETERMINED BY INIT. ROUTINE)
CLRNO = EVEN OR ODD COUNT. THE ACTUAL # IS NOT
SIGNIFICANT, HOWEVER IF BIT 0 IS SET
THEN THE MICROTTEST IS SET ALONG WITH
THE MASTER CLEAR. THIS ROUTINE WILL INCR.
THE VALUE. THIS WILL RESULT IN THE MICRO
TESTS BEING RUN ON EVERY OTHER MASTER CLEAR
EXIT CONDITIONS - 1. NO ERROR - DMR11 MICROPROCESSOR INITIALIZED
2. IF RUN BIT NOT SET BEFORE DELAY TIMEOUT, ERROR
WILL RESULT. ADDITONALLY THE ERROR MESSAGE WILL
RELAY THE RESULTS OF THE MICROTTESTS IF THE RUN
BIT IS NOT SET.
NOTE:                THERE IS A PATCH AREA TO ALLOW THESE DIAGNOSTICS
TO RUN ON A M8206 (INSTEAD OF M8207). THIS
SHOULD BE FOR DEVELOPMENT USE ONLY.
REGISTERS DESTROYED - RESTORED

```

```

*****
*****
$MSCLR:
MOV    (SP),SUBRPC    ;SAVE PC AFTER THE CALL TO $WAIT.
SUB    #4,SUBRPC      ;BACKUP TO THE PC OF THE ACTUAL CALL
MOV    R0,-(SP)       ;SAVE R0
MOV    R1,-(SP)       ;SAVE R1
CLRB   @BSEL3         ;CLEAR BSEL3
NOP
NOP
NOP
NOP
;*****
;** PATCH AREA FOR 8206 IF NEEDED **
;CLR @#SEL6 -
;*****
BIT    #BIT0,CLRNO    ;IS THIS AN ODD MASTER CLEAR.
BNE    7$             ;IF YES - BR
MOV    #MCLR,@SELO    ;ISSUE A MASTER CLEAR.
BR     8$
7$:
MOV    #MCLR!MDIAG,@SELO ;ISSUE THE MASTER CLEAR AND TOGGLE
;MICRO TEST SWITCH.
8$:
NOP
;*****

```

```

011070
011070 011637 002372
011074 162737 000004 002372
011102 010046
011104 010146
011106 105077 171132
011112 000240
011114 000240
011116 000240
011120 000240
011122 032737 000001 002376
011130 001004
011132 012777 040000 171072
011140 000403
011142
011142 012777 060000 171062
011150
011150 000240

```

```

58 011152 000240      NOP
59 011154 000240      NOP
60 011156 000240      NOP
61
62 011160 005237 002376  INC  CLRNO
63 011164 013701 002314  MOV  WAIT2,R1
64
65 011170          10$:
66 011170 005000          CLR  R0
67 011172          20$:
68 011172 032777 100000 171032  BIT  #RUN,@SELO
69 011200 001025          BNE  40$
70 011202          BREAK
71 011204 005300          DEC  R0
72 011206 001371          BNE  20$
73 011210          DELAY 1
74 011240 005301          DEC  R1
75 011242 001352          BNE  10$
76 011244          ERRDF 1,EMG1,ERRG3
77 011254          40$:
78 011254 012601          MOV  (SP)+,R1
79 011256 012600          MOV  (SP)+,R0
80 011260 005037 002372  CLR  SUBRPC
81 011264 000207          RETURN
82
83
; ** PATCH AREA FOR 8206 IF NEEDED **
; MOV #RUN,@SELO -
; *****
; INCR WORD (CHANGE ODD TO EVEN ETC.)
; GET THE # OF 100 MICRO SECOND DELAYS
; TO WAIT BEFORE EXITING THE ROUTINE.
; INNER LOOP COUNT
; IS THE RUN BIT SET IN SELO?
; YES - EXIT
; CALL SUPERVISOR - ALLOW CONSOLE INTERRUPT.
TRAP  CSBRK
; LOOP UNTIL R0 RETURNS TO 0
; DELAY 100 MICROSECONDS
MOV  #1,(PC)+
.WORD 0
MOV  LSDLY,(PC)+
.WORD 0
DEC  -6(PC)
BNE  -4
DEC  -22(PC)
BNE  -20
; REPEAT UNTIL MAX LOOP SATISFIED.
TRAP  CSERDF
.WORD 1
.WORD EMG1
.WORD ERRG3

```


GLOBAL SUBROUTINES

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

```

*****
*****
SUBROUTINE $BASEI
FUNCTION - TO PERFORM A BASE IN COMMAND

CALLING FORMAT:      JSR    PC,    $BASEI
                     .WORD A (SELO MAINTENANCE BITS)
                     .WORD B (SEL4 - ADDRESS)
                     .WORD C (SEL6 - MODE AND/OR RESUME)
                     (MACRO CALL -- BASEIN OR BASEIN A,B,C)

NESTING LEVEL - MAY ONLY BE CALLED FROM IN-LINE CODE (TEST,
SUBTEST OR TEST SEGMENT)

ENTRY CONDITIONS - A = MAINTENANCE BITS (I.E. LINE UNIT LOOP BACK)
                  B = BASE TABLE ADDRESS (SEL4)
                  C = MODE + RESUME (SEL6)
                  INFACE = 0 - NO INTERFACE WRITE REQUIRED
                          1 - WRITE INTERFACE (AX3-15)

EXIT CONDITIONS - 1. IF NO ERROR - DMR11 BASE TABLE ASSIGNED
                 2. IF IN DMR MODE, AND INTERFACE WRITE REQUESTED
                   WRITE REQUESTED AX3-15.
                 3. TIMEOUT ERRORS ARE DETECTED IN WAIT SUBROUTINES.
                   DMRFLG = -1 DMR MODE REQUESTED (USED IN CONTROL IN
                               ROUTINE)
                          0 DMC MODE OR RESUME REQUESTED.

REGISTERS DESTROYED - RESTORED
*****
*****
$BASEI:
MOV    (SP),SUBRPC    ;SAVE PC AFTER THE CALL TO $WAIT.
SUB    #4,SUBRPC      ;BACKUP TO THE PC OF THE ACTUAL CALL

MOVB   #ROI!BASEI,@BSELO ;ISSUE THE BASE IN COMMAND.
MOV    #1,NESTPC      ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
WAIT   RDI            ;WAIT FOR RDI
                     ;**** MACRO EXPANSION ****
                     ;CALL WAIT ROUTINE
                     ;FLAG THAT WE'RE WAITING FOR RDI
                     ;****
BNERROR 10$          ;IF NO ERROR, RDI SET - PROCEED
ADD    #6,(SP)        ;CORRECT STACK FOR ERROR EXIT
BR     30$           ;EXIT
10$:
BIS    @(SP),@SELO    ;SET ANY MAINTENANCE BITS
ADD    #2,(SP)        ;INC. POINTER.
MOV    @(SP),@SEL4    ;SET UP BASE ADDRESS
ADD    #2,(SP)        ;INC. POINTER AGAIN
MOV    @(SP),@SEL6    ;SET UP RESUME BIT AND THE HIGH 2 BITS
                     ;OF THE BASE TABLE ADDRESS

```

36	011266			
37	011266	011637	002372	
38	011272	162737	000004	002372
39				
40	011300	112777	000043	170724
41	011306	012737	000001	002374
42	011314			
	011314	004737	010272	
	011320	000000		
43	011322			
	011322	103003		
44	011324	062716	000006	
45	011330	000467		
46	011332			
47	011332	057677	000000	170672
48	011340	062716	000002	
49	011344	017677	000000	170664
50	011352	062716	000002	
51	011356	017677	000000	170654
52				

GLOBAL SUBROUTINES

```

53 011364 062716 000002      ADD    #2,(SP)      ;INC. POINTER AGAIN (SHOULD BE AT RETURN PC)
54 011370      WAIT    RQI      ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
                        ;**** MACRO EXPANSION ****
                        ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
                        ;****
011370 004737 010706      JSR    PC,$CLRQI
55 011374      BERROR  30$      ;IF ERROR, EXIT
011374 103445      ;
56 011376 122777 000122 170634  CMPB   #122,@BSEL6 ;WAS THIS A DMR BASE IN?
57 011404 001004      BNE    15$         ;IF NOT, CLEAR DMR FLAG (DMC MODE)
58 011406 032777 010000 170624  BIT    #RES,@SEL6 ;IS THIS A RESUME?
59 011414 001403      BEQ    16$         ;IF NOT, PROCEED
60 011416      15$:
61 011416 005037 002260      CLR    DMRFLG      ;CLEAR DMR FLAG (NO DMR RUN ACKNOWLEDGE).
62 011422 000432      BR     30$         ;SKIP - TO END
63 011424      16$:
64 011424 012737 177777 002260  MOV    #-1,DMRFLG ;FLAG THAT DMR MODE WAS REQUESTED.
65 011432 005737 002262      TST    INFACE     ;IS AN INTERFACE WRITE REQUIRED?
66 011436 001424      BEQ    30$         ;IF NOT - SKIP TO END
67 011440 022737 000001 002364  CMP    #CNTRL,ERROR ;ARE WE EXPECTING AN ERROR (IN TEST THAT
68                                ;FORCES AN ERROR)
69 011446 001004      BNE    17$         ;IF NOT PROCEED
70 011450 032777 000200 170554  BIT    #RDO,@SELO ;IF EXPECTING AN ERROR - IS RDO SET
71 011456 001014      BNE    30$         ;IF YES - DON'T BOTHER CHANGING THE INTERFACE.
72 011460      17$:
73 011460 112777 000055 170544  MOVB   #RQI!INTER,@BSELO ;ISSUE WRITE INTERFACE COMMAND.
74 011466      WAIT    RDI      ;WAIT FOR RDI
                        ;**** MACRO EXPANSION ****
011466 004737 010272      JSR    PC,$WAIT    ;CALL WAIT ROUTINE
011472 000000      .WORD  0      ;FLAG THAT WE'RE WAITING FOR RDI
                        ;****
75 011474      BERROR  30$      ;IF ERROR, BR TO END.
011474 103405      ;
76 011476 113777 002304 170544  MOVB   AX3,@BSEL7 ;WRITE AX3-15. INTERFACE SELECTED
77                                ;BY AX3 DETERMINED IN INIT. CODE.
78 011504      WAIT    RQI      ;CLEAR RQI AND WAIT FOR RDI TO CLEAR.
                        ;**** MACRO EXPANSION ****
011504 004737 010706      JSR    PC,$CLRQI ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
                        ;****
79 011510      30$:
80 011510 005037 002374      CLR    NESTPC     ;CLEAR THE NEST FLAG
81 011514 005037 002372      CLR    SUBRPC    ;TIDY UP SUBRPC
82 011520 000207      RETURN
83
84

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

*****
*****
SUBROUTINE $CNTIN
FUNCTION - TO PERFORM A CONTROL IN COMMAND

CALLING FORMAT:      JSR    PC,    $CNTIN
                     .WORD  A (SEL6 - MAINTENANCE MODE & HDX)
                     (MACRO CALL -- CNTRIN OR CNTRIN A)

NESTING LEVEL      - MAY ONLY BE CALLED FROM IN-LINE CODE (TEST,
                     SUBTEST OR TEST SEGMENT)

ENTRY CONDITIONS - DMRFLG = -1 EXPECT CONTROL OUT IF IN DMR MODE
                  = 0 NO CONTROL OUT, IN DMC MODE OR RESUME.

EXIT CONDITIONS - 1. IF NO ERROR - DMR11 CONTROL IN PERFORMED
                  2. TIMEOUTS REPORTED IN WAIT SUBROUTINES
                  3. IF THIS IS A DMR MODE START UP CONTROL IN,
                     THIS ROUTINE WILL WAIT FOR A CONTROL
                     OUT - DMR RUN. IF THIS CONTROL OUT IS
                     NOT RECEIVED, THIS WILL RESULT IN AN ERROR
                     MESSAGE AND A REMINDER TO CHECK THE BAUD RATE,
                     INTERFACE AND TURNAROUND (PROBABLE REASON).
    
```

REGISTERS DESTROYED

```

*****
*****
$CNTIN:
MOV    (SP),SUBRPC      ;SAVE PC FROM WHERE THIS SUBR. WAS CALLED.
SUB    #4,SUBRPC        ;BACKUP TO PC OF ACTUAL CALL
MOVB  #RQI+CNTRL,@BSEL0 ;SET UP CONTROL IN COMMAND
MOV   #1,NESTPC        ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
WAIT  RDI              ;WAIT FOR SETTING OF RDI
                     ;**** MACRO EXPANSION ****
JSR   PC,$WAIT         ;CALL WAIT ROUTINE
                     .WORD  0
                     ;FLAG THAT WE'RE WAITING FOR RDI
                     ;****
BNERROR 1$            ;IF NO ERROR - PROCEED
ADD   #2,(SP)          ;CORRECT RETURN ADDRESS
BR    20$              ;ERROR - EXIT
1$:
MOV   @(SP),@SEL6      ;SET MODE DESIRED
ADD   #2,(SP)          ;INC. RETURN PC LEFT ON STACK.
BIT   #MAINT,@SEL6    ;WAS MAINTENANCE MODE REQUESTED?
BEQ   5$               ;IF NOT, LEAVE DMRFLG AS IS.
CLR   DMRFLG          ;CLEAR FLAG - NO RUN MODE CONTROL OUT.
5$:
WAIT  RQI              ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
                     ;**** MACRO EXPANSION ****
JSR   PC,$CLRQI       ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
    
```

```

011522 011637 002372
011522 162737 000004 002372
011534 112777 000041 170470
011542 012737 000001 002374
011550 004737 010272
011554 000000
011556 103003
011556 062716 000002
011564 000463
011566 017677 000000 170444
011574 062716 000002
011600 032777 000400 170432
011606 001402
011610 005037 002260
011614
011614 004737 010706
    
```


012020	103	110	105
012023	103	113	040
012026	111	116	124
012031	105	122	106
012034	101	103	105
012037	054	040	102
012042	101	125	104
012045	040	101	116
012050	104	040	124
012053	125	122	116
012056	101	122	117
012061	125	116	104
012064	051	045	116
012067	000		

77
78
79

.EVEN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52

```

*****
*****
SUBROUTINE $DMRIN
FUNCTION - TO PERFORM A DMR MODE INPUT COMMAND

CALLING FORMAT:      JSR      PC,      $DMRIN
                     .WORD    COMMAND
                     .WORD    B
                     .WORD    C
                     (MACRO CALL -- DMRIN A,B,C)

NESTING LEVEL - MAY BE CALLED FROM IN-LINE CODE (TEST,
SUBTEST OR TEST SEGMENT) OR FROM THE $LOOP
SUBROUTINE

ENTRY CONDITIONS - MUST BE IN DMR MODE
FOR ALL COMMANDS EXCEPT WRITE MODEM
B = SEL4
C = SEL6
FOR MODEM WRITE
B = BITS TO CLEAR IN SEL6
C = BITS TO SET IN SEL6
NESTPC = 1 - SUBROUTINE NESTED WITHIN ANOTHER SUB.
= 0 - SUBROUTINE NOT NESTED.

EXIT CONDITIONS - IF NO ERROR - DMR11 MODE INPUT COMMAND PERFORMED.

REGISTERS DESTROYED
*****
*****
$DMRIN:
TST      NESTPC      ;IS THIS SUBROUTINE NESTED?
BNE      1$          ;IF YES - DON'T CHANGE SUBRPC.
MOV      (SP),SUBRPC ;SAVE PC FROM WHERE THIS SUBR. WAS CALLED.
SUB      #4,SUBRPC   ;BACKUP TO PC OF ACTUAL CALL

1$:
MOVB     @(SP),SAVE  ;SAVE DMR INPUT COMMAND
MOVB     @(SP),@BSELO ;SET UP DMR INPUT COMMAND.
ADD      #2,(SP)     ;INC RETURN PC LEFT ON STACK.
BIS      #RQI,@SELO  ;REQUEST INPUT.
MOV      NESTPC,-(SP) ;SAVE THE CURRENT NEST FLAG.
MOV      #1,NESTPC   ;USE THE FLAG TO SHOW THE WAIT
WAIT     RDI         ;ROUTINE IS NESTED.
JSR      PC,$WAIT    ;WAIT FOR SETTING OF RDI
                     ;**** MACRO EXPANSION ****
                     ;CALL WAIT ROUTINE
                     ;FLAG THAT WE'RE WAITING FOR RDI
                     ;****
MOV      (SP)+,NESTPC ;RESTORE THE ORIGINAL NEST FLAG.
BNERROR  5$          ;IF NO ERROR, OK - PROCEED.

ADD      #4,(SP)     ;UPDATE RETURN ADDRESS.
BR       10$         ;ERROR EXIT.
BCC      5$

```

```

012070
012070 005737 002374
012074 001005
012076 011637 002372
012102 162737 000004 002372
012110
012110 117637 000000 002340
012116 117677 000000 170106
012124 062716 000002
012130 052777 000040 170074
012136 013746 002374
012142 012737 000001 002374
012150
012150 004737 010272
012154 000000
012156 012637 002374
012162
012162 103003
012164 062716 000004
012170 000433

```

```

53 012172
54 012172 122737 000005 002340 5$: CMPB #WMODEM,SAVE ;IS THIS A MODEM WRITE?
55 012200 001413 BEQ 6$ ;IF YES - SET/CLEAR BITS.
56 012202 017677 000000 170026 MOV @ (SP),@SEL4 ;PASS VALUE FOR SEL4 (VALUE, IF ANY,
57 ;DEPENDS ON THE DMR COMMAND)
58 012210 062716 000002 ADD #2,(SP) ;INC. RETURN PC LEFT ON STACK.
59 012214 C17677 000000 170016 MOV @ (SP),@SEL6 ;PASS VALUE FOR SEL6 (VALUE, IF ANY,
60 ;DEPENDS ON THE DMR COMMAND)
61 012222 062716 000002 ADD #2,(SP) ;INC. RETURN PC LEFT ON STACK.
62 012226 000412 BR 7$
63 012230 6$:
64 012230 047677 000000 170002 BIC @ (SP),@SEL6 ;CLEAR MODEM BITS
65 012236 062716 000002 ADD #2,(SP) ;INC. RETURN PC LEFT ON STACK
66 012242 057677 000000 167770 BIS @ (SP),@SEL6 ;SET MODEM BITS
67 012250 062716 000002 ADD #2,(SP) ;INC. RETURN PC LEFT ON STACK.
68 012254 7$:
69 012254 WAIT RQI ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
012254 004737 010706 JSR PC, $CLRQI ;**** MACRO EXPANSION ****
;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
;****

70 012260 10$:
71 012260 005737 002374 TST NESTPC ;WAS THIS ROUTINE NESTED?
72 012264 001002 BNE 15$ ;BR IF YES
73 012266 005037 002372 CLR SUBRPC ;CLEAR PC
74 012272 15$:
75 012272 005037 002340 CLR SAVE ;RESTORE TEMP VALUE
76 012276 000207 RETURN
77
78
79
80

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

36
37
38
39
40
41
42
43
44
45

46
47
48
49

SUBROUTINE \$BACC

FUNCTION - TO PERFORM A BUFFER ADDRESS/CHARACTER
COUNT IN COMMAND

CALLING FORMAT: JSR PC, \$BACC
 .WORD SEL0 ;BA/CC IN COMMAND
 .WORD SEL4 ;BUFFER ADDRESS
 .WORD SEL6 ;BA BITS 16 & 17 AND
 ;CHARACTER COUNT
 (MACRO CALL -- BACCIT OR BACCIT A,B)
 OR (MACRO CALL -- BACCIR OR BACCIR A,B)

NESTING LEVEL - MAY ONLY BE CALLED FROM IN-LINE CODE (TEST,
SUBTEST OR TEST SEGMENT)

ENTRY CONDITIONS -

EXIT CONDITIONS - IF NO ERROR - DMR11 BA/CC COMMAND IN PERFORMED

REGISTERS DESTROYED - NOT AFFECTED


```

$BACC:
MOV    (SP),SUBRPC    ;SAVE PC FROM WHERE THIS SUBR. WAS CALLED.
SUB    #4,SUBRPC      ;BACKUP TO PC OF ACTUAL CALL
MOVB  @ (SP),@BSEL0   ;SET UP BA/CC COMMAND IN (TRANSMIT OR RECEIVE)
ADD   #2,(SP)         ;INC POINTER ON STACK
MOV   #1,NESTPC      ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
WAIT  RDI             ;WAIT FOR SETTING OF RDI
      ;**** MACRO EXPANSION ****
      ;CALL WAIT ROUTINE
      ;FLAG THAT WE'RE WAITING FOR RDI
      ;****          ****
BNERROR 10$          ;IF NO ERROR - PROCEED
ADD    #4,(SP)        ;CORRECT STACK FOR ERROR EXIT.
BR     20$            ;EXIT
10$:
MOV    @ (SP),@SEL4   ;SET BUFFER ADDRESS
ADD   #2,(SP)         ;INC POINTER ON STACK
MOV   @ (SP),@SEL6   ;SET UP BUFFER COUNT AND BUFFER ADDRESS
      ;BITS 16 & 17
ADD   #2,(SP)         ;INC POINTER ON STACK
WAIT  RQI             ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
      ;**** MACRO EXPANSION ****
      ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
      ;****          ****
20$:
CLR    NESTPC         ;CLEAR THE NEST FLAG
CLR    SUBRPC         ;CLEAR PC
RETURN
    
```

012300 011637 002372
 012304 162737 000004 002372
 012312 117677 000000 167712
 012320 062716 000002
 012324 012737 000001 002374
 012332 004737 010272
 012336 000000

 012340 103003
 012342 062716 000004
 012346 000414
 012350 017677 000000 167660
 012356 062716 000002
 012362 017677 000000 167650

 012370 062716 000002
 012374 004737 010706

 012400 005037 002374
 012404 005037 002372
 012410 000207

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

012412
012412 005737 002374
012416 001005
012420 011637 002372
012424 162737 000004 002372

012432
012432 010046
012434 010146
012436 012700 002637
012442 012701 000006
012446
012446 105720
012450 001016
012452 005301
012454 001374
012456 122010
012460 001012
012462 022737 000022 002114
012470 001403
012472 105710
012474 001004
012476 000407
012500
012500 122710 000001
012504 002004

012506
012506
012506 104457
012510 000005
012512 012536

SUBROUTINE \$ERROR

FUNCTION - TO CHECK THE FIRST 8. BASE TABLE ERROR COUNTS FOR NON-ZERO VALUES.

CALLING FORMAT: JSR PC, \$ERROR

NESTING LEVEL - CAN BE NESTED WITHIN ANOTHER ROUTINE

ENTRY CONDITIONS - SHOULD BE DONE AFTER PROPER SHUTDOWN
NESTPC = 1 - SUBROUTINE NESTED WITHIN ANOTHER SUB.
= 0 - SUBROUTINE NOT NESTED.

EXIT CONDITIONS - IF ANY NON-ZERO VALUE FOUND IN THE BASE TABLE A SOFT ERROR IS DECLARED.

REGISTERS DESTROYED - RESTORED


```

$ERROR:
TST    NESTPC      ;IS THIS ROUTINE NESTED?
BNE    10$         ;BR IF YES (PC ALREADY SAVED)
MOV    (SP),SUBRPC ;SAVE PC AFTER THE CALL TO $WAIT.
SUB    #4,SUBRPC   ;BACKUP TO THE PC OF THE ACTUAL CALL
                           ;THE INSTRUCTION AFTER THE CALL.

10$:
MOV    R0,-(SP)    ;SAVE R0
MOV    R1,-(SP)    ;SAVE R1
MOV    #BASE+3,R0  ;POINTER TO ACTUAL BASE TABLE COUNTS.
MOV    #6.,R1      ;CHECK THE 6 NAK BYTES IN THE TABLE

20$:
TSTB   (R0)+       ;IS THE NAK COUNT NON-ZERO?
BNE    30$         ;IF YES - REPORT SOFT ERROR
DEC    R1          ;LOOP UNTIL DONE.
BNE    20$
CMPB   (R0)+,(R0)  ;ARE THE REPS THE SAME?
BNE    30$         ;IF NOT - REPORT ERROR.
CMP    #18.,L$TEST ;IS THIS TEST 18 (LARGE BUFFER TEST)
BEQ    25$         ;IF YES - ALLOW 1 REP
TSTB   (R0)        ;IF NOT TEST 18 - REPORT IF 'NON ZERO.
BNE    30$
BR     40$         ;IF ZERO - OK.

25$:
CMPB   #1,(R0)     ;IS THE REP 0 OR 1?
BGE    40$         ;IF YES - OK (WE ALLOW 1 REP BECAUSE
                           ;IN TEST 18 AT LOW BAUD RATES 1 REP IS
                           ;EXPECTED.)

30$:
ERRSOFT 5,EMS3,ERRG4 ;REPORT SOFT ERROR
    
```

TRAP C\$ERSOFT
.WORD 5
.WORD EMS3

```

GLOBAL SUBROUTINES

012514 015522
55 012516
56 012516 005737 002374
57 012522 001002
58 012524 005037 002372
59 012530
60 012530 012601
61 012532 012600
62 012534 000207
63
64 012536 105 122 122 EMS3: .ASCIZ /ERRORS IN BASE TABLE/
   012541 117 122 123
   012544 040 111 116
   012547 040 102 101
   012552 123 105 040
   012555 124 101 102
   012560 114 105 000

65 .EVEN
66

```

```

40$: TST NESTPC ;IS THE ROUTINE NESTED?
     BNE 45$ ;BR IF YES
     CLR SUBRPC ;CLEAR SAVED PC

45$: MOV (SP)+,R1 ;RESTORE R1
     MOV (SP)+,R0 ;RESTORE R0
     RETURN

```

.WORD ERRG4

GLOBAL SUBROUTINES

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

24
25

26
27

28
29
30
31

32
33
34
35
36

```

012564
012564 011637 002372
012570 162737 000004 002372
012576 112777 000042 167426
012604 105077 167424
012610 012737 000001 002374
012616 004737 010272
012622 000000
012624 103430
012626 004737 010706
012632 103425
012634 004737 010272
012640 000001
012642 103421
012644 032777 000001 167362
012652 001005
012654 104455
012656 000004
012660 012726
012662 015126
012664 000410
012666 103421
012666 032777 001000 167344
012674 001004
012676 104455
012700 000004
012702 012726
    
```

```

*****
*****
SUBROUTINE $HALT
FUNCTION - TO SHUTDOWN THE DMR11
ENTRY CONDITIONS - NONE
EXIT CONDITIONS - DMR SHUTDOWN
REGISTERS - NO EFFECT
*****
*****
$HALT:
MOV (SP),SUBRPC ;SAVE THE PC WHEN THE SUBROUTINE WAS CALLED.
SUB #4,SUBRPC ;BACK UP TO THE ADDRESS OF THE ACTUAL CALL.
MOVB #RQI!HLT,@SELO ;ISSUE A HALT
CLRB @SEL2 ;CLEAR ANY OUTPUT PENDING
MOV #1,NESTPC ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
WAIT RDI ;WAIT FOR RDI
;**** MACRO EXPANSION ****
JSR PC,$WAIT ;CALL WAIT ROUTINE
.WORD 0 ;FLAG THAT WE'RE WAITING FOR RDI
;****
;****
BERROR 20$ ;IF ERROR, EXIT
;****
WAIT RQI ;CLEAR RQI AND WAIT FOR RDI TO CLEAR
;**** MACRO EXPANSION ****
JSR PC,$CLRQI ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
;****
;****
BERROR 20$ ;IF ERROR, EXIT
;****
WAIT RDO ;WAIT FOR RDO
;**** MACRO EXPANSION ****
JSR PC,$WAIT ;CALL WAIT ROUTINE
.WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
;****
;****
BERROR. 20$ ;IF ERROR, EXIT
;****
BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
BNE 10$ ;IF YES - PROCEED
ERRDF 4,EMS4,ERRG2 ;ERROR
;****
TRAP C$ERDF
.WORD 4
.WORD EMS4
.WORD ERRG2
BR 20$
10$:
BIT #HALTC,@SEL6 ;IS THE DMR HALTED?
BNE 20$ ;IF YES - EXIT
ERRDF 4,EMS4,ERRG2 ;ERROR - NOT EXPECTED CONTROL OUT.
;****
TRAP C$ERDF
.WORD 4
.WORD EMS4
    
```

.WORD ERRG2

```
012704 015126
37 012706
38 012706 042777 000207 167320 20$: BIC #RDO.CMD,@SEL2 ;CLEAR RDO AND COMMAND BITS.
39 012714 005037 002374 CLR NESTPC ;CLEAR THE NEST FLAG
40 012720 005037 002372 CLR SUBRPC ;CLEAR THE PC.
41 012724 000207 RETURN
42
43 012726 123 110 125 EMS4: .ASCIZ /SHUTDOWN ERROR/
012731 124 104 117
012734 127 116 040
012737 105 122 122
012742 117 122 000
44 .EVEN
```

GLOBAL SUBROUTINES

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

SUBROUTINE \$ROMO

FUNCTION - TO READ THE CONTENTS OF THE ROM

ENTRY CONDITIONS - ROMADR = ROM ADDRESS

EXIT CONDITIONS - BSEL6 = CONTENTS OF ROM ADDRESS

REGISTERS - NO EFFECT

\$ROMO:

```

CLR      @SELO      ;INIT
MOVB    ROMADR+1,@SEL2 ;SET HIGH BYTE OF ROM ADDRESS
BIS     #ROMI,@SELO ;ENABLE SEL6 TO BE USED AS MAINTENANCE REG.
MOV     #121053,@SEL6 ;SET UP MICROINSTRUCTION TO
                        ;MOVE IBUS* 2 TO OBUS* 13
                        ;(OBUS* 13 IS A SHADOW REGISTER FOR
                        ;BITS 8-11 OF THE PC)
BIS     #STUP,@SELO ;CLOCK THE INSTRUCTION
BIC     #ROMI!STUP,@SELO ;CLEAR
BIC     #377,1$     ;CLEAR ADDRESS FIELD OF BRANCH INST.
BISB   ROMADR,1$   ;ADD ADDRESS OF BRANCH.
BIS     #ROMI,@SELO ;ENABLE SEL6
MOV     1$,@SEL6   ;SET UP MICROINSTRUCTION TO
                        ;BRANCH IMMEDIATELY TO PC. BRANCH IS
                        ;NECESSARY TO TRANSFER PC SHADOW REG TO PC
BIS     #STUP,@SELO ;CLOCK THE INSTRUCTION
                        ;ROM PC = ROM ADDRESS
BIC     #ROMI!STUP,@SELO ;CLEAR
BIS     #ROMO,@SELO ;CLOCK IN A MAINTENANCE ROM OUT
                        ;ROM CONTENTS ARE NOW IN SEL6.
RETURN
1$:     .WORD      100000 ;MICRO INSTRUCTION OP CODE FOR IMMEDIATE
                        ;BRANCH (ROM ADDRESS IS ADDED INTO BITS 0-7)
    
```

```

17 012746
18 012746 005077 167260
19 012752 113777 002411 167254
20 012760 052777 001000 167244
21 012766 012777 121053 167244
25 012774 052777 000400 167230
26 013002 042777 001400 167222
27 013010 042737 000377 013064
28 013016 153737 002410 013064
29 013024 052777 001000 167200
30 013032 013777 013064 167200
33 013040 052777 000400 167164
35 013046 042777 001400 167156
36 013054 052777 002000 167150
38 013062 000207
40 013064 100000
    
```

GLOBAL SUBROUTINES

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

*****
*****
SUBROUTINE $LOOP
FUNCTION - TO ISSUE AN EXTENDED CONTROL IN TO SET
UP THE MODEM LOOPBACK DESIRED BY THE USER.

ENTRY CONDITIONS - WMAINT = 0 - DON'T WRITE MAINT. BITS
                   WMAINT = 1 - SET BITS
                   (WMAINT SET IN INIT CODE)
                   DMCMD = 0 - DMR MODE
                   DMTURN = TURN AROUND CONNECTOR

EXIT CONDITIONS -

REGISTERS - NOT DESTROYED

*****
*****
$LOOP:
TST    DMCMD     ;IS THE DMR IN DMC MODE?
BNE    30$      ;IF SO, EXIT (CAN'T DO DMR MODE INPUT)
TST    WMAINT   ;DO WE NEED TO WRITE THE MAINTENANCE BITS?
BEQ    30$      ;IF NOT - EXIT.
MOV    (SP),SUBRPC ;SAVE THE PC AFTER THE CALL TO $LOOP
SUB    #4,SUBRPC ;BACKUP TO THE PC OF THE ACTUAL CALL.
CMP    #LLOOP,DMTURN ;IS LOCAL MODEM LOOPBACK DESIRED?
BNE    10$      ;IF NOT - PROCEED.
MOV    #MAINT2,100$ ;ENSURE REMOTE LOOPBACK IS CLEAR.
MOV    #MAINT1,101$ ;SET MAINT BIT FOR LOCAL LOOPBACK
BR     20$

10$:
;IN ALL OTHER LOOPBACK CONFIGURATIONS
;SET MAINTENANCE 2 (CONFIG. TYPE 1,3,7)
;ENSURE REMOTE LOOPBACK IS CLEAR.
;SET MAINT BIT FOR REMOTE LOOPBACK
MOV    #MAINT1,100$
MOV    #MAINT2,101$

20$:
MOV    #1,NESTPC ;FLAG THAT THE NEXT SUBROUTINE IS NESTED.
CALL   $DMRIN   ;DMR MODE INPUT COMMAND
        .WORD   WMODEM ;WRITE MODEM COMMAND
100$:   .WORD   0      ;BITS TO CLEAR IN MODEM REGISTER
101$:   .WORD   0      ;BITS TO SET IN MODEM REGISTER

30$:
CLR    NESTPC   ;CLEAR THE NEST FLAG
CLR    SUBRPC   ;CLEAR PC.
RETURN

```

```

013066
013066 005737 002276
013072 001041
013074 005737 002306
013100 001436
013102 011637 002372
013106 162737 000004 002372
013114 022737 000006 002254
013122 001007
013124 012737 000004 013172
013132 012737 000010 013174
013140 000406
013142
013142 012737 000010 013172
013150 012737 000004 013174
013156
013156 012737 000001 002374
013164
013170 000005
013172 000000
013174 000000
013176
013176 005037 002374
013202 005037 002372
013206 000207

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

SUBROUTINE \$BUFFS

FUNCTION - TO DETERMINE BUFFERS FOR TEST 15 - 19. THIS
 SUBROUTINE WILL USE ONE OF THE FOLLOWING
 THREE BUFFER AREAS:
 1. IF MEMORY MANAGED, 32K - 48K
 2. FREE MEMORY, IF MORE THAN 4K BYTES.
 3. IF 2 OR 3 NOT POSSIBLE, DEFAULT 4K
 DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.

CALL - JSR PC,\$BUFFS
 NESTING LEVEL - CALLED ONLY BY TESTS 16-20
 ENTRY CONDITIONS - BUFNUM = # OF RCV & XMIT BUFFERS
 EXIT CONDITIONS - MMANAG = 1 MEMORY MANAGEMENT USED
 MMANAG = 0 MEMORY MANAGEMENT NOT USED
 RCVBUF = ADDRESS OF RECEIVE BUFFER (VIRTUAL)
 RCVBUF+2 = CHARACTER COUNT
 RCVBUF+4 = ADDRESS OF NEXT RECEIVE BUFFER
 (UP TO 64 ADDRESSES AND COUNTS)
 XMTBUF = ADDRESS OF TRANSMIT BUFFER (VIRTUAL)
 (UP TO 64 ADDRESSES AND COUNTS)
 REGISTERS - NOT DESTROYED


```

$BUFFS:
MOV    (SP),SUBRPC      ;SAVE PC AFTER THE CALL TO $BUFFS
SUB    #4,SUBRPC        ;BACKUP TO THE PC OF THE CALL.
CLR    NXMFLG
SETVEC #4,#NOXMEM,#PRI07 ;SET UP TRAP 4 (WILL SET FLAG FOR NXM)
                                MOV    #PRI07,-(SP)
                                MOV    #NOXMEM,-(SP)
                                MOV    #4,-(SP)
                                MOV    #3,-(SP)
                                TRAP   C$SVEC
                                ADD    #10,SP

TST    @#177572         ;ADDRESS MEMORY MANAGEMENT REG
TST    NXMFLG           ;IS THE FLAG STILL CLEARED?
                                ;NOTE: THE FLAG WILL BE SET BY THE TRAP
                                ;IF THERE IS NO MEMORY MANAGEMENT.
BNE    30$              ;BR TO USE NON-MEMORY MANAG. BUFFERS.
CMP    L$HIMEM,#3000    ;IS THERE AT LEAST 48K WORDS? (16K WORDS
                                ;FOR BUFFERS)
BLT    30$              ;IF NOT, USE NON-MEMORY MANAG. BUFFERS.
MOV    #1,MMANAG        ;FLAG THAT MEMORY MANAGEMENT IS USED
SETPRI #PRI07           ;MAKE SURE WE ARE IN KERNEL MODE.
                                MOV    #PRI07,R0
                                TRAP   C$SPRI

;SETTING PRI SHOULD SHOULD ALSO CLEAR
    
```

```

013210
013210 011637 002372
013214 162737 000004 002372
013222 005037 002350
013226
013226 012746 000340
013232 012746 023714
013236 012746 000004
013242 012746 000003
013246 104437
013250 062706 000010
013254 005737 177572
013260 005737 002350
013264 001143
013266 023727 002120 003000
013274 002537
013276 012737 000001 002302
013304
013304 012700 000340
013310 104441
    
```


107	013566	000511		BR	60\$	
108						
109	013570		29\$:			
110	013570	005037	177572	CLR	@177572	:TURN OFF MEMORY MANAGEMENT
111	013574		30\$:			
112	013574	005037	002302	CLR	MMANAG	:FLAG THAT MEMORY MANAGEMENT NOT USED.
113	013600			CLRVEC	#4	:RESTORE TRAP 4.
	013600	012700	000004			
	013604	104436				
114	013606			MEMORY	R2	:FIND THE FREE MEMORY AVAILABLE BETWEEN
	013606	104431				
	013610	010002				
115						
116	013612	021227	002000	CMP	@R2,#2000	:THE DIAGNOSTIC AND THE DRS (SUPERVISOR).
117						
118						
119	013616	003406		BLE	35\$:IS THERE AT LEAST 1K WORDS? (NOTE: CONTENTS
120	013620	010237	003234	MOV	R2,XMTBUF	:OF THE RETURNED ADDRESS OF THE START OF FREE
121	013624	011200		MOV	@R2,R0	:MEMORY CONTAIN THE AMOUNT OF AVAILABLE MEM.)
122	013626	042700	000001	BIC	#BIT0,R0	:IF NOT AT LEAST 1K, USE DEFAULT BUFFER.
123	013632	000405		BR	40\$:USE THE FREE MEMORY BUFFER.
124	013634					:SAVE THE WORD SIZE OF THE BUFFER.
125	013634	012737	004234	MOV	#BIGBUF,XMTBUF	:START WITH AN EVEN # OF WORDS.
126	013642	012700	002000	MOV	#2000,R0	:USE THE DEFAULT BUFFER (1ST HALF FOR XMIT).
127	013646					:1K WORD SIZE.
128	013646	013737	003234	MOV	XMTBUF,RCVBUF	:CALCULATE THE RECEIVE BUFFER ADDRESS
129	013654	060037	003634	ADD	R0,RCVBUF	:AS STARTING IN THE 2ND HALF OF THE BUFFER.
130	013660	010001		MOV	R0,R1	:BUFFER SIZE IN WORDS.
131	013662	022737	000001	CMP	#1,BUFNUM	:ARE WE SETTING UP 1 RECEIVE AND XMIT BUFFER?
132	013670	001415		BEQ	47\$:IF YES - R1 = BYTE SIZE FOR BOTH BUFFERS.
133	013672	022737	000007	CMP	#7,BUFNUM	:ARE WE SETTING UP 7 RCV & 7 XMIT BUFFERS?
134	013700	001004		BNE	45\$:IF NOT WE MUST NEED 64 RCV & 64 XMIT BUFFERS.
135	013702	006201		ASR	R1	:R1 = # BYTES IN THE BUFFERS/8
136	013704	006201		ASR	R1	
137	013706	006201		ASR	R1	
138	013710	000405		BR	47\$	
139	013712		45\$:			
140	013712	012704	000007	MOV	#7,R4	:DIVIDE BYTES BY 128.
141	013716		46\$:			
142	013716	006201		ASR	R1	:SHIFT RIGHT 7 TIMES
143	013720	005304		DEC	R4	
144	013722	001375		BNE	46\$	
145	013724		47\$:			
146	013724	010137	002322	MOV	R1,BUFSIZ	:SAVE THE BUFFER SIZE IN BYTES.
147	013730	162737	000002	SUB	#2,BUFSIZ	:ADJUST BUFFER SIZE BECAUSE WE
148						:WILL ADJUST BUFFER STARTING ADDRESS.
149	013736	042737	000001	BIC	#1,BUFSIZ	:ENSURE WE START WITH AN EVEN # OF BYTES.
150	013744	006200		ASR	R0	:# OF WORDS IN ALL XMIT BUFFERS.
151	013746	010001		MOV	R0,R1	:SAVE # OF WORDS IN ALL RCV BUFFERS.
152	013750	013702	003234	MOV	XMTBUF,R2	:ADDRESS OF START OF XMIT BUFFERS.
153	013754		50\$:			
154	013754	012703	002414	MOV	#SCCITT,R3	:ADDRESS OF TEST PATTERN
155	013760	012704	000040	MOV	#32.,R4	:# OF WORDS IN THE TEST PATTERN.
156	013764		51\$:			
157	013764	012312		MOV	(R3)+,(R2)	:WRITE TEST PATTERN INTO ALL XMIT BUFFERS.
158	013766	005300		DEC	R0	:ARE ALL THE XMIT BUFFERS WRITTEN?
159	013770	001403		BEQ	55\$:IF YES PROCEED.

GLOBAL SUBROUTINES

160	013772	005304		DEC	R4		:CONTINUE WITH TEST PATTERN TILL DONE.
161	013774	001373		BNE	51\$		
162	013776	000766		BR	50\$:START AT BEGINNING OF TEST PATTERN.
163	014000		55\$:				
164	014000	013702	003634	MOV	RCVBUF,R2		:ADDRESS OF PECEIVE BUFFERS
165	014004		56\$:				
166	014004	005022		CLR	(R2)+		:CLEAR ALL RECEIVE BUFFERS.
167	014006	005301		DEC	R1		
168	014010	001375		BNE	56\$		
169							
170							
171	014012		60\$:				
172	014012	013700	003634	MOV	RCVBUF,R0		:ADDRESS OF RECEIVE BUFFER
173	014016	012701	003634	MOV	#RCVBUF,R1		:TABLE ADDRESS OF RCV BUFFER POINTERS.
174	014022	013702	002324	MOV	BUFNUM,R2		:# OF RCV. BUFFERS.
175	014026		65\$:				
176	014026	010021		MOV	R0,(R1)+		:SAVE THE RECEIVE BUFFER ADDRESS
177	014030	013721	002322	MOV	BUFSIZ,(R1)+		:SAVE THE BUFFER SIZE
178	014034	063700	002322	ADD	BUFSIZ,R0		:CALCULATE THE NEXT BUFFER ADDRESS.
179	014040	005200		INC	R0		:CHANGE EVEN ADDRESS TO ODD & ODD TO EVEN.
180	014042	005302		DEC	R2		:CALCULATE ALL THE BUFFER ADDRESSES.
181	014044	001370		BNE	65\$		
182							
183	014046	013700	003234	MOV	XMTBUF,R0		:ADDRESS OF TRANSMIT BUFFERS
184	014052	012701	003234	MOV	#XMTBUF,R1		:TABLE OF XMIT BUFFER POINIERS.
185	014056	013702	002324	MOV	BUFNUM,R2		:#OF XMIT BUFFERS.
186	014062	012703	000004	MOV	#4,R3		:R3 IS USED TO VARY THE CHARACTER COUNT.
187	014066		70\$:				
188	014066	010021		MOV	R0,(R1)+		:SAVE THE XMIT BUFFER ADDRESS.
189	014070	013711	002322	MOV	BUFSIZ,(R1)		:SAVE THE BUFFER SIZE.
190	014074	160321		SUB	R3,(R1)+		:VARY THE BUFFER SIZE
191	014076	063700	002322	ADD	BUFSIZ,R0		:CALCULATE THE NEXT BUFFER ADDRESS
192	014102	005303		DEC	R3		:CHANGE THE CHARACTER COUNT VARIABLE.
193	014104	032703	000001	BIT	#BIT0,R3		:IS THE CONTENTS OF R3 ODD
194	014110	001001		BNE	72\$:IF YES, DON'T ADJUST BUFFER ADDRESS.
195	014112	005200		INC	R0		:CHANGE EVEN TO ODD ETC.
196	014114		72\$:				
197	014114	005703		TST	R3		:WHAT IS R3.
198	014116	002002		BGE	75\$:CONTINUE UNTIL R3 = -1
199	014120	012703	000004	MOV	#4,R3		:RE-INIT. THE R3 VARIABLE AGAIN.
200	014124		75\$:				
201	014124	005302		DEC	R2		:CALCULATE ALL THE XMIT BUFFERS.
202	014126	001357		BNE	70\$		
203							
204	014130	005037	002350	CLR	NXMFLG		:RESTORE FLAG USED IN TRAP VECTOR.
205	014134	005037	002372	CLR	SUBRPC		:CLEAR PC.
206	014140	000207		RETURN			

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

SUBROUTINE \$INOUT

FUNCTION - TO MANAGE THE INTERRUPT FROM BASE IN
TO BA/CC OUT IN THE INTERRUPT TESTS 15-19

ENTRY CONDITIONS - BUFNUM = # OF RCV AND XMIT BUFFERS
ALL BUFFERS SET UP IN THE \$BUFFS SUBROUTINE.
WAIT3 = # OF OUTER LOOP TIMEOUT COUNTERS.
THIS VALUE IS DETERMINED BY THE BAUD
RATE IN THE INIT. SECTION OF CODE.

