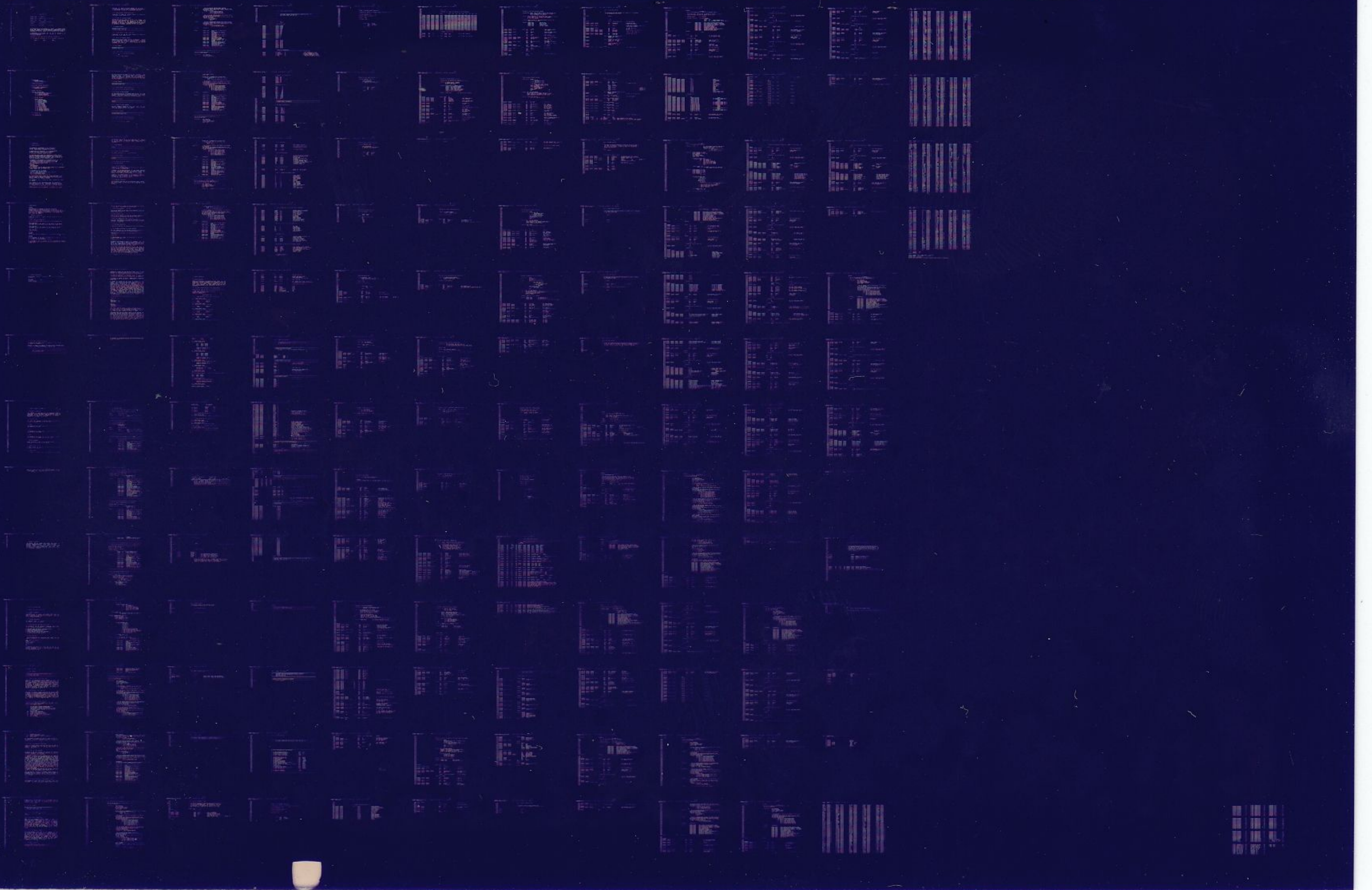


11/21+
KMV 11A

KMV 11-A FCTNL DIAG
CNKMCAO

COPYRIGHT (c) 1982-84
AH-T847A-MC
FICHE 01 OF 01

JUL 1984
digital
Made In USA



1
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

.NLIST TOC
.REM @

IDENTIFICATION

PRODUCT CODE: AC-T846A-MC
PRODUCT NAME: CNKMCAO KMV11-A FCTNL DIAG
PRODUCT DATE: APRIL 9, 1984
MAINTAINER: ISS/DIAGNOSTIC SERVICES
AUTHOR: ALAIN FEUILLET
MODIFIED BY: JAKI BERG 9-APR-1984

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

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

COPYRIGHT (C) 1982,1984 DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP QBUS
DEC DECUS DECTAPE

48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98

TABLE OF CONTENTS

- 1.0 INTRODUCTION
 - 1.1 PROGRAM ABSTRACT
 - 1.2 HARDWARE INTRODUCTION
- 2.0 HARDWARE REQUIREMENTS
- 3.0 PRELIMINARY PROGRAM REQUIREMENTS
- 4.0 GENERAL PROGRAM CONSIDERATIONS
 - 4.1 DIAGNOSTIC SUPERVISOR
 - 4.2 EXECUTION TIME
- 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
 - 6.3.2 RESTART COMMAND
 - 6.3.3 CONTINUE COMMAND
 - 6.3.4 PROCEED COMMAND
 - 6.3.5 ADD COMMAND
 - 6.3.6 DROP COMMAND
 - 6.3.7 PRINT COMMAND
 - 6.3.8 DISPLAY COMMAND
 - 6.3.9 FLAGS COMMAND
 - 6.3.10 ZFLAGS COMMAND
 - 6.3.11 CONTROL CHARACTERS
 - 6.3.12 HARDWARE PARAMETERS
 - 6.3.13 SOFTWARE PARAMETERS
 - 6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE
- 7.0 TEST DESCRIPTIONS
- 8.0 ERROR INFORMATION
 - 8.1 ERROR REPORTING

100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
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

1.0 INTRODUCTION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC IS DESIGNED TO TEST THE KMV11-A FIRMWARE WHICH HANDLES BASIC HDLC/SDLC FRAMING WITH FULL MODEM CONTROL ACCORDING TO DEC STD 052.

TO UNDERSTAND THIS DIAGNOSTIC, IT IS RECOMMENDED THAT THE FIRMWARE FUNCTIONAL SPECIFICATIONS (APPENDIX A OF THE KMV11-A TECHNICAL MANUAL) BE READ.

THE FUNCTIONAL DIAG. LOADS THE FIRMWARE AS A NORMAL APPLICATION MODE FIRMWARE PROGRAM AND PASSES CONTROL TO THE START ADDRESS OF THE BASIC HDLC/SDLC FIRMWARE. THE FIRMWARE DEFINES THE USE OF THE CSR'S OTHERS THAN BSEL1 (WHICH IS HARDWARE DEFINED AND DESCRIBED IN THE TECHNICAL MANUAL).

THE PURPOSE OF THIS PROGRAM IS TO PERFORM TESTING OF:
- SEQUENCING THE COMMANDS AND RESPONSES RECEIVED
- XMIT/RECEIVE PATHS USING THE LOOP BACK FACILITY
- ALL THE POSSIBLE STATUS RETURNED
DUE TO THE LOOP BACK MODE OF TEST, SOME STATUS CAN'T BE TESTED THESE ARE:
- MODEM DOWN(365)
- FCS ERROR(367)
DUE TO THE FACT THAT THE KMV11-A MODULE UNDER TEST IS CONSIDERED TO RUN CORRECTLY FROM A HARDWARE POINT OF VIEW, THE STATUS 370 (LATENCY ERROR) CAN'T BE TESTED EITHER.

THE XMIT/RECEIVE PATH IS TESTED :
AT SPEEDS 2.4K, 48K AND 64KBPS