EXIT CONDITIONS -

REGISTERS - R0 - R5 DESTROYED

\$INOUT:

```

MOV      (SP),SUBRPC      ;SAVE THE PC AFTER THE CALL TO $LOOP
SUB      #4,SUBRPC        ;BACKUP TO THE PC OF THE ACTUAL CALL.
MOV      #1,NESTPC        ;FLAG THAT ANY SUBROUTINE USED WILL BE NESTED.
MOV      BUFNUM,INRCV     ;# OF BA/CC IN RECEIVES
MOV      BUFNUM,INXMIT    ;# OF BA/CC IN TRANSMITS
MOV      BUFNUM,OUTRCV    ;# OF BA/CC OUT RECEIVES
MOV      BUFNUM,OUTXMT    ;# OF BA/CC OUT TRANSMITS
CLR      INFLAG           ;CLEAR INPUT BA/CC FLAG
CLR      OUTFLG           ;CLEAR OUTPUT BA/CC FLAG
CLR      START            ;CLEAR FLAG TO SHOW START UP NOT DONE (SET
                          ;AFTER CONTROL IN)
MOV      #RCVBUF,R2       ;ADDR OF RCV. BUFFER TABLE (FOR INPUT)
MOV      #XMTBUF,R3       ;ADDR OF XMIT BUFFER TABLE (FOR INPUT)
MOV      #RCVBUF,R4       ;ADDR OF RCV. BUFFER TABLE (OUTPUT CHECKING)
MOV      #XMTBUF,R5       ;ADDR OF XMIT BUFFER TABLE (OUTPUT CHECKING)
SETPRI   #PRI04           ;SET THE PRIORITY TO LEVEL 4 TO ALLOW THE
                          ;MOV      #PRI04,R0
                          ;TRAP   C$SPRI

MOV      WAIT3,WAIT4      ;DMR TO INTERRUPT AT LEVEL 5
MOV      #IFSET!RQI!BASEI, @BSELO ;TIMEOUT COUNTER DETERMINED BY BAUD RATE.
                          ;FIRST COMMAND - BASE IN.

8$:      MOV      #1000,R1 ;INNER LOOP COUNTER

10$:     BREAK            ;OPERATOR INTERRUPT ENABLE. CALL TO
                          ;TRAP   C$BRK

46:     ;THE SUPERVISOR TO ALLOW CONSOLE INTERRUPT
47:     ;(NOTE: INFLAG AND OUTFLG SET IN THE INTERRUPT
48:     ;SERVICE ROUTINES)
49:     TST      INFLAG    ;ARE THE INPUTS DONE? (INISR DONE?)
50:     BEQ     12$        ;IF NOT KEEP CHECKING.
51:     TST      OUTFLG    ;ARE THE OUTPUTS DONE? (OUTISR DONE?)
52:     BNE     20$        ;IF YES EXIT WAIT LOOP.

12$:    DELAY    1         ;WAIT 100 MICROSECONDS.
    
```

```

014142 011637 002372
014142 162737 000004 002372
014146 012737 000001 002374
014154 013737 002324 002326
014162 013737 002324 002330
014170 013737 002324 002332
014176 013737 002324 002334
014204 013737 002324
014212 005037 002352
014216 005037 002354
014222 005037 002272
014226 012702 003634
014232 012703 003234
014236 012704 003634
014242 012705 003234
014246 012700 000200
014246 104441
014254 013737 002316 002320
014262 112777 000143 165742
014270 012701 001000
014274 104422
014276 005737 002352
014302 001403
014304 005737 002354
014310 001026
014312
014312
    
```

```

GLOBAL SUBROUTINES

014312 012727 000001
014316 000000
014320 013727 002116
014324 000000
014326 005367 177772
014332 001375
014334 005367 177756
014340 001367
55 014342 005301
56 014344 001353
57 014346 005337 002320
58 014352 001346
59 014354
014354 104455
014356 000002
014360 020063
014362 014620

60
61
62 014364 000453
63 014366
64
65 014366 012700 003634
66 014372 012701 003234
67 014376 013702 002324
68 014402 005737 002302
69 014406 001403
70 014410 012737 000001 177572
71 014416
72 014416 012003
73 014420 012104
74 014422 011005
75 014424 022021
76 014426 001412
77 014430 005737 002302
78 014434 001402
79 014436 005037 177572
80 014442
81 014442
014442 104455
014444 000014
014446 020277
014450 016216
82 014452 000420
83 014454
84 014454 122324
85 014456 001005
86 014460 005305
87 014462 001374
88 014464 005302
89 014466 001353
90 014470 000411
91 014472
92 014472 005737 002302
93 014476 001402
94 014500 005037 177572
95 014504

;CONTINUE IN LOOP UNTIL R1 = 0.
;DECREMENT OUTER LOOP COUNTER
;IF NOT DONE - GO THROUGH INNER LOOP AGAIN.
;TIMEOUT MESSAGE.
;ALSO PRINT # OF BUFFERS NOT COMPLETE.
;EXIT
;RECEIVE BUFFER POINTER TABLE ADDRESS.
;TRANSMIT BUFFERS
;# OF RCV. AND XMIT BUFFERS.
;ARE THE BUFFERS MEMORY MANAGED?
;IF YES - PROCEED.
;TURN ON MEMORY MANAGEMENT
;ADDRESS OF A RECEIVE BUFFER.
;ADDRESS OF A TRANSMIT BUFFER.
;CHARACTER COUNT.
;ARE THE CHARACTER COUNTS THE SAME?
;IF YES - PROCEED.
;IS MEMORY MANAGEMENT TURNED ON?
;IF NOT - SKIP TURN OFF.
;TURN OFF MEMORY MANAGEMENT.
;EXIT
;ARE THE CHARACTERS THE SAME?
;IF NOT - ERROR EXIT
;CHECK ALL THE CHARACTERS
;CHECK ALL THE BUFFERS.
;IS MEMORY MANAGEMENT TURNED ON?
;IF NOT - SKIP TURN OFF.
;TURN OFF MEMORY MANAGEMENT.

MOV #1,(PC)+
.WORD 0
MOV L$DLY,(PC)+
.WORD 0
DEC -6(PC)
BNE -.4
DEC -22(PC)
BNE -.20

DEC R1
BNE 10$
DEC WAIT4
BNE 8$
ERRDF 2,EMG2,ERRG1

TRAP C$ERDF
.WORD 2
.WORD EMG2
.WORD ERRG1

BR 60$
20$:
MOV #RCVBUF,R0
MOV #XMTBUF,R1
MOV BUFNUM,R2
TST MMANAG
BEQ 40$
MOV #1,@#177572
40$:
MOV (R0)+,R3
MOV (R1)+,R4
MOV @R0,R5
CMP (R0)+,(R1)+
BEQ 45$
TST MMANAG
BEQ 41$
CLR @#177572
41$:
ERRDF 12,EMG12,ERRG10

TRAP C$ERDF
.WORD 12
.WORD EMG12
.WORD ERRG10

BR 60$
45$:
CMPB (R3)+,(R4)+
BNE 50$
DEC R5
BNE 45$
DEC R2
BNE 40$
BR 60$
50$:
TST MMANAG
BEQ 51$
CLR @#177572
51$:

```

```

96 014504          ERRDF  15,EMG15,ERRG12
   014504 104455
   014506 000017
   014510 020377
   014512 016302
97 014514          60$:
98 014514 005737 002302    TST  MMANAG      ;IS MEMORY MANAGEMENT TURNED ON?
99 014520 001402          BEQ  61$          ;IF NOT - SKIP TURN OFF.
100 014522 005037 177572    CLR  @#177572    ;TURN OFF MEMORY MANAGEMENT.
101 014526          61$:
102 014526 042777 000120 165476 BIC  #IESET!IECLR,@SELO ;DISABLE BOTH INPUT INTERRUPTS
103 014534 042777 000100 165472 BIC  #IEO,@SEL2    ;DISABLE OUTPUT INTERRUPT
104 014542 022737 000021 002114 CMP  #17.,L$TEST  ;IS THIS TEST 17, 18 OR 19 ?
105 014550 003011          BGT  62$          ;IF NOT - SHUTDOWN.
106
107
108
109
110
111 014552          DMRIN  UPDATE
   014552 004737 012070    JSR  PC, $DMRIN  ;DO A DMR UPDATE.
   014556 000011          ;**** MACRO EXPANSION ****
   014560 000000          ;CALL DMR MODE INPUT ROUTINE
   014562 000000          ;INPUT COMMAND
   ;NO SEL4
   ;NO SEL6
   ;****
112 014564          WAIT  RDO
   014564 004737 010272    JSR  PC, $WAIT   ;WAIT FOR RDO
   014570 000001          ;**** MACRO EXPANSION ****
   ;CALL WAIT ROUTINE
   ;FLAG THAT WE'RE WAITING FOR RDO
   ;****
113 014572 000402          BR   63$
114 014574          62$:
115 014574          SHUTDN
   014574 004737 012564    JSR  PC, $HALT   ;SHUT DOWN THE DMR
   ;**** MACRO EXPANSION ****
   ;DMR HALT ROUTINE.
   ;****
116 014600          63$:
117 014600          SETPRI #PRI07      ;RETURN PROCESSOR PRIORITY TO 7
   014600 012700 000340    MOV  #PRI07,R0
   014604 104441          TRAP C$SPRI
118 014606 005037 002374    CLR  NESTPC      ;CLEAR NESTED FLAG.
119 014612 005037 002372    CLR  SUBRPC      ;CLEAR PC.
120 014616 000207          RETURN
121
122
123
124
125
126
127
128
    
```

```

1      .SBTTL GLOBAL ERROR REPORT REPORT SECTION
2      :////////////////////////////////////////////////////////////////////
3      :// THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
4      :// THAT ARE USED IN MORE THAN ONE TEST.
5      :////////////////////////////////////////////////////////////////////
6      .EVEN
7
8 014620 BGNMSG ERRG1
9 014620 PRINTB #FMG3,SUBRPC ;PC THAT SUBROUTINE WAS CALLED. ERRG1::
014620 013746 002372 MOV SUBRPC,-(SP)
014624 012746 016420 MOV #FMG3,-(SP)
014630 C12746 000002 MOV #2,-(SP)
014634 010600 MOV SP,R0
014636 104414 TRAP C$PNTB
014640 062706 000006 ADD #6,SP
10 014644 PRINTB #FMG1,@SEL0,@SEL2 ;PRINT SEL0 AND SEL2 CONTENTS.
014644 017746 165364 MOV @SEL2,-(SP)
014650 017746 165356 MOV @SEL0,-(SP)
014654 012746 016334 MOV #FMG1,-(SP)
014660 012746 000003 MOV #3,-(SP)
014664 010600 MOV SP,R0
014666 104414 TRAP C$PNTB
014670 062706 000010 ADD #10,SP
11 014674 PRINTB #FMG2,@SEL4,@SEL6 ;PRINT SEL4 AND SEL2 CONIENTS.
014674 017746 165340 MOV @SEL6,-(SP)
014700 017746 165332 MOV @SEL4,-(SP)
014704 012746 016366 MOV #FMG2,-(SP)
014710 012746 000003 MOV #3,-(SP)
014714 010600 MOV SP,R0
014716 104414 TRAP C$PNTB
014720 062706 000010 ADD #10,SP
12 014724 PRINTB #FMG21,BUFNUM ;# OF BUFFERS
014724 013746 002324 MOV BUFNUM,-(SP)
014730 012746 017603 MOV #FMG21,-(SP)
014734 012746 000002 MOV #2,-(SP)
014740 010600 MOV SP,R0
014742 104414 TRAP C$PNTB
014744 062706 000006 ADD #6,SP
13 014750 PRINTB #FMG22,BUFSIZ ;BUFFER SIZE
014750 013746 002322 MOV BUFSIZ,-(SP)
014754 012746 017663 MOV #FMG22,-(SP)
014760 012746 000002 MOV #2,-(SP)
014764 010600 MOV SP,R0
014766 104414 TRAP C$PNTB
014770 062706 000006 ADD #6,SP
14 014774 NEG INRCV ;NEGATE BUFFER VALUES
15 015000 NEG INXMIT
16 015004 NEG OUTRCV
17 015010 NEG OUTXMT
18 015014 ADD BUFNUM,INRCV ;CALCULATE BUFFERS ASSIGNED.
19 015022 ADD BUFNUM,INXMIT
20 015030 ADD BUFNUM,OUTRCV ;CALCULATE BUFFERS RECEIVED.
21 015036 ADD BUFNUM,OUTXMT
22 015044 PRINTB #FMG23,INRCV,INXMIT
015044 013746 002330 MOV INXMIT,-(SP)
015050 013746 002326 MOV INRCV,-(SP)

```

015054	012746	017710							
015060	012746	000003						MOV	#FMG23,-(SP)
015064	010600							MOV	#3,-(SP)
015066	104414							MOV	SP,R0
015070	062706	000010						TRAP	C\$PNTB
23 015074			PRINTB	#FMG24,OUTRCV,OUTXMT				ADD	#10,SP
015074	013746	002334						MOV	OUTXMT,-(SP)
015100	013746	002332						MOV	OUTRCV,-(SP)
015104	012746	017771						MOV	#FMG24,-(SP)
015110	012746	000003						MOV	#3,-(SP)
015114	010600							MOV	SP,R0
015116	104414							TRAP	C\$PNTB
015120	062706	000010						ADD	#10,SP
24 015124			ENDMSG						
015124									
015124	104423							L10002:	TRAP C\$MSG
25									
26									
27 015126			BGNMSG	ERRG2					
015126									
28 015126	005737	002372							
29 015132	001412		TST	SUBRPC		ERRG2::			
30 015134			BEQ	10\$:IS THE ERROR IN A SUBROUTINE?			
015134			PRINTB	#FMG3,SUBRPC		:IF NOT, DON'T PRINT SUBR. PC			
						:PC THAT SUBROUTINE WAS CALLED.			
015134	013746	002372						MOV	SUBRPC,-(SP)
015140	012746	016420						MOV	#FMG3,-(SP)
015144	012746	000002						MOV	#2,-(SP)
015150	010600							MOV	SP,R0
015152	104414							TRAP	C\$PNTB
015154	062706	000006						ADD	#6,SP
31 015160			10\$:						
32 015160			PRINTB	#FMG1,@SEL0,@SEL2 ;PRINT SEL0 AND SEL2 CONTENTS.					
015160	017746	165050						MOV	@SEL2,-(SP)
015164	017746	165042						MOV	@SEL0,-(SP)
015170	012746	016334						MOV	#FMG1,-(SP)
015174	012746	000003						MOV	#3,-(SP)
015200	010600							MOV	SP,R0
015202	104414							TRAP	C\$PNTB
015204	062706	000010						ADD	#10,SP
33 015210			PRINTB	#FMG2,@SEL4,@SEL6 ;PRINT SEL4 AND SEL2 CONTENTS.					
015210	017746	165024						MOV	@SEL6,-(SP)
015214	017746	165016						MOV	@SEL4,-(SP)
015220	012746	016366						MOV	#FMG2,-(SP)
015224	012746	000003						MOV	#3,-(SP)
015230	010600							MOV	SP,R0
015232	104414							TRAP	C\$PNTB
015234	062706	000010						ADD	#10,SP
34 015240			ENDMSG						
015240									
015240	104423							L10003:	TRAP C\$MSG
35									
36 015242			BGNMSG	ERRG3					
015242									
37 015242	005737	002372							
38 015246	001412		TST	SUBRPC		ERRG3::			
39 015250			BEQ	10\$:IS THE ERROR IN A SUBROUTINE?			
015250			PRINTB	#FMG3,SUBRPC		:IF NOT, DON'T PRINT SUBR. PC			
						:PC THAT SUBROUTINE WAS CALLED.			
015250	013746	002372						MOV	SUBRPC,-(SP)
015254	012746	016420						MOV	#FMG3,-(SP)

	015260	012746	000002						MOV #2,-(SP)
	015264	010600							MOV SP,R0
	015266	104414							TRAP C\$PNTB
	015270	062706	000006						ADD #6,SP
40	015274			10\$:					
41	015274				PRINTB #FMG1,@SELO,@SEL2 ;PRINT SELO AND SEL2 CONTENTS.				
	015274	017746	164734						MOV @SEL2,-(SP)
	015300	017746	164726						MOV @SELO,-(SP)
	015304	012746	016334						MOV #FMG1,-(SP)
	015310	012746	000003						MOV #3,-(SP)
	015314	010600							MOV SP,R0
	015316	104414							TRAP C\$PNTB
	015320	062706	000010						ADD #10,SP
42	015324	032777	100000	164700	BIT #RUN,@SELO ;IS THE RUN BIT SET				
43	015332	001043			BNE 20\$;IF RUN SET, CHECK.				
44	015334	122777	000001	164702	CMPB #1,@SEL3 ;DID CF MICRO. FAIL?				
45	015342	001011			BNE 12\$;IF NOT SEE IF LU FAILED.				
46	015344				PRINTB #FMG4 ;CPU MICRO. FAILED.				
	015344	012746	016472						MOV #FMG4,-(SP)
	015350	012746	000001						MOV #1,-(SP)
	015354	010600							MOV SP,R0
	015356	104414							TRAP C\$PNTB
	015360	062706	000004						ADD #4,SP
47	015364	000455			BR 30\$				
48	015366			12\$:					
49	015366	122777	000002	164650	CMPB #2,@SEL3 ;DID LINE UNIT MICRO. FAIL?				
50	015374	001011			BNE 15\$				
51	015376				PRINTB #FMG5 ;LINE UNIT FAILED.				
	015376	012746	016523						MOV #FMG5,-(SP)
	015402	012746	000001						MOV #1,-(SP)
	015406	010600							MOV SP,R0
	015410	104414							TRAP C\$PNTB
	015412	062706	000004						ADD #4,SP
52	015416	000440			BR 30\$				
53	015420			15\$:					
54	015420				PRINTB #FMG5 ;NO RUN - MASTER CLEAR FAILED.				
	015420	012746	016523						MOV #FMG5,-(SP)
	015424	012746	000001						MOV #1,-(SP)
	015430	010600							MOV SP,R0
	015432	104414							TRAP C\$PNTB
	015434	062706	000004						ADD #4,SP
55	015440	000427			BR 30\$				
56	015442			20\$:					
57	015442	105777	164576		TSTB @SEL3 ;IS BSEL3 STILL 0?				
58	015446	001010			BNE 25\$;IF NOT - SEE IF MICRODIAG. RUN.				
59	015450				PRINTB #FMG19 ;DEVICE IS NOT DMR (DMC?)				
	015450	012746	017531						MOV #FMG19,-(SP)
	015454	012746	000001						MOV #1,-(SP)
	015460	010600							MOV SP,R0
	015462	104414							TRAP C\$PNTB
	015464	062706	000004						ADD #4,SP
60	015470			25\$:					
61	015470	122777	000100	164546	CMPB #100,@SEL3 ;WERE MICRODIAGNOSTICS DISABLE?				
62	015476	001010			BNE 30\$				
63	015500				PRINTB #FMG20 ;MICRODIAGNOSTICS DISABLED				
	015500	012746	017554						MOV #FMG20,-(SP)
	015504	012746	000001						MOV #1,-(SP)

91	015710			15\$:			
92	015710				PRINTB	#FMG9,<B,BASE+4>,<B,BASE+7>	
	015710	005046					CLR -(SP)
	015712	153716	002643				BISB BASE+7,(SP)
	015716	005046					CLR -(SP)
	015720	153716	002640				BISB BASE+4,(SP)
	015724	012746	016737				MOV #FMG9,-(SP)
	015730	012746	000003				MOV #3,-(SP)
	015734	010600					MOV SP,R0
	015736	104414					TRAP C\$PNTB
	015740	062706	000010				ADD #10,SP
93	015744			16\$:			
94	015744	105737	002646		TSTB	BASE+10.	
95	015750	001003			BNE	17\$	
96	015752	105737	002645		TSTB	BASE+9.	
97	015756	001416			BEQ	20\$	
98	015760			17\$:			
99	015760				PRINTB	#FMG10,<B,BASE+10.>,<B,BASE+9.>	
	015760	005046					CLR -(SP)
	015762	153716	002645				BISB BASE+9.,(SP)
	015766	005046					CLR -(SP)
	015770	153716	002646				BISB BASE+10.,(SP)
	015774	012746	017010				MOV #FMG10,-(SP)
	016000	012746	000003				MOV #3,-(SP)
	016004	010600					MOV SP,R0
	016006	104414					TRAP C\$PNTB
	016010	062706	000010				ADD #10,SP
100	016014			20\$:			
101	016014			ENDMSG			
	016014						
	016014	104423					L10005: TRAP C\$MSG
102							
103							
104							
105	016016			BGNMSG	ERRG7		
	016016						
106	016016				PRINTB	#FMG12 ;BA/CC OUT RECVD	ERRG7::
	016016	012746	017100				MOV #FMG12,-(SP)
	016022	012746	000001				MOV #1,-(SP)
	016026	010600					MOV SP,R0
	016030	104414					TRAP C\$PNTB
	016032	062706	000004				ADD #4,SP
107	016036				PRINTB	#FMG13,@SEL4,@SEL6 ;ACTUAL BA/CC	
	016036	017746	164176				MOV @SEL6,-(SP)
	016042	017746	164170				MOV @SEL4,-(SP)
	016046	012746	017131				MOV #FMG13,-(SP)
	016052	012746	000003				MOV #3,-(SP)
	016056	010600					MOV SP,R0
	016060	104414					TRAP C\$PNTB
	016062	062706	000010				ADD #10,SP
108	016066				PRINTB	#FMG15,-2(R4) ;EXPECTED BA/CC	
	016066	016446	177776				MOV -2(R4),-(SP)
	016072	012746	017261				MOV #FMG15,-(SP)
	016076	012746	000002				MOV #2,-(SP)
	016102	010600					MOV SP,R0
	016104	104414					TRAP C\$PNTB
	016106	062706	000006				ADD #6,SP

109	016112		ENDMSG				
	016112					L10006:	
	016112	104423				TRAP	C\$MSG
110							
111	016114		BGNMSG	ERRG8			
	016114					ERRG8::	
112	016114		PRINTB	#FMG11	;BA/CC OUT XMIT		
	016114	012746				MOV	#FMG11,-(SP)
	016120	012746				MOV	#1,-(SP)
	016124	010600				MOV	SP,R0
	016126	104414				TRAP	C\$PNTB
	016130	062706				ADD	#4,SP
113	016134		PRINTB	#FMG13,@SEL4,@SEL6	;ACTUAL BA/CC		
	016134	017746				MOV	@SEL6,-(SP)
	016140	017746				MOV	@SEL4,-(SP)
	016144	012746				MOV	#FMG13,-(SP)
	016150	012746				MOV	#3,-(SP)
	016154	010600				MOV	SP,R0
	016156	104414				TRAP	C\$PNTB
	016160	062706				ADD	#10,SP
114	016164		PRINTB	#FMG14,-4(R5),-2(R5)	;EXPECTED BA/CC		
	016164	016546				MOV	-2(R5),-(SP)
	016170	016546				MOV	-4(R5),-(SP)
	016174	012746				MOV	#FMG14,-(SP)
	016200	012746				MOV	#3,-(SP)
	016204	010600				MOV	SP,R0
	016206	104414				TRAP	C\$PNTB
	016210	062706				ADD	#10,SP
115	016214		ENDMSG				
	016214					L10007:	
	016214	104423				TRAP	C\$MSG
116							
117							
118	016216		BGNMSG	ERRG10			
	016216					ERRG10::	
119	016216		PRINTB	#FMG16,-2(R0),-2(R1)	;RCV CC & XMIT CC		
	016216	016146				MOV	-2(R1),-(SP)
	016222	016046				MOV	-2(R0),-(SP)
	016226	012746				MOV	#FMG16,-(SP)
	016232	012746				MOV	#3,-(SP)
	016236	010600				MOV	SP,R0
	016240	104414				TRAP	C\$PNTB
	016242	062706				ADD	#10,SP
120	016246		ENDMSG				
	016246					L10010:	
	016246	104423				TRAP	C\$MSG
121							
122	016250		BGNMSG	ERRG11			
	016250					ERRG11::	
123	016250		PRINTB	#FMG17,-4(R0),-4(R1)	;RCV BUFFER & XMIT BUFFER		
	016250	016146				MOV	-4(R1),-(SP)
	016254	016046				MOV	-4(R0),-(SP)
	016260	012746				MOV	#FMG17,-(SP)
	016264	012746				MOV	#3,-(SP)
	016270	010600				MOV	SP,R0
	016272	104414				TRAP	C\$PNTB
	016274	062706				ADD	#10,SP

```

124 016300          ENDMSG
    016300
    016300 104423          L10011:
                                TRAP    C$MSG
125
126 016302          BGNMSG  ERRG12
    016302
127 016302 005303          DEC    R3          ;BACKUP TO RECEIVE ADDRESS
128 016304 005304          DEC    R4          ;BACKUP TO TRANSMIT ADDRESS
129 016306          PRINTB #FMG18,R3,R4 ;PRINT OUT ADDRESS
    016306 010446          MOV    R4,-(SP)
    016310 010346          MOV    R3,-(SP)
    016312 012746 017442   MOV    #FMG18,-(SP)
    016316 012746 000003   MOV    #3,-(SP)
    016322 010600          MOV    SP,R0
    016324 104414          TRAP   C$PNTB
    016326 062706 000010   ADD    #10,SP
130 016332          ENDMSG
    016332
    016332 104423          L10012:
                                TRAP    C$MSG
131
132
133
134
135
136
137 016334 045 101 123 FMG1: .ASCIZ /%ASEL0: %06%A SEL2: %06%N/
    016337 105 114 060
    016342 072 040 045
    016345 117 066 045
    016350 101 040 123
    016353 105 114 062
    016356 072 040 045
    016361 117 066 045
    016364 116 000
138 016366 045 101 123 FMG2: .ASCIZ /%ASEL4: %06%A SEL6: %06%N/
    016371 105 114 064
    016374 072 040 045
    016377 117 066 045
    016402 101 040 123
    016405 105 114 066
    016410 072 040 045
    016413 117 066 045
    016416 116 000
139 016420 045 101 105 FMG3: .ASCIZ /%AERROR IN SUBROUTINE CALLED AT PC: %06%N/
    016423 122 122 117
    016426 122 040 111
    016431 116 040 123
    016434 125 102 122
    016437 117 125 124
    016442 111 116 105
    016445 040 103 101
    016450 114 114 105
    016453 104 040 101
    016456 124 040 120
    016461 103 072 040
    016464 045 117 066
    016467 045 116 000
    
```

140	016472	045	101	103	FMG4: .ASCIZ /%ACPU MICROTEST FAILED%N/
	016475	120	125	040	
	016500	115	111	103	
	016503	122	117	124	
	016506	105	123	124	
	016511	040	106	101	
	016514	111	114	105	
	016517	104	045	116	
	016522	000			
141	016523	045	101	114	FMG5: .ASCIZ /%ALU. MICROTEST FAILED%N/
	016526	125	056	040	
	016531	115	111	103	
	016534	122	117	124	
	016537	105	123	124	
	016542	040	106	101	
	016545	111	114	105	
	016550	104	045	116	
	016553	000			
142	016554	045	101	116	FMG6: .ASCIZ /%AND RUN - MASTER CLEAR FAILED%N/
	016557	117	040	122	
	016562	125	116	040	
	016565	055	040	115	
	016570	101	123	124	
	016573	105	122	040	
	016576	103	114	105	
	016601	101	122	040	
	016604	106	101	111	
	016607	114	105	104	
	016612	045	116	000	
143	016615	045	101	116	FMG7: .ASCIZ /%ANAKS-NO BUFFER RCV: %D3%A SENT: %D3%N/
	016620	101	113	123	
	016623	055	116	117	
	016626	040	102	125	
	016631	106	106	105	
	016634	122	040	040	
	016637	122	103	126	
	016642	072	040	045	
	016645	104	063	045	
	016650	101	040	123	
	016653	105	116	124	
	016656	072	040	045	
	016661	104	063	045	
	016664	116	000		
144	016666	045	101	116	FMG8: .ASCIZ /%ANAKS-BAD DATA RCV: %D3%A SENT: %D3%N/
	016671	101	113	123	
	016674	055	102	101	
	016677	104	040	104	
	016702	101	124	101	
	016705	040	040	040	
	016710	122	103	126	
	016713	072	040	045	
	016716	104	063	045	
	016721	101	040	123	
	016724	105	116	124	
	016727	072	040	045	
	016732	104	063	045	
	016735	116	000		

145	016737	045	101	116	FMG9: .ASCIZ /%ANAKS-BAD HEADER RCV: %D3%A SENT: %D3%N/
	016742	101	113	123	
	016745	055	102	101	
	016750	104	040	110	
	016753	105	101	104	
	016756	105	122	040	
	016761	122	103	126	
	016764	072	040	045	
	016767	104	063	045	
	016772	101	040	123	
	016775	105	116	124	
	017000	072	040	045	
	017003	104	063	045	
	017006	116	000		
146	017010	045	101	122	FMG10: .ASCIZ /%AREPS-RCV: %D3%A SENT: %D3%N/
	017013	105	120	123	
	017016	055	122	103	
	017021	126	072	040	
	017024	045	104	063	
	017027	045	101	040	
	017032	123	105	116	
	017035	124	072	040	
	017040	045	104	063	
	017043	045	116	000	
147	017046	045	101	130	FMG11: .ASCIZ /%AXMIT BACC OUT COMMAND%N/
	017051	115	111	124	
	017054	040	102	101	
	017057	103	103	040	
	017062	117	125	124	
	017065	040	103	117	
	017070	115	115	101	
	017073	116	104	045	
	017076	116	000		
148	017100	045	101	122	FMG12: .ASCIZ /%ARCV BACC OUT COMMAND%N/
	017103	103	126	040	
	017106	102	101	103	
	017111	103	040	117	
	017114	125	124	040	
	017117	103	117	115	
	017122	115	101	116	
	017125	104	045	116	
	017130	000			
149	017131	045	101	101	FMG13: .ASCIZ /%AACTUAL ADDR. %06%A ACTUAL COUNT %D5%N/
	017134	103	124	125	
	017137	101	114	040	
	017142	040	040	101	
	017145	104	104	122	
	017150	056	040	045	
	017153	117	066	045	
	017156	101	040	101	
	017161	103	124	125	
	017164	101	114	040	
	017167	103	117	125	
	017172	116	124	040	
	017175	040	040	045	
	017200	104	065	045	
	017203	116	000		

150 017205 045 101 105 FMG14: .ASCIZ /%AEXPECTED ADDR. %06%A EXPECTED COUNT %D5%N/
 017210 130 120 105
 017213 103 124 105
 017216 104 040 101
 017221 104 104 122
 017224 056 040 045
 017227 117 066 045
 017232 101 040 105
 017235 130 120 105
 017240 103 124 105
 017243 104 040 103
 017246 117 125 116
 017251 124 040 045
 017254 104 065 045
 017257 116 000

151 017261 045 101 105 FMG15: .ASCIZ /%AEXPECTED ADDR. %06%N/
 017264 130 120 105
 017267 103 124 105
 017272 104 040 101
 017275 104 104 122
 017300 056 040 045
 017303 117 066 045
 017306 116 000

152 017310 045 101 122 FMG16: .ASCIZ /%ARCV CHAR. COUNT %D5%A XMIT CHAR. COUNT %D5%N/
 017313 103 126 040
 017316 103 110 101
 017321 122 056 040
 017324 103 117 125
 017327 116 124 040
 017332 045 104 065
 017335 045 101 040
 017340 130 115 111
 017343 124 040 103
 017346 110 101 122
 017351 056 040 103
 017354 117 125 116
 017357 124 040 045
 017362 104 065 045
 017365 116 000

153 017367 045 101 122 FMG17: .ASCIZ /%ARCV BUFFER AT %06%A XMIT BUFFER AT %06%N/
 017372 103 126 040
 017375 102 125 106
 017400 106 105 122
 017403 040 101 124
 017406 040 045 117
 017411 066 045 101
 017414 040 130 115
 017417 111 124 040
 017422 102 125 106
 017425 106 105 122
 017430 040 101 124
 017433 040 045 117
 017436 066 045 116
 017441 000

154 017442 045 101 104 FMG18: .ASCIZ /%ADATA DIFFERS AT RCV ADDR. %06%A AND XMIT ADDR. %06%N/
 017445 101 124 101
 017450 040 104 111

	017453	106	106	105	
	017456	122	123	040	
	017461	101	124	040	
	017464	122	103	126	
	017467	040	101	104	
	017472	104	122	056	
	017475	040	045	117	
	017500	066	045	101	
	017503	040	101	116	
	017506	104	040	130	
	017511	115	111	124	
	017514	040	101	104	
	017517	104	122	056	
	017522	040	045	117	
	017525	066	045	116	
	017530	000			
155	017531	045	101	104	FMG19: .ASCIZ /%ADEVICE NOT DMR%/
	017534	105	126	111	
	017537	103	105	040	
	017542	116	117	124	
	017545	040	104	115	
	017550	122	045	116	
	017553	000			
156	017554	045	101	115	FMG20: .ASCIZ /%AMICROTEST DISABLED%/
	017557	111	103	122	
	017562	117	124	105	
	017565	123	124	040	
	017570	104	111	123	
	017573	101	102	114	
	017576	105	104	045	
	017601	116	000		
157	017603	045	101	123	FMG21: .ASCIZ /%ASTATUS OF BUFFERS %N%NUMBER OF BUFFERS:%D3%/
	017606	124	101	124	
	017611	125	123	040	
	017614	117	106	040	
	017617	102	125	106	
	017622	106	105	122	
	017625	123	040	045	
	017630	116	045	101	
	017633	116	125	115	
	017636	102	105	122	
	017641	040	117	106	
	017644	040	102	125	
	017647	106	106	105	
	017652	122	123	072	
	017655	045	104	063	
	017660	045	116	000	
158	017663	045	101	102	FMG22: .ASCIZ /%ABUFFER SIZE: %D5%/
	017666	125	106	106	
	017671	105	122	040	
	017674	123	111	132	
	017677	105	072	040	
	017702	045	104	065	
	017705	045	116	000	
159	017710	045	101	111	FMG23: .ASCIZ /%AIN - RCV ASSIGNED:%D3%A XMIT ASSIGNED:%D3%/
	017713	116	040	040	
	017716	055	040	122	

	017721	103	126	040	
	017724	101	123	123	
	017727	111	107	116	
	017732	105	104	072	
	017735	045	104	063	
	017740	045	101	040	
	017743	040	040	130	
	017746	115	111	124	
	017751	040	101	123	
	017754	123	111	107	
	017757	116	105	104	
	017762	072	045	104	
	017765	063	045	116	
	017770	000			
160	017771	045	101	117	FMG24: .ASCIZ /%AOUT - RCV RETURNED:%D3%A XMIT RETURNED:%D3%N/
	017774	125	124	040	
	017777	055	040	122	
	020002	103	126	040	
	020005	122	105	124	
	020010	125	122	116	
	020013	105	104	072	
	020016	045	104	063	
	020021	045	101	040	
	020024	040	040	130	
	020027	115	111	124	
	020032	040	122	105	
	020035	124	125	122	
	020040	116	105	104	
	020043	072	045	104	
	020046	063	045	116	
	020051	000			
161					
162					
163	020052	124	111	115	EMG1: .ASCIZ /TIME OUT/
	020055	105	040	117	
	020060	125	124	000	
164	020063	124	111	115	EMG2: .ASCIZ /TIME OUT - DURING INTERRUPT EXERCISE/
	020066	105	040	117	
	020071	125	124	040	
	020074	055	040	104	
	020077	125	122	111	
	020102	116	107	040	
	020105	111	116	124	
	020110	105	122	122	
	020113	125	120	124	
	020116	040	105	130	
	020121	105	122	103	
	020124	111	123	105	
	020127	000			
165	020130	105	130	120	EMG8: .ASCIZ /EXPECTED CONTROL OUT - NOT RECEIVED/
	020133	105	103	124	
	020136	105	104	040	
	020141	103	117	116	
	020144	124	122	117	
	020147	114	040	117	
	020152	125	124	040	
	020155	055	040	116	

	020160	117	124	040	
	020163	122	105	103	
	020166	105	111	126	
	020171	105	104	000	
166	020174	125	116	105	EMG9: .ASCIZ /UNEXPECTED CONTROL OUT/
	020177	130	120	105	
	020202	103	124	105	
	020205	104	040	103	
	020210	117	116	124	
	020213	122	117	114	
	020216	040	117	125	
	020221	124	000		
167	020223	105	122	122	EMG10: .ASCIZ /ERROR - MULTIPLE XMITS/
	020226	117	122	040	
	020231	055	040	115	
	020234	125	114	124	
	020237	111	120	114	
	020242	105	040	130	
	020245	115	111	124	
	020250	123	000		
168	020252	102	125	106	EMG11: .ASCIZ /BUFFER ADDRESS ERROR/
	020255	106	105	122	
	020260	040	101	104	
	020263	104	122	105	
	020266	123	123	040	
	020271	105	122	122	
	020274	117	122	000	
169	020277	103	110	101	EMG12: .ASCIZ /CHARACTER COUNT ERROR/
	020302	122	101	103	
	020305	124	105	122	
	020310	040	103	117	
	020313	125	116	124	
	020316	040	105	122	
	020321	122	117	122	
	020324	000			
170	020325	105	122	122	EMG13: .ASCIZ /ERROR - MULTIPLE RCVS/
	020330	117	122	040	
	020333	055	040	115	
	020336	125	114	124	
	020341	111	120	114	
	020344	105	040	122	
	020347	103	126	123	
	020352	000			
171	020353	122	105	103	EMG14: .ASCIZ /RECEIVED EXTRA DATA/
	020356	105	111	126	
	020361	105	104	040	
	020364	105	130	124	
	020367	122	101	040	
	020372	104	101	124	
	020375	101	000		
172	020377	104	101	124	EMG15: .ASCIZ /DATA ERROR/
	020402	101	040	105	
	020405	122	122	117	
	020410	122	000		
173	020412	125	116	105	EMG16: .ASCIZ /UNEXPECTED HALT RECEIVED/
	020415	130	120	105	
	020420	103	124	105	

	020423	104	040	110	
	020426	101	114	124	
	020431	040	122	105	
	020434	103	105	111	
	020437	126	105	104	
	020442	000			
174	020443	103	117	116	EMG17: .ASCIZ /CONTROL IN PROBLEM - IN INTERRUPT ROUTINE/
	020446	124	122	117	
	020451	114	040	111	
	020454	116	040	120	
	020457	122	117	102	
	020462	114	105	115	
	020465	040	055	040	
	020470	111	116	040	
	020473	111	116	124	
	020476	105	122	122	
	020501	125	120	124	
	020504	040	122	117	
	020507	125	124	111	
	020512	116	105	000	
175	020515	123	120	125	EMG18: .ASCIZ /SPURIOUS RDO INTERRUPT/
	020520	122	111	117	
	020523	125	123	040	
	020526	122	104	117	
	020531	040	111	116	
	020534	124	105	122	
	020537	122	125	120	
176	020542	124	000		.EVEN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

.SBTTL LOAD DEVICE PROTECTION TABLE
:////////////////////
:// THIS TABLE IDENTIFIES THE LOAD DEVICE TO THE SUPERVISOR, SO THAT IT CAN BE
:// PROTECTED FROM TESTING. IF DESIRED.
:////////////////////

020544
020544

BGNPROT

L\$PROT::

020544 177777
020546 177777
020550 177777

.WORD -1
.WORD -1
.WORD -1

:DON'T CHECK CSR ADDRESS
:DON'T CHECK MASSBUS UNIT NUMBER
:DON'T CHECK DRIVE NUMBER

020552

ENDPROT

INITIALIZE SECTION

```

1          .SBTTL INITIALIZE SECTION
2
3          :////////////////////
4          :// THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
5          :// AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.
6          :////////////////////
7
8 020552    BGNINIT
9
10          L$INIT::
11          020552    SETPRI #PRI07          ;SET DIAGNOSTIC PRIORITY = 7
12          020552    012700 000340
13          020556    104441
14          020560    010637 002370          MOV SP,PSTACK          ;STORE BASE LEVEL PROGRAM STACK POINTER
15          020564    005037 002372          CLR SUBRPC            ;CLEAR STORAGE WORD FOR SUBROUTINE PC CALL
16          020570    005037 002364          CLR ERROR            ;CLEAR ERROR FLAGS
17          020574    005037 002274          CLR RESUME           ;CLEAR FLAG USED TO ALLOW BASE IN - RESUME.
18          020600    005037 002276          CLR DMCMD           ;CLEAR FLAG USED TO INDICATE DMC MODE
19          020604    005037 002376          CLR CLRNO           ;CLEAR WORD USED TO RUN MICRO TESTS ON
20          020610    005037 002350          CLR NXMFLG          ;EVERY OTHER MASTER CLEAR.
21          020614    005737 002264          TST FRSTIM          ;FLAG USED TO MARK A NXM DMR ADDRESS.
22          020620    001005
23          020622    012737 000001 002264    BNE 1$              ;IS THIS THE TIME THROUGH AFTER LOAD?
24          020630    005037 002266          MOV #1,FRSTIM       ;IF NOT - ERROR TRAP VECTOR ALREADY SAVED
25          020634    005037 002266          CLR FRSPAS          ;FLAG THAT WE'VE BEEN THRU THE 1ST TIME
26          020634    1$:
27          020634    CLRVEC #4              ;ENSURE VECTOR 4 IS IN NORMAL STATE.
28          020640    012700 000004          MOV #4,RO
29          020642    104436
30          020646    104436          TRAP C$CVEC
31          020642    READEF #EF.START        ;IS THIS JUST STARTED?
32          020642    012700 000040
33          020646    104447          MOV #EF.START,RO
34          020650    BCOMPLETE STARST        ;IF YES - BRANCH.          TRAP C$REFG
35          020650    103416
36          020652    READEF #EF.RESTART      ;IS THIS A RESTART ?          BCS STARST
37          020652    012700 000037
38          020656    104447          MOV #EF.RESTART,RO
39          020660    BCOMPLETE STARST        ;IF YES - BRANCH.          TRAP C$REFG
40          020660    103412
41          020662    READEF #EF.NEW          ;IS THIS A NEW PASS?          BCS STARST
42          020662    012700 000035
43          020666    104447          MOV #EF.NEW,RO
44          020670    BCOMPLETE NEWST        ;IF YES - BRANCH          TRAP C$REFG
45          020670    103410
46          020672    READEF #EF.CONTINUE    ;IS THIS A CONTINUATION?          BCS NEWST
47          020672    012700 000036
48          020676    104447          MOV #EF.CONTINUE,RO
49          020700    BNCOMPLETE GETPRM      ;IF NOT - GET PARAMETERS          TRAP C$REFG
50          020700    103013
51          020702    000137 021610          JMP END              ;OTHERWISE - DON'T INITIALIZE.          BCC GETPRM
52          020706    STARST:
53          020706    005037 002270          CLR STARES          ;CLEAR THE FLAG TO SHOW START/RESTART.
54          020712    NEWST:

```

INITIALIZE SECTION

```

41 020712 012737 177777 002366      MOV    #-1,LOGDEV      ;INITIALIZE LOGICAL UNIT NUMBER.
42 020720 005237 002266              INC    FRSPAS          ;INCREMENT # OF PASSES AFTER LOAD.
43 020724 005237 002270              INC    STARES         ;INCREMENT # OF PASSES SINCE START/RESTART.
44 020730                                GETPRM:
45 020730 005237 002366              INC    LOGDEV         ;NEXT LOGICAL UNIT TO BE TESTED
46 020734 023737 002366 002012      CMP    LOGDEV,L$UNIT  ;IS THE MAXIMUM UNIT # EXCEEDED?
47 020742 002363                    BGE    NEWST          ;IF YES - DO A NEW START
48 020744                                GPHARD LOGDEV,R1     ;GET THE P-TABLE POINTER INTO R1
                                MOV    LOGDEV,R0
                                TRAP   C$GPHRD
                                MOV    R0,R1
49 020754                                BNCOMPLETE GETPRM   ;IF NOT AVAILABLE, GET THE NEXT ONE
                                BCC   GETPRM
50 020756 012137 002252      MOV    (R1)+,WTYPE    ;MICROPROCESSOR TYPE
51 020762 011100              MOV    (R1),R0       ;SAVE THE ADDRESS
52 020764 032700 000007      BIT    #7,R0         ;DOES THIS DEVICE ADDRESS END IN NON-ZERO?
53 020770 001414              BEQ    10$           ;IF NOT - OK (76XXX0)
54 020772 042711 000007      BIC    #7,(R1)       ;MAKE IT 76XXX0
55 020776                                PRINTB #FINIT1,(R1),R0 ;INFORM THE USER
                                MOV    R0,-(SP)
                                MOV    (R1),-(SP)
                                MOV    #FINIT1,-(SP)
                                MOV    #3,-(SP)
                                MOV    SP,R0
                                TRAP   C$PNTB
                                ADD    #10,SP
020776 010046
021000 011146
021002 012746 021612
021006 012746 000003
021012 010600
021014 104414
021016 062706 0C0010
56 021022                                10$:
57 021022 011137 002232      MOV    (R1),CSR      ;CSR ADDRESS
58 021026 011137 002242      MOV    (R1),BSEL1   ;HIGH BYTE ADDRESS OF CSR
59 021032 005237 002242      INC    BSEL1
60 021036 011137 002234      MOV    (R1),SEL2    ;CONTROL OUT REGISTER ADDRESS
61 021042 062737 000002 002234      ADD    #2,SEL2
62 021050 011137 002244      MOV    (R1),BSEL3   ;HIGH BYTE OF SEL2
63 021054 062737 000003 002244      ADD    #3,BSEL3
64 021062 011137 002236      MOV    (R1),SEL4    ;PORT REG (SEL 4) ADDRESS
65 021066 062737 000004 002236      ADD    #4,SEL4
66 021074 011137 002246      MOV    (R1),BSEL5   ;HIGH BYTE OF SEL4
67 021100 062737 000005 002246      ADD    #5,BSEL5
68 021106 011137 002240      MOV    (R1),SEL6    ;PORT REG (SEL 6) ADDRESS
69 021112 062737 000006 002240      ADD    #6,SEL6
70 021120 012137 002250      MOV    (R1)+,BSEL7  ;HIGH BYTE OF SEL6
71 021124 062737 000007 002250      ADD    #7,BSEL7
72 021132 011100              MOV    (R1),R0       ;GET VECTOR
73 021134 032700 000007      BIT    #7,R0         ;DOES THIS VECTOR END IN NON-7ERO?
74 021140 001414              BEQ    11$           ;IF NOT - OK (XX0)
75 021142 042711 000007      BIC    #7,(R1)       ;MAKE IT XX0
76 021146                                PRINTB #FINIT2,(R1),R0 ;INFORM THE USER
                                MOV    R0,-(SP)
                                MOV    (R1),-(SP)
                                MOV    #FINIT2,-(SP)
                                MOV    #3,-(SP)
                                MOV    SP,R0
                                TRAP   C$PNTB
                                ADD    #10,SP
021146 010046
021150 011146
021152 012746 021705
021156 012746 000003
021162 010600
021164 104414
021166 062706 000010
77 021172                                11$:
78 021172 011137 002226      MOV    (R1),DMRVEC   ;RCV. VECTOR
79 021176 011137 002230      MOV    (R1),DMTVEC   ;TRANSMIT VECTOR

```

INITIALIZE SECTION

```

80 021202 011100      MOV      (R1),R0      ;RCV. VECTOR
81 021204 105060 000003  CLRB     3(R0)      ;CLEAR HI BYTE OF PSW FOR RCV. VECTOR.
82 021210 105060 000007  CLRB     7(R0)      ;CLEAR HI BYTE OF PSW FOR XMIT. VECTOR.
83                                     ;THIS WILL ENSURE THAT WE DON'T PICK
84                                     ;UP ANY UNEXPECTED BITS IN PROCESSORS
85                                     ;WHICH USE BITS 11-15 OF THE PSW. IE
86                                     ;IF BIT 11 IS SET IN AN 11/70 ANOTHER
87                                     ;REGISTER SET MAY BE USED.
88 021214 062737 000004 002230  ADD      #4,DMTVEC   ;ADJUST XMIT VECTOR
89
90                                     ;SET UP ISRS FOR DMR. INTERRUPTS ENABLED IN
91                                     ;TESTS 15-19.
92
93 021222      SETVEC  DMRVEC,#INISR,#PRI05 ;INPUT ISR
94 021222 012746 000240      MOV      #PRI05,-(SP)
95 021226 012746 022174      MOV      #INISR,-(SP)
96 021232 013746 002226      MOV      DMRVEC,-(SP)
97 021236 012746 000003      MOV      #3,-(SP)
98 021242 104437      TRAP    C$SVEC
99 021244 062706 000010      ADD      #10,SP
100 021250      SETVEC  DMTVEC,#OUTISR,#PRI05 ;OUTPUT ISR
101 021250 012746 000240      MOV      #PRI05,-(SP)
102 021254 012746 023256      MOV      #OUTISR,-(SP)
103 021260 013746 002230      MOV      DMTVEC,-(SP)
104 021264 012746 000003      MOV      #3,-(SP)
105 021270 104437      TRAP    C$SVEC
106 021272 062706 000010      ADD      #10,SP
107
108 021276 062701 000014  ADD      #14,R1      ;INCR. P-TABLE POINTER.
109 021302 012137 002254  MOV      (R1)+,DMTURN ;TURNAROUND
110
111 021306 013700 002224  MOV      SPEED,R0    ;GET THE SOFTWARE P-TABLE VALUE GIVEN
112                                     ;BY THE USER
113
114 021312      13$:
115 021312 012701 000002  MOV      #2,R1      ;GET FIRST TIMER VALUE
116 021316 012702 000012  MOV      #10.,R2    ;GET SECOND TIMER VALUE
117 021322      14$:
118 021322 006301      ASL     R1          ; TIMER VALUES X 2
119 021324 006302      ASL     R2
120 021326 005300      DEC     R0          ; DEC SPEED VARIABLE
121 021330 001374      BNE    14$         ; CONTINUE UNTIL DONE INCREASING WAIT VALUES
122
123 021332 010137 002312  MOV      R1,WAIT1   ;SAVE TIMER VALUE FOR $WAIT
124 021336 006201      ASR     R1          ;HALF THAT VALUE
125 021340 006201      ASR     R1          ;HALF IT AGAIN.
126 021342 010137 002314  MOV      R1,WAIT2   ;SAVE TIMER VALUE FOR $MSCLR AND $CLRQI
127                                     ;SUBROUTINES.
128 021346 010237 002316  MOV      R2,WAIT3   ;TIMER VALUE FOR $INOUT SUBROUTINE.
129
130 021352 012737 000333 002304  MOV      #333,AX3   ;CHECK TURNAROUND.
131                                     ;MASK FOR AX3-15 - BIT CLEARED WILL
132                                     ;BE THE INTERFACE SELECTED.
133 021360 022737 000001 002254  CMP      #1,DMTURN  ;IS V.35 REQUESTED?
134 021366 001004      BNE    20$         ;IF NOT - CONTINUE

```


	021623	111	116	107	
	021626	040	055	040	
	021631	127	111	114	
	021634	114	040	101	
	021637	123	123	125	
	021642	115	105	040	
	021645	104	115	122	
	021650	040	101	104	
	021653	104	122	105	
	021656	123	123	040	
	021661	045	117	066	
	021664	045	101	040	
	021667	050	116	117	
	021672	124	040	045	
	021675	117	066	045	
	021700	101	051	045	
173	021703	116	000		
	021705	045	101	052	FINIT2: .ASCIZ /%A** WARNING - WILL ASSUME DMR VECTOR %03%A (NOT %03%A)%N/
	021710	052	040	127	
	021713	101	122	116	
	021716	111	116	107	
	021721	040	055	040	
	021724	127	111	114	
	021727	114	040	101	
	021732	123	123	125	
	021735	115	105	040	
	021740	104	115	122	
	021743	040	126	105	
	021746	103	124	117	
	021751	122	040	040	
	021754	045	117	063	
	021757	045	101	040	
	021762	050	116	117	
	021765	124	040	045	
	021770	117	063	045	
	021773	101	051	045	
	021776	116	000		
174	022000	045	101	052	FINIT3: .ASCIZ /%A** WITH REMOTE LOOPBACK USE TESTS 17 - 19 ONLY **%N/
	022003	052	040	127	
	022006	111	124	110	
	022011	040	122	105	
	022014	115	117	124	
	022017	105	040	114	
	022022	117	117	120	
	022025	102	101	103	
	022030	113	040	125	
	022033	123	105	040	
	022036	124	105	123	
	022041	124	123	040	
	022044	061	067	040	
	022047	055	040	061	
	022052	071	040	117	
	022055	116	114	131	
	022060	040	052	052	
	022063	045	116	000	
175					
176					.EVEN

177
178
179
180

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35

.SBTTL AUTO DROP UNIT SECTION

:/ THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
:/ WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.

BGNAUTO

L\$AUTO::

SETVEC #4,#NOXMEM,#PRI07 ;SET UP NON -EXISTENT MEMORY TRAP VECTOR.

MOV #PRI07,-(SP)
MOV #NOXMEM,-(SP)
MOV #4,-(SP)
MOV #3,-(SP)
TRAP C\$SVEC
ADD #10,SP

CLR NXMFLG ;CLEAR FLAG THAT WILL BE SET IF NXM OCCURS.
TST @CSR ;REFERENCE MEMORY ADDRESS FOR THE DEVICE
;TO SEE IF IT EXISTS.

: IF THE DEVICE DOESN'T EXIST THE RESULTANT TRAP TO VECTOR 04 WILL
: CAUSE THE DEVICE TO BE DROPPED (SEE INTERRUPT ROUTINE 'DROP04').
: OTHERWISE THE MEMORY REFERENCE IS UNEVENTFUL AND THE DEVICE IS READY.
:*****

CLRVEC #4 ;RETURN VECTOR 04 TO NORMAL STATE

MOV #4,R0
TRAP C\$CVEC

TST NXMFLG ;DID NXM OCCUR?
BEQ 1\$;IF NOT EXIT
DODU LOGDEV ;DROP THE DEVICE

MOV LOGDEV,R0
TRAP C\$DODU

DOCLN ;DO CLEAN UP - FORCE BACK TO INIT CODE.

TRAP C\$DCLN

CLR NXMFLG ;RESTORE FLAG.

1\$:
ENDAUTO

L10015:

TRAP C\$AUTO

022066
022066
022066 012746 000340
022072 012746 023714
022076 012746 000004
022102 012746 000003
022106 104437
022110 062706 000010
022114 005037 002350
022120 005777 160106

022124
022124 012700 000004
022130 104436
022132 005737 002350
022136 001406
022140
022140 013700 002366
022144 104451
022146
022146 104444
022150 005037 002350
022154
022154
022154 104461

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

.SBTTL CLEANUP CODING SECTION

////////////////////////////////////
:// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED AT THE
:// END OF THE TEST SEQUENCE ON A PARTICULAR UNIT. THIS SECTION IS REQUIRED
:// EVEN IF IT IS A NULL CLEANUP
////////////////////////////////////

BGNCLN

L\$CLEAN::

TST NXMFLG ;WAS THERE A NXM ERROR?
BNE 10\$;IF YES - SKIP MASTER CLEAR.
MOV #MCLR,@SELO ;ISSUE A MASTER CLEAR.

10\$:

ENDCLN

L10016: TRAP C\$CLEAN

022156
022156
022156 005737 002350
022162 001003
022164 012777 040000 160040
022172
022172
022172
022172 104412

```

1      .SBTTL GLOBAL INTERRUPT HANDLING ROUTINES
2
3      :////////////////////////////////////////////////////////////////////
4      :// THE INTERRUPT HANDLING SECTION CONTAINS CODING REQUIRED TO USE
5      :// THE 'SETVEC' MACRO. NOTE EVERY INTERRUPT ROUTINE SHOULD SAVE
6      :// AND RESTORE R0.
7      :////////////////////////////////////////////////////////////////////
8
9      BGNSRV INISR ;INPUT INTERRUPT SERVICE ROUTINE
10     022174 010046 MOV R0,-(SP) ;SAVE R0
11     022174 010146 MOV R1,-(SP) ;SAVE R1
12     022200 017701 160026 MOV @SELO,R1 ;SAVE THE CONTROL IN COMMAND.
13     022204 042701 177760 BIC #177760,R1 ;CLEAR ALL BUT THE COMMAND BITS (0-3)
14     022210 032777 000200 160014 BIT #RDI,@SELO ;IS RDI SET
15     022216 001002 BNE 1$ ;IF YES - PROCESS INPUT COMMAND.
16     022220 000137 022730 JMP NEXT ;ISSUE NEXT INPUT COMMAND.
17     :*****
18     :
19     : PROCESS INPUT COMMAND
20     :
21     :*****
22     022224 1$:
23     022224 022701 000004 CMP #BACCR,R1 ;IS THIS A RCV. BA/CC?
24     022230 001533 BEQ 29$ ;BR IF YES.
25     022232 022701 000000 CMP #BACCT,R1 ;IS THIS A XMIT. BA/CC?
26     022236 001537 BEQ 30$ ;BR IF YES.
27     022240 022701 000003 CMP #BASEI,R1 ;IS THIS A BASE IN?
28     022244 001461 BEQ 20$ ;BR IF YES.
29     022246 022701 000001 CMP #CNTRL,R1 ;IS THIS A CONTROL IN?
30     022252 001444 BEQ 15$ ;BR IF YES.
31     022254 022701 000005 CMP #WMODEM,R1 ;IS THIS A WRITE MODEM?
32     022260 001417 BEQ 10$ ;BR IF YES.
33     022262 022701 000015 CMP #INTER,R1 ;IS THIS AN INTERFACE WRITE.
34     022266 001410 BEQ 5$ ;BR IF YES.
35     022270 022701 000002 CMP #HLT,R1 ;IS THIS A HALT?
36     022274 001572 BEQ 70$ ;EXIT - IF YES (NOTHING TO SET UP)
37     022276 ERRDF 17,EMG17,ERRG2 ;PROBLEM IF IT'S NOT ONE OF THE ABOVE.
38     022276 104455 TRAP C$ERDF
39     022300 000021 .WORD 17
40     022302 020443 .WORD EMG17
41     022304 015126 .WORD ERRG2
42     022306 000565 BR 70$ ;EXIT
43
44     5$:
45     :
46     : WRITE AX3-15
47     :
48     :
49     :
50     022310 113777 002304 157732 MOVB AX3,@SEL7 ;WRITE NECESSARY AX3-15 INTERFACE.
51     : ;AX3 HAS BEEN DETERMINED IN THE INIT
52     : ;CODE.
53     BR 70$
54
55     10$:
56     :
57     : MODEM WRITE
    
```

```

53
54 022320 022737 000006 002254      ;CMP      #LLOOP,DMTURN      ;IS LOCAL MODEM LOOPBACK DESIRED?
55 022326 001007                      BNE      11$           ;BR IF NOT
56 022330 042777 000004 157702      BIC      #MAINT2,@SEL6  ;ENSURE REMOTE LOOPBACK IS CLEAR.
57 022336 052777 000110 157674      BIS      #DTR!MAINT1,@SEL6 ;SET MAINTENANCE 1 BIT AND DTR.
58 022344 000546                      BR       70$
59 022346                                11$:
60 022346 042777 000010 157664      BIC      #MAINT1,@SEL6  ;ENSURE LOCAL LOOPBACK IS CLEAR.
61 022354 052777 000104 157656      BIS      #DTR!MAINT2,@SEL6 ;SET MAINTENANCE 2 BIT AND DTR.
62 022362 000537                      BR       70$           ;CLEAR RQI
63 022364                                15$:
64
65      ;CONTROL IN
66
67 022364 005737 002300      TST      MNTMDE          ;IS MAINTENANCE MODE REQUESTED
68 022370 001404                      BEQ      17$           ;BR IF NOT
69 022372 012777 000400 157640      MOV      #MAINT,@SEL6   ;REQUEST MAINT. MODE
70 022400 000530                      BR       70$
71 022402                                17$:
72 022402 005077 157632      CLR      @SEL6           ;FULL DUPLEX - NON-MAINT. MODE.
73 022406 000525                      BR       70$
74 022410                                20$:
75
76      ;BASE IN
77
78 022410 012777 002634 157620      MOV      #BASE,@SEL4    ;BASE TABLE ADDRESS.
79
80 022416 005737 002276      TST      DMCMD          ;ARE WE IN DMC MODE?
81 022422 001004                      BNE      22$           ;BR IF YES
82 022424 012777 000522 157606      MOV      #DMR,@SEL6    ;DMR MODE.
83 022432 000402                      BR       23$           ;CHECK LOOPBACK.
84 022434                                22$:
85 022434 005077 157600      CLR      @SEL6           ;DMC MODE
86 022440                                23$:
87 022440 005737 002272      TST      START          ;IS THIS THE FIRST BASE IN?
88 022444 001004                      BNE      24$           ;IF NOT - SET RESUME.
89 022446 052777 000100 157560      BIC      #IEO,@SEL2     ;ON FIRST BASE IN SET RDO INT.ENABLE.
90 022454 000406                      BR       25$
91 022456                                24$:
92 022456 052777 010000 157554      BIS      #RES,@SEL6     ;SET RESUME
93 022464 012737 177777 002356      MOV      #-1,RESFLG    ;FLAG THAT THIS IS A BASE IN RESUME COMMAND
94                                ;(THIS WILL BE USED LATER IN THIS ISR TO
95                                ;DECIDE WHAT THE NEXT COMMAND WILL BE)
96 022472                                25$:
97 022472 005737 002254      TST      DMTURN          ;IS INTERNAL LOOPACK REQUESTED?
98 022476 001004                      BNE      27$           ;BR IF NOT - CLEAR LU LOOP
99 022500 052777 004000 157524      BIS      #LPLU,@SELO   ;SET THE LINE UNIT LOOPBACK BIT
100 022506 000465                      BR       70$           ;CLEAR RQI AND EXIT.
101 022510                                27$:
102 022510 042777 004000 157514      BIC      #LPLU,@SELO   ;CLEAR LINE UNIT LOOPBACK (CONNECTOR OR
103                                ;CABLE)
104 022516 000461                      BR       70$           ;CLEAR RQI AND EXIT
105
106      ;BA/CC IN RCV
107
108
109 022520                                29$:

```

```

110 022520 005337 002326      DEC      INRCV      ;DECREMENT COUNTER
111 022524 012277 157506      MOV      (R2)+,@SEL4 ;RCV BUFFER ADDRESS
112 022530 012277 157504      MOV      (R2)+,@SEL6 ;RCV CHARACTER COUNT
113 022534 000406              BR       40$
114
115                          ;BA/CC IN XMIT
116
117 022536              30$:
118 022536 005337 002330      DEC      INXMIT     ;DECREMENT COUNTER
119 022542 012377 157470      MOV      (R3)+,@SEL4 ;XMIT BUFFER ADDRESS.
120 022546 012377 157466      MOV      (R3)+,@SEL6 ;XMIT CHARACTER COUNT.
121 022552              40$:
122 022552 005737 002302      TST      MMANAG     ;ARE THE BUFFERS MEMORY MANAGED?
123 022556 001441              BEQ      70$        ;IF NOT SKIP CONVERTING VIRTUAL ADDR
124                          ;TO PHYSICAL ADDR.
125 022560 052777 040000 157452  BIS      #BIT14,@SEL6 ;SET BIT 16 OF PHYSICAL ADDRESS (I.E.
126                          ;VIRTUAL ADDR 60000 = PHYSICAL ADDR 200000
127 022566 010246              MOV      R2,-(SP)   ;SAVE R2 (NEXT RCV BUFFER ADDRESS)
128 022570 017702 157442      MOV      @SEL4,R2   ;SAVE THE VIRTUAL ADDRESS.
129 022574 042777 160000 157434  BIC      #160000,@SEL4 ;CLEAR BITS CORRESPONDING TO THE PAGE #
130                          ;IN THE VIRTUAL ADDRESS.
131 022602 042702 017777      BIC      #17777,R2  ;SAVE ONLY THE PAGE # IN THE SAVED ADDR.
132 022606 022702 060000      CMP      #60000,R2  ;IS THIS PAGE 3?
133 022612 001421              BEQ      44$        ;IF YES, PHYSICAL ADDRESS CALCULATED
134 022614 022702 100000      CMP      #100000,R2 ;IS THIS PAGE 4?
135 022620 001004              BNE      41$        ;IF NOT SEE IF IT'S PAGE 4 OR 5
136 022622 052777 020000 157406  BIS      #BIT13,@SEL4 ;SET BIT FOR PHYSICAL ADDR (I.E. VIRTUAL
137                          ;ADDR 100000 = PHYSICAL ADDR. 220000
138 022630 000412              BR       44$
139 022632              41$:
140 022632 022702 120000      CMP      #120000,R2 ;IS THIS PAGE 4?
141 022636 001004              BNE      42$        ;IF NOT, MUST BE PAGE 5.
142 022640 052777 040000 157370  BIS      #BIT14,@SEL4 ;SET BIT FOR PHYSICAL ADDR (I.E. VIRTUAL
143                          ;ADDR 120000 = PHYSICAL ADDR. 240000
144 022646 000403              BR       44$
145 022650              42$:
146 022650 052777 060000 157360  BIS      #BIT14!BIT13,@SEL4 ;SET BIT FOR PHYSICAL ADDR (I.E. VIRTUAL
147                          ;ADDR 140000 = PHYSICAL ADDR. 260000
148 022656              44$:
149 022656 012602              MOV      (SP)+,R2   ;RESTORE R2 (NEXT RCV BUFFER ADDRESS)
150 022660 000400              BR       70$        ;CLEAR RQI AND EXIT
151
152
153 022662              70$:
154 022662 010137 002362      MOV      R1, LAST   ;SAVE THE INPUT COMMAND (USED
155                          ;TO DETERMINE NEXT INPUT COMMAND)
156 022666 005737 002276      TST      DMCMD     ;ARE WE IN DMC MODE?
157 022672 001011              BNE      80$        ;IF YES - DON'T USE IECLR
158                          ;NOTE: INTERRUPT CAPABILITY FOR RQI
159                          ;CLEAR IS ONLY AVAILABLE IN DMR MODE.
160 022674 012601              MOV      (SP)+,R1   ;RESTORE R1
161 022676 012600              MOV      (SP)+,R0   ;RESTORE R0
162 022700 052777 000020 157324  BIS      #IECLR,@SELO ;SET INTERRUPT ENABLE FOR RDI CLEAR.
163 022706 042777 000040 157316  BIC      #RQI,@SELO ;CLEAR RQI - INT. GENERATED WHEN RDI
164                          ;CLEARS IN RESPONSE.
165 022714 000002              RTI               ;RETURN AND WAIT FOR RQI CLEAR INTERRUPT.
166

```

```

167 022716
168 022716 042777 000020 157306 80$: BIC #IECLR,@SELO ;ENSURE INTERRUPT ENABLE FOR RDI CLEAR IS CLR.
169 022724 CALL $CLRQI ;CLEAR RQI AND WAIT FOR RDI TO CLEAR.
170 :*****
171 :
172 : RDI CLEAR - DETERMINE NEXT INPUT COMMAND.
173 :
174 :*****
175 022730 NEXT:
176 022730 022737 000002 002362 CMP #HLT, LAST ;WAS THE LAST COMMAND A HALT?
177 022736 001015 BNE 110$ ;IF NOT - PROCEED.
178 022740 005737 002274 TST RESUME ;ARE WE TESTING RESUME?
179 022744 001541 BEQ 170$ ;IF NOT, DON'T ISSUE ANOTHER COMMAND.
180 022746 005737 002352 TST INFLAG ;INPUT BUFFER DONE?
181 022752 001403 BEQ 106$ ;IF NOT, BASE IN.
182 022754 005737 002354 TST OUTFLG ;OUTPUT BUFFER DONE?
183 022760 001133 BNE 170$ ;IF YES, DON'T ISSUE ANOTHER COMMAND.
184 022762
185 022762 112777 000143 157242 106$: MOVB #IESET!RQI!BASEI,@SELO ;ISSUE A BASE IN.
186 022770 000527 BR 170$ ;EXIT
187 022772
188 022772 005737 002276 110$: TST DMCMD E ;ARE WE IN DMC MODE?
189 022776 001032 BNE 130$ ;IF YES - DON'T BOTHER CHECKING MODEM
190 :WRITE AND AX3-15 WRITE COMMANDS
191 023000 022737 000003 002362 CMP #BASEI, LAST ;WAS THE LAST COMMAND A BASE IN ?
192 023006 001405 BEQ 115$ ;IF YES - SEE IF INTER. OR M. WRITE IS NEEDED?
193 023010 022737 000015 002362 CMP #INTER, LAST ;WAS THE LAST COMMAND AN AX3-15 WRITE?
194 023016 001413 BEQ 117$
195 023020 000421 BR 130$ ;KEEP CHECKING FOR NEXT COMMAND.
196 023022
197 023022 005737 002262 115$: TST INFACE ;IS AN AX3-15 WRITE NEEDED?
198 023026 001407 BEQ 117$ ;BR IF NOT
199 023030 005737 002272 TST START ;WAS CONTROL IN ISSUED?
200 023034 001004 BNE 117$ ;IF YES - NO NEED TO REWRITE AX3-15. THIS
201 :SHOULD HAVE BEEN DONE ON THE 1ST BASE IN.
202 023036 112777 000155 157166 MOVB #IESET!RQI!INTER,@SELO ;ISSUE AN AX3-15 WRITE COMMAND.
203 023044 000501 BR 170$
204 023046
205 023046 005737 002306 117$: TST WMAINT ;WRITE MAINT 1 OR 2?
206 023052 001404 BEQ 130$ ;IF NOT - SKIP WRITE MODEM COMMAND.
207 023054 112777 000145 157150 MOVB #IESET!RQI!WMODEM,@SELO ;ISSUE A MODEM WRITE COMMAND
208 023062 000472 BR 170$
209 023064
210 023064 005737 002272 130$: TST START ;WAS A CONTROL IN ISSUED?
211 023070 001006 BNE 150$ ;IF YES - SKIP
212 023072 005237 002272 INC START ;SET FLAG.
213 023076 112777 000141 157126 MOVB #IESET!RQI!CNTRL,@SELO ;ISSUE A CONTROL IN
214 023104 000461 BR 170$
215 023106
216 023106 005737 002326 150$: TST INRCV ;ARE ALL THE BA/CC IN RCVS DONE?
217 023112 001424 BEQ 160$ ;IF YES - BR TO SEE IF XMIT'S DONE.
218 023114 005737 002274 TST RESUME ;IS A TEST OF RESUME REQUESTED?
219 023120 001415 BEQ 153$ ;BR IF NOT.
220 023122 032737 000001 002326 BIT #BIT0, INRCV ;IS THIS AN ODD COUNT?
221 023130 001411 BEQ 153$ ;BR IF NOT.
222 023132 005737 002356 TST RESFLG ;WAS THE LAST COMMAND A BASE IN RESUME?
223 023136 001004 BNE 152$ ;IF YES, ISSUE BA/CC
    
```



```

224                                     ;HALT - TO TEST RESUME. NOTE: THIS WILL
225                                     ;OCCUR ONLY WHEN RESUME IS REQUESTED,
226                                     ;FOLLOWING EVERY OTHER BA/CC
227                                     ;COMMAND (NEVER FOLLOWING A RESUME)
228 023140 112777 000142 157064          MOVB  #IESET!RQI!HLT,@BSELO ;HALT IT
229 023146 000440                          BR    170$
230 023150                                152$:
231 023150 005037 002356                  CLR    RESFLG          ;CLEAR FLAG.
232 023154                                153$:
233 023154 112777 000144 157050          MOVB  #IESET!RQI!BACCR,@BSELO ;ISSUE A BA/CC IN RCV. COMMAND.
234 023162 000432                          BR    170$
235 023164                                160$:
236 023164 005737 002330                  TST   INXMIT          ;ARE ALL THE BA/CC IN XMITs DONE?
237 023170 001424                          BEQ   165$            ;IF YES, SET THE FLAG
238 023172 005737 002274                  TST   RESUME          ;IS A TEST OF RESUME REQUESTED?
239 023176 001415                          BEQ   163$            ;BR IF NOT.
240 023200 032737 000001 002330          BIT   #BIT0,INXMIT    ;IS THIS AN ODD COUNT?
241 023206 001411                          BEQ   163$            ;BR IF NOT.
242 023210 005737 002356                  TST   RESFLG          ;WAS THE LAST COMMAND A BASE IN RESUME?
243 023214 001004                          BNE   162$            ;IF YES, ISSUE BA/CC
244                                     ;HALT - TO TEST RESUME. NOTE: THIS WILL
245                                     ;OCCUR ONLY WHEN RESUME IS REQUESTED,
246                                     ;FOLLOWING EVERY OTHER BA/CC
247                                     ;COMMAND (NEVER FOLLOWING A RESUME)
248 023216 112777 000142 157006          MOVB  #IESET!RQI!HLT,@BSELO ;HALT IT
249 023224 000411                          BR    170$
250 023226                                162$:
251 023226 005037 002356                  CLR    RESFLG          ;CLEAR BASE IN RESUME FLAG.
252 023232                                163$:
253 023232 112777 000140 156772          MOVB  #IESET!RQI!BACCT,@BSELO ;ISSUE A BA/CC IN XMIT COMMAND.
254 023240 000403                          BR    170$
255 023242                                165$:
256 023242 012737 177777 002352          MOV   #-1,INFLAG      ;FLAG THAT ALL BA/CC INS DONE.
257
258 023250                                170$:
259 023250 012601                          MOV   (SP)+,R1        ;RESTORE R1
260 023252 012600                          MOV   (SP)+,R0        ;RESTORE R0
261
262 023254                                ENDSRV
263                                     ;*****
264                                     ;*****
265                                     ;*****
266                                     ;*****
267 023256                                BGNSRV  OUTISR          ;OUTPUT INTERRUPT SERVICE ROUTINE
268 023256                                OUTISR::
269 023256 010046                          MOV   R0,-(SP)        ;SAVE R0
270 023260 032777 000200 156746          BIT   #RDO,@SEL2      ;IS THE RDO OUT BIT SET?
271 023266 001006                          BNE   5$              ;IF YES - OK TO PROCEED.
272 023270                                ERRDF  18,EMG18,ERRG2  ;OTHERWISE REPORT SPURIOUS INTERRUPT
273 023270 104455                                TRAP  C$ERDF
274 023272 000022                                .WORD 18
275 023274 020515                                .WORD EMG18
276 023276 015126                                .WORD ERRG2
277 023300 000137 023652                  JMP   60$

```

```

274 023304
275 023304 032777 000001 156722 5$: BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT
276 023312 001436 BEQ 20$ ;IF NOT - PROCESS BA/CC OUT
277 023314 032777 001000 156716 BIT #HALTC,@SEL6 ;IS THIS CONTROL OUT A HALT?
278 023322 001013 BNE 10$ ;IF IT IS - SEE IF WE SHOULD RESUME.
279 023324 032777 000040 156706 BIT #DMRRUN,@SEL6 ;IS THIS DMR RUN MODE ACKNOWLEDGE?
280 023332 001407 BEQ 10$ ;IF NOT - REPORT ERROR
281 023334 000137 023702 JMP 65$ ;EXIT
282 023340 7$:
283 023340 ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT.
    023340 104455 TRAP C$ERDF
    023342 000011 .WORD 9
    023344 020174 .WORD EMG9
    023346 015126 .WORD ERRG2
284 023350 000554 BR 65$ ;EXIT ROUTINE
285 023352 10$:
286 023352 005737 002352 TST INFLAG ;ARE THE INPUTS DONE?
287 023356 001403 BEQ 15$ ;BR IF NOT
288 023360 005737 002354 TST OUTFLG ;ARE THE OUTPUTS DONE?
289 023364 001132 BNE 60$ ;IF YES - ALL DONE, EXIT
290 023366 15$:
291 023366 005737 002274 TST RESUME ;IS A RESUME REQUESTED?
292 023372 001143 BNE 65$ ;IF YES - OK, BR TO EXIT
293 023374 16$:
294 023374 ERRDF 16,EMG16 ;ERROR - UNEXPECTED HALT.
    023374 104455 TRAP C$ERDF
    023376 000020 .WORD 16
    023400 020412 .WORD EMG16
    023402 000000 .WORD 0
295 023404 000137 023652 JMP 60$
296 023410 20$:
297 023410 005737 002302 TST MMANAG ;ARE THE BUFFERS MEMORY MANAGED?
298 023414 001452 BEQ 40$ ;IF NOT - NO NEED TO DETERMINE PHYS. ADDR.
299 023416 032777 040000 156614 BIT #BIT14,@SEL6 ;IS BIT 16 OF THE PHYSICAL ADDR SET?
300 ;(I.E. BUFFER SHOULD BE IN PHYSICAL
301 ;ADDRESS RANGE: 200000 - 277776)
302 023424 001005 BNE 21$ ;PROCEED - IF BIT SET.
303 023426 ERRDF 11,EMG11,ERRG2
    023426 104455 TRAP C$ERDF
    023430 000013 .WORD 11
    023432 020252 .WORD EMG11
    023434 015126 .WORD ERRG2
304 023436 000505 BR 60$
305 023440 21$:
306 023440 042777 140000 156572 BIC #BIT15!BIT14,@SEL6 ;CLEAR THE EXTENDED ADDRESS BITS.
307 023446 017702 156564 MOV @SEL4,R2 ;SAVE BITS 0-15 OF THE PHYSICAL ADDRESS.
308 023452 042702 017777 BIC #17777,R2 ;SAVE ONLY PAGE ADDRESS BITS.
309 023456 042777 160000 156552 BIC #160000,@SEL4 ;CLEAR PAGE ADDRESS BITS IN SEL4
310 ;DETERMINE PAGE # FOR VIRTUAL ADDRESS.
311 023464 005702 TST R2 ;IS THIS PAGE 3?
312 023466 001004 BNE 22$ ;IF NOT CHECK FOR OTHER PAGES
313 023470 052777 060000 156540 BIS #60000,@SEL4 ;SET BITS FOR PAGE 3.
314 023476 000421 BR 40$
315 023500 22$:
316 023500 022702 020000 CMP #20000,R2 ;IS THIS PAGE 4?
317 023504 001004 BNE 23$ ;IF NOT - KEEP CHECKING
318 023506 052777 100000 156522 BIS #100000,@SEL4 ;SET BITS FOR PAGE 4.
    
```

```

319 023514 000412          BR      40$
320 023516                23$:
321 023516 022702 040000    CMP     #40000,R2      ;IS THIS PAGE 5?
322 023522 001004          BNE     24$          ;IF NOT - MUST BE PAGE 6
323 023524 052777 120000 156504  BIS     #120000,@SEL4 ;SET BITS FOR PAGE 5.
324 023532 000403          BR      40$
325 023534                24$:
326 023534 052777 140000 156474  BIS     #140000,@SEL4 ;SET BITS FOR PAGE 6.
327 023542                40$:
328 023542 032777 000004 156464  BIT     #RCV,@SEL2   ;IS THIS A RECV. BUFFER?
329 023550 001023          BNE     50$          ;IF YES - PROCESS THE BUFFER.
330 023552 005337 002334    DEC     OUTXMT       ;DECREMENT BA/CC OUT XMIT.
331 023556 022577 156454    CMP     (R5)+,@SEL4  ;IS THE XMIT BUFFER ADDRESS CORRECT?
332 023562 001406          BEQ     41$          ;IF YES - PROCEED.
333 023564 005725          TST     (R5)+        ;INCR. POINTER FOR ERROR MESSAGE.
334 023566 104455          ERRDF   11,EMG11,ERRG8 ;IF NOT - ERROR
                                TRAP   C$ERDF
                                .WORD  11
                                .WORD  EMG11
                                .WORD  ERRG8
335 023576 000425          BR      60$
336 023600                41$:
337 023600 022577 156434    CMP     (R5)+,@SEL6  ;IS THE CHAR. COUNT CORRECT?
338 023604 001422          BEQ     60$          ;IF OK - EXIT ROUTINE.
339 023606 104455          ERRDF   12,EMG12,ERRG8 ;IF NOT - ERROR
                                TRAP   C$ERDF
                                .WORD  12
                                .WORD  EMG12
                                .WORD  ERRG8
340 023616 000415          BR      60$
341 023620                50$:
342 023620 005337 002332    DEC     OUTRCV       ;DECREMENT BA/CC OUT RCV
343 023624 022477 156406    CMP     (R4)+,@SEL4  ;IS THE RCV BUFFER ADDRESS CORRECT?
344 023630 001406          BEQ     51$          ;IF OK - PROCEED
345 023632 104455          ERRDF   11,EMG11,ERRG7
                                TRAP   C$ERDF
                                .WORD  11
                                .WORD  EMG11
                                .WORD  ERRG7
346 023642 005724          TST     (R4)+        ;UPDATE POINTER
347 023644 000402          BR      60$          ;EXIT ROUTINE
348 023646                51$:
349 023646 017724 156366    MOV     @SEL6,(R4)+  ;CHANGE THE CHARACTER COUNT TO WHAT
350                                ;WAS RECEIVED.
351 023652                60$:
352 023652 005737 002334    TST     OUTXMT       ;HAVE ALL THE XMITs BEEN DONE?
353 023656 001011          BNE     65$          ;IF NOT, CONTINUE
354 023660 005737 002332    TST     OUTRCV       ;HAVE ALL THE RECEIVES BEEN DONE?
355 023664 001006          BNE     65$          ;IF NOT, CONTINUE
356 023666                61$:
357 023666 042777 000100 156340  BIC     #IEO,@SEL2   ;CLEAR THE OUTPUT INTERRUPT
358 023674 012737 177777 002354  MOV     #-1,OUTFLG   ;FLAG AS DONE.
359 023702                65$:
360 023702 042777 000207 156324  BIC     #RDO!CMD,@SEL2 ;CLEAR THE RDO BIT.
361 023710 012600          MOV     (SP)+,R0     ;RESTORE R0
362 023712                ENDSRV
    
```

L10020:

GLOBAL INTERRUPT HANDLING ROUTINES

```

023712 000002
363
364
365
366
367 023714          BGNSRV NOXMEM
    023714
                                NOXMEM::
368
369 023714 012737 000001 002350      MOV    #1,NXMFLG      ;SET FLAG IF MEMORY ADDRESSED IS NON-EXISTENT.
370
371 023722          ENDSRV
    023722
    023722 000002
                                L10021:
372
373
                                RTI

```

```

1      .SBTTL DROP UNIT SECTION
2
3      :////////////////////////////////////////////////////////////////////
4      :// THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
5      :// TO NO LONGER BE TESTED.
6      :////////////////////////////////////////////////////////////////////
7
8      023724      BGNDU
9      023724
10     023724      BRESET      ;ISSUE UNIBUS RESET TO CLEAN UP
11     023724      104433
12     023726      013746      002366
13     023732      012746      023754
14     023736      012746      000002
15     023742      010600
16     023744      104417
17     023746      062706      000006
18
19     ENDDU
20
21     L10022:
22     TRAP      C$RESET
23
24     MOV      LOGDEV,-(SP)
25     MOV      #FMDROP,-(SP)
26     MOV      #2,-(SP)
27     MOV      SP,R0
28     TRAP      C$PNTF
29     ADD      #6,SP
30
31     FMDROP: .ASCIZ /%N%AUNIT %D2%A DROPPED/
32
33     .EVEN
    
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19 024004
024004
20
21 024004
024004 012746 000340
024010 012746 024126
024014 012746 000004
024020 012746 000003
024024 104437
024026 062706 000010
22 024032 005037 002350
23 024036 005001
24
25
26
27
28
29
30
31
32 024040 005777 156166
33 024044 012701 000002
34 024050 005777 156160
35 024054 012701 000004
36 024060 005777 156152
37 024064 012701 000006
38 024070 005777 156144
39 024074 005737 002350
40 024100 001406
41 024102
024102 013700 002366
024106 104451
42 024110
024110 104444
43 024112 005037 002350
44 024116
45 024116
024116 012700 000004
024122 104436

```

.SBTTL          TEST 1 - DMR CSR VERIFICATION
*****
TEST 1 - DMR-11
* VERIFY THAT ADDRESSING THE 4 UNIBUS CSRS DOES NOT CAUSE A NON-
* EXISTENT MEMORY TRAP.
*
* THE DMR IS AN NPR DEVICE RESIDING ON A UNIBUS. COMMUNICATION
* BETWEEN THE MAIN CPU AND THE DMR IS ACCOMPLISHED THROUGH A
* SET OF FOUR 16-BIT UNIBUS CONTROL AND STATUS REGISTERS (CSRS).
* THE FOUR REGISTERS ARE ASSIGNED ADDRESSES IN THE I/O PAGE
* FLOATING ADDRESS SPACE: 76XXX0 - 76XXX6
*
* NOTE: THIS TEST IS REDUNDANT IN THAT STATIC LOGIC TESTS SHOULD
* HAVE BEEN RUN BEFORE THESE FREE-RUNNING TESTS WERE STARTED, AND
* THEY SHOULD HAVE DETECTED ANY CSR ADDRESSING PROBLEMS.
* BUT JUST IN CASE THOSE STATIC TESTS AREN'T RUN, WE'LL BE SAFE.
*****
BGNTST
T1::
SETVEC #4,#LOCATE,#PRI07 ;SET UP NON -EXISTENT MEMORY TRAP VECTOR.
MOV #PRI07,-(SP)
MOV #LOCATE,-(SP)
MOV #4,-(SP)
MOV #3,-(SP)
TRAP C$SVEC
ADD #10,SP

CLR NXMFLG ;FLAG USED IN THE TRAP ROUTINE.
CLR R1 ;USE REGISTER TO REMEMBER WHICH OF THE
;4 CSRS WE ARE ADDRESSING.

*****
; IF ADDRESSING ANY ONE OF THE CSRS RESULTS IN A TRAP TO VECTOR 04, THE TRAP
; WILL REPORT THE ERROR (SEE INTERRUPT ROUTINE 'LOCATE'). OTHERWISE THE
; MEMORY REFERENCE IS UNEVENTFUL AND THE DEVICE IS READY FOR FURTHER TESTS
*****

TST @SEL0 ;TEST THE CSR AT 76XXX0
MOV #2,R1 ;SAVE THE OFFSET OF THE NEXT CSR
TST @SEL2 ;TEST THE CSR AT 76XXX2
MOV #4,R1 ;SAVE THE OFFSET OF THE NEXT CSR
TST @SEL4 ;TEST THE CSR AT 76XXX4
MOV #6,R1 ;SAVE THE OFFSET OF THE NEXT CSR
TST @SEL6 ;TEST THE CSR AT 76XXX6
TST NXMFLG ;WAS THERE A TRAP?
BEQ 10$ ;IF NOT - EXIT.
DODU LOGDEV ;DROP THE DEVICE
MOV LOGDEV,R0
TRAP C$DODU

DOCLN ;DO CLEAN UP - FORCE BACK TO INIT CODE.
TRAP C$DCLN

CLR NXMFLG ;RESTORE THE FLAG.

10$: CLRVEC #4 ;RETURN VECTOR 04 TO NORMAL STATE
MOV #4,R0
TRAP C$CVEC

```

```

46
47 024124          ENDTST
    024124
    024124 104401          L10023:
48                                     TRAP   C$ETST
49
50 024126          BGNSRV LOCATE          ;INTERRUPT SERVICE ROUTINE
    024126                                     LOCATE::
51 024126 010046    MOV     R0,-(SP)      ;SAVE R0
52 024130 005737 002350  TST     NXMFLG      ;HAVE WE HAD AT LEAST 1 PREVIOUS TRAP?
53 024134 001006    BNE     10$          ;IF YES, DON'T BOTHER DECLARING ANOTHER
54                                     ;DEVICE FATAL ERROR
55 024136          ERRDF  6,EMTO        ;NON-EXISTENT DEVICE ERROR
    024136 104455
    024140 000006
    024142 024204
    024144 000000
56 024146 005237 002350
57 024152          10$: INC     NXMFLG      ;SET THE FLAG
58 024152          PRINTX #FMT0,R1,CSR(R1) ;PRINT THE CSR THAT DOESN'T RESPOND.
    024152 016146 002232
    024156 010146
    024160 012746 024242
    024164 012746 000003
    024170 010600
    024172 104415
    024174 062706 000010
59 024200 012600
60 024202          ENDSRV MOV     (SP)+,R0      ;RESTORE R0
    024202
    024202 000002          L10024:
61                                     RTI
62 024204          103    123    122    EMT0:  .ASCIZ  /CSR ADDRESSING ERROR - TRAP 4/
    024207          040    101    104
    024212          104    122    105
    024215          123    123    111
    024220          116    107    040
    024223          105    122    122
    024226          117    122    040
    024231          055    040    124
    024234          122    101    120
    024237          040    064    000
63 024242          045    123    063    FMT0:  .ASCIZ  /%S3%ACSR (SEL%D1%A) AT %06%A DOES NOT RESPOND%/
    024245          045    101    103
    024250          123    122    040
    024253          050    123    105
    024256          114    045    104
    024261          061    045    101
    024264          051    040    101
    024267          124    040    045
    024272          117    066    045
    024275          101    040    104
    024300          117    105    123
    024303          040    116    117
    024306          124    040    122
    024311          105    123    120
    024314          117    116    104
    
```

64 024317 045 116 000

.EVEN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

.SBTTL TEST 2 - ROM CHECK

```

*****
*          TEST 2 - DMR-11
* ROM CRC/CCITT - CHECK ROM POSITION AND CALCULATE CRC/CCITT. THE
* LAST 4 BYTES CONTAIN INFORMATION ABOUT THE ROM TO CHECK. THE 1ST
* OF THESE BYTES CONTAINS THE ASCII VERSION NUMBER. THE 2ND BYTE
* CONTAINS THE ROM NUMBER. THE 3RD AND 4TH BYTES CONTAIN A NEGATIVE
* CRC/CCITT WORD FOR THE ROM.
*          CHIP ADDRESS RANGE
*          LOCATION  CHIP NO.      BYTE    ADDRESS RANGE
*          E03       0            LOW     0000 - 1777
*          E02       1            HIGH    0000 - 1777
*          E04       2            LOW     2000 - 3777
*          E01       3            HIGH    2000 - 3777
*          E05       4            LOW     4000 - 5777
*          E14       5            HIGH    4000 - 5777
*
***** IMPORTANT !!!!!!!! *****
* FOR THIS TEST TO RUN CORRECTLY, ENSURE THAT SWITCH 1 AT LOCATION
* E85 ON THE M8207 IS ON. IF THIS SWITCH IS OFF, BSEL1 WILL BE
* LOCKED OUT AND THE MAINTENANCE FEATURES WILL NOT BE ENABLED.
*****
*          SUBTEST 1 - ON THE FIRST PASS PRINT THE VERSION # IN EACH ROM
*          SUBTEST 2 - GENERATE THE CRC-CCITT IN EACH ROM AND COMPARE IT
*                      IT AGAINST THE CRC BLASTED IN THE ROM
*          SUBTEST 3 - COMPARE THE ROM # BLASTED IN THE ROM AGAINST THE
*                      EXPECTED ROM #.
*****
    
```

```

BGNTST
BGNSUB
T2::
T2.1:
TRAP C$BSUB
CMP #1,STARES ;IS THIS THE FIRST PASS?
BNE S$ ;IF NOT - SKIP THIS SUBROUTINE.
CLR R4 ;GET VERSION # FROM EACH ROM AND PRINT IT OUT
MOV #1,R5 ;# OF THE 1ST ROM
MOV #1774,ROMADR ;# OF NEXT ROM
PRINTB #FMT1,LOGDEV ;ADDRESS OF BYTE CONTAINING # IN ROMS 0 & 1
MOV LOGDEV,-(SP) ;MICROCODE VERSION
MOV #FMT1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #6,SP

1$:
CALL $ROMO ;GET ROM CONTENTS.
MOVB @BSEL6,REV1 ;SAVE THE ASCII REVISION # OF THE ROM
MOVB @BSEL7,REV2 ;SAVE THE REV. # OF THE NEXT ROM
PRINT ;PRINT
PRINTB #FMT2,R4,#REV1,R5,#REV2
MOV #REV2,-(SP)
MOV R5,-(SP)
    
```

```

024322
024322
024322
024322 104402
024324 022737 000001 002270
024332 001061
024334 005004
024336 012705 000001
024342 012737 001774 002410
024350
024350 013746 002366
024354 012746 025276
024360 012746 000002
024364 010600
024366 104414
024370 062706 000006
024374
024374
024400 117737 155634 025632
024406 117737 155636 025634
024414
024414 012746 025634
024420 010546
    
```



```

93
94
95
96
97 024610 012703 000020
98 024614
99 024614 000241
100 024616 006037 002400
101 024622 006037 002404
102
103
104 024626 102011
105 024630 012702 102010
106 024634 043702 002400
107 024640 042737 102010 002400
108 024646 050237 002400
109 024652
110 024652 000241
111 024654 006037 002402
112 024660 006037 002406
113
114 024664 102011
115
116 024666 012702 102010
117 024672 043702 002402
118 024676 042737 102010 002402
119 024704 050237 002402
120 024710
121 024710 005303
122 024712 001340
123 024714 000707
124 024716
125
126
127
128
129
130 024716 005137 002400
131 024722 023737 002400 002404
132
133 024730 001427
134 024732 005737 002342
135
136 024736 001007
137 024740 012737 000001 002342
138 024746
    024746 104455
    024750 000007
    024752 025564
    024754 000000
139 024756
140 024756
    024756 013746 002404
    024762 013746 002400
    024766 010446
    024770 012746 025422
    024774 012746 000004

```

```

: CRC/CCITT CALCULATION - CONVERT THE WORD (LOWORD & HIWORD) TO
: A SERIAL STREAM FOR CALCULATION.
MOV #16,R3 ;16 BITS TO CONSIDER
25$: CLC ;CLEAR THE CARRY
ROR LOCRC ;ROTATE BIT 0 INTO THE CARRY BIT
ROR LOWORD ;ROTATE BIT 0 INTO C AND THE OLD C INTO BIT15
;ARE THE BITS 15 & BITS 0 THE SAME?
;IF YES (V IS CLEAR), DON'T DO THE CRC
;NOTE: V IS THE EXCLUSIVE OR OF BIT 0 & BIT15.
;CRC/CCITT POLYNOMIAL
BVC 30$
MOV #102010,R2
BIC LOCRC,R2
BIC #102010,LOCRC
BIS R2,LOCRC
30$: CLC ;CLEAR THE CARRY
ROR HICRC ;ROTATE BIT 0 INTO C
ROR HIWORD ;ROTATE OLD C INTO BIT15 (SIGN) & BIT 0 INTO C
;ARE THE BITS 0 OF HICRC & HIWORD THE SAME?
;IF YES (V IS CLEAR), DON'T DO THE CRC.
;NOTE: V IS THE EXCLUSIVE OR OF BIT 0 & BIT15.
;CRC/CCITT POLYNOMIAL
MOV #102010,R2
BIC HICRC,R2
BIC #102010,HICRC
BIS R2,HICRC
35$: DEC R3 ;DO ALL 16 BITS
BNE 25$
BR 20$ ;GET THE CONTENTS OF THE NEXT 2 ROM ADDRESSES.
40$:
:
: AT THIS POINT WE'VE READ THE CONTENTS AND CALCULATED THE CRC FOR
: 2 ROM ROMS (ONE LOW BYTE & ONE HIGH BYTE). ALSO WE'VE READ THE
: CRC BLASTED INTO THE LAST 2 BYTES OF THE ROM (IN LOWORD/HIWORD)
:
COM LOCRC ;COMPLEMENT THE CALCULATED CRC
CMP LOCRC,LOWORD ;IS THE CRC IN ROM THE SAME AS THE
;CALCULATED CRC?
BEQ 50$ ;IF YES - CHECK THE HIGH BYTE CRC (NEXT ROM)
TST FLAG ;HAS AN ERRDF ALREADY BEEN DECLARED (REMEMBER
;WE'RE IN A LOOP)
BNE 41$ ;IF YES, DON'T BOTHER WITH ANOTHER ERRDF.
MOV #1,FLAG ;FLAG THAT ERRDF HAS BEEN DETECTED.
ERRDF 7,EMT1 ;ROM ERROR

TRAP C$ERRDF
.WORD 7
.WORD EMT1
.WORD 0
41$: PRINTB #FMT3,R4,LOCRC,LOWORD

MOV LOWORD,-(SP)
MOV LOCRC,-(SP)
MOV R4,-(SP)
MOV #FMT3,-(SP)
MOV #4,-(SP)

```

```

025000 010600
025002 104414
025004 062706 000012
141 025010 50$:
142 025010 005204 INC R4 ; INCR ROM #
143 025012 005137 002402 COM HICRC ; COMPLEMENT THE CALCULATED CRC FOR THE HI BYTE
144 025016 023737 002402 002406 CMP HICRC,HIWORD ; ROM CRC AND CALCULATED CRC THE SAME?
145 025024 001427 BEQ 60$ ; IF YES - CHECK THE ROM LOCATIONS.
146 025026 005737 002342 TST FLAG ; HAS AN ERRDF ALREADY BEEN DECLARED (REMEMBER
147 ; WE'RE IN A LOOP)
148 025032 001007 BNE 51$ ; IF YES, DON'T BOTHER WITH ANOTHER ERRDF.
149 025034 012737 000001 002342 MOV #1,FLAG ; FLAG THAT ERRDF HAS BEEN DETECTED.
150 025042 ERRDF 7,EMT1 ; ROM ERROR
025042 104455 TRAP C$ERRDF
025044 000007 .WORD 7
025046 025564 .WORD EMT1
025050 000000 .WORD 0
151 025052 51$:
152 025052 PRINTB #FMT3,R4,HICRC,HIWORD
025052 013746 002406 MOV HIWORD,-(SP)
025056 013746 002402 MOV HICRC,-(SP)
025062 010446 MOV R4,-(SP)
025064 012746 025422 MOV #FMT3,-(SP)
025070 012746 000004 MOV #4,-(SP)
025074 010600 MOV SP,RO
025076 104414 TRAP C$PNTB
025100 062706 000012 ADD #12,SP
153 025104 60$:
154 025104 022704 000005 CMP #5,R4 ; IF WE'VE DONE ROMS 0-5, WE'RE DONE.
155 025110 001403 BEQ 70$ ; EXIT WHEN DONE
156 025112 005204 INC R4 ; CHECK THE NEXT ROM.
157 025114 000137 024514 JMP 10$
158 025120 70$:
159 ENDSUB
160 025120
025120 104403 L10027: TRAP C$ESUB
161 BGNSUB
162 025122
025122 T2.3: TRAP C$BSUB
025122 104402
163 025124 005037 002342 CLR FLAG ; CLEAR FLAG
164 025130 005004 CLR R4 ; BEGIN AT ROM 0
165 025132 012737 001775 002410 MOV #1775,ROMADR ; ADDRESS OF BYTE CONTAINING ROM #
166 025140 10$:
167 025140 CALL $ROMO ; GET ROM CONTENTS
168 025144 117701 155070 MOVB @BSEL6,R1 ; SAVE THE CONTENTS OF THE LOW BYTE
169 ; FOR ROMS 0,2,4
170 025150 000402 BR 17$
171 025152 15$:
172 025152 117701 155072 MOVB @BSEL7,R1 ; SAVE THE CONTENTS OF THE HIGH BYTE
173 ; FOR ROMS 1,3,5
174 025156 17$:
175 025156 042701 177760 BIC #^C17,R1 ; CONVERT THE ASCII BYTE TO AN OCTAL WORD.
176 025162 020104 CMP R1,R4 ; IS THIS THE EXPECTED ROM #
177 025164 001427 BEQ 20$ ; IF YES - OK.
178 025166 005737 002342 TST FLAG ; HAS AN ERRDF ALREADY BEEN DECLARED (REMEMBER
    
```

```

179
180 025172 001007
181 025174 012737 000001 002342
182 025202
    025202 104455
    025204 000007
    025206 025604
    025210 000000
183 025212
184 025212
    025212 010446
    025214 010146
    025216 005046
    025220 156416 025623
    025224 012746 025506
    025230 012746 000004
    025234 010600
    025236 104414
    025240 062706 000012
185 025244
186 025244 022704 000005
187 025250 001410
188 025252 005204
189 025254 032704 000001
190 025260 001334
191
192 025262 062737 002000 002410
193 025270 000723
194 025272
195 025272
    025272
    025272 104403
196
197 025274
    025274
    025274 104401
198 025276 045 116 045
    025301 101 115 111
    025304 103 122 117
    025307 103 117 104
    025312 105 040 122
    025315 105 126 111
    025320 123 111 117
    025323 116 040 111
    025326 116 040 125
    025331 116 111 124
    025334 045 104 063
    025337 045 101 072
    025342 045 116 000
199 025345 045 101 122
    025350 117 115 045
    025353 104 062 045
    025356 101 040 055
    025361 040 122 105
    025364 126 056 040
    025367 045 124 045
    025372 116 045 101

;WE'RE IN A LOOP)
;IF YES, DON'T BOTHER WITH ANOTHER ERRDF.
;FLAG THAT ERRDF HAS BEEN DETECTED.
;ROM ERROR
TRAP C$ERRDF
.WORD 7
.WORD EMT2
.WORD 0
18$:
PRINTB #FMT4,<B,ROMLOC(R4)>,R1,R4
MOV R4,-(SP)
MOV R1,-(SP)
CLR -(SP)
BISB ROMLOC(R4),(SP)
MOV #FMT4,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #12,SP
20$:
CMP #5,R4
BEQ 30$
INC R4
BIT #BIT0,R4
BNE 15$
;DID WE FINISH THE LAST ROM?
;IF YES - SKIP TO THE END
;POINT TO THE NEXT ROM #
;IS THIS AN ODD #
;IF YES GO BACK AND READ THE HIGH BYTE
ADD #2000,ROMADR
BR 10$
;INCR. ADDRESS POINTER TO NEXT ROM #.
30$:
ENDSUB
L10030:
TRAP C$ESUB
ENDTST
L10025:
TRAP C$ETST
FMT1: .ASCIZ /%N%AMICROCODE REVISION IN UNIT%D3%A:%N/
FMT2: .ASCIZ /%AROM%D2%A - REV. %T%N%AROM%D2%A - REV. %T%N/

```

	025375	122	117	115
	025400	045	104	062
	025403	045	101	040
	025406	055	040	122
	025411	105	126	056
	025414	040	045	124
	025417	045	116	000
200	025422	045	101	122
	025425	117	115	045
	025430	104	062	045
	025433	101	072	040
	025436	103	101	114
	025441	103	125	101
	025444	124	105	104
	025447	040	103	122
	025452	103	040	075
	025455	045	117	066
	025460	045	101	040
	025463	040	103	122
	025466	103	040	111
	025471	116	040	122
	025474	117	115	040
	025477	075	045	117
	025502	066	045	116
	025505	000		
201	025506	045	101	105
	025511	045	104	062
	025514	045	101	040
	025517	111	123	040
	025522	122	117	115
	025525	040	045	104
	025530	061	045	101
	025533	040	050	123
	025536	110	117	125
	025541	114	104	040
	025544	102	105	040
	025547	122	117	115
	025552	040	045	104
	025555	061	045	101
	025560	051	045	116
	025563	000		
202				
203	025564	103	122	103
	025567	055	103	103
	025572	111	124	124
	025575	040	105	122
	025600	122	117	122
	025603	000		
204	025604	114	117	103
	025607	101	124	111
	025612	117	116	040
	025615	105	122	122
	025620	117	122	000
205				
206	025623	003	002	004
	025626	001	005	016
207				

FMT3: .ASCIZ /%AROM%D2%A: CALCUATED CRC =%06%A CRC IN ROM -%06%N/

FMT4: .ASCIZ /%AE%D2%A IS ROM %D1%A (SHOULD BE ROM %D1%A)%N/

EMT1: .ASCIZ /CRC-CCITT ERROR/

EMT2: .ASCIZ /LOCATION ERROR/

ROMLOC: .BYTE 3,2,4,1,5,14. ;ROM 0 = ROM LOCATION 3 ETC.

.EVEN

208 025632 000000
209 025634 000000
210
211
212

REV1: .WORD 0
REV2: .WORD 0

;ASCII VALUE OF THE REV. NUMBER
;ASCII VALUE OF THE REV. NUMBER

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

```
.SBTTL          TEST 3 - MASTER CLEAR AND MICROTST
*****
*              TEST 3 - DMR-11
* MASTER CLEAR
* THIS TEST WILL ISSUE 2 MASTER CLEARS.  EACH CALL TO THE MASTER
* CLEAR ROUTINE WILL ENSURE THAT THE RUN BIT WILL BE SET.  ALSO
* THE MASTER CLEAR WILL CAUSE THE DIAGNOSTIC MICROTSTES TO BE
* RUN WHEN THE MICRODIAGNOSTIC BIT (BIT 13 IN SEL0) IS CORRECTLY
* SET OR CLEARED.  BECAUSE THE RUNNING OF MICROTSTES DEPENDS ON THE
* EXCLUSIVE OR OF THE HARDWARE SWITCH 10 ON E134 OF THE M8203 AND
* THE MICRODIAGNOSTIC BIT, WE CAN'T KNOW WHETHER THE SETTING OR
* CLEARING OF BIT 13 WILL RESULT IN THE RUNNING OF MICROTSTES.
* THEREFORE THE MASTER CLEAR SUBROUTINE WILL TOGGLE (I.E. SET
* BIT 13 ONLY ON EVERY OTHER MASTER CLEAR) THE SOFTWARE BIT.
* THIS WILL ENSURE THAT REGARDLESS OF THE POSITION OF THE
* HARDWARE SWITCH, MICROTSTES WILL BE RUN EVERY OTHER MASTER CLEAR.
* WHEN RUNNING THIS TEST, WE EXPECT TO ADD THE RESULTS OF BSEL3
* AFTER EACH MASTER CLEAR.
* BSEL3 = 100      - MICROTSTES DISABLED
* BSEL3 = 200      - MICROTSTES RUN SUCCESFULLY
* IF THE RESULT OF THE 2 MASTER CLEARS IS NOT 300, AN ERROR IS
* REPORTED.
*
* ADDITIONALLY THIS ROUTINE WILL REPORT WHENEVER THE RESULT OF
* BSEL3 IS 0.  THIS WILL MEAN THAT THE DEVICE IS NOT A DMR
* (I.E. DMC)
*****
BGNTST
```

```
025636          CLEAR          ;MACRO FOR MASTER CLEAR
025636          JSR          PC, $MSCLR      ;**** MACRO EXPANSION ****
025636          ;ISSUE A DMR MASTER CLEAR
025636          ;****

025636 004737 011070          ;IF ERROR, BR TO TEST END.

025642          ESCAPE TST          ;IF ERROR, BR TO TEST END.
025642 104410          TRAP          C$ESCAPE
025644 000072          .WORD          L10031-.

025646 105777 154372          TSTB          @BSEL3          ;IS THERE A DMR RESPONSE?
025652 001011          BNE          1$
025654          PRINTB          #FMG19          ;REPORT DEVICE NOT DMR.
025654 012746 017531          MOV          #FMG19,-(SP)
025660 012746 000001          MOV          #1,-(SP)
025664 010600          MOV          SP,R0
025666 104414          TRAP          C$PNTB
025670 062706 000004          ADD          #4,SP

025674 000420          BR          5$

1$:
025676 117701 154342          MOVB          @BSEL3,R1          ;SAVE THE RESULT OF THE FIRST MASTER CLEAR.
025702          CLEAR          ;MASTER CLEAR AGAIN.
025702          JSR          PC, $MSCLR      ;**** MACRO EXPANSION ****
025702          ;ISSUE A DMR MASTER CLEAR
025702          ;****

025706          ESCAPE TST          ;IF ERROR, BR TO TEST END.
025706 104410          TRAP          C$ESCAPE
025710 000026          .WORD          L10031-.
```



```

42 025712 117702 154326      MOVB  @SEL3,R2      ;SAVE THE RESULTS OF THE SECOND MASTER CLEAR
43 025716 060102      ADD   R1,R2        ;ADD THE RESULTS OF THE 2 CLEARS
44                               ;NOTE: ONE SHOULD BE 100 - MICRO TESTS NOT
45                               ;ENABLED AND ONE SHOULD BE 200 - MICRO TESTS
46                               ;SUCCESFULLY RUN.
47 025720 122702 000300      CMPB  #300,R2      ;WAS THE MICROTEST COMPLETED?
48 025724 001404      BEQ   S$          ;IF YES - OK
49 025726      ERRDF 3,EMT3,ERRG3 ;MICROTEST NOT COMPLETED
    025726 104455      TRAP  C$ERDF
    025730 000003      .WORD 3
    025732 025740      .WORD EMT3
    025734 015242      .WORD ERRG3
50 025736      S$:
51 025736      ENDTST
    025736      L10031:
    025736 104401      TRAP  C$ETST
52
53 025740      115      111      103  EMT3: .ASCIZ /MICROTEST NOT COMPLETED/
    025743      122      117      124
    025746      105      123      124
    025751      040      116      117
    025754      124      040      103
    025757      117      115      120
    025762      114      105      124
    025765      105      104      000
54                               .EVEN
    
```

```

1      .SBTTL          TEST 4 - BASE IN COMMAND
2
3      :*****
4      :*              TEST 4 - DMR-11
5      :*      BASE IN COMMANDS
6      :*
7      :*      SUBTEST 1 - ISSUE A BASE IN - DMR MODE.
8      :*              ENSURE THAT THE DMR MODE BIT (BIT 4) IS SET IN
9      :*              THE MICROCODE SCRATCH PAD 7 AND THAT THE DDCMP
10     :*              MESSAGE VARIABLES ARE PROPERLY INITIALIZED.
11     :*      SUBTEST 2 - ISSUE A BASE IN - DMC MODE.
12     :*              ENSURE THAT THE DMC MODE BIT (BIT 4) IS CLEAR IN
13     :*              THE MICROCODE SCRATCH PAD 7 AND THAT THE DDCMP
14     :*              MESSAGE VARIABLES ARE PROPERLY INITIALIZED.
15     :*
16     :*****
17     BGNTST
18     BGNSUB
19     CLEAR          ;MACRO FOR MASTER CLEAR COMMAND
20     JSR    PC, $MSCLR ;**** MACRO EXPANSION ****
21     ;ISSUE A DMR MASTER CLEAR
22     ;****
23     ESCAPE TST     ;IF ERROR, BR TO TEST END
24     TRAP      C$ESCAPE
25     .WORD    L10032-.
26     BASEIN 0,BASE,DMR ;BASE IN COMMAND WITH NO MAINTENANCE,
27     ;BASE=BASE TABLE ADDRESS, AND DMR-11 MODE
28     ;**** MACRO EXPANSION ****
29     JSR    PC, $BASEI ;CALL BASE IN ROUTINE
30     .WORD  0          ;MAINTENANCE MODE BITS TO SET IN BSEL1
31     .WORD  BASE       ;BASE TABLE ADDRESS
32     .WORD  DMR        ;MODE
33     ;****
34     ESCAPE TST     ;IF ERROR, BR TO TEST END
35     TRAP      C$ESCAPE
36     .WORD    L10032-.
37     SHUTDN
38     ;**** MACRO EXPANSION ****
39     ;DMR HALT ROUTINE.
40     ;****
41     JSR    PC, $HALT ;SEE IF THE DMR MODE BIT IS SET IN THE
42     ;DMR SCRATCH PAD REGISTER 7 (BASE TABLE
43     ;LOCATION CONTAINS AN IMAGE OF SP7)
44     ;OK IF SET - BR
45     BNE    10$
46     ERRDF  20,EMT4
47     TRAP      C$ERRDF
48     .WORD    20
49     .WORD    EMT4
50     .WORD    0
51     10$:
52     ;CHECK MESSAGE EXCHANGE VALUES
53     ;IN THE BASE TABLE.

```

TEST 4 - BASE IN COMMAND

35	026044	105737	002676	TSTB	BASE+R								
36	026050	001015		BNE	20\$								
37	026052	105737	002677	TSTB	BASE+N								
38	026056	001012		BNE	20\$								
39	026060	105737	002700	TSTB	BASE+A								
40	026064	001007		BNE	20\$								
41	026066	122737	000001 002701	CMPB	#1,BASE+T								
42	026074	001003		BNE	20\$								
43	026076	105737	002702	TSTB	BASE+X								
44	026102	001404		BEQ	30\$								
45	026104					20\$:							
46	026104			ERRDF	20,EMT5,ERRT1								
	026104	104455									TRAP	C\$ERDF	
	026106	000024									.WORD	20	
	026110	026513									.WORD	EMT5	
	026112	026246									.WORD	ERRT1	
47	026114					30\$:							
48	026114			ENDSUB									
	026114												
	026114	104403											
49													
50	026116			BGNSUB									
	026116												
	026116	104402											
51	026120			CLEAR									
	026120	004737	011070	JSR	PC, \$MSCLR								
52													
53	026124			ESCAPE	TST								
	026124	104410											
	026126	000116									TRAP	C\$ESCAPE	
54											.WORD	L10032-	
55													
56	026130			BASEIN	0,BASE,0								
	026130	004737	011266	JSR	PC, \$BASEI								
	026134	000000			.WORD 0								
	026136	002634			.WORD BASE								
	026140	000000			.WORD 0								
57													
58	026142			ESCAPE	TST								
	026142	104410											
	026144	000100									TRAP	C\$ESCAPE	
59	026146			SHUTDN							.WORD	L10032-	
	026146	004737	012564	JSR	PC, \$HALT								
60	026152	132737	000020 002726	BITB	#BIT4,BASE+ISP7								
61													
62													
63	026160	001404		BEQ	10\$								
64	026162			ERRDF	20,EMT6								
	026162	104455									TRAP	C\$ERDF	
	026164	000024									.WORD	20	
	026166	026561									.WORD	EMT6	

```

026170 000000 .WORD 0
65 026172 10$:
66
67 ;CHECK MESSAGE EXCHANGE VALUES
68 026172 105737 002676 TSTB BASE+R ;IN THE BASE TABLE.
69 026176 001015 BNE 20$ ;#R (MESSAGE RECEIVED) = 0?
70 026200 105737 002677 TSTB BASE+N ;ERROR IF NON ZERO
71 026204 001012 BNE 20$ ;#N (MESSAGE TRANSMITTED) = 0?
72 026206 105737 002700 TSTB BASE+A ;ERROR IF NON ZERO
73 026212 001007 BNE 20$ ;#A (MESSAGE ACKNOWLEDGED) = 0?
74 026214 122737 000001 002701 CMPB #1,BASE+T ;#T (NEXT MESSAGE # TRANSMITTED) - 1?
75 026222 001003 BNE 20$ ;ERROR IF NOT EQUAL TO 1.
76 026224 105737 002702 TSTB BASE+X ;#X (LAST MESSAGE TRANSMITTED) = 0?
77 026230 001404 BEQ 30$
78 026232 20$:
79 026232 ERRDF 20,EMT5,ERRT1
026232 104455 TRAP C$ERDF
026234 000024 .WORD 20
026236 026513 .WORD EMT5
026240 026246 .WORD ERRT1
80 026242 30$:
81 026242 ENDSUB
026242 L10034:
026242 104403 TRAP C$ESUB
82
83 026244 ENDTST
026244 L10032:
026244 104401 TRAP C$ETST
84
85 026246 BGNMSG ERRT1
026246 ERRT1::
86 026246 105737 002676 TSTB BASE+R ;IS #R = 0?
87 026252 001413 BEQ 1$ ;OK - IF ZERO
88 026254 PRINTB #FMT5,<B,BASE+R> ;PRINT #R
026254 005046 CLR -(SP)
026256 153716 002676 BISB BASE+R,(SP)
026262 012746 026625 MOV #FMT5,-(SP)
026266 012746 000002 MOV #2,-(SP)
026272 010600 MOV SP,R0
026274 104414 TRAP C$PNTB
026276 062706 000006 ADD #6,SP
89 026302 1$:
90 026302 105737 002677 TSTB BASE+N ;IS #N = 0?
91 026306 001413 BEQ 2$ ;OK - IF ZERO
92 026310 PRINTB #FMT6,<B,BASE+N> ;PRINT #N
026310 005046 CLR -(SP)
026312 153716 002636 BISB BASE+2,(SP)
026316 012746 026656 MOV #FMT6,-(SP)
026322 012746 000002 MOV #2,-(SP)
026326 010600 MOV SP,R0
026330 104414 TRAP C$PNTB
026332 062706 000006 ADD #6,SP
93 026336 2$:
94
95 026336 105737 002700 TSTB BASE+A ;IS #A = 0?
96 026342 001413 BEQ 3$ ;OK - IF ZERO
97 026344 PRINTB #FMT7,<B,BASE+A> ;PRINT #A
    
```

026344	005046								CLR	-(SP)
026346	153716	002700							BISB	BASE+A,(SP)
026352	012746	026707							MOV	#FMT7,-(SP)
026356	012746	000002							MOV	#2,-(SP)
026362	010600								MOV	SP,R0
026364	104414								TRAP	C\$PNTB
026366	062706	000006							ADD	#6,SP
98 026372				3\$:						
99 026372	122737	000001	002701		CMPB	#1,BASE+T		:IS #T = 1?		
100 026400	001413				BEQ	4\$:OK - IF ONE		
101 026402					PRINTB	#FMT8,<B,BASE+T>		:PRINT #T		
026402	005046								CLR	-(SP)
026404	153716	002701							BISB	BASE+T,(SP)
026410	012746	026740							MOV	#FMT8,-(SP)
026414	012746	000002							MOV	#2,-(SP)
026420	010600								MOV	SP,R0
026422	104414								TRAP	C\$PNTB
026424	062706	000006							ADD	#6,SP
102 026430				4\$:						
103 026430	105737	002702			TSTB	BASE+X		:IS #X = 0?		
104 026434	001413				BEQ	5\$:OK - IF ZERO		
105 026436					PRINTB	#FMT9,<B,BASE+X>		:PRINT #X		
026436	005046								CLR	-(SP)
026440	153716	002702							BISB	BASE+X,(SP)
026444	012746	027003							MOV	#FMT9,-(SP)
026450	012746	000002							MOV	#2,-(SP)
026454	010600								MOV	SP,R0
026456	104414								TRAP	C\$PNTB
026460	062706	000006							ADD	#6,SP
106 026464				5\$:						
107 026464				ENDMSG						
026464										
026464	104423							L10035:	TRAP	C\$MSG
108 026466										
109 026466	104	115	122	EMT4:	.ASCIZ	/DMR MODE BIT NOT SET/				
026471	040	115	117							
026474	104	105	040							
026477	102	111	124							
026502	040	116	117							
026505	124	040	123							
026510	105	124	000							
110 026513	104	104	103	EMT5:	.ASCIZ	/DDCMP MESSAGE VARIABLE(S) NOT CORRECT/				
026516	115	120	040							
026521	115	105	123							
026524	123	101	107							
026527	105	040	126							
026532	101	122	111							
026535	101	102	114							
026540	105	050	123							
026543	051	040	116							
026546	117	124	040							
026551	103	117	122							
026554	122	105	103							
026557	124	000								
111 026561	104	115	103	EMT6:	.ASCIZ	/DMC MODE - DMR MODE BIT NOT CLEARED/				
026564	040	115	117							
026567	104	105	040							

	026572	055	040	104	
	026575	115	122	040	
	026600	115	117	104	
	026603	105	040	102	
	026606	111	124	040	
	026611	116	117	124	
	026614	040	103	114	
	026617	105	101	122	
	026622	105	104	000	
112					
113	026625	045	101	043	FMT5: .ASCIZ /%A#R (MSG. RCVD) = %D3%N/
	026630	122	040	050	
	026633	115	123	107	
	026636	056	040	122	
	026641	103	126	104	
	026644	051	040	075	
	026647	040	045	104	
	026652	063	045	116	
	026655	000			
114	026656	045	101	043	FMT6: .ASCIZ /%A#N (MSG. XMIT) = %D3%N/
	026661	116	040	050	
	026664	115	123	107	
	026667	056	040	130	
	026672	115	111	124	
	026675	051	040	075	
	026700	040	045	104	
	026703	063	045	116	
	026706	000			
115	026707	045	101	043	FMT7: .ASCIZ /%A#A (MSG. ACK) = %D3%N/
	026712	101	040	050	
	026715	115	123	107	
	026720	056	040	101	
	026723	103	113	051	
	026726	040	040	075	
	026731	040	045	104	
	026734	063	045	116	
	026737	000			
116	026740	045	101	043	FMT8: .ASCIZ /%A#T (NEXT MSG TO XMIT) = %D3%N/
	026743	124	040	050	
	026746	116	105	130	
	026751	124	040	115	
	026754	123	107	040	
	026757	124	117	040	
	026762	130	115	111	
	026765	124	051	040	
	026770	040	040	040	
	026773	075	040	045	
	026776	104	063	045	
	027001	116	000		
117	027003	045	101	043	FMT9: .ASCIZ /%A#X (LAST COMPLETED XMIT) = %D3%N/
	027006	130	040	050	
	027011	114	101	123	
	027014	124	040	103	
	027017	117	115	120	
	027022	114	105	124	
	027025	105	104	040	
	027030	130	115	111	

027033	124	051	040
027036	075	040	045
027041	104	063	045
027044	116	000	

118
119
120
121

.EVEN

```

1      .SBTTL          TEST 5 - DMR COMMANDS
2
3      :*****
4      :*              TEST 5 - DMR-11
5      :* DMR COMMANDS
6      :* SUBTEST 1 - ISSUE AN ENABLE EXTENDED ERROR COMMAND AND CHECK THAT
7      :*              THE EXT. ENABLE BIT IS SET IS SCRATCH PAD 13. THEN
8      :*              DISABLE EXTENDED ERROR AND CHECK THAT THE ENABLE BIT
9      :*              IS CLEAR.
10     :* SUBTEST 2 - SET REP/SEL TIMER VALUE AND SET THE DMR THRESHOLD
11     :*              VALUES. CHECK THAT THE VALUES ARE CORRECT IN
12     :*              THE BASE TABLE AFTER HALTING THE DMR.
13     :*
14     :*
15     :*****
16     BGNTST
17     BGNSUB
18     CLEAR          ;MACRO FOR MASTER CLEAR COMMAND
19     JSR            PC, $MSCLR
20     ESCAPE TST     ;IF ERROR, BR TO TEST END
21     BASEIN        ;BASE IN COMMAND WITH LINE UNIT LOOP,
22     JSR            PC, $BASEI
23     ESCAPE TST     ;IF ERROR, BR TO TEST END
24     DMRIN         EXERR ;ENABLE EXTENDED ERROR NOTIFICATION
25     JSR            PC, $DMRIN
26     ESCAPE TST     ;IF ERROR, BR TO TEST END
27     SHUTDN        ;HALT THE DMR
28     JSR            PC, $HALT
29     ESCAPE TST     ;IF ERROR, BR TO TEST END
    
```

027046
 027046
 027046
 027046 104402
 027050 004737 011070
 027054 104410
 027056 000330
 027060 004737 011266
 027064 004000
 027066 002634
 027070 000522
 027072 104410
 027074 000312
 027076 004737 012070
 027102 000006
 027104 000000
 027106 000000
 027110 104410
 027112 000274
 027114 004737 012564
 027120 104410

T5::
 T5.1:
 TRAP C\$BSUB
 TRAP C\$ESCAPE
 .WORD L10036-.
 TRAP C\$ESCAPE
 .WORD L10036-.
 TRAP C\$ESCAPE
 .WORD L10036-.
 TRAP C\$ESCAPE


```

30 027122 000264
31 027124 132737 000001 002732 BITB #BIT0,BASE+ISP13 ;CHECK EXT ENABLE BIT IN THE BASE TABLE.
32 027132 001005 BNE 10$ ;IMAGE OF SCRATCH PAD 13.
33 027134 104455 ERRDF 24,EMT7 ;BIT SET - OK.
    027134 104455 ;ERROR EXT ENABLE CLEAR
    027136 000030 TRAP C$ERDF
    027140 027716 .WORD 24
    027142 000000 .WORD EMT7
34 027144 000430 BR 20$ .WORD 0
35 027146 10$:
36 027146 BASEIN LPLU,BASE,RES!DMR ;BASE IN COMMAND WITH RESUME SET.
    027146 004737 011266 JSR PC, $BASEI ;**** MACRO EXPANSION ****
    027152 004000 .WORD LPLU ;CALL BASE IN ROUTINE
    027154 002634 .WORD BASE ;MAINTENANCE MODE BITS TO SET IN BSEL1
    027156 010522 .WORD RES!DMR ;BASE TABLE ADDRESS
    ;MODE
    ;****
37
38 027160 DMRIN DXERR ;DISABLE EXTENDED ERROR NOTIFICATION.
    027160 004737 012070 JSR PC, $DMRIN ;**** MACRO EXPANSION ****
    027164 000007 .WORD DXERR ;CALL DMR MODE INPUT ROUTINE
    027166 000000 .WORD 0 ;INPUT COMMAND
    027170 000000 .WORD 0 ;NO SEL4
    ;NO SEL6
    ;****
39
40 027172 ESCAPE TST ;IF ERROR, BR TO TEST END
    027172 104410
    027174 000212 TRAP C$ESCAPE
41 027176 SHUTDN ;HALT THE DMR
    027176 004737 012564 JSR PC, $HALT ;**** MACRO EXPANSION ****
    ;DMR HALT ROUTINE.
    ;****
42 027202 ESCAPE TST ;IF ERROR, BR TO TEST END.
    027202 104410 TRAP C$ESCAPE
    027204 000202 .WORD L10036-.
43 027206 132737 000001 002732 BITB #BIT0,BASE+ISP13 ;CHECK EXT ENABLE BIT IN THE BASE TABLE.
44 ;IMAGE OF SCRATCH PAD 13.
45 027214 001404 BEQ 20$ ;IF CLEAR OK
46 027216 104455 ERRDF 24,EMT7 ;ERROR EXT ENABLE SET
    027216 104455 TRAP C$ERDF
    027220 000030 .WORD 24
    027222 027716 .WORD EMT7
    027224 000000 .WORD 0
47 027226 20$:
48 027226 ENDSUB
    027226 104403 L10037: TRAP C$ESUB
49
50 027230 BGNSUB
    027230 104402 T5.2: TRAP C$BSUB
51 027232 CLEAR ;MACRO FOR MASTER CLEAR COMMAND
    027232 004737 011070 JSR PC, $MSCLR ;**** MACRO EXPANSION ****
    ;ISSUE A DMR MASTER CLEAR
    ;****

```

TEST 5 - DMR COMMANDS

```

52
53 027236          ESCAPE TST          ;IF ERROR, BR TO TEST END
    027236 104410
    027240 000146          TRAP          C$ESCAPE
                                .WORD          L10036-.
54
55 027242          BASEIN              ;BASE IN COMMAND WITH LINE UNIT LOOP,
    027242 004737 011266  JSR      PC, $BASEI      ;**** MACRO EXPANSION ****
    027246 004000          .WORD      LPLU      ;CALL BASE IN ROUTINE WITH DEFAULTS
    027250 002634          .WORD      BASEF     ;SET LINE UNIT LOOP
    027252 000522          .WORD      DMR      ;BASE TABLE ADDRESS
                                ;DMR-11 MODE
                                ;****          ****
56
57 027254          ESCAPE TST          ;IF ERROR, BR TO TEST END
    027254 104410
    027256 000130          TRAP          C$ESCAPE
                                .WORD          L10036-.
58 027260          DMRIN      TIMER,0,54 ;SET REP/SELECT TIMER VALUE
    027260 004737 012070  JSR      PC, $DMRIN      ;**** MACRO EXPANSION ****
    027264 000012          .WORD      TIMER     ;CALL DMR MODE INPUT ROUTINE
    027266 000000          .WORD      0          ;INPUT COMMAND
    027270 000054          .WORD      54         ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
                                ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
                                ;****          ****
59
60 027272          ESCAPE TST          ;IF ERROR, BR TO TEST END
    027272 104410
    027274 000112          TRAP          C$ESCAPE
                                .WORD          L10036-.
61
62
63
64
65
66
67 027276          DMRIN      THRESH,5403,2015 ;SET THRESHOLD VALUES AS FOLLOWS:
    027276 004737 012070  JSR      PC, $DMRIN      ;**** MACRO EXPANSION ****
    027302 000013          .WORD      THRESH     ;CALL DMR MODE INPUT ROUTINE
    027304 005403          .WORD      5403       ;INPUT COMMAND
    027306 002015          .WORD      2015       ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
                                ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
                                ;****          ****
68
69 027310          ESCAPE TST          ;IF ERROR, BR TO TEST END
    027310 104410
    027312 000074          TRAP          C$ESCAPE
                                .WORD          L10036-.
70 027314          SHUTDN              ;HALT THE DMR.
    027314 004737 012564  JSR      PC, $HALT      ;**** MACRO EXPANSION ****
                                ;DMR HALT ROUTINE.
                                ;****          ****
71 027320          ESCAPE TST          ;IF ERROR, BR TO TEST END
    027320 104410
    027322 000064          TRAP          C$ESCAPE
                                .WORD          L10036-.
72 027324          CMPB      #54,BASE+PRETIM ;CHECK REP/SEL TIME IN BASE TABLE.
    027324 122737 000054 002711 BNE      10$          ;IF NOT 54, BR TO ERROR.
73 027332          CMPB      #15,BASE+TH3L ;CHECK REP. THRESH. IN BASE TABLE.
    027332 122737 000015 002720 BNE      10$          ;IF NOT 15, BR TO ERROR.
74 027334          CMPB      #3,BASE+TH1L ;CHECK NAK RCVD. THRESH. IN BASE TABLE.
    027334 122737 000003 002714 BNE      10$          ;IF NOT 3, BR TO ERROR.
75 027342
76 027344
77 027352

```

78	027354	122737	000013	002716	CMPB	#13,BASE+TH2L	;CHECK NAK SENT THRESH. IN BASE TABLE.		
79	027362	001004			BNE	10\$;IF NOT 13, BR TO ERROR		
80	027364	122737	000004	002722	CMPB	#4,BASE+TH4L	;CHECK NO BUF. THRESH. IN BASE TABLE.		
81	027372	001404			BEQ	20\$;IF 4, ALL CHECKS OK - EXIT		
82	027374					10\$:			
83	027374				ERRDF	24,EMT8,ERRT3			
	027374	104455						TRAP	C\$ERDF
	027376	000030						.WORD	24
	027400	027747						.WORD	EMT8
	027402	027410						.WORD	ERRT3
84	027404					20\$:			
85	027404				ENDSUB				
	027404							L10040:	
86	027406	104403			ENDTST			TRAP	C\$ESUB
	027406							L10036:	
	027406	104401						TRAP	C\$ETST
87	027410				BGNMSG	ERRT3			
88	027410							ERRT3::	
89	027410				PRINTB	#FMG1,@SELO,@SEL2	;PRINT SELO AND SEL2		
	027410	017746	152620					MOV	@SEL2,-(SP)
	027414	017746	152612					MOV	@SELO,-(SP)
	027420	012746	016334					MOV	#FMG1,-(SP)
	027424	012746	000003					MOV	#3,-(SP)
	027430	010600						MOV	SP,R0
	027432	104414						TRAP	C\$PNTB
	027434	062706	000010					ADD	#10,SP
90	027440				PRINTB	#FMT11,<B,BASE+ISP13>	;PRINT OUT THE IMAGE OF SCRATCH PAD 13.		
	027440	005046						CLR	-(SP)
	027442	153716	002732					BISB	BASE+ISP13,(SP)
	027446	012746	030004					MOV	#FMT11,-(SP)
	027452	012746	000002					MOV	#2,-(SP)
	027456	010600						MOV	SP,R0
	027460	104414						TRAP	C\$PNTB
	027462	062706	000006					ADD	#6,SP
91	027466	122737	00C054	002711	CMPB	#54,BASE+PRETIM	;IS REP/SEL TIME OK?		
92	027474	001413			BEQ	1\$;BR IF OK		
93	027476				PRINTB	#FMT12,<B,BASE+PRETIM>	;PRINT IT OUT.		
	027476	005046						CLR	-(SP)
	027500	153716	002711					BISB	BASE+PRETIM,(SP)
	027504	012746	030035					MOV	#FMT12,-(SP)
	027510	012746	000002					MOV	#2,-(SP)
	027514	010600						MOV	SP,R0
	027516	104414						TRAP	C\$PNTB
	027520	062706	000006					ADD	#6,SP
94	027524					1\$:			
95	027524	122737	000003	002714	CMPB	#3,BASE+TH1L	;IS NAK RCVD OK?		
96	027532	001413			BEQ	2\$;BR IF OK.		
97	027534				PRINTB	#FMT13,<B,BASE+TH1L>	;PRINT IT OUT		
	027534	005046						CLR	-(SP)
	027536	153716	002714					BISB	BASE+TH1L,(SP)
	027542	012746	030072					MOV	#FMT13,-(SP)
	027546	012746	000002					MOV	#2,-(SP)
	027552	010600						MOV	SP,R0
	027554	104414						TRAP	C\$PNTB
	027556	062706	000006					ADD	#6,SP

98	027562			2\$:				
99	027562	122737	000013	002716	CMPB	#13,BASE+TH2L	;IS NAK SENT OK?	
100	027570	001413			BEQ	3\$;BR IF OK.	
101	027572				PRINTB	#FMT14,<B,BASE+TH2L>	;PRINT IT OUT	
	027572	005046						CLR -(SP)
	027574	153716	002716					BISB BASE+TH2L,(SP)
	027600	012746	030127					MOV #FMT14,-(SP)
	027604	012746	000002					MOV #2,-(SP)
	027610	010600						MOV SP,R0
	027612	104414						TRAP C\$PNTB
	027614	062706	000006					ADD #6,SP
102	027620			3\$:				
103	027620	122737	000015	002720	CMPB	#15,BASE+TH3L	;IS REP LEVEL OK?	
104	027626	001413			BEQ	4\$;BR IF OK.	
105	027630				PRINTB	#FMT15,<B,BASE+TH3L>	;PRINT IT OUT	
	027630	005046						CLR -(SP)
	027632	153716	002720					BISB BASE+TH3L,(SP)
	027636	012746	030164					MOV #FMT15,-(SP)
	027642	012746	000002					MOV #2,-(SP)
	027646	010600						MOV SP,R0
	027650	104414						TRAP C\$PNTB
	027652	062706	000006					ADD #6,SP
106	027656			4\$:				
107	027656	122737	000004	002722	CMPB	#4,BASE+TH4L	;IS NO BUFFER LEVEL OK?	
108	027664	001413			BEQ	5\$;BR IF OK.	
109	027666				PRINTB	#FMT16,<B,BASE+TH4L>	;PRINT IT OUT	
	027666	005046						CLR -(SP)
	027670	153716	002722					BISB BASE+TH4L,(SP)
	027674	012746	030221					MOV #FMT16,-(SP)
	027700	012746	000002					MOV #2,-(SP)
	027704	010600						MOV SP,R0
	027706	104414						TRAP C\$PNTB
	027710	062706	000006					ADD #6,SP
110	027714			5\$:				
111	027714			ENDMSG				
	027714							
	027714	104423						L10041: TRAP C\$MSG
112								
113								
114	027716	105	130	124	EMT7:	.ASCIZ	/EXT. ERROR BIT INCORRECT/	
	027721	056	040	105				
	027724	122	122	117				
	027727	122	040	102				
	027732	111	124	040				
	027735	111	116	103				
	027740	117	122	122				
	027743	105	103	124				
	027746	000						
115	027747	104	115	122	EMT8:	.ASCIZ	/DMR MODE INPUT COMMAND ERROR/	
	027752	040	115	117				
	027755	104	105	040				
	027760	111	116	120				
	027763	125	124	040				
	027766	103	117	115				
	027771	115	101	116				
	027774	104	040	105				
	027777	122	122	117				

	030002	122	000	
116				
117	030004	045	101	111 FMT11: .ASCIZ /%AIMAGE OF SP 13 = %D3%N/
	030007	115	101	107
	030012	105	040	:17
	030015	106	040	123
	030020	120	040	061
	030023	063	040	075
	030026	040	045	104
	030031	063	045	116
	030034	000		
118	030035	045	101	122 FMT12: .ASCIZ /%AREP-SEL TIME VALUE = %D3%N/
	030040	105	120	055
	030043	123	105	114
	030046	040	124	111
	030051	115	105	040
	030054	126	101	114
	030057	125	105	040
	030062	075	040	045
	030065	104	063	045
	030070	116	000	
119	030072	045	101	116 FMT13: .ASCIZ /%ANAK RCVD THRESHOLD = %D3%N/
	030075	101	113	040
	030100	122	103	126
	030103	104	040	124
	030106	110	122	105
	030111	123	110	117
	030114	114	104	040
	030117	075	040	045
	030122	104	063	045
	030125	116	000	
120	030127	045	101	116 FMT14: .ASCIZ /%ANAK SENT THRESHOLD = %D3%N/
	030132	101	113	040
	030135	123	105	116
	030140	124	040	124
	030143	110	122	105
	030146	123	110	117
	030151	114	104	040
	030154	075	040	045
	030157	104	063	045
	030162	116	000	
121	030164	045	101	122 FMT15: .ASCIZ /%AREP SENT THRESHOLD = %D3%N/
	030167	105	120	040
	030172	123	105	116
	030175	124	040	124
	030200	110	122	105
	030203	123	110	117
	030206	114	104	040
	030211	075	040	045
	030214	104	063	045
	030217	116	000	
122	030221	045	101	116 FMT16: .ASCIZ /%ANO BUFFER THRESHOLD = %D3%N/
	030224	117	040	102
	030227	125	106	106
	030232	105	122	040
	030235	124	110	122
	030240	105	123	110

030243	117	114	104
030246	040	075	040
030251	045	104	063
030254	045	116	000

123
124
125

.EVEN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32

```

.SBTTL          TEST 6 - CONTROL IN COMMAND

:*****
:          TEST 6 - DMR-11
:* CONTROL IN COMMAND TEST -
:* SUBTEST 1 - CONTROL IN, FULL DUPLEX, DDCMP MODE. ENSURE THAT
:*              THE HALF-DUPLEX BIT IS CLEAR IN THE MODEM STATUS WORD,
:*              ALSO ENSURE THAT DDCMP MODE BIT IS SET IN SCRATCH PAD 7.
:* SUBTEST 2 - CONTROL IN, HALF DUPLEX. ENSURE THAT THE HALF DUPLEX
:*              BIT IS SET.
:* SUBTEST 3 - CONTROL IN, MAINTENANCE MODE. ENSURE THAT MAINT. MODE
:*              BIT IS SET IN SCRATCH PAD 7.
:* SUBTEST 4 - CONTROL IN USING SELECTED LOOPBACK. ISSUE A CONTROL IN
:*              USING THE USER SELECTED LOOPBACK. IF THE LOOPBACK IS
:*              NOT CORRECT, DMR RUN MODE ACKNOWLEDGE WILL NOT BE
:*              RECEIVED.
:*****

BGNTST

BGNSUB

T6::
T6.1:
TRAP C$BSUB

CLEAR          ;MACRO FOR MASTER CLEAR
               ;**** MACRO EXPANSION ****
               ;ISSUE A DMR MASTER CLEAR
               ;****          ****

JSR PC, $MSCLR

ESCAPE TST     ;IF ERROR, BR TO TEST END.
TRAP          C$ESCAPE
               .WORD L10042-.

BASEIN        ;MACRO FOR BASE IN COMMAND
               ;**** MACRO EXPANSION ****
               ;CALL BASE IN ROUTINE WITH DEFAULTS
               ;SET LINE UNIT LOOP
               ;BASE TABLE ADDRESS
               ;DMR-11 MODE
               ;****          ****

JSR PC, $BASEI
               .WORD LPLU
               .WORD BASE
               .WORD DMR

ESCAPE TST     ;IF ERROR, BR TO TEST END.
TRAP          C$ESCAPE
               .WORD L10042-.

CNTRIN        ;MACRO FOR CONTROL IN (FULL DUPLEX)
               ;**** MACRO EXPANSION ****
               ;CALL CONTROL IN ROUTINE WITH DEFAULT
               ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
               ;****          ****

JSR PC, $CNTIN
               .WORD 0

ESCAPE TST     ;IF ERROR, BR TO TEST END.
TRAP          C$ESCAPE
               .WORD L10042-.

BIS #RQI!RMODEM,@SELO ;SET RQI AND READ MODEM COMMAND
WAIT RDI       ;WAIT FOR RDI TO BE SET
               ;**** MACRO EXPANSION ****
               ;CALL WAIT ROUTINE
               ;FLAG THAT WE'RE WAITING FOR RDI

JSR PC, $WAIT
               .WORD 0
    
```

```

030260
030260
030260
030260 104402
030262
030262 004737 011070
030266
030266 104410
030270 000404
030272
030272 004737 011266
030276 004000
030300 002634
030302 000522
030304
030304 104410
030306 000366
030310
030310 004737 011522
030314 000000
030316
030316 104410
030320 000354
030322 052777 000057 151702
030330
030330 004737 010272
030334 000000
    
```

```

33 030336 032777 000020 151672      BIT      #BIT4,@SEL4      :*****
34 030344 001404                      BEQ      10$            :IS THE HDX BIT SET IN MODEM STATUS REG?
35 030346 104455                      ERRDF   21,EMT9        :OK - IF BIT CLEAR
                                :ERROR HDX BIT SET
                                TRAP      C$ERDF
                                .WORD    21
030350 000025                      :
030352 030676                      :
030354 000000                      :
36 030356 104455                      :
37 030356 004737 010706      10$:    WAIT      RQI          :CLEAR RQI AND WAIT FOR RDI TO CLEAR.
                                :***** MACRO EXPANSION *****
                                JSR      PC, $CLRQI      :CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
                                :*****
                                SHUTDN
                                :HALT DMR
                                :***** MACRO EXPANSION *****
                                JSR      PC, $HALT      :DMR HALT ROUTINE.
                                :*****
                                ESCAPE  TST          :IF ERROR, EXIT.
                                :*****
                                TRAP      C$ESCAPE
                                .WORD    L10042-
40 030372 132737 000020 002726      BITB    #BIT4,BASE+ISP7 :IS THE DDCMP RUN BIT SET IN IMAGE OF SP 7.
41 030400 001004                      BNE     20$
42 030402 104455                      ERRDF   21,EMT10      :ERROR DDCMP RUN BIT NOT SET
                                TRAP      C$ERDF
                                .WORD    21
                                .WORD    EMT10
                                .WORD    0
43 030412 104403                      :
44 030412 004737 011266      20$:    ENDSUB
                                :
                                L10043:
45 030414 104402                      BGNSUB
                                :
                                T6.2:
46 030414 004737 011266      BASEIN  LPLU,BASE,RES!DMR ;BASE IN WITH RESUME.
47 030416 004737 011266      JSR     PC, $BASEI      :***** MACRO EXPANSION *****
                                .WORD    LPLU          :CALL BASE IN ROUTINE
                                .WORD    BASE          :MAINTENANCE MODE BITS TO SET IN BSEL1
                                .WORD    RES!DMR       :BASE TABLE ADDRESS
                                .WORD    RES!DMR       :MODE
                                :*****
                                :*****
48 030430 004737 011522      CNTRIN  HDX            :CONTROL IN COMMAND WITH HDX.
49 030430 002000                      JSR     PC, $CNTIN      :***** MACRO EXPANSION *****
                                .WORD    HDX          :CALL CONTROL IN ROUTINE
                                :SEL6 - (DUPLEX, MODE)
                                :*****
                                :*****
50 030436 104410                      ESCAPE  TST            :IF ERROR, BR TO TEST END.
51 030436 000234                      :
52 030442 052777 000057 151562      BIS     #RQI!RMODEM,@SELO ;SET RQI AND READ MODEM COMMAND
53 030450 004737 010272      WAIT    RDI            :WAIT FOR RDI TO BE SET
                                :***** MACRO EXPANSION *****
                                JSR     PC, $WAIT      :CALL WAIT ROUTINE
                                .WORD    0            :FLAG THAT WE'RE WAITING FOR RDI
                                TRAP      C$ESCAPE
                                .WORD    L10042-
    
```



```

54 030456 032777 000020 151552      BIT      #BIT4,@SEL4      ;*****
55 030464 001004                      BNE      10$            ;IS THE HDX BIT SET IN MODEM STATUS REG2
56 030466                      ERRDF   21,EMT11      ;OK - IF BIT SET
                                ;*****
                                ;ERROR HDX BIT CLEAR.
                                TRAP   C$ERDF
                                .WORD 21
                                .WORD EMT11
                                .WORD 0
57 030476                      10$:
58 030476                      SHUTDN
                                ;HALT THE DMR.
                                ;***** MACRO EXPANSION *****
                                ;DMR HALT ROUTINE.
                                ;*****
                                TRAP   C$ERDF
                                .WORD 21
                                .WORD EMT11
                                .WORD 0
                                030476 004737 012564      JSR      PC, $HALT
59
60 030502                      ENDSUB
                                L10044:
60 030502 104403                      TRAP   C$ESUB
61
62 030504                      BGNSUB
                                T6.3:
63 030506                      CLEAR
                                ;MACRO FOR MASTER CLEAR
                                ;***** MACRO EXPANSION *****
                                ;ISSUE A DMR MASTER CLEAR
                                ;*****
                                TRAP   C$BSUB
                                .WORD 0
                                030506 004737 011070      JSR      PC, $MSCLR
64
65 030512                      ESCAPE  TST
                                ;IF ERROR, BR TO TEST END.
                                TRAP   C$ESCAPE
                                .WORD L10042-
66 030516                      BASEIN
                                ;MACRO FOR BASE IN COMMAND
                                ;***** MACRO EXPANSION *****
                                ;CALL BASE IN ROUTINE WITH DEFAULTS
                                ;SET LINE UNIT LOOP
                                ;BASE TABLE ADDRESS
                                ;DMR-11 MODE
                                ;*****
                                TRAP   C$ESCAPE
                                .WORD L10042-
                                030516 004737 011266      JSR      PC, $BASEI
                                .WORD LPLU
                                .WORD BASE
                                .WORD DMR
67
68 030530                      ESCAPE  TST
                                ;IF ERROR, BR TO TEST END.
                                TRAP   C$ESCAPE
                                .WORD L10042-
69 030534                      CNTRIN  MAINT
                                ;MACRO FOR CONTROL IN (MAINT. MODE)
                                ;***** MACRO EXPANSION *****
                                ;CALL CONTROL IN ROUTINE
                                ;SEL6 - (DUPLEX, MODE)
                                ;*****
                                TRAP   C$ESCAPE
                                .WORD L10042-
                                030534 004737 011522      JSR      PC, $CNTIN
                                .WORD MAINT
70
71 030542                      ESCAPE  TST
                                ;IF ERROR, BR TO TEST END.
                                TRAP   C$ESCAPE
                                .WORD L10042-
72 030546                      SHUTDN
                                ;HALT
                                ;***** MACRO EXPANSION *****
                                ;DMR HALT ROUTINE.
                                ;*****
                                TRAP   C$ESCAPE
                                .WORD L10042-
                                030546 004737 012564      JSR      PC, $HALT
73 030552                      ESCAPE  TST
                                ;IF ERROR, BR TO TEST END.
                                TRAP   C$ESCAPE
                                .WORD L10042-
74 030556 132737 000002 002726      BITB    #BIT1,BASE+ISP7 ;IS THE MAINTENANCE BIT SET IN IMAGE OF SP 7.

```

```

75 030564 001004          BNE      10$
76 030566          ERRDF   21,EMT12      ;ERROR - MAINT. BIT NOT SET.
    030566 104455          .WORD   C$ERDF
    030570 000025          .WORD   21
    030572 031010          .WORD   EMT12
    030574 000000          .WORD   0
77 030576          10$:
78 030576          ENDSUB
    030576 104403          L10045:
    030576          TRAP    C$ESUB
79
80 030600          BGNSUB
    030600          T6.4:
    030600 104402          TRAP    C$BSUB
81
82 030602          CLEAR
    030602 004737 011070   JSR     PC, $MSCLR      ;MACRO FOR MASTER CLEAR
    ;***** MACRO EXPANSION *****
    ;ISSUE A DMR MASTER CLEAR
    ;*****
83
84 030606          ESCAPE  TST
    030606 104410          .WORD   C$ESCAPE
    030610 000064          .WORD   L10042-.
85 030612 005737 002254   TST     DMTURN
86 030616 001004          BNE     1$
87 030620 052737 004000 030642  BIS     #LPLU,100$
88 030626 000403          BR      2$
89 030630          1$:
90 030630 042737 004000 030642  BIC     #LPLU,100$
91 030636          2$:
92 030636          CALL    $BASEI
93 030642 000000          .WORD   0
94 030644 002634          .WORD   BASE
95 030646 000522          .WORD   DMR
96 030650          ESCAPE  TST
    030650 104410          .WORD   C$ESCAPE
    030652 000022          .WORD   L10042-.
97 030654          CALL    $LOOP
98
99
100 030660          ESCAPE  TST
    030660 104410          .WORD   C$ESCAPE
    030662 000012          .WORD   L10042-.
101 030664          CNTRIN
    030664 004737 011522   JSR     PC, $CNTIN
    030670 000000          .WORD   0
    ;MACRO FOR CONTROL IN (FULL DUPLEX)
    ;***** MACRO EXPANSION *****
    ;CALL CONTROL IN ROUTINE WITH DEFAULT
    ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
    ;*****
102 030672          ENDSUB
    030672 104403          L10046:
    030672          TRAP    C$ESUB
103
104
105
106 030674          ENDTST
    030674 104401          L10042:
    030674          TRAP    C$ETST
    
```

```

107
108 030676      110      104      130  EMT9:  .ASCIZ  /HDX BIT SET WHEN IN FDX/
      030701      040      102      111
      030704      124      040      123
      030707      105      124      040
      030712      127      110      105
      030715      116      040      111
      030720      116      040      106
      030723      104      130      000
109 030726      104      104      103  EMT10: .ASCIZ  /DDCMP RUN BIT NOT SET/
      030731      115      120      040
      030734      122      125      116
      030737      040      102      111
      030742      124      040      116
      030745      117      124      040
      030750      123      105      124
      030753      000
110 030754      110      104      130  EMT11: .ASCIZ  /HDX BIT NOT SET WHEN IN HDX/
      030757      040      102      111
      030762      124      040      116
      030765      117      124      040
      030770      123      105      124
      030773      040      127      110
      030776      105      116      040
      031001      111      116      040
      031004      110      104      130
      031007      000
111 031010      115      101      111  EMT12: .ASCIZ  /MAINT. MODE BIT NOT SET/
      031013      116      124      056
      031016      040      115      117
      031021      104      105      040
      031024      102      111      124
      031027      040      116      117
      031032      124      040      123
      031035      105      124      000
112
113
114
115

```

.EVEN

```

1          .SBTTL          TEST 7 - MODEM WRITE COMMAND
2
3          *****
4          TEST 7 - DMR-11
5          * MODEM WRITE COMMAND
6          * SUBTEST 1 - WRITE DATA PATTERNS INTO THE MODEM WRITE REGISTER.
7          * ENSURE THAT ON THE NEXT MODEM READ THAT THE
8          * MICROCODE RETURNS THE PATTERN WRITTEN INTO BSEL6.
9          * SUBTEST 2 - ATTEMPT TO WRITE BOTH THE HALF-DUPLEX BIT AND THE
10         * RTS HOLD BIT. THE MICROCODE SHOULD NOT ALLOW THIS
11         * TO HAPPEN. WHEN READING THE MODEM STATUS, ONLY
12         * THE HALF-DUPLEX SHOULD BE SET.
13         *
14         *****
15 031040  BGNTST
16         T7::
17 031040  BGNSUB
18 031040  104402          CLEAR          ;MACRO FOR MASTER CLEAR
19         ;***** MACRO EXPANSION *****
20 031040  004737 011070  JSR          PC, $MSCLR  ;ISSUE A DMR MASTER CLEAR
21         ;*****
22         ESCAPE TST          ;IF ERROR, BR TO TEST END.
23 031040  104410          TRAP          C$SUB
24 031050  000232          .WORD          L10047-.
25 031052  004737 011266  BASE IN
26         ;BASE IN COMMAND.
27 031056  004000          JSR          PC, $BASEI  ;***** MACRO EXPANSION *****
28         ;CALL BASE IN ROUTINE WITH DEFAULTS
29 031060  002634          .WORD          LPLU      ;SET LINE UNIT LOOP
30 031062  000522          .WORD          BASE      ;BASE TABLE ADDRESS
31         ;DMR-11 MODE
32         ;*****
33         ESCAPE TST          ;IF ERROR, BR TO TEST END.
34 031064  104410          TRAP          C$ESCAPE
35 031066  000214          .WORD          L10047-.
36 031070  012701 000005  MOV          #5,R1          ;COUNTER
37 031074  012702 031304  MOV          #MODEM,R2     ;PATTERN TO WRITE INTO MODEM
38 10$:
39 031100  012237 031114  MOV          (2)+,15$      ;WRITE PATTERN
40 031104  004737 012070  JSR          PC,$DMRIN     ;ISSUE DMR MODE COMMAND
41 031110  000005          .WORD          WMODEM    ;WRITE MODEM COMMAND
42 031112  000377          .WORD          377      ;CLEAR ALL BITS IN BSEL6
43 031114  000000          .WORD          0        ;SET THE BITS IN BSEL6 (FROM PATTERN)
44 031116  104410          ESCAPE TST          ;IF ERROR, BR TO TEST END.
45 031120  000162          TRAP          C$ESCAPE
46         .WORD          L10047-.
47 031122  052777 000057 151102  BIS          #RQI!RMODEM,@SELO ;SET RQI AND READ MODEM COMMAND
48 031130  WAIT          RDI          ;WAIT FOR RDI TO BE SET.
49         ;***** MACRO EXPANSION *****
50 031130  004737 010272  JSR          PC,$WAIT      ;CALL WAIT ROUTINE
51 031134  000000          .WORD          0        ;FLAG THAT WE'RE WAITING FOR RDI
    
```

TEST 7 - MODEM WRITE COMMAND

```

37 031136          ESCAPE TST          :*****
      031136 104410          :IF ERROR, EXIT TEST.          *****
      031140 000142          TRAP      C$ESCAPE
38 031142          .WORD      L10047-.
39 031142 127737 151072 031114 20$:  -
40 031150 001406          CMPB      @BSEL6,15$  :DID THE MICROCODE COPY THE BITS?
41 031152 013703 031114          BEQ      25$      :IF YES CONTINUE
42 031156          MOV      15$,R3      :SAVE THE PATTERN FOR THE ERROR MESSAGE.
      031156 104455          ERRDF     22,EMT13,ERRT2 :WRITE MODEM ERROR
      031160 000026          TRAP      C$ERDF
      031162 031350          .WORD      22
      031164 031316          .WORD      EMT13
43 031165          .WORD      ERRT2
44 031166          25$:
      031166 004737 010706          WAIT     RQI      :CLEAR RQI AND WAIT FOR RDI TO CLEAR.
      JSR      PC, $CLRQI          :***** MACRO EXPANSION *****
      JSR      PC, $CLRQI          :CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
45 031172          ESCAPE TST          :*****
      031172 104410          :IF ERROR, EXIT TEST.          *****
      031174 000106          TRAP      C$ESCAPE
46 031176 005301          .WORD      L10047-.
47 031200 001337          DEC      R1
48 031202          BNE      10$      :DECREMENT COUNTER
49          30$:
50 031202          ENDSUB
      031202 104403          L10050:
51          TRAP      C$ESUB
52 031204          BGNSUB
      031204 104402          T7.2:
53          TRAP      C$BSUB
54 031206          DMRIN     WMODEM,377,21 :ATTEMPT TO WRITE MODEM HDX AND RTS.
      031206 004737 012070          JSR      PC, $DMRIN :***** MACRO EXPANSION *****
      031212 000005          .WORD      WMODEM :CALL DMR MODE INPUT ROUTINE
      031214 000377          .WORD      377  :INPUT COMMAND
      031216 000021          .WORD      21  :SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
      :SEL6 VALUE (OR BITS TO SET IN BSEL6)
      :*****
55          *****
56 031220          ESCAPE TST          :IF ERROR, BR TO END.
      031220 104410          TRAP      C$ESCAPE
      031222 000060          .WORD      L10047-.
57 031224 052777 000057 151000          BIS      #RQI!RMODEM,@SELO :SET RQI AND READ MODEM COMMAND.
58 031232          WAIT     RDI      :WAIT FOR RDI TO BE SET
      031232 004737 010272          JSR      PC, $WAIT :***** MACRO EXPANSION *****
      031236 000000          .WORD      0 :CALL WAIT ROUTINE
      :FLAG THAT WE'RE WAITING FOR RDI
      :*****
59 031240          ESCAPE TST          :IF ERROR, EXIT TEST.
      031240 104410          TRAP      C$ESCAPE
      031242 000040          .WORD      L10047-.
60
61 031244 122777 000020 150766          CMPB      #20,@BSEL6 :IS ONLY HDX SET?
62 031252 001406          BEQ      10$      :IF YES - OK
63 031254 012703 000021          MOV      #21,R3      :SAVE THE PATTERN FOR THE ERROR MESSAGE.

```

```

64 031260          ERRDF  22,EMT13,ERRT2
    031260 104455
    031262 000026
    031264 031350
    031266 031316
65 031270          10$:
66 031270          WAIT   RQI           ;CLEAR RQI AND WAIT FOR RDI TO CLEAR.
    031270 004737 010706          JSR   PC, $CLRQI       ;**** MACRO EXPANSION ****
    ;*****                               ;CLEAR RQI AND WAIT FOR IT TO BE CLEARED.
    ;*****                               ;*****
67 031274          SHUTDN
    031274 004737 012564          JSR   PC, $HALT       ;**** MACRO EXPANSION ****
    ;*****                               ;DMR HALT ROUTINE.
    ;*****                               ;*****
68
69 031300          ENDSUB
    031300
    031300 104403
70
71 031302          ENDTST
    031302
    031302 104401
72
73 031304 000000 000376 000001 MODEM: .WORD 0,376,1,252,357 ;PATTERN TO WRITE INTO MODEM
    031312 000252 000357
74
75 031316          BGNMSG  ERRT2
    031316
76 031316          PRINTB #FMT19,R3,<B,@BSEL6>
    031316 005046
    031320 157716 150714
    031324 010346
    031326 012746 031374
    031332 012746 000003
    031336 010600
    031340 104414
    031342 062706 000010
77 031346          ENDMSG
    031346
    031346 104423
78
79
80 031350          EMT13: .ASCIZ /WRITE MODEM ERROR /
    031353 127 122 111
    031356 124 105 040
    031361 115 117 104
    031364 105 115 040
    031367 105 122 122
    031372 117 122 040
    031372 000
81
82
83 031374          .EVEN
    031377 045 101 127 FMT19: .ASCIZ /%AWROTE IN BSEL6: %03%A MODEM FORMAT IN BSEL6: %03%N/
    031402 122 117 124
    031405 105 040 111
    031410 116 040 102
    031413 123 105 114
    031413 066 072 040
    L10051: TRAP C$ESUB
    L10047: TRAP C$ETST
    ERRT2::
    CLR -(SP)
    BISB @BSEL6,(SP)
    MOV R3,-(SP)
    MOV #FMT19,-(SP)
    MOV #3,-(SP)
    MOV SP,R0
    TRAP C$PNTB
    ADD #10,SP
    L10052: TRAP C$MSG
    
```

031416	045	117	063
031421	045	101	040
031424	040	115	117
031427	104	105	115
031432	040	106	117
031435	122	115	101
031440	124	040	111
031443	116	040	102
031446	123	105	114
031451	066	072	040
031454	045	117	063
031457	045	116	000

84
85
86
87

.EVEN

TEST 8 - NO BUFFER ERROR

```

1      .SBTTL          TEST 8 - NO BUFFER ERROR
2
3      ;*****
4      ;*          TEST 8 - DMR-11
5      ;* SUBTEST 1 - TRANSMIT A BUFFER THREE TIMES WIHOUT ASSIGNING A
6      ;*          RECEIVE BUFFER. BY ASSIGNING A NO BUFFER THRESHOLD
7      ;*          OF THREE, ENSURE THAT A NO BUFFER ERROR IS RECEIVED
8      ;*          AFTER THE THIRD THRANSMISSION.
9      ;* SUBTEST 2 - TRANSMIT A BUFFER WITHOUT A RECEIVE BUFFER.
10     ;*          ASSIGN THE NAKS THRESHOLD OF 3 AND A NO BUFFER
11     ;*          THRESHOLD OF 7. CHECK THAT THE NAKS ERROR COUNT IS
12     ;*          THREE AFTER SHUTDOWN.
13     ;*****
14     BGNTST
15     BGNSUB          T8::
16     CLEAR          ;MACRO FOR MASTER CLEAR
17     JSR PC, $MSCLR ;**** MACRO EXPANSION ****
18     ESCAPE TST     ;ISSUE A DMR MASTER CLEAR
19     BASEIN         ;****
20     JSR PC, $BASEI ;MACRO FOR BASE IN COMMAND
21     .WORD LPLU     ;**** MACRO EXPANSION ****
22     .WORD BASE     ;CALL BASE IN ROUTINE WITH DEFAULTS
23     .WORD DMR      ;SET LINE UNIT LOOP
24     ;DMR-11 MODE  ;BASE TABLE ADDRESS
25     ;*****      ;DMR-11 MODE
26     ESCAPE TST     ;*****
27     CNTRIN MAINT   ;IF ERROR, BR TO TEST END.
28     JSR PC, $CNTIN ;MACRO FOR CONTROL IN (FULL DUPLEX AND MAINT)
29     .WORD MAINT    ;**** MACRO EXPANSION ****
30     ;SEL6 - (DUPLEX, MODE)
31     DMRIN THRESH,177777,1777 ;CALL CONTROL IN ROUTINE
32     JSR PC, $DMRIN ;SEL6 - (DUPLEX, MODE)
33     .WORD THRESH  ;****
34     .WORD 177777 ;NO BUFFER = 3
35     ;**** MACRO EXPANSION ****
36     ;CALL DMR MODE INPUT ROUTINE
37     ;INPUT COMMAND
38     ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)

```


TEST 8 - NO BUFFER ERROR

```

031534 001777 .WORD 1777 ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
;****
32
33 031536 ESCAPE TST ;IF ERROR, BR TO TEST END.
031536 104410
031540 000350 TRAP C$ESCAPE
34 031542 012700 000003 MOV #3,R0 ;SET UP A COUNTER .WORD L10053-.
35 031546 1$: BACCIT ;BA/CC IN COMMAND FOR TRANSMIT
36 031546 ;**** MACRO EXPANSION ****
031546 004737 012300 JSR PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
031552 000040 .WORD RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
031554 002516 .WORD TBUF ;TRANSMIT BUFFER ADDRESS
031556 000044 .WORD TCOUNT ;TRANSMIT CHARACTER COUNT
;****

37
38 031560 WAIT RDO ;WAIT FOR RDO TO BE SET
;**** MACRO EXPANSION ****
031560 004737 010272 JSR PC, $WAIT ;CALL WAIT ROUTINE
031564 000001 .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
;****

39 031566 ESCAPE TST ;IF RDO NOT SET, BR TO TEST END.
031566 104410 TRAP C$ESCAPE
031570 000320 .WORD L10053-.
40 031572 005300 DEC R0 ;DEC COUNTER
41 031574 001404 BEQ 10$ ;TRANSMIT FOR 3 TIMES.
42 031576 042777 000207 150430 BIC #RDO!CMD,@SEL2 ;CLEAR BACC OUT TRANSMIT.
43 031604 000760 BR 1$ ;TRANSMIT AGAIN
44 031606 10$:
45 031606 032777 000001 150420 BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
46 031614 001005 BNE 20$ ;IF YES, PROCEED.
47 031616 ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT NOT RECEIVED.
031616 104455 TRAP C$ERDF
031620 000010 .WORD 8
031622 020130 .WORD EMG8
031624 015126 .WORD ERRG2
48 031626 000410 BR 30$ ;EXIT
49 031630 20$:
50 031630 032777 000004 150402 BIT #NOBFR,@SEL6 ;IS THE NO BUFFER FLAG SET?
51 031636 001004 BNE 30$ ;IF YES - OK, PROCEED.
52 031640 ERRDF 9,EMG9,ERRG2 ;WE'RE NOT GETTING EXPECTED RESULT
031640 104455 TRAP C$ERDF
031642 000011 .WORD 9
031644 020174 .WORD EMG9
031646 015126 .WORD ERRG2
53
54 031650 ;(EITHER CONTROL OUT OR NOBUF/NAKS)
55 031650 042777 000207 150356 30$: BIC #RDO!CMD,@SEL2 ;CLEAR CONTROL OUT
56 031656 WAIT RDO ;EXPECT ANOTHER BACC OUT.
031656 004737 010272 JSR PC, $WAIT ;**** MACRO EXPANSION ****
031662 000001 .WORD 1 ;CALL WAIT ROUTINE
;FLAG THAT WE'RE WAITING FOR RDO
;****

57 031664 ESCAPE TST ;IF ERROR, BR TO END.
031664 104410 TRAP C$ESCAPE
031666 000222 .WORD L10053-.
58 031670 042777 000207 150336 BIC #RDO!CMD,@SEL2 ;CLEAR BACC OUT.

```

```

59 031676          SHUTDN          ;HALT DMR
      031676 004737 012564        JSR   PC, $HALT          ;**** MACRO EXPANSION ****
                                          ;DMR HALT ROUTINE.
                                          ;****          ****
60 031702          50$:
61 031702          ENDSUB
      031702          L10054:
      031702 104403          TRAP   C$ESUB
62
63 031704          BGNSUB
      031704          T8.2:
      031704 104402          TRAP   C$BSUB
64 031706          CLEAR          ;MACRO FOR MASTER CLEAR
      031706 004737 011070        JSR   PC, $MSCLR          ;**** MACRO EXPANSION ****
                                          ;ISSUE A DMR MASTER CLEAR
                                          ;****          ****
65
66 031712          ESCAPE TST          ;IF ERROR, BR TO TEST END.
      031712 104410          TRAP   C$ESCAPE
      031714 000174          .WORD  L10053-.
67 031716          BASEIN          ;MACRO FOR BASE IN COMMAND
      031716 004737 011266        JSR   PC, $BASEI          ;**** MACRO EXPANSION ****
      031722 004000          .WORD  LPLU          ;CALL BASE IN ROUTINE WITH DEFAULTS
      031724 002634          .WORD  BASE          ;SET LINE UNIT LOOP
      031726 000522          .WORD  DMR          ;BASE TABLE ADDRESS
                                          ;DMR-11 MODE
                                          ;****          ****
68
69 031730          ESCAPE TST          ;IF ERROR, BR TO TEST END.
      031730 104410          TRAP   C$ESCAPE
      031732 000156          .WORD  L10053-.
70 031734          CNTRIN          ;MACRO FOR CONTROL IN (FULL DUPLEX)
      031734 004737 011522        JSR   PC, $CNTIN          ;**** MACRO EXPANSION ****
      031740 000000          .WORD  0          ;CALL CONTROL IN ROUTINE WITH DEFAULT
                                          ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
                                          ;****          ****
71
72 031742          ESCAPE TST          ;IF ERROR, BR TO TEST END.
      031742 104410          TRAP   C$ESCAPE
      031744 000144          .WORD  L10053-.
73
74
75
76
77
78
79 031746          DMRIN THRESH,1403,3777 ;SET THRESHOLDS:
      031746 004737 012070        JSR   PC, $DMRIN          ;NAKS RCVD = 3
      031752 000013          .WORD  THRESH          ;NAKS SENT = 3
      031754 001403          .WORD  1403          ;REP SENT = 377
      031756 003777          .WORD  3777          ;NO BUFFER = 7
                                          ;**** MACRO EXPANSION ****
                                          ;CALL DMR MODE INPUT ROUTINE
                                          ;INPUT COMMAND
                                          ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
                                          ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
                                          ;****          ****
80
81 031760          ESCAPE TST          ;IF ERROR, BR TO TEST END.
      031760 104410          TRAP   C$ESCAPE
      031762 000126          .WORD  L10053-.
    
```

TEST 8 - NO BUFFER ERROR

```

82 031764          BACCIT          ;BA/CC IN COMMAND FOR TRANSMIT
      031764 004737 012300          ;**** MACRO EXPANSION ****
      031770 000040          JSR    PC, $BACC          ;CALL BA/CC IN ROUTINE WITH DEFAULTS
      031772 002516          .WORD  RQI!BACC          ;BA/CC IN TRANSMIT COMMAND
      031774 000044          .WORD  TBUF          ;TRANSMIT BUFFER ADDRESS
      .WORD  TCOUNT          ;TRANSMIT CHARACTER COUNT
      .WORD  TCOUNT          ;****          ****

83 031776          10$:
84 031776          WAIT    RDO          ;WAIT FOR RDO TO BE SET
      031776 004737 010272          ;**** MACRO EXPANSION ****
      032002 000001          JSR    PC, $WAIT          ;CALL WAIT ROUTINE
      .WORD  1          ;FLAG THAT WE'RE WAITING FOR RDO
      .WORD  1          ;****          ****

85 032004          ESCAPE  TST          ;IF RDO NOT SET, BR TO TEST END.
      032004 104410          .WORD  1          TRAP    C$ESCAPE
      032006 000102          .WORD  1          .WORD  L10053-

86 032010          BIT     #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
87 032016          BNE    20$          ;IF YES, PROCEED.
88 032020          ERRDF  8,EMG8,ERRG2 ;EXPECTED CONTROL OUT NOT RECEIVED.
      032020 104455          .WORD  8          TRAP    C$ERDF
      032022 000010          .WORD  8          .WORD  8
      032024 020130          .WORD  8          .WORD  EMG8
      032026 015126          .WORD  8          .WORD  ERRG2

89 032030          BR     30$          ;EXIT
90 032032          20$:
91 032032 032777 000004 150200 BIT  #NOBFR,@SEL6 ;IS THE NO BUFFER FLAG SET?
92 032040 001004          BNE    30$          ;IF YES - OK, PROCEED.
93 032042          ERRDF  9,EMG9,ERRG2 ;WE'RE NOT GETTING EXPECTED RESULT
      032042 104455          .WORD  9          TRAP    C$ERDF
      032044 000011          .WORD  9          .WORD  9
      032046 020174          .WORD  9          .WORD  EMG9
      032050 015126          .WORD  9          .WORD  ERRG2

94          ;(EITHER CONTROL OUT OR NOBUF/NAKS)
95 032052          30$:
96 032052          SHUTDN
      032052 004737 012564          ;**** MACRO EXPANSION ****
      JSR    PC, $HALT          ;DMR HALT ROUTINE.
      .WORD  1          ;****          ****

97 032056 123727 002637 000003 CMPB BASE+3,#3 ;NAKS REC. - NO BUFFER = 3?
98 032064 001004          BNE    35$          ;IF NOT ERROR
99 032066 123727 002642 000003 CMPB BASE+6,#3 ;NAKS SENT - NO BUFFER = 3?
100 032074 001404          BEQ    40$          ;IF OK - SKIP.
101 032076          35$:
102 032076          ERRDF  23,EMT20,ERRT4
      032076 104455          .WORD  23          TRAP    C$ERDF
      032100 000027          .WORD  23          .WORD  23
      032102 032150          .WORD  23          .WORD  EMT20
      032104 032112          .WORD  23          .WORD  ERRT4

103          40$:
104 032106          ENDSUB
105 032106          L10055:
      032106 104403          TRAP    C$ESUB
106 032110          ENDTST          L10053:
      032110          TRAP    C$ETST
107 032110 104401          TRAP    C$ETST

```

```
108
109 032112          BGNMSG  ERR4
      032112
110 032112          PRINTB  #FMG7,<B,BASE+3>,<B,BASE+6>
      032112 005046
      032114 153716 002642
      032120 005046
      032122 153716 002637
      032126 012746 016615
      032132 012746 000003
      032136 010600
      032140 104414
      032142 062706 000010
111 032146          ENDMSG
      032146
      032146 104423
112
113 032150          116      101      113  EMT20: .ASCIZ  /NAKS ERROR/
      032153          123      040      105
      032156          122      122      117
      032161          122      000
114
115          .EVEN
```

```
ERR4::
      CLR      -(SP)
      BISB    BASE+6,(SP)
      CLR      -(SP)
      BISB    BASE+3,(SP)
      MOV     #FMG7,-(SP)
      MOV     #3,-(SP)
      MOV     SP,R0
      TRAP    C$PNTB
      ADD     #10,SP

L10056:
      TRAP    C$MSG
```

```

1      .SBTTL          TEST 9 - NON-EXISTENT MEMORY ERROR
2
3      :*****
4      :*              TEST 9 - DMR-11
5      :* NON-EXISTENT MEMORY (NXM) ERROR CHECK
6      :* PERFORM DMR COMMANDS USING NXM ADDRESSES; VERIFY THAT NXM ERROR IS
7      :* REPORTED IN EACH OF THE FOLLOWING SUBTESTS:
8      :* SUBTEST 1 - BASE IN RESUME COMMAND - BASE TABLE ADDRESS IS NXM
9      :* SUBTEST 2 - BA/CC IN RECEIVE COMMAND - BA/CC IN ADDRESS IS NXM
10     :* SUBTEST 3 - BA/CC IN TRANSMIT COMMAND - BA/CC IN ADDRESS IS NXM
11     :*
12     :*****
13     BGNTST
14     BGNSUB          T9::
15     032164          T9.1:      TRAP      C$BSUB
16     032164          CLEAR      ;MASTER CLEAR MACRO
17     032164          JSR        PC, $MSCLR ;**** MACRO EXPANSION ****
18     032164          ;ISSUE A DMR MASTER CLEAR
19     032164          ;****
20     032164          ESCAPE    TST      ;IF ERROR, BR TO TEST END
21     032164          ;TRAP      C$ESCAPE
22     032164          ;.WORD    L10057-.
23     032164          BASEIN     ;BASE IN COMMAND - DMR MODE
24     032164          JSR        PC, $BASEI ;**** MACRO EXPANSION ****
25     032164          ;CALL BASE IN ROUTINE WITH DEFAULTS
26     032164          ;.WORD    LPLJ    ;SET LINE UNIT LOOP
27     032164          ;.WORD    BASE    ;BASE TABLE ADDRESS
28     032164          ;.WORD    DMR     ;DMR-11 MODE
29     032164          ;****
30     032164          ESCAPE    TST      ;IF ERROR, BR TO TEST END
31     032164          ;TRAP      C$ESCAPE
32     032164          ;.WORD    L10057-.
33     032164          SHUTDN     ;HALT
34     032164          JSR        PC, $HALT ;**** MACRO EXPANSION ****
35     032164          ;DMR HALT ROUTINE.
36     032164          ;****
37     032164          ESCAPE    TST      ;IF ERROR, BR TO TEST END.
38     032164          ;TRAP      C$ESCAPE
39     032164          ;.WORD    L10057-.
40     032164          MOV        #CNTRL,ERROR ;THIS FLAG WILL INHIBIT CONTROL OUT
41     032164          ;ERROR REPORTING - BECAUSE WE EXPECT ONE.
42     032164          ;BASE IN RESUME COMMAND WITH NXM BASE TABLE.
43     032164          BASEIN     0,160000,BIT15!BIT14!RES!DMR
44     032164          JSR        PC, $BASEI ;**** MACRO EXPANSION ****
45     032164          ;CALL BASE IN ROUTINE
46     032164          ;.WORD    0      ;MAINTENANCE MODE BITS TO SET IN BSEL1
47     032164          ;.WORD    160000 ;BASE TABLE ADDRESS
48     032164          ;.WORD    BIT15!BIT14!RES!DMR ;MODE
49     032164          ;****
50     032164          WAIT      RDO      ;WAIT FOR RDO TO BE SET
51     032164          ;**** MACRO EXPANSION ****

```

TEST 9 - NON-EXISTENT MEMORY ERROR

```

032244 004737 010272          JSR    PC, $WAIT          ;CALL WAIT ROUTINE
032250 000001                   .WORD    1                ;FLAG THAT WE'RE WAITING FOR RDO
                                ;*****
30 032252 032777 000001 147754  BIT    #CNTRL,@SEL2        ;IS THERE A CONTROL OUT REPORTED ?
31 032260 001005                   BNE    10$                ;IF YES, PROCEED.
32 032262                   ERRDF   8,EMG8,ERRG2      ;EXPECTED CONTROL OUT
                                TRAP    C$ERDF
                                .WORD    8
                                .WORD    EMG8
                                .WORD    ERRG2
032262 104455
032264 000010
032266 020130
032270 015126
33 032272 000410          BR     20$                ;EXIT
34 032274                   10$:
35 032274 032777 000400 147736  BIT    #NXM,@SEL6        ;IS THE NXM FLAG SET?
36 032302 001004                   BNE    20$                ;IF YES - ERROR REPORTED CORRECTLY
37 032304                   ERRDF   9,EMG9,ERRG2      ;UNEXPECTED CONTROL OUT RECEIVED
                                TRAP    C$ERDF
                                .WORD    9
                                .WORD    EMG9
                                .WORD    ERRG2
032304 104455
032306 000011
032310 020174
032312 015126
38 032314                   20$:
39 032314 042777 000207 147712  BIC    #RDO!CMD,@SEL2    ;CLEAR RDO AND THE COMMAND BITS
40 032322 005037 002364          CLR    ERROR              ;ALLOW ERROR REPORTING
41 032326                   ENDSUB
                                L10060:
42 032326 104403                   TRAP    C$ESUB
43 032330                   BGNSUB
                                T9.2:
44 032332                   CLEAR          ;MACRO FOR MASTER CLEAR
                                ;***** MACRO EXPANSION *****
                                ;ISSUE A DMR MASTER CLEAR
                                ;*****
                                TRAP    C$BSUB
032332 004737 011070          JSR    PC, $MSCLR
45
46 032336                   ESCAPE  TST              ;IF ERROR, BR TO TEST END.
                                TRAP    C$ESCAPE
                                .WORD    L10057-
032336 104410
032340 000334
47 032342                   BASEIN          ;MACRO FOR BASE IN COMMAND
                                ;***** MACRO EXPANSION *****
                                ;CALL BASE IN ROUTINE WITH DEFAULTS
                                ;SET LINE UNIT LOOP
                                ;BASE TABLE ADDRESS
                                ;DMR-11 MODE
                                ;*****
032342 004737 011266          JSR    PC, $BASEI
032346 004000                   .WORD    LPLU
032350 002634                   .WORD    BASE
032352 000522                   .WORD    DMR
48
49 032354                   ESCAPE  TST              ;IF ERROR, BR TO TEST END.
                                TRAP    C$ESCAPE
                                .WORD    L10057-
032354 104410
032356 000316
50 032360                   CNTRIN          ;MACRO FOR CONTROL IN (FULL DUPLEX)
                                ;***** MACRO EXPANSION *****
                                ;CALL CONTROL IN ROUTINE WITH DEFAULT
                                ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
                                ;*****
032360 004737 011522          JSR    PC, $CNTIN
032364 000000                   .WORD    0
51
52 032366                   ESCAPE  TST              ;IF ERROR, BR TO TEST END.
                                TRAP    C$ESCAPE
                                .WORD    L10057-
032366 104410
032370 000304

```

```

53 032372 012737 000001 002364      MOV      #CNTRL,ERROR      ;INHIBIT CONTROL OUT ERROR REPORTING AGAIN.
54
55
56
57 032400      BACCIR  160000,BIT15.BIT14.RCOUNT
                                     ;BA/CC IN REC. COMMAND WITH NXM
                                     ;ADDR = 760000 AND A CHARACTER COUNT = 3.
                                     ;**** MACRO EXPANSION ****
032400 004737 012300      JSR      PC, $BACC        ;CALL BA/CC IN ROUTINE
032404 000044      .WORD   RQI!BACCR        ;BA/CC IN RECEIVE COMMAND
032406 160000      .WORD   160000          ;BUFFER ADDRESS BITS 0-15
032410 140044      .WORD   BIT15!BIT14!RCOUNT ;BA BITS 16/17 AND CHAR. COUNT
                                     ;****
58
59 032412      BACCIT
                                     ;BA/CC IN XMIT
                                     ;**** MACRO EXPANSION ****
032412 004737 012300      JSR      PC, $BACC        ;CALL BA/CC IN ROUTINE WITH DEFAULTS
032416 000040      .WORD   RQI!BACCT       ;BA/CC IN TRANSMIT COMMAND
032420 002516      .WORD   TBUF            ;TRANSMIT BUFFER ADDRESS
032422 000044      .WORD   TCOUNT        ;TRANSMIT CHARACTER COUNT
                                     ;****
60
61 032424      WAIT    RDO
                                     ;WAIT FOR RDO
                                     ;**** MACRO EXPANSION ****
032424 004737 010272      JSR      PC, $WAIT       ;CALL WAIT ROUTINE
032430 000001      .WORD   1               ;FLAG THAT WE'RE WAITING FOR RDO
                                     ;****
62 032432 032777 000001 147574      BIT      #CNTRL,@SEL2    ;IS THERE A CONTROL OUT REPORTED ?
63 032440 001005      BNE     10$             ;IF YES, PROCEED.
64 032442      ERRDF  8,EMG8,ERRG2   ;EXPECTED CONTROL OUT
032442 104455
032444 000010      TRAP   C$ERDF
032446 020130      .WORD   8
032450 015126      .WORD   EMG8
65 032452 000410      BR      20$            ;EXIT
032454
66 032454      10$:
67 032454 032777 000400 147556      BIT      #NXM,@SEL6     ;IS THE NXM FLAG SET?
68 032462 001004      BNE     20$             ;IF YES - ERROR REPORTED CORRECTLY
69 032464      ERRDF  9,EMG9,ERRG2   ;UNEXPECTED CONTROL OUT RECEIVED
032464 104455      TRAP   C$ERDF
032466 000011      .WORD   9
032470 020174      .WORD   EMG9
032472 015126      .WORD   ERRG2
70
71 032474      20$:
72 032474 042777 000207 147532      BIC     #RDO!CMD,@SEL2  ;CLEAR RDO AND THE COMMAND BITS.
73 032502 005037 002364      CLR     ERROR           ;ENABLE ERROR REPORTING
74 032506      ENDSUB
032506
032506 104403      L10061:
75
76 032510      BGNSUB
032510
032510 104402      T9.3:
77 032512      CLEAR
                                     ;MACRO FOR MASTER CLEAR
032512 004737 011070      JSR      PC, $MSCLR     ;**** MACRO EXPANSION ****
                                     ;ISSUE A DMR MASTER CLEAR
                                     ;****
78
    
```



```

032634 015126
99 032636 000410
100 032640
101 032640 032777 000400 147372
102 032646 001004
103 032650
    032650 104455
    032652 000011
    032654 020174
    032656 015126
104 032660
105 032660 042777 000207 147346
106 032666 005037 002364
107 032672
    032672
    032672 104403
108
109 032674
    032674
    032674 104401
110
111
112
113
114
115
    
```

```

10$:
BR 20$ ;EXIT
BIT #NXM,@SEL6 ;IS THE NXM FLAG SET?
BNE 20$ ;IF YES - ERROR REPORTED CORRECTLY
ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
TRAP C$ERDF
.WORD 9
.WORD EMG9
.WORD ERRG2

20$:
BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
CLR ERROR ;DON'T INHIBIT CONTROL OUT ERRORS
ENDSUB
L10062:
TRAP C$ESUB

ENDTST
L10057:
TRAP C$ETST
    
```

```

1          .SBTTL          TEST 10 - TIME OUT ERROR
2
3          :*****
4          :*          TEST 10 - DMR-11
5          :* TIME OUT - FORCE A TIMEOUT AND VERIFY THAT THE ERROR IS REPORTED
6          :*
7          :*****
8 032676   BGNTST
9 032676   CLEAR          ;MACRO FOR MASTER CLEAR
          032676 004737 011070 JSR PC, $MSCLR          ;**** MACRO EXPANSION ****
          ;ISSUE A DMR MASTER CLEAR
          ;****
10
11 032702   ESCAPE TST          ;IF ERROR, BR TO TEST END.
          032702 104410 TRAP C$ESCAPE
          032704 000172 .WORD L10063-.
12 032706   BASEIN          ;MACRO FOR BASE IN COMMAND
          032706 004737 011266 JSR PC, $BASEI          ;**** MACRO EXPANSION ****
          032712 004000 .WORD LPLU          ;CALL BASE IN ROUTINE WITH DEFAULTS
          032714 002634 .WORD BASE          ;SET LINE UNIT LOOP
          032716 000522 .WORD DMR          ;BASE TABLE ADDRESS
          ;DMR-11 MODE
          ;****
13
14          ;SET THRESHOLD VALUES AS FOLLOWS:
15          ;BSEL4 = NAKS RECEIVED (377)
16          ;BSEL5 = NAKS TRANSMITTED (377)
17          ;BSEL6 = REP/SEL SENT (1)
18          ;BSEL7 = NO BUFFFER (377)
19 032720   DMRIN THRESH,177777,177401
          032720 004737 012070 JSR PC, $DMRIN          ;**** MACRO EXPANSION ****
          032724 000013 .WORD THRESH          ;CALL DMR MODE INPUT ROUTINE
          032726 177777 .WORD 177777          ;INPUT COMMAND
          032730 177401 .WORD 177401          ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
          ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
          ;****
20
21 032732   ESCAPE TST          ;IF ERROR, BR TO TEST END
          032732 104410 TRAP C$ESCAPE
          032734 000142 .WORD L10063-.
22 032736   DMRIN TIMER,0,1          ;SET REP/SEL TIMER TO MINIMUM (100 MSEC)
          032736 004737 012070 JSR PC, $DMRIN          ;**** MACRO EXPANSION ****
          032742 000012 .WORD TIMER          ;CALL DMR MODE INPUT ROUTINE
          032744 000000 .WORD 0          ;INPUT COMMAND
          032746 000001 .WORD 1          ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
          ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
          ;****
23
24 032750   ESCAPE TST          ;IF ERROR, BR TO TEST END.
          032750 104410 TRAP C$ESCAPE
          032752 000124 .WORD L10063-.
25 032754   CNTRIN          ;MACRO FOR CONTROL IN (FULL DUPLEX)
          032754 004737 011522 JSR PC, $CNTIN          ;**** MACRO EXPANSION ****
          032760 000000 .WORD 0          ;CALL CONTROL IN ROUTINE WITH DEFAULT
          ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
          ;****

```

```

26
27 032762          ESCAPE TST          ;IF ERROR, BR TO TEST END.
   032762 104410
   032764 000112          TRAP C$ESCAPE
                              .WORD L10063-.
28
29
30 032766          DMRIN WMODEM,0,BIT4 ;BLIND THE RECEIVER BY GOING INTO HDX.
                              ;USE WRITE MODEM COMMAND TO SET HALF DUPLEX.
   032766 004737 012070 JSR PC, $DMRIN          ;**** MACRO EXPANSION ****
   032772 000005          .WORD WMODEM          ;CALL DMR MODE INPUT ROUTINE
   032774 000000          .WORD 0              ;INPUT COMMAND
   032776 000020          .WORD BIT4           ;SEL4 VALUE (OR BITS TO CLEAR IN BSEL6)
                              ;SEL6 VALUE (OR BITS TO SET IN BSEL6)
                              ;****          ****
31
32 033000          BACCIT          ;BA/CC IN XMIT BUFFER
   033000 004737 012300 JSR PC, $BACC          ;**** MACRO EXPANSION ****
   033004 000040          .WORD RQI!BACCT          ;CALL BA/CC IN ROUTINE WITH DEFAULTS
   033006 002516          .WORD TBUF           ;BA/CC IN TRANSMIT COMMAND
   033010 000044          .WORD TCOUNT        ;TRANSMIT BUFFER ADDRESS
                              ;TRANSMIT CHARACTER COUNT
                              ;****          ****
33
34 033012          ESCAPE TST          ;IF ERROR, EXIT
   033012 104410
   033014 000062          TRAP C$ESCAPE
                              .WORD L10063-.
35 033016          WAIT RDO          ;WAIT FOR THE READY OUT.
   033016 004737 010272 JSR PC, $WAIT          ;**** MACRO EXPANSION ****
   033022 000001          .WORD 1              ;CALL WAIT ROUTINE
                              ;FLAG THAT WE'RE WAITING FOR RDO
                              ;****          ****
36 033024          ESCAPE TST          ;IF ERROR, EXIT.
   033024 104410
   033026 000050          TRAP C$ESCAPE
                              .WORD L10063-.
37 033030          BIT #CNTRL,@SEL2    ;IS THIS A CONTROL OUT
   033030 032777 000001 147176 BNE 10$              ;IF YES, PROCEED.
38 033036          ERRDF 8,EMG8,ERRG2 ;EXPECTED A CONTROL OUT.
   033036 001005
39 033040          TRAP C$ERDF
   033040 104455          .WORD 8
   033042 000010          .WORD EMG8
   033044 020130          .WORD ERRG2
   033046 015126
40 033050          BR 20$              ;EXIT
   033050 000410
41 033052          10$:
42 033052          BIT #TOUT,@SEL6    ;WAS THE TIME OUT REPORTED?
   033052 032777 000002 147160 BNE 20$              ;IF YES, EXIT
43 033060          ERRDF 9,EMG9,ERRG2 ;UNEXPECTED ERROR.
   033060 001004
44 033062          TRAP C$ERDF
   033062 104455          .WORD 9
   033064 000011          .WORD EMG9
   033066 020174          .WORD ERRG2
   033070 015126
45 033072          20$:
46 033072          SHUTDN
   033072 004737 012564 JSR PC, $HALT          ;**** MACRO EXPANSION ****
                              ;DMR HALT ROUTINE.
                              ;****          ****
47
48 033076          ENDTST
   033076

```

L10063:

033076 104401

TRAP C\$ETST

TEST 11 - MESSAGE TOO LONG ERROR

```

1          .SBTTL          TEST 11 - MESSAGE TOO LONG ERROR
2
3          :*****
4          :*          TEST 11 - DMR-11
5          :* MESSAGE TOO LONG - TRANSMIT A MESSAGE THAT IS TOO LONG FOR THE
6          :* RECEIVE BUFFER AND VERIFY THAT THE 'TOO LONG' ERROR IS RECEIVED.
7          :*
8          :*****
9 033100    BGNTST
10 033100    CLEAR          ;MACRO FOR MASTER CLEAR          T11::
          033100 004737 011070 JSR      PC, $MSCLR      ;**** MACRO EXPANSION ****
          ;ISSUE A DMR MASTER CLEAR
          ;****          ****
11
12 033104    ESCAPE TST          ;IF ERROR, BR TO TEST END.
          033104 104410          TRAP      C$ESCAPF
          033106 000150          .WORD   L10064-.
13 033110    BASEIN          ;MACRO FOR BASE IN COMMAND
          033110 004737 011266 JSR      PC, $BASEI      ;**** MACRO EXPANSION ****
          ;CALL BASE IN ROUTINE WITH DEFAULTS
          033114 004000          .WORD   LPLU      ;SET LINE UNIT LOOP
          033116 002634          .WORD   BASE      ;BASE TABLE ADDRESS
          033120 000522          .WORD   DMR      ;DMR-11 MODE
          ;****          ****
14
15 033122    ESCAPE TST          ;IF ERROR, BR TO TEST END.
          033122 104410          TRAP      C$ESCAPE
          033124 000132          .WORD   L10064-.
16 033126    CNTRIN          ;MACRO FOR CONTROL IN (FULL DUPLEX)
          033126 004737 011522 JSR      PC, $CNTIN      ;**** MACRO EXPANSION ****
          ;CALL CONTROL IN ROUTINE WITH DEFAULT
          033132 000000          .WORD   0      ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
          ;****          ****
17
18 033134    ESCAPE TST          ;IF ERROR, BR TO TEST END.
          033134 104410          TRAP      C$ESCAPE
          033136 000120          .WORD   L10064-.
19 033140    BACCIR RBUF,RCOUNT/2 ;SET UP THE RECEIVE BUFFER WITH 1/2 BUF. SPACE
          033140 004737 012300 JSR      PC, $BACC      ;**** MACRO EXPANSION ****
          ;CALL BA/CC IN ROUTINE
          033144 000044          .WORD   RQI!BACCR ;BA/CC IN RECEIVE COMMAND
          033146 002566          .WORD   RBUF      ;BUFFER ADDRESS BITS 0-15
          033150 000022          .WORD   RCOUNT/2 ;BA BITS 16/17 AND CHAR. COUNT
          ;****          ****
20
21 033152    MOV      #CNTRL,ERROR ;THIS FLAG WILL DISABLE ANY CONTROL OUT ERROR
22          ;REPORTING BECAUSE WE ARE INTENTIONALLY
23          ;CAUSING ONE IN THIS TEST.
24 033160    BACCIT          ;BA/CC IN XMIT COMMAND
          033160 004737 012300 JSR      PC, $BACC      ;**** MACRO EXPANSION ****
          ;CALL BA/CC IN ROUTINE WITH DEFAULTS
          033164 000040          .WORD   RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
          033166 002516          .WORD   TBUF      ;TRANSMIT BUFFER ADDRESS
          033170 000044          .WORD   TCOUNT ;TRANSMIT CHARACTER COUNT
          ;****          ****
25 033172    10$:

```

```

26 033172          WAIT  RDO          ;WAIT FOR RDO TO BE SET
      033172 004737 010272          ;**** MACRO EXPANSION ****
      033176 000001          JSR  PC, $WAIT ;CALL WAIT ROUTINE
      ;FLAG THAT WE'RE WAITING FOR RDO
      ;****          ****
27 033200          ESCAPE TST          ;IF RDO NOT SET, BR TO TEST END.
      033200 104410          TRAP  C$ESCAPE
      033202 000054          .WORD L10064-.
28 033204          BIT    #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
29 033212          BNE   20$          ;IF YES, PROCEED
30 033214          ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT.
      033214 104455          TRAP  C$ERDF
      033216 C00010          .WORD 8
      033220 020130          .WORD EMG8
      033222 015126          .WORD ERRG2
31 033224          BR    40$          ;EXIT
      033226          20$:
32 033226          BIT    #TOLONG,@SEL6 ;IS THE TOO LONG BIT SET?
33 033226 032777 000020 147004      BNE   40$          ;IF YES, TEST OK - FINISH UP.
34 033234          ERRDF 9,EMG9,ERRG2 ;WE'RE NOT GETTING EXPECTED RESULT
35 033236          30$:
      033236 104455          TRAP  C$ERDF
      033240 000011          .WORD 9
      033242 020174          .WORD EMG9
      033244 015126          .WORD ERRG2
37
38 033246          40$:
39 033246 005037 002364          CLR  ERROR          ;RESTORE ERROR FLAG TO NORMAL STATE.
40 033252          SHUTDN ;HALT THE DMR.
      033252 004737 012564          JSR  PC, $HALT      ;**** MACRO EXPANSION ****
      ;DMR HALT ROUTINE.
      ;****          ****
41
42
43
44 033256          ENDTST
      033256          L10064:
      033256 104401          TRAP  C$ETST
45
46
  
```

TEST 12 - PROCEDURE ERRORS

```

1      .SBTTL          TEST 12 - PROCEDURE ERRORS
2
3      :*****
4      :*              TEST 12 - DMR-11
5      :*  PROCEDURE ERRORS -
6      :*  THE FOLLOWING SHOULD CAUSE THE DMR-11 TO HALT AND RESPOND WITH
7      :*  A PROCEDURE ERROR:
8      :*  SUBTEST 1 - A SECOND BASE IN COMMAND
9      :*  SUBTEST 2 - A CONTROL IN BEFORE A BASE IN
10     :*  SUBTEST 3 - A BA/CC IN BEFORE A BASE IN
11     :*  SUBTEST 4 - A BA/CC IN RCV WITH A BUFFER LENGTH OF 0
12     :*  SUBTEST 5 - A BA/CC IN XMIT. WITH A BUFFER LENGTH OF 0
13     :*
14     :*****
15     BGNTST
16     033260          BGNSUB          T12::
17     033260          104402          T12.1:      TRAP      C$BSUB
18     033262          CLEAR          ;MASTER CLEAR MACRO
19     033262 004737 011070          JSR      PC, $MSCLR      ;**** MACRO EXPANSION ****
20     033266          BASEIN          ;ISSUE A DMR MASTER CLEAR
21     033266 004737 011266          JSR      PC, $BASEI      ;****
22     033272 004000          .WORD    LPLU          ;**** MACRO EXPANSION ****
23     033274 002634          .WORD    BASE          ;CALL BASE IN ROUTINE WITH DEFAULTS
24     033276 000522          .WORD    DMR           ;SET LINE UNIT LOOP
25     033300 012737 000001 002364  MOV     #CNTRL,ERROR ;BASE TABLE ADDRESS
26     033306          BASEIN          ;DMR-11 MODE
27     033306 004737 011266          JSR      PC, $BASEI      ;****
28     033312 004000          .WORD    LPLU          ;**** MACRO EXPANSION ****
29     033314 002634          .WORD    BASE          ;CALL BASE IN ROUTINE WITH DEFAULTS
30     033316 000522          .WORD    DMR           ;SET LINE UNIT LOOP
31     033320          WAIT    RDO      ;BASE TABLE ADDRESS
32     033320 004737 010272          JSR      PC, $WAIT      ;DMR-11 MODE
33     033324 000001          .WORD    1             ;****
34     033326          ESCAPE  TST      ;WAIT FOR RDO TO BE SET
35     033326 104410          .WORD    1             ;**** MACRO EXPANSION ****
36     033330 000632          .WORD    1             ;CALL WAIT ROUTINE
37     033332 032777 000001 146674  BIT     #CNTRL,@SEL2    ;FLAG THAT WE'RE WAITING FOR RDO
38     033334 001005          BNE     10$           ;****
39     033342          ERRDF  8,EMG8,ERRG2 ;IF RDO NOT SET, BR TO TEST END.
40     033342 104455          .WORD    8             TRAP      C$ESCAPE
41     033344 000010          .WORD    8             .WORD    L10065-

```

TEST 12 - PROCEDURE ERRORS

```

033346 020130
033350 015126
32 033352 000410
33 033354
34 033354 032777 001000 146656
35 033362 001004
36 033364
033364 104455
033366 000011
033370 020174
033372 015126
37 033374
38 033374 042777 000207 146632
39 033402 005037 002364
40 033406
033406
033406 104403
41
42 033410
033410
033410 104402
43
44 033412
033412 004737 011070
45
46 033416 012737 000001 002364
47
48
49 033424 005037 002260
50
51
52
53 033430
033430 004737 011522
033434 000000
54
55 033436
033436 004737 010272
033442 000001
56 033444
033444 104410
033446 000514
57 033450 032777 000001 146556
58 033456 001005
59 033460
033460 104455
033462 000010
033464 020130
033466 015126
60 033470 000410
61 033472

```

10\$: BR 15\$;EXIT

10\$: BIT #HALTC,@SEL6 ;IS THE HALT - PROCEDURE ERROR BIT SET?
BNE 15\$;IF YES - ERROR REPORTED CORRECTLY
ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED

15\$: BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS
CLR ERROR ;RESTORE FLAG

ENDSUB

L10066: TRAP C\$ESUB

BGNSUB

T12.2: TRAP C\$BSUB

CLEAR ;MASTER CLEAR MACRO
;**** MACRO EXPANSION ****
JSR PC, \$MSCLR ;ISSUE A DMR MASTER CLEAR
;****

MOV #CNTRL,ERROR ;THIS FLAG WILL DISABLE ANY CONTROL OUT ERROR
;REPORTING BECAUSE WE ARE INTENTIONALLY
;CAUSING ONE IN THIS TEST.
CLR DMRFLG ;CLEAR FLAG THAT IS SET IN BASEIN IN ORDER
;TO FLAG THAT A CONTROL OUT-DMR RUN MODE
;COMMAND IS EXPECTED (THIS FLAG WAS SET IN
;THE PREVIOUS SUBTEST BASEIN)
CNTRIN ;CONTROL IN
;**** MACRO EXPANSION ****
JSR PC, \$CNTIN ;CALL CONTROL IN ROUTINE WITH DEFAULT
.WORD 0 ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
;****

WAIT RDO ;WAIT FOR RDO TO BE SET
;**** MACRO EXPANSION ****
JSR PC, \$WAIT ;CALL WAIT ROUTINE
.WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
;****

ESCAPE TST ;IF RDO NOT SET, BR TO TEST END.

TRAP C\$ESCAPE
.WORD L10065-

BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
BNE 10\$;IF YES - PROCEED.
ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT

TRAP C\$ERDF
.WORD 8
.WORD EMG8
.WORD ERRG2

10\$: BR 15\$;EXIT


```

TEST 12 - PROCEDURE ERRORS

      033626 015126
89 033630
90 033630 042777 000207 146376 15$:
91 033636 005037 002364      BIC      #RDO.CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
92 033642      EP.DSUB      CLR      ERROR      ;RESTORE FLAG
      033642
      033642 104403      L10070:
93      TRAP      C$ESUB
94 033644      BGNSUB
      033644
      033644 104402      T12.4:
95 033646      CLEAR      ;MASTER CLEAR
      ;**** MACRO EXPANSION ****
      033646 004737 011070      JSR      PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
      ;****
96
97 033652      ESCAPE TST      ;IF ERROR, EXIT.
      033652 104410      TRAP      C$ESCAPE
      033654 000306      .WORD      L10065-.
98 033656      BASEIN      ;BASE IN COMMAND
      ;**** MACRO EXPANSION ****
      033656 004737 011266      JSR      PC, $BASEI ;CALL BASE IN ROUTINE WITH DEFAULTS
      .WORD      LPLU      ;SET LINE UNIT LOOP
      .WORD      BASE      ;BASE TABLE ADDRESS
      .WORD      DMR      ;DMR-11 MODE
      ;****
99
100 033670      ESCAPE TST      ;IF ERROR, EXIT.
      033670 104410      TRAP      C$ESCAPE
      033672 000270      .WORD      L10065-.
101 033674      BACCIR      ;ASSIGN A BA/CC IN RECEIVE BUFFER
      ;**** MACRO EXPANSION ****
      033674 004737 012300      JSR      PC, $BACC ;CALL BA/CC IN ROUTINE WITH DEFAULTS
      .WORD      RQI!BACCR ;BA/CC IN RECEIVE COMMAND
      .WORD      RBUF      ;RECEIVE BUFFER
      .WORD      RCOUNT ;RECEIVE CHARACTER COUNT
      ;****
102
103 033706      ESCAPE TST      ;IF ERROR, EXIT.
      033706 104410      TRAP      C$ESCAPE
      033710 000252      .WORD      L10065-.
104 033712 012737 000001 002364      MOV      #CNTRL,ERROR ;THIS FLAG WILL DISABLE ANY CONTROL OUT
105      ;ERROR REPORTING BECAUSE WE ARE INTENTIONALLY
106      ;CAUSING ONE.
107 033720      BACCIT TBUF,0 ;ASSIGN A BA/CC IN XMIT BUFFER LENGTH = 0.
      ;**** MACRO EXPANSION ****
      033720 004737 012300      JSR      PC, $BACC ;CALL BA/CC IN ROUTINE
      .WORD      RQI!BACCT ;BA/CC IN TRANSMIT COMMAND
      .WORD      TBUF      ;BUFFER ADDRESS BITS 0-15
      .WORD      0 ;BA BITS 16 & 17 AND CHAR. COUNT
      ;****
108
109 033732      WAIT      RDO      ;WAIT FOR RDO TO BE SET
      ;**** MACRO EXPANSION ****
      033732 004737 010272      JSR      PC, $WAIT ;CALL WAIT ROUTINE
      .WORD      1 ;FLAG THAT WE'RE WAITING FOR RDO
      ;****

```

TEST 12 - PROCEDURE ERRORS

```

110 033740          ESCAPE TST          ;IF RDO NOT SET, BR TO TEST END.
    033740 104410
    033742 000220          TRAP C$ESCAPE
111 033744 032777 000001 146262      BIT #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
112 033752 001005          BNE 10$          ;IF YES - PROCEED.
113 033754          ERRDF 8,EMG8,ERRG2 ;EXPECTED CONTROL OUT
    033754 104455          TRAP C$ERDF
    033756 000010          .WORD 8
    033760 020130          .WORD EMG8
    033762 015126          .WORD ERRG2
114 033764 000410          BR 15$          ;EXIT
115 033766          10$:
116 033766 032777 001000 146244      BIT #HALTC,@SEL6 ;IS THE HALT - PROCEDURE ERROR BIT SET?
117 033774 001004          BNE 15$          ;IF YES - ERROR REPORTED CORRECTLY
118 033776          ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
    033776 104455          TRAP C$ERDF
    034000 000011          .WORD 9
    034002 020174          .WORD EMG9
    034004 015126          .WORD ERRG2
119 034006          15$:
120 034006 042777 000207 146220      BIC #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
121 034014 005037 002364          CLR ERROR ;RESTORE FLAG
122 034020          ENDSUB
    034020 104403          L10071: TRAP C$ESUB
123 034022          BGNSUB
124 034022 104402          T12.5: TRAP C$BSUB
125 034024          CLEAR ;MASTER CLEAR
    034024 004737 011070          JSR PC, $MSCLR ;**** MACRO EXPANSION ****
    ;ISSUE A DMR MASTER CLEAR
    ;****
126 034030          ESCAPE TST          ;IF ERROR, EX.T.
127 034030 104410          TRAP C$ESCAPE
    034032 000130          .WORD L10065-.
128 034034          BASEIN
    034034 004737 011266          JSR PC, $BASEI ;BASE IN COMMAND
    034040 004000          .WORD LPLU ;**** MACRO EXPANSION ****
    034042 002634          .WORD BASE ;CALL BASE IN ROUTINE WITH DEFAULTS
    034044 000522          .WORD DMR ;SET LINE UNIT LOOP
    ;BASE TABLE ADDRESS
    ;DMR-11 MODE
    ;****
129 034046          ESCAPE TST          ;IF ERROR, EXIT.
130 034046 104410          TRAP C$ESCAPE
    034050 000112          .WORD L10065-.
131 034052 012737 000001 002364      MOV #CNTRL,ERROR ;THIS FLAG WILL DISABLE ANY CONTROL OUT
132 034052          ;ERROR REPORTING BECAUSE WE ARE INTENTIONALLY
133 034052          ;CAUSING ONE.
134 034060          BACCIR RBUF,0 ;ASSIGN A BA/CC IN REC. BUFFER LENGTH = 0
    034060 004737 012300          JSR PC, $BACC ;**** MACRO EXPANSION ****
    034064 000044          .WORD RQI!BACCR ;CALL BA/CC IN ROUTINE
    034066 002566          .WORD RBUF ;BA/CC IN RECEIVE COMMAND
    034070 000000          .WORD 0 ;BUFFER ADDRESS BITS 0-15
    ;BA BITS 16/17 AND CHAR. COUNT

```

```

135
136 034072          WAIT  RDO          ;*****          *****
      034072 004737 010272          ;WAIT FOR RDO TO BE SET
      034076 000001          JSR    PC, $WAIT ;***** MACRO EXPANSION *****
      ;CALL WAIT ROUTINE
      ;FLAG THAT WE'RE WAITING FOR RDO
137 034100          ESCAPE TST          ;*****          *****
      034100 104410          ;IF RDO NOT SET, BR TO TEST END.
      034102 000060          TRAP   C$ESCAPE
138 034104 032777 000001 146122      BIT    #CNTRL,@SEL2 ;IS THIS A CONTROL OUT?
139 034112 001005          BNE   10$ ;IF YES - PROCEED.
140 034114          ERRDF  8,EMG8,ERRG2 ;EXPECTED CONTROL OUT
      034114 104455          TRAP   C$ERDF
      034116 000010          .WORD  8
      034120 020130          .WORD  EMG8
      034122 015126          .WORD  ERRG2
141 034124 000410          BR     15$ ;EXIT
142 034126          10$:
143 034126 032777 001000 146104      BIT    #HALTC,@SEL6 ;IS THE HALT - PROCEDURE ERROR BIT SET?
144 034134 001004          BNE   15$ ;IF YES - ERROR REPORTED CORRECTLY
145 034136          ERRDF  9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
      034136 104455          TRAP   C$ERDF
      034140 000011          .WORD  9
      034142 020174          .WORD  EMG9
      034144 015126          .WORD  ERRG2
146 034146          15$:
147 034146 042777 000207 146060      BIC   #RDO!CMD,@SEL2 ;CLEAR RDO AND THE COMMAND BITS.
148 034154 005037 002364          CLR   ERROR ;RESTORE FLAG
149 034160          ENDSUB
      034160          L10072:
150 034160 104403          TRAP   C$ESUB
151 034162          ENDTST
      034162          L10065:
      034162 104401          TRAP   C$ETST

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

.SBTTL          TEST 13 - DATA TEST
:*****
:*              TEST 13 - DMR-11
:* FREE RUNNING FLAG MODE DATA TEST
:* TRANSMIT A MESSAGE AND VERIFY THE RECEIVED DATA IS CORRECT.
:* IN THIS TEST NO INTERRUPTS ARE USED AND THE LINE UNIT IS IN
:* INTERNAL (TTL) LOOPBACK. THIS TEST IS THE FIRST TEST IN WHICH
:* THE DMR IS USED IN A DATA TRANSMISSION MODE.
:*****
BGNTST
                                T13::
12 034164 013700 000044      MOV    RCOUNT,R0      ;BYTE COUNT FOR RECEIVE BUFFER
13 034170 062700 000002      ADD    #2,R0          ;2 ADDITIONAL BYTES AT END OF BUFFER ARE
                                ;USED FOR DELIMITOR
15 034174 012701 002566      MOV    #RBUF,R1      ;ADDRESS OF RECEIVE BUFFER
16 034200 105021              10$: CLRB  (R1)+          ;CLEAR A BYTE IN THE BUFFER
18 034202 005309              DEC    R0              ;CONTINUE - UNTIL ENTIRE BUFFER DONE
19 034204 001375              BNE   10$
21 034206 005037 002514      CLR    TFLAG         ;CLEAR TRANSMIT FLAG
22 034212 005037 002564      CLR    RFLAG         ;CLEAR RECEIVER FLAG
23 034216 004737 011070      CLEAR                ;MACRO FOR MASTER CLEAR
                                ;**** MACRO EXPANSION ****
                                ;ISSUE A DMR MASTER CLEAR
                                ;****
24 034216 004737 011070      JSR    PC,$MSCLR     ;
25 034222 104410              ESCAPE TST           ;IF ERROR, BR TO TEST END.
                                TRAP   C$ESCAPE
                                .WORD  L10073-.
26 034226 005737 002254      TST    DMTURN        ;IS INTERNAL LOOPBACK DESIRED?
27 034232 001004              BNE   11$            ;IF NOT, CLEAR INTERNAL LOOPBACK.
28 034234 052737 004000 034256  BIS    #LPLU,100$    ;SET LINE UNIT LOOPBACK.
29 034242 000403              BR    12$
30 034244 042737 004000 034256 11$: BIC    #LPLU,100$    ;CLEAR LINE UNIT LOOPBACK.
31 034252 100000              12$: CALL  $BASEI      ;BASE IN COMMAND.
32 034252 000000              100$: .WORD  0         ;MAINTENANCE BITS (LINE UNIT LOOP)
34 034256 002634              .WORD  BASE         ;BASE TABLE ADDRESS
35 034260 000522              .WORD  DMR          ;DMR MODE
36 034262 104410              ESCAPE TST           ;IF ERROR, BR TO TEST END.
37 034264 000424              TRAP   C$ESCAPE
38 034266 000424              .WORD  L10073-.
39 034270 104410              CALL  $LOOP          ;DMR COMMAND TO SET MAINT. BITS
40 034274 000414              ESCAPE TST           ;IF ERROR, BR TO TEST END.
                                TRAP   C$ESCAPE
                                .WORD  L10073-.
41 034274 104410
42 034276 000414
43 034300 004737 011522      CNTRIN              ;MACRO FOR CONTROL IN (FULL DUPLEX)
                                ;**** MACRO EXPANSION ****
                                ;CALL CONTROL IN ROUTINE WITH DEFAULT
                                ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
                                ;****
                                ;IF ERROR, BR TO TEST END.
                                ;****
                                ;IF ERROR, BR TO TEST END.
034300 004737 011522      JSR    PC,$CNTIN    ;
034304 000000              .WORD  0
034306 000000      ESCAPE TST
    
```

```

034306 104410
034310 000402                                TRAP  C$ESCAPE
44                                           .WORD  L10073-.
45 034312                                BACCIR      ;BUFFER ADDRESS/CHARACTER COUNT REC. IN
                                           ;**** MACRO EXPANSION ****
034312 004737 012300                        JSR  PC, $BACC      ;CALL BA/CC IN ROUTINE WITH DEFAULTS
034316 000044                                .WORD  RQI!BACCR    ;BA/CC IN RECEIVE COMMAND
034320 002566                                .WORD  RBUF         ;RECEIVE BUFFER
034322 000044                                .WORD  RCOUNT     ;RECEIVE CHARACTER COUNT
                                           ;****                ****

46
47 034324                                ESCAPE TST   ;IF ERROR (I.E. RDI NOT SET), ESCAPE
034324 104410                                TRAP  C$ESCAPE
034326 000364                                .WORD  L10073-.

48
49 034330                                BACCIT      ;BUFFER ADDRESS/CHARACTER COUNT XMIT. IN
                                           ;**** MACRO EXPANSION ****
034330 004737 012300                        JSR  PC, $BACC      ;CALL BA/CC IN ROUTINE WITH DEFAULTS
034334 000040                                .WORD  RQI!BACCT   ;BA/CC IN TRANSMIT COMMAND
034336 002516                                .WORD  TBUF        ;TRANSMIT BUFFER ADDRESS
034340 000044                                .WORD  TCOUNT     ;TRANSMIT CHARACTER COUNT
                                           ;****                ****

50
51 034342                                ESCAPE TST   ;IF ERROR (I.E. RDI NOT SET), ESCAPE
034342 104410                                TRAP  C$ESCAPE
034344 000346                                .WORD  L10073-.

52
53 034346                                20$:
54 034346                                WAIT  RDO     ;WAIT FOR RDO
                                           ;**** MACRO EXPANSION ****
034346 004737 010272                        JSR  PC, $WAIT     ;CALL WAIT ROUTINE
034352 000001                                .WORD  1           ;FLAG THAT WE'RE WAITING FOR RDO
                                           ;****                ****

55 034354                                BERROR 52$    ;IF ERROR - RDO NOT SET, END TEST
034354 103552                                BCS  52$
56 034356 032777 000001 145650            BIT  #CNTRL,@SEL2 ;IS THIS A CONTROL OUT COMMAND ?
57 034364 001405                                BEQ  25$         ;IF NOT - PROCEED
58 034366                                ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT RECEIVED
034366 104455                                TRAP  C$ERDF
034370 000011                                .WORD  9
034372 020174                                .WORD  EMG9
034374 015126                                .WORD  ERRG2

59 034376 000541                                BR  52$
60 034400                                25$:
61 034400 032777 000004 145626            BIT  #RCV,@SEL2   ;TRANSMIT OR RECEIVE ?
62 034406 001035                                BNE  40$         ;BR FOR RECEIVE

63
64 ;CHECK TRANSMIT
65
66 034410 005737 002514                        TST  TFLAG        ;IS THIS THE FIRST TRANSMIT DONE?
67 034414 001405                                BEQ  30$         ;YES - OK
68 034416                                ERRDF 10,EMG10,ERRG2 ;ERROR MULTIPLE TRANSMITS
034416 104455                                TRAP  C$ERDF
034420 000012                                .WORD  10
034422 020223                                .WORD  EMG10
034424 015126                                .WORD  ERRG2

69 034426 000525                                BR  52$

```

TEST 13 - DATA TEST

```

70 034430
71 034430 012737 177777 002514 30$: MOV # -1, TFLAG ; FLAG THAT TRANSMIT CHECK IS DONE.
72 034436 022777 002516 145572 CMP #TBUF, @SEL4 ; TRANSMIT BUFFER ADDRESS CORRECT?
73 034444 001405 BEQ 32$ ; YES - PROCEED
74 034446 ERRDF 11, EMG11, ERRG2 ; BUFFER ADDRESS ERROR
    034446 104455 TRAP C$ERDF
    034450 000013 .WORD 11
    034452 020252 .WORD EMG11
    034454 015126 .WORD ERRG2
75 034456 000511 BR 52$
76 034460
77 034460 022777 000044 145552 32$: CMP #TCOUNT, @SEL6 ; COUNT CORRECT ?
78 034466 001470 BEQ 50$ ; YES - PROCEED
79 034470 ERRDF 12, EMG12, ERRG2 ; CHARACTER COUNT ERROR
    034470 104455 TRAP C$ERDF
    034472 000014 .WORD 12
    034474 020277 .WORD EMG12
    034476 015126 .WORD ERRG2
80 034500 BR 52$
81
82 ; CHECK RECEIVE
83
84 034502
85 034502 005737 002564 40$: TST RFLAG ; IS THIS THE FIRST RECEIVE DONE ?
86 034506 001405 BEQ 41$ ; YES - PROCEED
87 034510 ERRDF 13, EMG13, ERRG2 ; MULTIPLE RECEIVES
    034510 104455 TRAP C$ERDF
    034512 000015 .WORD 13
    034514 020325 .WORD EMG13
    034516 015126 .WORD ERRG2
88 034520 000470 BR 52$
89 034522
90 034522 012737 177777 002564 41$: MOV # -1, RFLAG ; FLAG THAT RECEIVE CHECK HAS BEEN DONE.
91 034530 022777 002566 145500 CMP #RBUF, @SEL4 ; IS THE RECEIVE BUFFER ADDRESS CORRECT?
92 034536 001405 BEQ 43$ ; YES - PROCEED
93 034540 ERRDF 11, EMG11, ERRG2 ; BUFFER ADDRESS ERROR
    034540 104455 TRAP C$ERDF
    034542 000013 .WORD 11
    034544 020252 .WORD EMG11
    034546 015126 .WORD ERRG2
94 034550 000454 BR 52$
95 034552
96 034552 022777 000044 145460 43$: CMP #RCOUNT, @SEL6 ; IS THE BUFFER COUNT CORRECT?
97 034560 001405 BEQ 44$ ; YES - PROCEED
98 034562 ERRDF 12, EMG12, ERRG2 ; CHARACTER COUNT ERROR
    034562 104455 TRAP C$ERDF
    034564 000014 .WORD 12
    034566 020277 .WORD EMG12
    034570 015126 .WORD ERRG2
99 034572 000443 BR 52$
100 034574
101 034574 012700 000044 44$: MOV #RCOUNT, R0 ; SET UP FOR DATA CHECK (CHARCATER COUNT)
102 034600 012701 002516 MOV #TBUF, R1 ; GOOD DATA POINTER
103 034604 012702 002566 MOV #RBUF, R2 ; RECEIVE DATA POINTER
104 034610
105 034610 122122 45$: CMPB (R1)+, (R2)+ ; IS THE DATA THE SAME ?
106 034612 001011 BNE 46$ ; IF NOT, BRANCH TO DATA ERROR MESSAGE
    
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

.SBTTL          TEST 14 - EXTENDED ADDRESSING DATA TEST
:*****
:*              TEST 14 - DMR-11
:* IN THIS TEST - SEE IF WE HAVE MEMORY MANAGEMENT, IF SO SEE IF WE
:* HAVE THE MEMORY TO CHECK BITS 16 & 17 IN SEL6. THIS WILL ALLOW
:* US TO TRANSFER DATA USING THOSE EXTENDED ADDRESSING BITS. AS IN
:* TEST 13 THE TEST IS NON-INTERRUPT AND INTERNAL (TTL) LOOPBACK IS
:* USED.
:*
:*****
BGNTST
:
:              T14::
.ENABL  LSB          ;ENABLE LOCAL BLOCK - NEEDED BECAUSE OF
SETVEC  #4,#NOXMEM,#PRI07 ;USE OF SYMBOLIC LABELS 'RSEL4' ETC.
:              ;SET UP TRAP VECTOR 4
:              MOV      #PRI07,-(SP)
:              MOV      #NOXMEM,-(SP)
:              MOV      #4,-(SP)
:              MOV      #3,-(SP)
:              TRAP     C$SVEC
:              ADD      #10,SP
:
:              CLR     NXMFLG          ;CLEAR FLAG - SET IF TRAP TO 4.
:              TST     @#177572        ;ADDRESS MEMORY MANAGEMENT REGISTER.
:              CLRVEC  #4              ;RESTORE TRAP VECTOR 4.
:
:              MOV      #4,R0
:              TRAP     C$CVEC
:
:              TST     NXMFLG          ;IS THE FLAG STILL CLEARED?
:              ;NOTE: THE FLAG WILL BE SET BY TRAP 4
:              ;IF THERE IS NO MEMORY MANAGEMENT.
:              ;IF FLAG IS CLEARED, PROCEED WITH TEST.
:              BEQ     10$
:              CLR     NXMFLG          ;RESTORE FLAG
:              JMP     85$             ;EXIT - CAN'T TEST WITHOUT MEM. MANAG.
:
:              ;NOTE: L$HIMEM IS SIZE OF TOTAL MEMORY IN
:              ;PAGE ADDRESS REGISTER FORM - DETERMINED BY
:              ;BY DIAGNOSTIC SUPERVISOR AT STARTUP.
:              ;DO WE HAVE ENOUGH MEMORY TO ADDRESS BIT 16?
:              ;IF YES - PROCEED WITH TEST
:              ;IF NOT - EXIT
:
:              CMP     L$HIMEM,#2200
:              BGE     15$
:              JMP     85$
:
:              SETPRI  #PRI07          ;MAKE SURE WE ARE IN KERNEL MODE.
:              MOV      #PRI07,R0
:              TRAP     C$SPRI
:              ;SETTING PRI SHOULD ALSO CLEAR BITS 14 & 15
:              ;IN PSW WHICH PLACES PROCESSOR IN KERNEL MODE.
:              ;GET ADDRESS OF KERNEL PDR REG 0
:              ;GOING TO WRITE PDR REG 0-7
:
:              MOV     #172300,R1
:              MOV     #8.,R0
:
:              MOV     #77406,(R1)+    ;WRITE BITS FOR THE FOLLOWING PAGE DESCRIPTION
:              ;READ/WRITE ACCESS, 128. BLOCK PAGE LENGTH.
:              ;WRITE ALL PDRS
:
:              DEC     R0
:              BNE     20$
:              MOV     #172340,R1
:              CLR     (R1)
:              MOV     #200,2(R1)
:              MOV     #400,4(R1)
:              ;GET ADDRESS OF KERNAL PAR 0
:              ;PAR 0,  ADDRS  0 - 17776
:              ;PAR 1,  ADDRS 20000 - 37776
:              ;PAR 2,  ADDRS 40000 - 57776
    
```

```

47 035062 012761 000600 000006      MOV      #600,6(R1)      ;PAR 3,  ADDR  60000 - 77776
48 035070 012761 001000 000010      MOV      #1000,10(R1)   ;PAR 4,  ADDR  100000 - 117776
49 035076 012761 002000 000012      MOV      #2000,12(R1)   ;PAR 5,  ADDR  200000 - 217776
50 035104 012761 004000 000014      MOV      #4000,14(R1)   ;PAR 6,  ADDR  400000 - 417776
51 035112 012761 007600 000016      MOV      #7600,16(R1)   ;PAR 7,  ADDR  160000 - 177776 (I/O PAGE)
52
53 035120 012703 000100                MOV      #64.,R3        ;COUNTER FOR OUTER LOOP OF TEST PATTERN GEN.
54 035124 012704 120000                MOV      #120000,R4     ;USE VIRTUAL ADDRESS TO MAP TO PAR 5
55                                     ;GENERATE A TEST PATTERN IN THE 1ST 4K
56                                     ;BYTES OF PAR 5 (VIRTUAL ADDR 120000 - 127776)
57 035130 005037 002350                CLR      NXMFLG        ;ENSURE FLAG IS CLEARED
58 035134                SETVEC  #4,#NOXMEM,#PRI07 ;SET UP TRAP VECTOR 4 (WILL SET FLAG)
    035134 012746 000340                MOV      #PRI07,-(SP)
    035140 012746 023714                MOV      #NOXMEM,-(SP)
    035144 012746 000004                MOV      #4,-(SP)
    035150 012746 000003                MOV      #3,-(SP)
    035154 104437                        TRAP    C$SVEC
    035156 062706 000010                ADD     #10,SP
59 035162 012737 000001 177572      MOV      #1,@#177572   ;ENABLE MEMORY MANAGEMENT
60 035170                21$:
61 035170 012701 000040                MOV      #32.,R1       ;COUNTER FOR INNER LOOP OF TEST PATTERN GEN.
62 035174 012702 002414                MOV      #5CCITT,R2    ;ADDRESS FOR 32. WORD TEST PATTERN
63 035200                22$:
64 035200 012224                MOV      (R2)+,(R4)+   ;WRITE TEST PATTERN INTO 4K BYTES
65                                     ;(PHYSICAL ADDRESS 200000 - 207776)
66 035202 005737 002350                TST     NXMFLG        ;NXM TRAP 4?
67 035206 001014                BNE    24$            ;IF YES - EXIT
68 035210 005301                DEC     R1             ;DO THE INNER LOOP 32. TIMES
69 035212 001372                BNE    22$
70 035214 005303                DEC     R3             ;DO THE OUTER LOOP 128. TIMES
71 035216 001364                BNE    21$
72 035220 012701 004000                MOV      #4000,R1     ;COUNTER TO CLEAR THE NEXT 4K BYTES.
73 035224                23$:
74 035224 005024                CLR     (R4)+         ;CLEAR OUT THE ENTIRE PAR
75                                     ;(PHYSICAL ADDRESS 210000 - 217776)
76 035226 005737 002350                TST     NXMFLG        ;NXM TRAP 4?
77 035232 001002                BNE    24$            ;IF YES - EXIT
78 035234 005301                DEC     R1
79 035236 001372                BNE    23$
80 035240                24$:
81 035240 005037 177572                CLR     @#177572      ;TURN OFF MEMORY MANAGEMENT
82 035244                CLRVEC #4             ;RESTORE TRAP 4 TO SUPERVISOR
    035244 012700 000004                MOV     #4,R0
    035250 104436                TRAP   C$CVEC
83 035252 005737 002350                TST     NXMFLG        ;WAS THIS AN ERROR EXIT
84 035256 001417                BEQ    25$            ;IF NOT, PROCEED.
85 035260                ERRDF 19,EMT22
    035260 104455                TRAP   C$ERRDF
    035262 000023                .WORD 19
    035264 036340                .WORD EMT22
    035266 000000                .WORD 0
86 035270                PRINTB #FMT25,R4
    035270 010446                MOV     R4,-(SP)
    035272 012746 036376                MOV     #FMT25,-(SP)
    035276 012746 000002                MOV     #2,-(SP)
    035302 010600                MOV     SP,R0
    035304 104414                TRAP   C$PNTB

```

```

87 035306 062706 000006
88 035312 000137 036336
89 035316
      25$: JMP      85$
      CLEAR
      JSR      PC, $MSCLR
      ESCAPE  TST
      TST     DMTURN
      BNE     30$
      BIS     #LPLU,100$
      BR      32$
      30$:   BIC     #LPLU,100$
      32$:   CALL    $BASEI
      100$:  .WORD   0
      .WORD   BASE
      .WORD   DMR
      ESCAPE  TST
      105    CALL    $LOOP
      107    ESCAPE TST
      108    CNTRIN
      109    JSR      PC, $CNTIN
      .WORD   0
      110    ESCAPE TST
      111    CLR     TFLAG
      112    CLR     RFLAG
      113    CLR     SFLAG
      114    CLR     SFLAG
      115    ;IF SFLAG = 0, THEN THIS IS A TEST OF BIT 16
      116    ;IF SFLAG = -1, THEN THIS IS A TEST OF BIT 17
      117    MOV     #10000,RSEL4
      118    MOV     #BIT14!10000,RSEL6
      119    ;RECEIVE BUFFER ADDRESS (BITS 0-15)
      120    CLR     TSEL4
      121    MOV     #BIT14!10000,TSEL6
      122    ;TRANSMIT BUFFER ADDRESS (BITS 0-15)
      123    ;BYTE RECEIVE CHARACTER COUNT
      124    ;BYTE XMIT CHARACTER COUNT
      35$:   CALL    $BACC
      .WORD   RQI.BACCR
      RSEL4: .WORD   0
      RSEL6: .WORD   0
      ;ISSUE THE BUFFER ADDR/ CHAR COUNT COMMAND
      ;COMMAND FOR BA/CC IN RECEIVE
      ;BUFFER ADDRESS BITS 0-15
      ;BUFFER ADDR BIT 16 + CHAR. COUNT
      90     ;MACRO FOR MASTER CLEAR
      91     ;**** MACRO EXPANSION ****
      92     ;ISSUE A DMR MASTER CLEAR
      93     ;****
      94     ;IF ERROR, BR TO TEST END.
      95     TRAP   C$ESCAPE
      96     .WORD  L10074-
      97     ;IS INTERNAL LOOPBACK DESIRED?
      98     ;IF NOT, CLEAR INTERNAL LOOPBACK.
      99     ;SET LINE UNIT LOOPBACK.
      100    ;BASE IN COMMAND.
      101    ;MAINTENANCE BITS (LINE UNIT LOOP)
      102    ;BASE TABLE ADDRESS
      103    ;DMR MODE
      104    ;IF ERROR, BR TO TEST END.
      106    TRAP   C$ESCAPE
      107    .WORD  L10074-
      108    ;DMR COMMAND TO SET MAINT. BITS
      109    ;IF ERROR, BR TO TEST END.
      110    TRAP   C$ESCAPE
      111    .WORD  L10074-
      112    ;MACRO FOR CONTROL IN (FULL DUPLEX)
      113    ;**** MACRO EXPANSION ****
      114    ;CALL CONTROL IN ROUTINE WITH DEFAULT
      115    ;SEL6 - FULL DUPLEX, RUN MODE, 1 SEC START.
      116    ;****
      117    ;IF ERROR, BR TO TEST END.
      118    TRAP   C$ESCAPE
      119    .WORD  L10074-
      120    ;RECEIVE BUFFER ADDRESS (BITS 0-15)
      121    ;REC BUFFER ADDR BIT 16 SET AND 4K
      122    ;BYTE RECEIVE CHARACTER COUNT
      123    ;TRANSMIT BUFFER ADDRESS (BITS 0-15)
      124    ;XMIT BUFFER ADDR BIT 16 SET AND 4K
      125    ;BYTE XMIT CHARACTER COUNT
    
```

```

128 035466          ESCAPE TST          ;IF ERROR, END TEST
    035466 104410
    035470 000646          TRAP C$ESCAPE
                          .WORD L10074-.
129
130 035472          CALL  $BACC          ;ISSUE THE BUFFER ADDR/ CHAR COUNT COMMAND
131 035476 000040          .WORD RQI!BACCT ;COMMAND FOR BA/CC IN TRANSMIT
132 035500 000000          TSEL4: .WORD 0 ;BUFFER ADDRESS BITS 0-15
133 035502 000000          TSEL6: .WORD 0 ;BUFFER ADDR BIT 16 + CHAR. COUNT
134 035504          ESCAPE TST          ;IF ERROR, END TEST
    035504 104410
    035506 000630          TRAP C$ESCAPE
                          .WORD L10074-.
135 035510          40$:
136 035510          WAIT  RDO          ;WAIT FOR RDO TO BE SET
                          ;**** MACRO EXPANSION ****
    035510 004737 010272          JSR  PC, $WAIT          ;CALL WAIT ROUTINE
    035514 000001          .WORD 1 ;FLAG THAT WE'RE WAITING FOR RDO
                          ;****
137 035516          ESCAPE TST          ;IF RDO NOT SET BEFORE TIMEOUT, END TEST
    035516 104410          TRAP C$ESCAPE
    035520 000616          .WORD L10074-.
138
139 035522 032777 000001 144504          BIT  #CNTRL,@SEL2          ;IS THIS A CONTROL OUT COMMAND?
140 035530 001406          BEQ  50$          ;NO - PROCEED
141 035532          ERRDF 9,EMG9,ERRG2 ;UNEXPECTED CONTROL OUT.
    035532 104455          TRAP C$ERDF
    035534 000011          .WORD 9
    035536 020174          .WORD EMG9
    035540 015126          .WORD ERRG2
142 035542 000137 036326          JMP  80$          ;EXIT
143 035546          50$:
144 035546 032777 000004 144460          BIT  #RCV,@SEL2          ;IS THIS A TRANSMIT OR RECEIVE?
145 035554 001040          BNE  60$          ;BR FOR RECEIVE
146 035556 005737 002514          TST  TFLAG          ;IS THIS THE 1ST TRANSMIT DONE
147 035562 001406          BEQ  55$          ;IF YES, PROCEED
148 035564          ERRDF 10,EMG10,ERRG2 ;MULTIPLE TRANSMITS
    035564 104455          TRAP C$ERDF
    035566 000012          .WORD 10
    035570 020223          .WORD EMG10
    035572 015126          .WORD ERRG2
149 035574 000137 036326          JMP  80$          ;EXIT
150 035600          55$:
151 035600 012737 177777 002514          MOV  #-1,TFLAG          ;FLAG THAT THE TRANSMIT IS DONE.
152 035606 023777 035500 144422          CMP  TSEL4,@SEL4          ;IS THE BUFFER ADDRESS CORRECT?
153 035614 001406          BEQ  56$          ;IF OK, PROCEED WITH CHECK.
154 035616          ERRDF 11,EMG11,ERRG2 ;BUFFER ADDRESS ERROR
    035616 104455          TRAP C$ERDF
    035620 000013          .WORD 11
    035622 020252          .WORD EMG11
    035624 015126          .WORD ERRG2
155 035626 000137 036326          JMP  80$          ;EXIT
156 035632          56$:
157 035632 023777 035502 144400          CMP  TSEL6,@SEL6          ;IS THE CHAR. COUNT CORRECT?
158 035640 001502          BEQ  70$          ;IF OK, PROCEED
159 035642          ERRDF 12,EMG12,ERRG2 ;CHARACTER COUNT ERROR - OR EXT MEM PROBLEM
    035642 104455          TRAP C$ERDF
    035644 000014          .WORD 12
    035646 020277          .WORD EMG12
    
```

```

160 035650 015126
160 035652 000137 036326
161 035656
162 035656 005737 002564
163 035662 001406
164 035664
    035664 104455
    035666 000015
    035670 020325
    035672 015126
165 035674 000137 036326
166 035700
167 035700 012737 177777 002564
168 035706 023777 035462 144322
169 035714 001405
170 035716
    035716 104455
    035720 000013
    035722 020252
    035724 015126
171 035726 000577
172 035730
173 035730 023777 035464 144302
174 035736 001404
175 035740
    035740 104455
    035742 000014
    035744 020277
    035746 015126
176 035750
177 035750 005737 002344
178 035754 001007
179 035756 012700 004000
180 035762 012701 120000
181
182 035766 012702 130000
183 035772 000406
184 035774
185 035774 012700 010000
186 036000 012701 120000
187 036004 012702 140000
188 036010
189 036010 012737 000001 177572
190 036016
191 036016 022122
192 036020 001003
193 036022 005300
194 036024 001374
195 036026 000407
196 036030
197 036030 005037 177572
198 036034
    036034 104455
    036036 000017
    036040 020377
    036042 015126
199 036044 000530

    60$: JMP 80$ ;EXIT
        TST RFLAG ;IS THIS THE 1ST RECEIVE DONE
        BEQ 61$ ;IF YES, PROCEED
        ERRDF 13,EMG13,ERRG2 ;MULTIPLE RECEIVES
        TRAP C$ERDF
        .WORD 13
        .WORD EMG13
        .WORD ERRG2

    61$: JMP 80$ ;EXIT
        MOV #-1,RFLAG ;FLAG THAT THE RECEIVE IS DONE.
        CMP RSEL4,@SEL4 ;IS THE BUFFER ADDRESS CORRECT?
        BEQ 62$ ;IF OK, PROCEED WITH CHECK.
        ERRDF 11,EMG11,ERRG2 ;BUFFER ADDRESS ERROR
        TRAP C$ERDF
        .WORD 11
        .WORD EMG11
        .WORD ERRG2

    62$: BR 80$ ;EXIT
        CMP RSEL6,@SEL6 ;IS THE CHAR. COUNT CORRECT?
        BEQ 63$ ;IF OK, PROCEED
        ERRDF 12,EMG12,ERRG2 ;CHARACTER COUNT ERROR
        TRAP C$ERDF
        .WORD 12
        .WORD EMG12
        .WORD ERRG2

    63$: TST SFLAG ;WHICH EXTENDED ADDRESS ARE WE CHECKING?
        BNE 65$ ;BR FOR BIT 17 CHECK
        MOV #4000,R0 ;IN THE TEST OF BIT 16 WE SENT 4K BYTES
        MOV #120000,R1 ;THE XMIT BUFFER BEGAN AT THIS VIRTUAL ADDR
        ;WHICH WILL MAP TO 200000
        MOV #130000,R2 ;THE REC BUF. MAPS TO 210000
        BR 66$ ;GO COMPARE THE XMIT AND REC BUFFERS

    65$: MOV #10000,R0 ;IN THE TEST OF BIT 17 WE SENT 8K BYTES
        MOV #120000,R1 ;THE XMIT BUFFER MAPS TO 200000
        MOV #140000,R2 ;THE REC BUF. MAPS TO 400000

    66$: MOV #1,@#177572 ;TURN MEMORY MANAGEMENT BACK ON.

    67$: CMP (R1)+,(R2)+ ;IS THE DATA THE SAME?
        BNE 68$ ;IF NOT THERE IS A DATA ERROR.
        DEC R0 ;CHECK ENTIRE BUFFER
        BNE 67$
        BR 70$ ;IF DATA OK, PROCEED WITH TEST.

    68$: CLR @#177572 ;TURN MEMORY MANAGEMENT OFF.
        ERRDF 15,EMG15,ERRG2 ;DATA ERROR
        TRAP C$ERDF
        .WORD 15
        .WORD EMG15
        .WORD ERRG2

    80$: BR 80$ ;EXIT
    
```

```

200 036046          70$:
201 036046 005037 177572          CLR      @#177572          ;TURN MEMORY MANAGEMENT OFF.
202 036052 042777 000213 144154  BIC      #RDO+RCV+CMD,@SEL2 ;CLEAR RDO, RCV & COMMAND BITS (0,1)
203 036060 005737 002564          TST      RFLAG          ;IS THE RECEIVE DONE ? (IF DONE, FLAG = -1)
204 036064 001002          BNE      71$          ;YES - SEE IF TRANSMIT DONE
205 036066 000137 035510          JMP      40$          ;NO - GO BACK AND DO IT.
206 036072          71$:
207 036072 005737 002514          TST      TFLAG          ;IS THE TRANSMIT DONE ?
208 036076 001002          BNE      72$          ;YES - SEE IF THERE IS MORE
209 036100 000137 035510          JMP      40$          ;NO - DO IT
210 036104          72$:
211 036104 005737 002344          TST      SFLAG          ;HAVE WE ALREADY TESTED BIT 17
212 036110 001106          BNE      80$          ;IF SO - END OF TEST
213
214 036112 012737 177777 002344  MOV      #-1,SFLAG          ;FLAG SO WE DON'T COME THIS WAY AGAIN.
215 036120 023727 002120 004200  CMP      L$HIMEM,#4200      ;IS THERE ENOUGH MEMORY TO TEST BIT 17?
216 036126 002477          BLT      80$          ;IF NOT - END OF TEST.
217 036130 005037 002514          CLR      TFLAG          ;CLEAR FLAGS FOR NEXT TEST
218 036134 005037 002564          CLR      RFLAG
219
220          ;
221          ;SET UP TO TEST BIT 17, IF THERE IS ENOUGH MEMORY.
222          ;THIS TEST WILL TRANSMIT 8K BYTES STARTING AT PHYSICAL ADDRESS 200000
223          ;TO PHYSICAL ADDRESS 400000. THE TRANSMITTED BUFFER STILL CONTAINS
224          ;THE TEST PATTERN GENERATED IN THE BIT 16 TEST.
225 036140 005037 035462          CLR      RSEL4          ;RECEIVE BUFFER ADDRESS (BITS 0-15)
226 036144 012737 120000 035464  MOV      #BIT15!20000,RSEL6 ;REC BUFFER ADDR BIT 17 SET AND 8K
227          ;BYTE RECEIVE CHARACTER COUNT
228 036152 005037 035500          CLR      TSEL4          ;TRANSMIT BUFFER ADDRESS (BITS 0-15)
229 036156 012737 060000 035502  MOV      #BIT14!20000,TSEL6 ;XMIT BUFFER ADDR BIT 16 SET AND 8K
230          ;BYTE XMIT CHARACTER COUNT
231 036164 012701 010000          MOV      #10000,R1          ;COUNTER TO CLEAR 8K BYTES
232 036170 012704 140000          MOV      #140000,R4         ;VIRTUAL ADDRESS THAT WILL MAP INTO PAR 6
233          ;WITH THE PHYSICAL ADDRESS 400000
234 036174 005037 002350          CLR      NXMFLG          ;ENSURE FLAG IS CLEAR
235 036200          SETVEC  #4,#NOXMEM,#PRI07 ;SET UP TRAP TO VECTOR 4 (WILL SET FLAG)
236 036200          MOV      #PRI07,-(SP)
237 036204          MOV      #NOXMEM,-(SP)
238 036210          MOV      #4,-(SP)
239 036214          MOV      #3,-(SP)
240 036220          TRAP   C$SVEC
241 036222          ADD      #10,SP
242 036226 012737 000001 177572  MOV      #1,@#177572      ;TURN ON MEMORY MANAGEMENT
243 036234          74$:
244 036234 005024          CLR      (R4)+          ;CLEAR 400000 - 417776
245 036236 005737 002350          TST      NXMFLG          ;DOES A NXM TRAP 4 OCCUR?
246 036242 001002          BNE      75$          ;IF YES, EXIT
247 036244 005300          DEC      R0
248 036246 001372          BNE      74$
249 036250          75$:
250 036250 005037 177572          CLR      @#177572      ;TURN OFF MEMORY MANAGEMENT
251 036254          CLRVEC #4              ;RESTORE TRAP 4
252 036254          MOV      #4,R0
253 036260          TRAP   C$CVEC
254 036262 005737 002350          TST      NXMFLG          ;WAS THIS AN ERROR EXIT?
255 036266 001002          BNE      76$          ;IF YES - REPORT ERROR
256 036270 000137 035454          JMP      35$          ;START THE SECOND TEST
    
```

```

249 036274
250 036274 76$: ERRDF 19,EMT22
    036274 104455
    036276 000023 TRAP C$ERDF
    036300 036340 .WORD 19
    036302 000000 .WORD EMT22
    251 036304 .WORD 0
    036304 010446 PRINTB #FMT25,R4
    036306 012746 036376 MOV R4,-(SP)
    036312 012746 000002 MOV #FMT25,-(SP)
    036316 010600 MOV #2,-(SP)
    036320 104414 MOV SP,R0
    036322 062706 000006 TRAP C$PNTB
252 036326 ADD #6,SP
253 036326 80$: SHUTDN ;SHUTDOWN DMR
    036326 004737 012564 JSR PC,$HALT ;**** MACRO EXPANSION ****
    ;DMR HALT ROUTINE.
    254 036332 CALL $ERROR ;****
    ;CHECK BASE TABLE AND REPORT ANY SOFT ERRORS
255
256 036336 85$:
257 .DSABL LSB ;DISABLE LOCAL SYMBOL BLOCK
258 036336 ENDTST
    036336 104401 L10074:
259 TRAP C$ETST
260
261 036340 103 101 116 EMT22: .ASCIZ /CAN'T ADDRESS EXTENDED MEMORY/
    036343 047 124 040
    036346 101 104 104
    036351 122 105 123
    036354 123 040 105
    036357 130 124 105
    036362 116 104 105
    036365 104 040 115
    036370 105 115 117
    036373 122 131 000
262 036376 045 101 115 FMT25: .ASCIZ /%MEMORY ADDRESS %06% DOES NOT RESPOND - TRAP 4%N/
    036401 105 115 117
    036404 122 131 040
    036407 101 104 104
    036412 122 105 123
    036415 123 040 045
    036420 117 066 045
    036423 040 104 117
    036426 105 123 040
    036431 116 117 124
    036434 040 122 105
    036437 123 120 117
    036442 116 104 040
    036445 055 040 124
    036450 122 101 120
    036453 040 064 045
    036456 116 000
263 .EVEN
    
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

```
.SBTTL          TEST 15 - DMC MODE (RESUME) INTERRUPT TEST

:*****
:*              TEST 15 - DMR-11
:* RESUME BASE IN - DMC MODE
:* ** WILL NOT RUN IF MODEM LOOPBACK IS SELECTED **
:* IN THIS TEST THE DMR WILL TRANSMIT AND RECEIVE 7 BUFFERS. DURING THE
:* TEST THE DMR WILL BE HALTED AND RESTARTED BY A BASE-IN RESUME IN THE
:* FOLLOWING MANNER:
:*   BASE IN
:*   CONTROL IN
:*   HALT - BASE IN RESUME
:*   2 BA/CC IN RECEIVE
:*   HALT - BASE IN RESUME
:*   2 BA/CC IN RECEIVE
:*   HALT - BASE IN RESUME
:*   2 BA/CC IN RECEIVE
:*   HALT - BASE IN RESUME
:*   1 BA/CC IN RECEIVE
:*   HALT - BASE IN RESUME
:*   2 BA/CC IN TRANSMIT
:*   HALT - BASE IN RESUME
:*   2 BA/CC IN TRANSMIT
:*   HALT - BASE IN RESUME
:*   2 BA/CC IN TRANSMIT
:*   HALT - BASE IN RESUME
:*   1 BA/CC IN TRANSMIT
:*   HALT - BASE IN RESUME
:*
:* ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
:* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
:* THE RECEIVE/TRANSMIT TABLE.
:*
:* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
:* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
:* SEVEN RECEIVE AND SEVEN TRANSMIT BUFFERS. THE ROUTINE WILL
:* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
:* HIERARCHY:
:*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
:*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
:*      THAN 2K BYTES, USE THAT MEMORY
:*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
:*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
:******
BGNTST
T15::
TST      WMAINT      ;DO WE NEED TO WRITE MODEM
          ;MAINTENACE 1 OR 2?
BNE      40$        ;IF YES WE CAN'T RUN THIS TEST
          ;(NOTE: CAN'T WRITE MODEM IN DMC MODE)
MOV      #7,BUFNUM  ;# OF RCV & XMIT BUFFERS.
MOV      #1,RESUME  ;FLAG SET TO REQUEST USE OF RESUME.
MOV      #1,DMCMDE  ;FLAG SET TO REQUEST DMC MODE.
CLR      MNTMDE     ;FLAG NOT TO REQUEST MAINTENANCE MODE.

CALL     $BUFFS     ;DETERMINE 7 RCV & 7 XMIT BUFFERS
```

```
036460
036460
036460 005737 002306
036464 001036
036466 012737 000007 002324
036474 012737 000001 002274
036502 012737 000001 002276
036510 005037 002300
036514
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```
.SBTTL          TEST 16 - DMR MODE (RESUME) INTERRUPT TEST

*****
*          TEST 16 - DMR-11
* RESUME BASE IN - DMR MODE
* IN THIS TEST THE DMR WILL TRANSMIT AND RECEIVE 7 BUFFERS. DURING THE
* TEST THE DMR WILL BE HALTED AND RESTARTED BY A BASE-IN RESUME IN THE
* FOLLOWING MANNER:
*   BASE IN
*   CONTROL IN
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   1 BA/CC IN RECEIVE
*   HALT - BASE IN RESUME
*   2 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*   2 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*   2 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*   1 BA/CC IN TRANSMIT
*   HALT - BASE IN RESUME
*
* ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
* THE RECEIVE/TRANSMIT TABLE.
*
* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
* SEVEN RECEIVE AND SEVEN TRANSMIT BUFFERS. THE ROUTINE WILL
* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
* HIERARCHY:
*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
*      THAN 2K BYTES, USE THAT MEMORY
*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
*****
```

```
BGNTST
T16::
MOV #7,BUFNUM ;# OF RCV & XMIT BUFFERS.
MOV #1,RESUME ;FLAG SET TO REQUEST USE OF RESUME.
CLR DMCMD     ;FLAG CLEARED - DMR MODE.
CLR MNTMDE    ;FLAG NOT TO REQUEST MAINTENANCE MODE.

CALL $BUFFS   ;DETERMINE 7 RCV & 7 XMIT BUFFERS

CLEAR        ;MASTER CLEAR
***** MACRO EXPANSION ****
JSR PC, $MSCLR ;ISSUE A DMR MASTER CLEAR
```

```
036564
036564
47 036564 012737 000007 002324
48 036572 012737 000001 002274
49 036600 005037 002276
50 036604 005037 002300
51
52 036610
53
54 036614
036614 004737 011070
```

```
55                                     ;****          ****
56 036620          ESCAPE TST          ;IF ERROR, EXIT TEST
   036620 104410
   036622 000006                                     TRAP C$ESCAPE
57                                     .WORD L10076-.
58 036624          CALL $INOUT          ;THIS ROUTINE WILL MANAGE ALL THE DMR
59                                     ;COMMANDS ISSUED IN THE INTERRUPT ROUTINES
60                                     ;(FROM BASE IN UNTIL SHUT DOWN). BESIDES
61                                     ;CONTROLLING THE SOFTWARE TIMEOUT, THIS
62                                     ;ROUTINE WILL ALSO CHECK THAT BUFFER
63                                     ;CHARACTER COUNTS AND ADDRESSES ARE CORRECT
64                                     ;AND THAT THE DATA IS CORRECT IN THOSE BUFFERS
65
66
67
68 036630          FNDTST
   036630
   036630 104401                                     L10076:
69                                     TRAP C$ETST
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24 036632
036632
25 036632 012737 000100 002324
26
27 036640 005037 002274
28 036644 005037 002276
29 036650 005037 002300
30
31 036654
32
33 036660
036660 004737 011070
34
35 036664
036664 104410
036666 000012
36
37 036670
38
39
40
41
42
43
44
45 036674
46
47
48 036700
036700
036700 104401
49

```
.SBTTL          TEST 17 - DMR MODE INTERRUPT EXERCISE

*****
*          TEST 17 - DMR-11
* INTERRUPT DRIVEN EXERCISE
* IN THIS TEST 64 BUFFERS WILL BE TRANSMITTED AND RECEIVED
*
* ALL BA/CC OUTS RECEIVES AND TRANSMITS WILL BE ACCOUNTED FOR AND
* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
* THE RECEIVE/TRANSMIT TABLE.
*
* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
* 64 RECEIVE AND 64 TRANSMIT BUFFERS. THE ROUTINE WILL
* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
* HIERARCHY:
*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
*      THAN 2K BYTES, USE THAT MEMORY
*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
*****
BGNTST
                                T1/::
MOV      #64.,BUFNUM           ;# OF RCV & XMIT BUFFERS.
CLR      RESUME                 ;FLAG CLEARED IN ORDER NOT TO USE RESUME.
CLR      DMCMD                  ;FLAG CLEARED TO ALLOW DMR MODE.
CLR      MNTMDE                 ;FLAG NOT TO REQUEST MAINTENANCE MODE.
CALL     $BUFFS                 ;DETERMINE 64 RCV & 64 XMIT BUFFERS
CLEAR
                                ;MASTER CLEAR
                                ;**** MACRO EXPANSION ****
JSR      PC, $MSCLR             ;ISSUE A DMR MASTER CLEAR
                                ;****
                                ;
ESCAPE   TST                    ;IF ERROR, EXIT TEST
                                TRAP   C$ESCAPE
                                .WORD  L10077-.
CALL     $INOUT                 ;THIS ROUTINE WILL MANAGE ALL THE DMR
                                ;COMMANDS ISSUED IN THE INTERRUPT ROUTINES
                                ;(FROM BASE IN UNTIL SHUT DOWN). BESIDES
                                ;CONTROLLING THE SOFTWARE TIMEOUT, THIS
                                ;ROUTINE WILL ALSO CHECK THAT BUFFER
                                ;CHARACTER COUNTS AND ADDRESSES ARE CORRECT
                                ;AND THAT THE DATA IS CORRECT IN THOSE BUFFERS
CALL     $ERROR                 ;CHECK BASE TABLE FOR SOFT ERRORS

ENDTST
                                L10077:
                                TRAP   C$ETST
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51

```
.SBTTL          TEST 18 - DMR MODE LARGE MESSAGE

:*****
:*              TEST 18 - DMR-11
:*  LARGE MESSAGE
:*  IN THIS MODE TRANSMIT AND RECEIVE 1 LARGE BUFFER
:*
:*  THE BA/CC OUT RECEIVE AND TRANSMIT WILL BE ACCOUNTED FOR AND
:*  THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
:*  THE RECEIVE/TRANSMIT TABLE.
:*
:*  THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS.  THIS
:*  SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
:*  ONE RECEIVE AND ONE TRANSMIT BUFFER.  THE ROUTINE WILL
:*  ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
:*  HIERARCHY:
:*      A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
:*      B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
:*         THAN 2K BYTES, USE THAT MEMORY
:*      C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
:*         THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
:*
:*****
BGNTST
T18::

MOV    #1,BUFNUM      ;# OF RCV & XMIT BUFFERS.
CLR    RESUME         ;FLAG CLEARED IN ORDER NOT TO USE RESUME.
CLR    DMCMD          ;FLAG CLEARED TO ALLOW DMR MODE.
CLR    MNTMDE         ;FLAG NOT TO REQUEST MAINTENANCE MODE.
CALL   $BUFFS        ;DETERMINE 1 RCV & 1 XMIT BUFFER
CLEAR  ;MASTER CLEAR
;**** MACRO EXPANSION ****
JSR    PC, $MSCLR    ;ISSUE A DMR MASTER CLEAR
;****

ESCAPE TST           ;IF ERROR, EXIT TEST
TRAP   C$ESCAPE     .WORD L10100-.

CALL   $INOUT        ;THIS ROUTINE WILL MANAGE ALL THE DMR
;COMMANDS ISSUED IN THE INTERRUPT ROUTINES
;(FROM BASE IN UNTIL SHUT DOWN).  BESIDES
;CONTROLLING THE SOFTWARE TIMEOUT, THIS
;ROUTINE WILL ALSO CHECK THAT BUFFER
;CHARACTER COUNTS AND ADDRESSES ARE CORRECT
;AND THAT THE DATA IS CORRECT IN THOSE BUFFERS

CALL   $ERROR        ;CHECK BASE TABLE FOR SOFT ERRORS
;NOTE:  NORMALLY ANY NON-ZERO ERROR COUNT IS
;REPORTED;  HOWEVER IN THIS TEST A REP COUNT
;OF 1 IS ALLOWED, BECAUSE AT LOW BAUD RATES
;WE WOULD EXPECT 1 REP.
```

036702
036702
036702 012737 000001 002324
036710 005037 002274
036714 005037 002276
036720 005037 002300
036724
036730
036730 004737 011070
036734
036734 104410
036736 000012
036740
036744

52
53 036750
036750
54 036750 104401

ENDTST

L10100:
TRAP C\$ETST

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

```

.SBTTL          TEST 19 - DMR MAINTENANCE MODE MESSAGE
:*****
:          TEST 19 - DMR-11
:* MAINTENANCE MODE OPERATION
:*
:* THE BA/CC OUT RECEIVE AND TRANSMIT WILL BE ACCOUNTED FOR AND
:* THE CHARACTER COUNTS AND BUFFER ADDRESSES WILL BE CHECKED AGAINST
:* THE RECEIVE/TRANSMIT TABLE.
:*
:* THE BUFFERS ARE DETERMINED IN THE SUBROUTINE $BUFFS. THIS
:* SUBROUTINE WILL DETERMINE THE ADDRESS AND CHARACTER COUNT OF
:* ONE RECEIVE AND ONE TRANSMIT BUFFER. THE ROUTINE WILL
:* ATTEMPT TO USE AS LARGE BUFFERS AS POSSIBLE IN THE FOLLOWING
:* HIERARCHY:
:*   A. IF THERE IS MEMORY MANAGEMENT, USE A PAGE ABOVE 32K.
:*   B. IF THERE IS FREE MEMORY ABOVE THE SUPERVISOR GREATER
:*      THAN 2K BYTES, USE THAT MEMORY
:*   C. IF NEITHER OF THE PRECEEDING TWO ARE POSSIBLE, USE
:*      THE 2K BYTE DEFAULT BUFFER WITHIN THIS DIAGNOSTIC.
:*****
BGNTST
          T19::
24 036752 012737 000001 002324      MOV    #1, BUFNUM      ;# OF RCV & XMIT BUFFERS.
25
26 036760 005037 002274              CLR    RESUME         ;DON'T ALLOW RESUME
27 036764 005037 002276              CLR    DMCMD         ;FLAG CLEARED TO ALLOW DMR MODE.
28 036770 012737 000001 002300      MOV    #1, MNTMDE     ;FLAG SET TO REQUEST MAINTENANCE MODE.
29
30 036776              CALL   $BUFFS         ;DETERMINE 1 RCV & 1 XMIT BUFFER
31
32 037002              CLEAR                ;MASTER CLEAR
          037002 004737 011070        JSR    PC, $MSCLR     ;**** MACRO EXPANSION ****
          ;ISSUE A DMR MASTER CLEAR
          ;****
33
34 037006              ESCAPE TST          ;IF ERROR, EXIT TEST
          037006 104410              TRAP   C$ESCAPE
          037010 000012              .WORD  L10101-.
35
36 037012              CALL   $INOUT        ;THIS ROUTINE WILL MANAGE ALL THE DMR
37 ;COMMANDS ISSUED IN THE INTERRUPT ROUTINES
38 ;(FROM BASE IN UNTIL SHUT DOWN). BESIDES
39 ;CONTROLLING THE SOFTWARE TIMEOUT, THIS
40 ;ROUTINE WILL ALSO CHECK THAT BUFFER
41 ;CHARACTER COUNTS AND ADDRESSES ARE CORRECT
42 ;AND THAT THE DATA IS CORRECT IN THOSE BUFFERS
43
44 037016              CALL   $ERROR        ;CHECK BASE TABLE FOR SOFT ERRORS
45
46
47 037022              ENDTST
          037022 104401              L10101:
          037022              TRAP   C$ETST
48
49

```

50

.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.
:*****
    
```

```

1
2
3
4
5
6
7
8
9
10
11
12
13 037024          BGNHRD
    037024 000015
    037026
                                .WORD L10102-L$HARD/2
                                L$HARD::
14
15 037026          GPRMA  P1,2,0,160000,177776,YES
    037026 001031
    037030 037060
    037032 160000
    037034 177776
                                .WORD  T$CODE
                                .WORD  P1
                                .WORD  T$LLOLIM
                                .WORD  T$HILIM
16 037036          GPRMA  P2,4,0,0,776,YES
    037036 002031
    037040 037076
    037042 000000
    037044 000776
                                .WORD  T$CODE
                                .WORD  P2
                                .WORD  T$LLOLIM
                                .WORD  T$HILIM
17 037046          GPRMD  P3,20,0,7,0,7,YES
    037046 010032
    037050 037117
    037052 000007
    037054 000000
    037056 000007
                                .WORD  T$CODE
                                .WORD  P3
                                .WORD  7
                                .WORD  T$LLOLIM
                                .WORD  T$HILIM
18
19 037060          ENDHRD
                                .EVEN
                                L10102:
20 037060
21 037060          103    123    122  P1:  .ASCIZ  /CSR ADDRESS: /
    037063          040    101    104
    037066          104    122    105
    037071          123    123    072
    037074          040    000
22 037076          126    105    103  P2:  .ASCIZ  /VECTOR ADDRESS: /
    037101          124    117    122
    037104          040    101    104
    037107          104    122    105
    037112          123    123    072
    037115          040    000
23 037117          124    105    123  P3:  .ASCII  /TEST CONFIGURATION -/<CR><LF>
    037122          124    040    103
    037125          117    116    106
    037130          111    107    125
    037133          122    101    124
    037136          111    117    116
    037141          040    055    015
    037144          012
24 037145          040    040    060
                                .ASCII  / 0 = INTERNAL (NO CONNECTOR)/<CR><LF>
    
```

	037150	040	075	040	
	037153	111	116	124	
	037156	105	122	116	
	037161	101	114	040	
	037164	050	116	117	
	037167	040	103	117	
	037172	116	116	105	
	037175	103	124	117	
	037200	122	051	015	
	037203	012			
25	037204	040	040	061	.ASCII / 1 = H3254 - V.35 (NOTE: MODE 1-4 ALLOWS/<CR><LF>
	037207	040	075	040	
	037212	110	063	062	
	037215	065	064	040	
	037220	055	040	126	
	037223	056	063	065	
	037226	040	040	040	
	037231	040	040	040	
	037234	050	116	117	
	037237	124	105	072	
	037242	040	040	115	
	037245	117	104	105	
	037250	040	061	055	
	037253	064	040	101	
	037256	114	114	117	
	037261	127	123	015	
	037264	012			
26	037265	040	040	062	.ASCII / 2 = H3254 - INTEGRAL PROGRAM INTERFACE SELECTION)/
	037270	040	075	040	
	037273	110	063	062	
	037276	065	064	040	
	037301	055	040	111	
	037304	116	124	105	
	037307	107	122	101	
	037312	114	040	040	
	037315	040	120	122	
	037320	117	107	122	
	037323	101	115	040	
	037326	111	116	124	
	037331	105	122	106	
	037334	101	103	105	
	037337	040	123	105	
	037342	114	105	103	
	037345	124	111	117	
	037350	116	051		
27	037352	015	012	040	.ASCII <CR><LF>/ 3 = H3255 - RS232C/<57>/423/<CR><LF>
	037355	040	063	040	
	037360	075	040	110	
	037363	063	062	065	
	037366	065	040	055	
	037371	040	122	123	
	037374	062	063	062	
	037377	103	057	064	
	037402	062	063	015	
	037405	012			
28					
29	037406	040	040	064	.ASCII / 4 = H3255 - RS422/<CR><LF>

	037411	040	075	040	
	037414	110	063	062	
	037417	065	065	040	
	037422	055	040	122	
	037425	123	064	062	
	037430	062	015	012	
30	037433	040	040	065	.ASCII / 5 = CABLE AND SW PACK INTERFACE SELECTED/<CR><LF>
	037436	040	075	040	
	037441	103	101	102	
	037444	114	105	040	
	037447	101	116	104	
	037452	040	123	127	
	037455	040	120	101	
	037460	103	113	040	
	037463	111	116	124	
	037466	105	122	106	
	037471	101	103	105	
	037474	040	123	105	
	037477	114	105	103	
	037502	124	105	104	
	037505	015	012		
31	037507	040	040	040	.ASCII / (V.35-H3250, INTEGRAL-BC55A-10, /
	037512	040	040	040	
	037515	050	126	056	
	037520	063	065	055	
	037523	110	063	062	
	037526	065	060	054	
	037531	040	111	116	
	037534	124	105	107	
	037537	122	101	114	
	037542	055	102	103	
	037545	065	065	101	
	037550	055	061	060	
	037553	054			
32	037554	040	122	123	.ASCII / RS232C-H325, RS423/<57>/422-H3251)/<CR><LF>
	037557	062	063	062	
	037562	103	055	110	
	037565	063	062	065	
	037570	054	040	122	
	037573	123	064	062	
	037576	063	057	064	
	037601	062	062	055	
	037604	110	063	062	
	037607	065	061	051	
	037612	015	012		
33	037614	052	040	123	.ASCII /* SELECT THE FOLLOWING ONLY IF THE MODEM SUPPORTS LOOPBACK */
	037617	105	114	105	
	037622	103	124	040	
	037625	124	110	105	
	037630	040	106	117	
	037633	114	114	117	
	037636	127	111	116	
	037641	107	040	117	
	037644	116	114	131	
	037647	040	111	106	
	037652	040	124	110	
	037655	105	040	115	

	037660	117	104	105	
	037663	115	040	123	
	037666	125	120	120	
	037671	117	122	124	
	037674	123	040	114	
	037677	117	117	120	
	037702	102	101	103	
	037705	113	040	052	
34	037710	015	012	040	.ASCII <CR><LF>/ 6 = LOCAL LOOP/<CR><LF>
	037713	040	066	040	
	037716	075	040	114	
	037721	117	103	101	
	037724	114	040	114	
	037727	117	117	120	
	037732	015	012		
35	037734	040	040	067	.ASCIIZ / 7 = REMOTE LOOP/<CR><LF>
	037737	040	075	040	
	037742	122	105	115	
	037745	117	124	105	
	037750	040	114	117	
	037753	117	120	015	
	037756	012	000		
36					.EVEN
37					

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

.SBTTL SOFTWARE PARAMETER CODING SECTION

```

:*****
: THE SOFTWARE 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.
:*****
    
```

037760
037760 000005
037762

037762
037762 000032
037764 037774
037766 000007
037770 000001
037772 000005

BGNSFT

.WORD L10103-L\$SOFT/2
L\$SOFT::

GPRMD S1,0,0,7,1,5,YES

.WORD T\$CODE
.WORD S1
.WORD 7
.WORD T\$LOLIM
.WORD T\$HILIM

ENDSFT

.EVEN
L10103:

037774
037774 123 105 114
037777 105 103 124
040002 101 102 114
040005 105 040 120
040010 122 117 107
040013 122 101 115
040016 040 114 117
040021 117 120 040
040024 124 111 115
040027 105 055 117
040032 125 124 040
040035 126 101 122
040040 111 101 102
040043 114 105 015
040046 012
040047 133 122 105
040052 106 105 122
040055 040 124 117
040060 040 114 111
040063 123 124 111
040066 116 107 040
040071 066 056 063
040074 056 061 063
040077 135 040 040
040102 040 050 115
040105 101 130 075
040110 065 073 040
040113 115 111 116
040116 075 061 051
040121 040 000

S1: .ASCII /SELECTABLE PROGRAM LOOP TIME-OUT VARIABLE/<CR><LF>

.ASCIZ /[REFER TO LISTING 6.3.13] (MAX=5; MIN=1) /

21
 22
 23
 24 040124
 25 040224 040224
 26 040224 000240
 27 040226 000240
 28 040230 000240
 29
 30 040232
 31
 32 040232
 040232 000000
 040234 000000
 040236
 33 000001

.EVEN
 :*****
 PATCH: PATCH AREA *****
 . = +100
 NOP
 NOP
 NOP
 :*****
 ENDMOD
 LASTAD
 L\$LAST::
 .END
 .EVEN
 .WORD 0
 .WORD 0

A = 000044	C\$BRK = 000022	DMRFLG 002260	ERRT3 027410 G	FRSTIM 002264
ADR = 000020 G	C\$BSEG= 000004	DMRRUN= 000040	ERRT4 032112 G	F\$AU - 000015
ASSEMB= 000010	C\$BSUB= 000002	DMRVEC 002226	EVL = 000004 G	F\$AUU= 000020
AX3 002304	C\$CEFG= 000045	DMTURN 002254	EXERR = 000006	F\$BGN = 000040
BACCR = 000004	C\$CLCK= 000062	DMTVEC 002230	E\$END = 002100	F\$CLEA= 000007
BACCT = 000000	C\$CLEA= 000012	DTR = 000100	E\$LOAD= 000035	F\$DU = 000016
BASE 002634	C\$CLOS= 000035	DXERR = 000007	FINIT1 021612	F\$END = 000041
BASE1 = 000003	C\$CLP1= 000006	EF.CON= 000036 G	FINIT2 021705	F\$HARD= 000004
BASEUP= 020000	C\$CVEC= 000036	EF.NEW= 000035 G	FINIT3 022000	F\$HW = 000013
BIGBUF 004234	C\$DCLN= 000044	EF.PWR= 000034 G	FLAG 002342	F\$INIT= 000006
BIT0 = 000001 G	C\$DODU= 000051	EF.RES= 000037 G	FMDROP 023754	F\$JMP = 000050
BIT00 = 000001 G	C\$DRPT= 000024	EF.STA= 000040 G	FMG1 016334	F\$MOD = 000000
BIT01 = 000002 G	C\$DU = 000053	EMG1 020052	FMG10 017010	F\$MSG = 000011
BIT02 = 000004 G	C\$EDIT= 000003	EMG10 020223	FMG11 017046	F\$PROT= 000021
BIT03 = 000010 G	C\$ERDF= 000055	EMG11 020252	FMG12 017100	F\$PWR = 000017
BIT04 = 000020 G	C\$ERHR= 000056	EMG12 020277	FMG13 017131	F\$RPT = 000012
BIT05 = 000040 G	C\$ERRO= 000060	EMG13 020325	FMG14 017205	F\$SEG = 000003
BIT06 = 000100 G	C\$ERSF= 000054	EMG14 020353	FMG15 017261	F\$SOFT= 000005
BIT07 = 000200 G	C\$ERSO= 000057	EMG15 020377	FMG16 017310	F\$SRV = 000010
BIT08 = 000400 G	C\$ESCA= 000010	EMG16 020412	FMG17 017367	F\$SUB = 000602
BIT09 = 001000 G	C\$ESEG= 000005	EMG17 020443	FMG18 017442	F\$SW = 000014
BIT1 = 000002 G	C\$ESUB= 000003	EMG18 020515	FMG19 017531	F\$TEST= 000001
BIT10 = 002000 G	C\$ETST= 000001	EMG2 020063	FMG2 016366	GETPRM 020730
BIT11 = 004000 G	C\$EXIT= 000032	EMG8 020130	FMG20 017554	G\$CNT0= 000200
BIT12 = 010000 G	C\$GETB= 000026	EMG9 020174	FMG21 017603	G\$DELM= 000372
BIT13 = 020000 G	C\$GETW= 000027	EMS3 012536	FMG22 017663	G\$DISP= 000003
BIT14 = 040000 G	C\$GMAN= 000043	EMS4 012726	FMG23 017710	G\$EXCP= 000400
BIT15 = 100000 G	C\$GPHR= 000042	EMT0 024204	FMG24 017771	G\$HILI= 000002
BIT2 = 000004 G	C\$GPLO= 000030	EMT1 025564	FMG3 016420	G\$LOLI= 000001
BIT3 = 000010 G	C\$GPRI= 000040	EMT10 030726	FMG4 016472	G\$NO = 000000
BIT4 = 000020 G	C\$INIT= 000011	EMT11 030754	FMG5 016523	G\$OFFS= 000400
BIT5 = 000040 G	C\$INLP= 000020	EMT12 031010	FMG6 016554	G\$OFSI= 000376
BIT6 = 000100 G	C\$MANI= 000050	EMT13 031350	FMG7 016615	G\$PRMA= 000001
BIT7 = 000200 G	C\$MEM = 000031	EMT2 025604	FMG8 016666	G\$PRMD= 000002
BIT8 = 000400 G	C\$MSG = 000023	EMT20 032150	FMG9 016737	G\$PRML= 000000
BIT9 = 001000 G	C\$OPEN= 000034	EMT22 036340	FMS1 010604	G\$RADA= 000140
BOE = 000400 G	C\$PNTB= 000014	EMT3 025740	FMS2 010645	G\$RADB= 000000
BSEL0 = 002232	C\$PNTF= 000017	EMT4 026466	FMS3 011746	G\$RADD= 000040
BSEL1 002242	C\$PNTS= 000016	EMT5 026513	FMT0 024242	G\$RADL= 000120
BSEL2 = 002234	C\$PNTX= 000015	EMT6 026561	FMT1 025276	G\$RADO= 000020
BSEL3 002244	C\$QIO = 000377	EMT7 027716	FMT11 030004	G\$XFER= 000004
BSEL4 = 002236	C\$RDBU= 000007	EMT8 027747	FMT12 030035	G\$YES = 000010
BSEL5 002246	C\$REFG= 000047	EMT9 030676	FMT13 030072	HALTC = 001000
BSEL6 = 002240	C\$RESE= 000033	END 021610	FMT14 030127	HDX = 002000
BSEL7 002250	C\$REVI= 000003	ERRFLG 002360	FMT15 030164	HELP = 000000
BUFNUM 002324	C\$RFLA= 000021	ERRG1 014620 G	FMT16 030221	HICRC 002402
BUFSIZ 002322	C\$RPT = 000025	ERRG10 016216 G	FMT19 031374	HIWORD 002406
CD = 002000	C\$SEFG= 000046	ERRG11 016250 G	FMT2 025345	HLT = 000002
CHIPNO 002412	C\$SPRI= 000041	ERRG12 016302 G	FMT25 036376	HOE = 100000 G
CLRNO 002376	C\$SVEC= 000037	ERRG2 015126 G	FMT3 025422	IBE = 010000 G
CMD = 000007	C\$TPRI= 000013	ERRG3 015242 G	FMT4 025506	IDU = 000040 G
CNTRL = 000001	DDMC = 000010	ERRG4 015522 G	FMT5 026625	IECLR = 000020
CR = 000015	DFPTBL 002174 G	ERRG7 016016 G	FMT6 026656	IEO = 000100
CSR 002232	DIAGMC= 000000	ERRG8 016114 G	FMT7 026707	IER = 020000 G
CTS = 010000	DISCON= 000100	ERROR 002364	FMT8 026740	IESET = 000100
C\$AU = 000052	DMCMDE 002276	ERRI1 026246 G	FMT9 027003	INFACE 002262
C\$AUTO= 000061	DMR = 000522	ERRT2 031316 G	FRSPAS 002266	INFLAG 002352

INISR 022174 G	L\$ETP 002102 G	L10033 026114	NOBFR = 000004	RUN = 100000
INRCV 002326	L\$EXP1 002046 G	L10034 026242	NOXMEM 023714 G	SAVE = 002340
INTER = 000015	L\$EXP4 002064 G	L10035 026464	NXM = 000400	SECN = 004000
INXMIT 002330	L\$EXP5 002066 G	L10036 027406	NXMFLG 002350	SELO = 002232
ISP13 = 000076	L\$HARD 037026 G	L10037 027226	OUTFLG 002354	SEL2 = 002234
ISP7 = 000072	L\$HIME 002120 G	L10040 027404	OUTISR 023256 G	SEL4 = 002236
ISR = 000100 G	L\$HPCP 002016 G	L10041 027714	OUTRCV 002332	SEL6 = 002240
IXE = 004000 G	L\$HPTP 002022 G	L10042 030674	OUTXMT 002334	SFLAG 002344
ISAU = 000041	L\$HW 002174 G	L10043 030412	OSAPTS= 000000	SFPTBL 002224 G
ISAUTO= 000041	L\$ICP 002104 G	L10044 030502	OSAU = 000000	SKIP 002346
ISCLN = 000041	L\$INIT 020552 G	L10045 030576	OSBGNR= 000000	SPEED 002224
ISDU = 000041	L\$SLADP 002026 G	L10046 030672	OSBGN5= 000001	STARES 002270
ISHRD = 000041	L\$SLAST 040236 G	L10047 031302	OSDU = 000001	STARST 020706
ISINIT= 000041	L\$LOAD 002100 G	L10050 031202	OSERRT= 000000	START 002272
ISMOD = 000041	L\$SLUN 002074 G	L10051 031300	OSGNSW= 000001	STLU = 010000
ISMSG = 000041	L\$SMREV 002050 G	L10052 031346	OSPOIN= 000001	STREC = 000200
ISPROT= 000040	L\$NAME 002000 G	L10053 032110	OSSETU= 000000	STUP = 000400
ISPTAB= 000041	L\$PRIO 002042 G	L10054 031702	PATCH 040124	SUBRPC 002372
ISPR = 000041	L\$PROT 020544 G	L10055 032106	PNT = 001000 G	SVCGBL= 000000
ISRPT = 000041	L\$PRT 002112 G	L10056 032146	PRETIM= 000055	SVCINS= 000001
ISSEG = 000041	L\$REPP 002062 G	L10057 032674	PRI = 002000 G	SVCSUB= 000001
ISSETU= 000041	L\$REV 002010 G	L10060 032326	PRI00 = 000000 G	SVCTAG= 000001
ISSFT = 000041	L\$SOFT 037762 G	L10061 032506	PRI01 = 000040 G	SVCTST= 000001
ISSRV = 000041	L\$SPC 002056 G	L10062 032672	PRI02 = 000100 G	SW00 = 000001
ISSUB = 000041	L\$SPCP 002020 G	L10063 033076	PRI03 = 000140 G	SW01 = 000002
ISTST = 000041	L\$SPTP 002024 G	L10064 033256	PRI04 = 000200 G	SW02 = 000004
JSJMP = 000167	L\$STA 002030 G	L10065 034162	PRI05 = 000240 G	SW03 = 000010
LAST 002362	L\$SW 002224 G	L10066 033406	PRI06 = 000300 G	SW04 = 000020
LF = 000012	L\$TEST 002114 G	L10067 033524	PRI07 = 000340 G	SW05 = 000040
LLOOP = 000006	L\$TIML 002014 G	L10070 033642	PSTACK 002370	SW06 = 000100
LOCATE 024126 G	L\$UNIT 002012 G	L10071 034020	P1 037060	SW07 = 000200
LOCRC 002400	L10000 002222	L10072 034160	P2 037076	SW08 = 000400
LOE = 040000 G	L10001 002226	L10073 034712	P3 037117	SW09 = 001000
LOGDEV 002366	L10002 015124	L10074 036336	R = 000042	SW10 = 002000
LOT = 000010 G	L10003 015240	L10075 036562	RBUF 002566	SW11 = 004000
LOWORD 002404	L10004 015520	L10076 036630	RCOUNT= 000044	SW12 = 010000
LPLU = 004000	L10005 016014	L10077 036700	RCV = 000004	SW13 = 020000
L\$ACP 002110 G	L10006 016112	L10100 036750	RCVBUF 003634	SW14 = 040000
L\$APT 002036 G	L10007 016214	L10101 037022	RDI = 000200	SW15 = 100000
L\$AUT 002070 G	L10010 016246	L10102 037060	RDO = 000200	S\$LSYM= 010000
L\$AUTO 022066 G	L10011 016300	L10103 037774	RES = 010000	S1 037774
L\$CCP 002106 G	L10012 016332	MAINT = 000400	RESFLG 002356	T = 000045
L\$CLEA 022156 G	L10014 021610	MAINT1= 000010	RESUME 002274	TBUF 002516
L\$CO 002032 G	L10015 022154	MAINT2= 000004	RETURN= 000207	TCOUNT= 000044
L\$DEPO 002011 G	L10016 022172	MANUF 002310	REV1 025632	TEMP 002336
L\$DESC 010242 G	L10017 023254	MCLR = 040000	REV2 025634	TFLAG 002514
L\$DESC 002076 G	L10020 023712	MDIAG = 020000	RFLAG 002564	THRESH= 000013
L\$DEVP 002060 G	L10021 023722	MICRO 002256	RLOOP = 000007	TH1L = 000060
L\$DISP 002124 G	L10022 023752	MMANAG 002302	RMODEM= 000017	TH2L = 000062
L\$DLY 002116 G	L10023 024124	MNTMDE 002300	ROMADR 002410	TH3L = 000064
L\$DTP 002040 G	L10024 024202	MNTREC= 000010	ROMI = 001000	TH4L = 000066
L\$DTYP 002034 G	L10025 025274	MODEM 031304	ROMLOC 025623	TIMER = 000012
L\$DU 023724 G	L10026 024476	N = 000043	ROMO = 002000	TOLONG= 000020
L\$DUT 002072 G	L10027 025120	NAKS = 000001	RQI = 000040	TOUT = 000002
L\$DVTY 010234 G	L10030 025272	NESTPC 002374	RRAM = 000014	TSEL4 035500
L\$EF 002052 G	L10031 025736	NEWST 020712	RSEL4 035462	TSEL6 035502
L\$ENVI 002044 G	L10032 026244	NEXT 022730	RSEL6 035464	T\$ARGC= 000002

TS CODE= 000032
 TSERRN= 000023
 TSEXCP= 000000
 TSFLAG= 000040
 TSGMAN= 000000
 TSHILI= 000005
 TSLAST= 000001
 TSLOLI= 000001
 TSLSYM= 010000
 TSLTNO= 000023
 TSNEST= 177777
 TSNSO = 000000
 TSNS1 = 000005
 TSNS2 = 000002
 TSPTNU= 000000
 TSSAVL= 177777
 TSSEGL= 177777
 TSSUBN= 000000
 TSTAGL= 177777
 TSTAGN= 010104
 TSTEMP= 000000
 TSTEST= 000023

TSTSTM= 177777
 TSTSTS= 000001
 TSSAUT= 010015
 TSSCLE= 010016
 TSSDU = 010022
 TSSHAR= 010102
 TSSHW = 010000
 TSSINI= 010014
 TSSMSG= 010056
 TSSPRO= 010013
 TSSSOF= 010103
 TSSSRV= 010024
 TSSSUB= 010072
 TSSSW = 010001
 TSSTES= 010101
 T1 024004 G
 T10 032676 G
 T11 033100 G
 T12 033260 G
 T12.1 033260
 T12.2 033410
 T12.3 033526

T12.4 033644
 T12.5 034022
 T13 034164 G
 T14 034714 G
 T15 036460 G
 T16 036564 G
 T17 036632 G
 T18 036702 G
 T19 036752 G
 T2 024322 G
 T2.1 024322
 T2.2 024500
 T2.3 025122
 T3 025636 G
 T4 025770 G
 T4.1 025770
 T4.2 026116
 T5 027046 G
 T5.1 027046
 T5.2 027230
 T6 030260 G
 T6.1 030260

T6.2 030414
 T6.3 030504
 T6.4 030600
 T7 031040 G
 T7.1 031040
 T7.2 031204
 T8 031462 G
 T8.1 031462
 T8.2 031704
 T9 032164 G
 T9.1 032164
 T9.2 032330
 T9.3 032510
 UAM = 000200 G
 UPDATE= 000011
 WAIT1 002312
 WAIT2 002314
 WAIT3 002316
 WAIT4 002320
 WMAINT 002306
 WMODEM= 000005
 WTYPE 002252

X = 000046
 XMTBUF 003234
 XSALWA= 000000
 XSFALS= 000040
 XSOFFS= 000400
 XSTRUE= 000020
 SBACC 012300
 SBASEI 011266
 SBUFFS 013210
 SCCITT 002414
 SCLRQI 010706
 SCNTIN 011522
 SDMRIN 012070
 \$ERROR 012412
 \$HALT 012564
 \$INOUT 014142
 \$LOOP 013066
 \$LSTIN= 000001
 \$LSTTA= 000001
 SMSCLR 011070
 \$ROMO 012746
 \$WAIT 010272

. ABS. 040236 000
 000000 001
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 23272 WORDS (91 PAGES)
 DYNAMIC MEMORY AVAILABLE FOR 69 PAGES
 CZDMIB.BIN,DB2:CZDMIB.SEQ/C/N:TOC=#SVC34R.MLB,CZDMIB.P11

\$BACC	34-29#	60-36	60-82	61-57	61-50	61-86	61-93	62-32	64-19	64-24	65-77	65-101	65-107	65-134
	67-45	67-49	68-124	68-130										
\$BASEI	31-36#	56-23	56-56	57-22	57-36	57-55	58-25	58-47	58-66	58-92	59-21	60-19	60-67	61-18
	61-27	61-47	61-80	62-12	64-13	65-20	65-25	65-98	65-128	67-33	68-100	70-64		
\$BUFFS	40-34#	70-56	71-52	72-31	73-32	74-30								
\$CCITT	20-150#	40-75	40-154	68-62										
\$CLRQI	29-29#	31-54	31-78	32-50	33-69	34-45	37-25	48-169	58-37	59-44	59-66			
\$CNTIN	32-34#	58-28	58-49	58-69	58-101	60-22	60-70	61-50	61-83	62-25	64-16	65-53	67-42	68-109
\$DMRIN	33-35#	39-41	41-111	57-25	57-38	57-58	57-67	59-29	59-54	60-31	60-79	62-19	62-22	62-30
\$ERROR	35-25#	67-132	68-254	72-45	73-46	74-44								
\$HALT	37-17#	41-115	56-26	56-59	57-28	57-41	57-70	58-38	58-58	58-72	59-67	60-59	60-96	61-21
	62-46	64-40	67-130	68-253	70-67									
\$INOUT	41-22#	70-74	71-58	72-37	73-38	74-36								
\$LOOP	39-22#	58-97	67-39	68-106										
\$LSTIN	14-22#													
\$LSTTA	14-23#													
\$MSCLR	30-36#	54-30	54-39	56-19	56-51	57-18	57-51	58-22	58-63	58-82	59-18	60-16	60-64	61-15
	61-44	61-77	62-9	64-10	65-18	65-44	65-72	65-95	65-125	67-23	68-89	70-58	71-54	72-33
	73-34	74-32												
\$ROMO	38-17#	53-42	53-81	53-85	53-167									
\$WAIT	28-37#	31-42	31-74	32-39	32-54	33-48	34-35	37-23	37-27	41-112	58-32	58-53	59-36	59-58
	60-38	60-56	60-84	61-29	61-61	61-95	62-35	64-26	65-27	65-55	65-79	65-109	65-136	67-54
	68-136													
A	19-121#	56-39	56-72	56-95	56-97									
ADR	19-8#													
ASSEMB	14-16	14-16												
AX3	20-59#	31-76	45-121*	45-125*	45-130*	45-135*	45-140*	45-146*	48-44					
BACCR	19-93#	48-23	48-233	61-57	61-86	64-19	65-77	65-101	65-134	67-45	68-125			
BACCT	19-89#	48-25	48-253	60-36	60-82	61-59	61-93	62-32	64-24	65-107	67-49	68-131		
BASE	20-184#	35-34	42-73	42-75	42-78	42-78	42-80	42-82	42-85	42-85	42-87	42-89	42-92	42-92
	42-94	42-96	42-99	42-99	48-78	56-23	56-27	56-35	56-37	56-39	56-41	56-43	56-56	56-60
	56-68	56-70	56-72	56-74	56-76	56-86	56-88	56-90	56-92	56-95	56-97	56-99	56-101	56-103
	56-105	57-22	57-30	57-36	57-43	57-55	57-72	57-74	57-76	57-78	57-80	57-90	57-91	57-93
	57-95	57-97	57-99	57-101	57-103	57-105	57-107	57-109	58-25	58-40	58-47	58-66	58-74	58-94
	59-21	60-19	60-67	60-97	60-99	60-110	60-110	61-18	61-47	61-80	62-12	64-13	65-20	65-25
	65-98	65-128	67-35	68-102	70-64									
BASEI	19-92#	31-40	41-41	48-27	48-185	48-191								
BASEUP	19-57#													
BIGBUF	20-196#	40-125												
BIT0	19-8#	19-82	30-49	40-122	40-193	48-220	48-240	53-189	57-30	57-43				
BIT00	19-8	19-8#												
BIT01	19-8	19-8#												
BIT02	19-8	19-8#												
BIT03	19-8	19-8#												
BIT04	19-8	19-8#												
BIT05	19-8	19-8#												
BIT06	19-8	19-8#												
BIT07	19-8	19-8#												
BIT08	19-8	19-8#												
BIT09	19-8	19-8#												
BIT1	19-8#	19-81	58-74											
BIT10	19-8#	19-39	19-64	19-65										
BIT11	19-8#	19-38	19-62											
BIT12	19-8#	19-37	19-59	19-61										
BIT13	19-8#	19-36	19-57	48-136	48-146									
BIT14	19-8#	19-35	48-125	48-142	48-146	48-299	48-306	61-27	61-57	61-93	68-118	68-121	68-229	

BIT15	19-8#	19-34	48-306	61-27	61-57	61-93	68-226							
BIT2	19-8#	19-50	19-79	19-80										
BIT3	19-8#	19-77	19-78	45-130										
BIT4	19-8#	19-47	19-76	45-125	56-27	56-60	58-33	58-40	58-54	62-30				
BIT5	19-8#	19-46	19-75											
BIT6	19-8#	19-44	19-54	19-73	19-74	45-135								
BIT7	19-8#	19-43	19-53	19-72	45-140									
BIT8	19-8#	19-41	19-67	19-68	19-71									
BIT9	19-8#	19-40	19-66											
BOE	19-8#													
BSELO	20-20#	31-40*	31-73*	32-37*	33-42*	34-32*	41-41*	48-202*	48-207*	48-213*	48-228*	48-233*	48-248*	48-253*
BSEL1	20-21#	45-58*	45-59*											
BSEL2	20-22#													
BSEL3	20-23#	30-42*	42-44	42-49	42-57	42-61	45-62*	45-63*	54-33	54-38	54-42			
BSEL4	20-24#													
BSEL5	20-25#	45-66*	45-67*											
BSEL6	20-26#	31-56	53-43	53-82	53-86	53-168	59-39	59-61	59-76					
BSEL7	20-27#	31-76*	45-70*	45-71*	48-44*	53-44	53-83	53-87	53-172					
BUFNUM	20-88#	40-96	40-101	40-131	40-133	40-174	40-185	41-26	41-27	41-28	41-29	41-67	42-12	42-18
	42-19	42-20	42-21	70-51*	71-47*	72-25*	73-26*	74-24*						
BUFSIZ	20-87#	40-98*	40-103*	40-106*	40-146*	40-147*	40-149*	40-177	40-178	40-189	40-191	42-13		
C\$AU	14-16#													
C\$AUTO	14-16#	46-28												
C\$BRK	14-16#	28-58	29-46	30-70	41-45									
C\$BSEG	14-16#													
C\$BSUB	14-16#	53-33	53-59	53-162	56-18	56-50	57-17	57-50	58-21	58-46	58-62	58-80	59-17	59-52
	60-15	60-63	61-14	61-43	61-76	65-16	65-42	65-70	65-94	65-124				
C\$CEFG	14-16#													
C\$CLCK	14-16#													
C\$CLEA	14-16#	47-16												
C\$CLOS	14-16#													
C\$CLP1	14-16#													
C\$CVEC	14-16#	40-113	45-25	46-21	51-45	68-18	68-82	68-245						
C\$DCLN	14-16#	46-25	51-42											
C\$DODU	14-16#	46-24	51-41											
C\$DRPT	14-16#													
C\$DU	14-16#	49-13												
C\$EDIT	14-16#	15-11												
C\$ERDF	14-16#	28-65	28-76	29-52	30-76	32-61	32-66	37-31	37-36	41-59	41-81	41-96	48-37	48-272
	48-283	48-294	48-303	48-334	48-339	48-345	51-55	53-138	53-150	53-182	54-49	56-31	56-46	56-64
	56-79	57-33	57-46	57-83	58-35	58-42	58-56	58-76	59-42	59-64	60-47	60-52	60-88	60-93
	60-102	61-32	61-37	61-64	61-69	61-98	61-103	62-39	62-44	64-30	64-36	65-31	65-36	65-59
	65-64	65-83	65-88	65-113	65-118	65-140	65-145	67-58	67-68	67-74	67-79	67-87	67-93	67-98
	67-112	67-115	68-85	68-141	68-148	68-154	68-159	68-164	68-170	68-175	68-198	68-250		
C\$ERHR	14-16#													
C\$ERRO	14-16#													
C\$ERSF	14-16#													
C\$ERSO	14-16#	35-54												
C\$ESCA	14-16#	54-32	54-41	56-21	56-25	56-53	56-58	57-20	57-24	57-27	57-29	57-40	57-42	57-53
	57-57	57-60	57-69	57-71	58-24	58-27	58-30	58-39	58-51	58-65	58-68	58-71	58-73	58-84
	58-96	58-100	59-20	59-23	59-33	59-37	59-45	59-56	59-59	60-18	60-21	60-24	60-33	60-39
	60-57	60-66	60-69	60-72	60-81	60-85	61-17	61-20	61-22	61-46	61-49	61-52	61-79	61-82
	61-85	61-88	62-11	62-21	62-24	62-27	62-34	62-36	64-12	64-15	64-18	64-27	65-28	65-56
	65-80	65-97	65-100	65-103	65-110	65-127	65-130	65-137	67-25	67-37	67-40	67-43	67-47	67-51
	68-91	68-104	68-107	68-110	68-128	68-134	68-137	70-60	70-66	70-68	71-56	72-35	73-36	74-34
C\$ESEG	14-16#													
C\$ESUB	14-16#	53-56	53-160	53-195	56-48	56-81	57-48	57-85	58-44	58-60	58-78	58-102	59-50	59-69

CSETST	60-61 14-16# 68-258	60-105 51-47 70-83	61-41 53-197 71-68	61-74 54-51 72-48	61-107 56-83 73-53	65-40 57-86 74-47	65-68 58-106	65-92 59-71	65-122 60-106	65-149 61-109	62-48	64-44	65-151	67-134
CSEXIT	14-16#													
CSGETB	14-16#													
CSGETW	14-16#													
CSGMAN	14-16#													
CSGPHR	14-16#	45-48												
CSGPLO	14-16#													
CSGPRI	14-16#													
CSINIT	14-16#	45-171												
CSINLP	14-16#													
CSMANI	14-16#													
CSMEM	14-16#	40-114												
CSMSG	14-16#	42-24	42-34	42-65	42-101	42-109	42-115	42-120	42-124	42-130	56-107	57-111	59-77	60-111
CSOPEN	14-16#													
CSPNTB	14-16#	28-73	28-82	32-56	42-9	42-10	42-11	42-12	42-13	42-22	42-23	42-30	42-32	42-33
		42-39	42-41	42-46	42-51	42-54	42-59	42-63	42-71	42-78	42-85	42-92	42-99	42-106
		42-108	42-112	42-113	42-114	42-119	42-123	42-129	45-55	45-76	45-163	53-40	53-46	53-140
		53-184	54-35	56-88	56-92	56-97	56-101	56-105	57-89	57-90	57-93	57-97	57-101	57-109
		59-76	60-110	68-86	68-251									
CSPNTF	14-16#	49-11												
CSPNTS	14-16#													
CSPNTX	14-16#	51-58												
CSQIO	14-16#													
CSRDBU	14-16#													
CSREFG	14-16#	45-27	45-29	45-31	45-33									
CSRESE	14-16#	14-16#	49-10											
CSREVI	14-16#	15-11												
CSRFLA	14-16#													
CSRPT	14-16#													
CSSEFG	14-16#													
CSSPRI	14-16#	40-48	41-38	41-117	45-10	68-33								
CSSVEC	14-16#	40-38	45-93	45-94	46-10	51-21	68-15	68-58	68-235					
CSTPRI	14-16#													
CD	19-65#													
CHIPNO	20-135#													
CLRNO	20-125#	30-49	30-62*	45-16*										
CMD	19-110#	32-69	37-38	48-360	60-42	60-55	60-58	61-39	61-72	61-105	65-38	65-66	65-90	65-120
		65-147	67-121	68-202										
CNTRL	19-90#	28-71	28-74	31-67	32-37	32-59	37-29	48-29	48-213	48-275	60-45	60-86	61-23	61-30
		61-53	61-62	61-89	61-96	62-37	64-21	64-28	65-22	65-29	65-46	65-57	65-74	65-81
		65-111	65-131	65-138	67-56	68-139								
CR	19-143#	75-23	75-24	75-25	75-27	75-27	75-29	75-30	75-32	75-34	75-34	75-35	76-19	
CSR	20-15#	20-19	20-20	45-57*	46-12	51-58								
CTS	19-61#													
DDMC	19-97#													
DFPTBL	17-10#													
DIAGMC	14-16	14-16												
DISCON	19-73#													
DMCMDE	20-54#	39-23	45-15*	48-80	48-156	48-188	70-53*	71-49*	72-28*	73-29*	74-27*			
DMR	19-68#	48-82	56-23	57-22	57-36	57-55	58-25	58-47	58-66	58-95	59-21	60-19	60-67	61-18
		61-27	61-47	61-80	62-12	64-13	65-20	65-25	65-98	65-128	67-36	68-103	70-64	
DMRFLG	20-41#	31-61*	31-64*	32-48*	32-51	32-53*	65-49*							
DMRRUN	19-75#	32-64	48-279											
DMRVEC	20-13#	45-78*	45-93											
DMTURN	20-33#	39-29	45-97*	45-123	45-128	45-133	45-138	45-152	45-154	45-157	45-159	48-54	48-97	58-85

L\$DLY	15-11#	28-61	29-49	30-73	41-54
L\$DTP	15-11#				
L\$DTYP	15-11#				
L\$DU	15-11	49-8#			
L\$DUT	15-11#				
L\$DVTY	15-11	22-13#			
L\$EF	15-11#				
L\$ENVI	15-11#				
L\$ETP	15-11#				
L\$EXP1	15-11#				
L\$EXP4	15-11#				
L\$EXP5	15-11#				
L\$HARD	15-11	75-13	75-13#		
L\$HIME	15-11#	40-44	68-29	68-215	
L\$HPCP	15-11#				
L\$HPTP	15-11#				
L\$HW	15-11	17-10	17-10#		
L\$IICP	15-11#				
L\$INIT	15-11	45-8#			
L\$LADP	15-11#				
L\$LAST	15-11	76-32#			
L\$LOAD	15-11#				
L\$LUN	15-11#				
L\$MREV	15-11#				
L\$NAME	15-11#				
L\$PRIO	15-11#				
L\$PROT	15-11	44-8#			
L\$PRT	15-11#				
L\$REPP	15-11#				
L\$REV	15-11#				
L\$SOFT	15-11	76-13	76-13#		
L\$SPC	15-11#				
L\$SPCP	15-11#				
L\$SPTP	15-11#				
L\$STA	15-11#				
L\$SW	15-11	18-8	18-8#		
L\$TEST	15-11#	35-43	41-104		
L\$TIML	15-11#				
L\$UNIT	15-11#	45-46			
L10000	17-10	17-24#			
L10001	18-8	18-12#			
L10002	42-24#				
L10003	42-34#				
L10004	42-65#				
L10005	42-101#				
L10006	42-109#				
L10007	42-115#				
L10010	42-120#				
L10011	42-124#				
L10012	42-130#				
L10014	45-171#				
L10015	46-28#				
L10016	47-16#				
L10017	48-262#				
L10020	48-362#				
L10021	48-371#				
L10022	49-13#				



LOWORD	20-132#	53-82*	53-86*	53-101*	53-131	53-140								
LPLU	19-38#	48-99	48-102	57-22	57-36	57-55	58-25	58-47	58-66	58-87	58-90	59-21	60-19	60-67
	61-18	61-47	61-80	62-12	64-13	65-20	65-25	65-98	65-128	67-28	67-31	68-95	68-98	70-64
MAINT	19-67#	32-46	48-69	58-69	60-22									
MAINT1	19-77#	39-32	39-37	48-57	48-60									
MAINT2	19-80#	39-31	39-38	48-56	48-61									
MANUF	20-70#	45-149												
MCLR	19-35#	30-51	30-54	47-13										
MDIAG	19-36#	30-54												
MICRO	20-34#													
MMANAG	20-56#	40-47*	40-112*	41-68	41-77	41-92	41-98	48-122	48-297					
MNTMDE	20-55#	48-67	70-54*	71-50*	72-29*	73-30*	74-28*							
MNTREC	19-78#													
MODEM	59-26	59-73#												
N	19-120#	56-37	56-70	56-90										
NAKS	19-82#													
NESTPC	20-123#	28-39	28-84	29-32	29-55	31-41*	31-80*	32-38*	32-71*	33-36	33-45	33-46*	33-49*	33-71
	34-34*	34-47*	35-26	35-56	37-22*	37-39*	39-40*	39-47*	41-25*	41-118*				
NEWST	45-32	45-40#	45-47											
NEXT	48-16	48-175#												
NOBFR	19-79#	60-50	60-91											
NOXMEM	40-38	46-10	48-367#	68-15	68-58	68-235								
NXM	19-71#	61-35	61-67	61-101										
NXMFLG	20-104#	40-37*	40-40	40-79	40-88	40-204*	45-18*	46-11*	46-22	46-26*	47-11	48-369*	51-22*	51-39
	51-43*	51-52	51-56*	68-16*	68-19	68-23*	68-57*	68-66	68-76	68-83	68-234*	68-239	68-246	
OSAPTS	14-16#	15-11												
OSAU	14-16#	15-11												
OSBGR	14-16#	15-11												
OSBGNS	14-16#	14-35#	15-11											
OSDU	14-16#	14-35#	15-11											
OSERRT	14-16#	15-11												
OSGNSW	14-16#	14-35#	15-11											
OSPOIN	14-16#	14-35	14-35#	14-35#	14-35#	15-11								
OSSETU	14-16#	15-11	76-32											
OUTFLG	20-109#	41-31*	41-51	48-182	48-288	48-358*								
OUTISR	45-94	48-267#												
OUTRCV	20-93#	41-28*	42-16*	42-20*	42-23	48-342*	48-354							
OUTXMT	20-94#	41-29*	42-17*	42-21*	42-23	48-330*	48-352							
P1	75-15	75-21#												
P2	75-16	75-22#												
P3	75-17	75-23#												
PATCH	76-24#													
PNT	19-8#													
PRETIM	19-124#	57-72	57-91	57-93										
PRI	19-8#													
PRI00	19-8#													
PRI01	19-8#													
PRI02	19-8#													
PRI03	19-8#													
PRI04	19-8#	41-38												
PRI05	19-8#	45-93	45-94											
PRI06	19-8#													
PRI07	19-8#	40-38	40-48	41-117	45-10	46-10	51-21	68-15	68-33	68-58	68-235			
PSTACK	20-121#	45-11*												
R	19-119#	56-35	56-68	56-86	56-88									
RBUF	20-177#	61-86	64-19	65-77	65-101	65-134	67-15	67-45	67-91	67-103				
RCOUNT	20-176#	61-57	61-86	64-19	65-77	65-101	67-12	67-45	67-96	67-101				

T6.2	58-46#										
T6.3	58-62#										
T6.4	58-80#										
T7	16-8	59-15#									
T7.1	59-17#										
T7.2	59-52#										
T8	16-8	60-14#									
T8.1	60-15#										
T8.2	60-63#										
T9	16-8	61-13#									
T9.1	61-14#										
T9.2	61-43#										
T9.3	61-76#										
TBUF	20-168#	60-36	60-82	61-59	62-32	64-24	65-107	67-49	67-72	67-102	
TCOUNT	20-167#	60-36	60-82	61-59	62-32	64-24	67-49	67-77			
TEMP	20-98#	28-44*	28-69	28-80							
TFLAG	20-166#	67-21*	67-66	67-71*	67-126	68-112*	68-146	68-151*	68-207	68-217*	
TH1L	19-125#	57-76	57-95	57-97							
TH2L	19-126#	57-78	57-99	57-101							
TH3L	19-127#	57-74	57-103	57-105							
TH4L	19-128#	57-80	57-107	57-109							
THRESH	19-100#	57-67	60-31	60-79	62-19						
TIMER	19-99#	57-58	62-22								
TOLONG	19-76#	64-33									
TOUT	19-81#	62-42									
TSEL4	68-120*	68-132#	68-152	68-228*							
TSEL6	68-121*	68-133#	68-157	68-229*							
UAM	19-8#										
UPDATE	19-98#	41-111									
WAIT1	20-81#	28-49	45-113*								
WAIT2	20-83#	29-39	30-63	45-116*							
WAIT3	20-85#	41-40	45-118*								
WAIT4	20-86#	41-40*	41-57*								
WMAINT	20-67#	39-25	45-166*	45-169*	48-205	70-47					
WMODEM	19-94#	33-54	39-42	48-31	48-207	59-30	59-54	62-30			
WTYPE	20-32#	45-50*									
X	19-123#	56-43	56-76	56-103	56-105						
X\$ALWA	14-16#										
X\$FALS	14-16#										
X\$OFFS	14-16#										
X\$TRUE	14-16#										
XMTBUF	20-189#	40-94*	40-120*	40-125*	40-128	40-152	40-183	40-184	41-35	41-37	41-66

53-59#	53-59#	53-59#	53-138#	53-140#	53-150#	53-152#	53-160#	53-162	53-162	53-162	53-162#	53-162#	53-162#	53-162#
53-182#	53-184#	53-195#	53-197#	54-29	54-29	54-29	54-29#	54-29#	54-29#	54-32#	54-35#	54-41#	54-49#	54-49#
54-51#	56-17	56-17	56-17	56-17#	56-17#	56-17#	56-18	56-18	56-18	56-18#	56-18#	56-18#	56-21#	56-21#
56-25#	56-31#	56-46#	56-48#	56-50	56-50	56-50	56-50#	56-50#	56-50#	56-53#	56-58#	56-64#	56-79#	56-79#
56-81#	56-83#	56-85	56-85	56-85#	56-85#	56-88#	56-92#	56-97#	56-101#	56-105#	56-107#	57-16	57-16	57-16
57-16	57-16#	57-16#	57-16#	57-17	57-17	57-17	57-17#	57-17#	57-17#	57-20#	57-24#	57-27#	57-29#	57-29#
57-33#	57-40#	57-42#	57-46#	57-48#	57-50	57-50	57-50	57-50#	57-50#	57-50#	57-53#	57-57#	57-60#	57-60#
57-69#	57-71#	57-83#	57-85#	57-86#	57-88	57-88	57-88#	57-88#	57-89#	57-90#	57-93#	57-97#	57-101#	57-101#
57-105#	57-109#	57-111#	58-20	58-20	58-20	58-20#	58-20#	58-20#	58-20#	58-21	58-21	58-21#	58-21#	58-21#
58-21#	58-24#	58-27#	58-30#	58-35#	58-39#	58-42#	58-44#	58-46	58-46	58-46	58-46#	58-46#	58-46#	58-46#
58-51#	58-56#	58-60#	58-62	58-62	58-62	58-62#	58-62#	58-62#	58-62#	58-65#	58-68#	58-71#	58-73#	58-76#
58-78#	58-80	58-80	58-80	58-80#	58-80#	58-80#	58-84#	58-96#	58-100#	58-102#	58-106#	59-15	59-15	59-15
59-15	59-15#	59-15#	59-15#	59-17	59-17	59-17	59-17#	59-17#	59-17#	59-20#	59-23#	59-33#	59-37#	59-37#
59-42#	59-45#	59-50#	59-52	59-52	59-52	59-52#	59-52#	59-52#	59-52#	59-56#	59-59#	59-64#	59-69#	59-71#
59-75	59-75	59-75#	59-75#	59-76#	59-77#	60-14	60-14	60-14	60-14#	60-14#	60-14#	60-15	60-15	60-15
60-15	60-15#	60-15#	60-15#	60-18#	60-21#	60-24#	60-33#	60-39#	60-47#	60-52#	60-57#	60-61#	60-63	60-63
60-63	60-63	60-63#	60-63#	60-63#	60-66#	60-69#	60-72#	60-81#	60-85#	60-88#	60-93#	60-102#	60-105#	60-105#
60-106#	60-109	60-109	60-109#	60-109#	60-110#	60-111#	61-13	61-13	61-13	61-13#	61-13#	61-13#	61-14	61-14
61-14	61-14	61-14#	61-14#	61-14#	61-17#	61-20#	61-22#	61-32#	61-37#	61-41#	61-43	61-43	61-43	61-43
61-43#	61-43#	61-43#	61-46#	61-49#	61-52#	61-64#	61-69#	61-74#	61-76	61-76	61-76	61-76#	61-76#	61-76#
61-76#	61-79#	61-82#	61-85#	61-88#	61-98#	61-103#	61-107#	61-109#	62-8	62-8	62-8	62-8#	62-8#	62-8#
62-8#	62-11#	62-21#	62-24#	62-27#	62-34#	62-36#	62-39#	62-44#	62-48#	64-9	64-9	64-9	64-9#	64-9#
64-9#	64-9#	64-12#	64-15#	64-18#	64-27#	64-30#	64-36#	64-44#	65-15	65-15	65-15	65-15#	65-15#	65-15#
65-15#	65-16	65-16	65-16	65-16#	65-16#	65-16#	65-28#	65-31#	65-36#	65-40#	65-42	65-42	65-42	65-42
65-42#	65-42#	65-42#	65-56#	65-59#	65-64#	65-68#	65-70	65-70	65-70	65-70#	65-70#	65-70#	65-80#	65-80#
65-83#	65-88#	65-92#	65-94	65-94	65-94	65-94#	65-94#	65-94#	65-97#	65-100#	65-103#	65-110#	65-113#	65-113#
65-118#	65-122#	65-124	65-124	65-124	65-124#	65-124#	65-127#	65-127#	65-130#	65-137#	65-140#	65-145#	65-149#	65-149#
65-151#	67-11	67-11	67-11	67-11#	67-11#	67-11#	67-25#	67-37#	67-40#	67-43#	67-47#	67-51#	67-58#	67-58#
67-68#	67-74#	67-79#	67-87#	67-93#	67-98#	67-112#	67-115#	67-134#	68-12	68-12	68-12	68-12#	68-12#	68-12#
68-12#	68-15#	68-18#	68-33#	68-58#	68-82#	68-85#	68-86#	68-91#	68-104#	68-107#	68-110#	68-128#	68-134#	68-134#
68-137#	68-141#	68-148#	68-154#	68-159#	68-164#	68-170#	68-175#	68-198#	68-235#	68-245#	68-250#	68-251#	68-258#	68-258#
70-46	70-46	70-46	70-46#	70-46#	70-46#	70-60#	70-66#	70-68#	70-83#	71-46	71-46	71-46	71-46#	71-46#
71-46#	71-46#	71-56#	71-68#	72-24	72-24	72-24	72-24#	72-24#	72-24#	72-35#	72-48#	73-24	73-24	73-24
73-24	73-24#	73-24#	73-24#	73-36#	73-53#	74-23	74-23	74-23	74-23#	74-23#	74-23#	74-34#	74-47#	74-47#
75-13	75-13	75-13#	75-13#	76-13	76-13	76-13#	76-13#							
MSIOSE MSLDRO	1-A00#	14-16#												
	1-C42#	14-16#	40-48	40-48#	40-113	40-113#	41-38	41-38#	41-117	41-117#	45-10	45-10#	45-25	45-25#
	45-27	45-27#	45-29	45-29#	45-31	45-31#	45-33	45-33#	45-48	45-48#	46-21	46-21#	46-24	46-24#
	51-41	51-41#	51-45	51-45#	68-18	68-18#	68-33	68-33#	68-82	68-82#	68-245	68-245#		
MSMASK	1-a71#	14-16#												
MSMCHI	1-4#	14-16	14-16#	14-16#										
MSMCLO	1-a24#	14-16	14-16#	14-16#										
MSMSK1	1-a77#	14-16#												
MSPOP	1-881#	14-16#	17-24	17-24#	18-12	18-12#	42-24	42-24#	42-34	42-34#	42-65	42-65#	42-101	42-101#
	42-109	42-109#	42-115	42-115#	42-120	42-120#	42-124	42-124#	42-130	42-130#	44-14	44-14#	45-171	45-171#
	46-28	46-28#	47-16	47-16#	48-262	48-262#	48-362	48-362#	48-371	48-371#	49-13	49-13#	51-47	51-47#
	51-60	51-60#	53-56	53-56#	53-160	53-160#	53-195	53-195#	53-197	53-197#	54-51	54-51#	56-48	56-48#
	56-81	56-81#	56-83	56-83#	56-107	56-107#	57-48	57-48#	57-85	57-85#	57-86	57-86#	57-111	57-111#
	58-44	58-44#	58-60	58-60#	58-78	58-78#	58-102	58-102#	58-106	58-106#	59-50	59-50#	59-69	59-69#
	59-71	59-71#	59-77	59-77#	60-61	60-61#	60-105	60-105#	60-106	60-106#	60-111	60-111#	61-41	61-41#
	61-74	61-74#	61-107	61-107#	61-109	61-109#	62-48	62-48#	64-44	64-44#	65-40	65-40#	65-68	65-68#
	65-92	65-92#	65-122	65-122#	65-149	65-149#	65-151	65-151#	67-134	67-134#	68-258	68-258#	70-83	70-83#
	71-68	71-68#	72-48	72-48#	73-53	73-53#	74-47	74-47#	75-19	75-19#	76-17	76-17#	76-30	76-30#
MSPRIN	1-a36#	14-16#	28-73	28-73#	28-82	28-82#	32-56	32-56#	42-9	42-9#	42-10	42-10#	42-11	42-11#
	42-12	42-12#	42-13	42-13#	42-22	42-22#	42-23	42-23#	42-30	42-30#	42-32	42-32#	42-33	42-33#
	42-39	42-39#	42-41	42-41#	42-46	42-46#	42-51	42-51#	42-54	42-54#	42-59	42-59#	42-63	42-63#
	42-71	42-71#	42-78	42-78#	42-85	42-85#	42-92	42-92#	42-99	42-99#	42-106	42-106#	42-107	42-107#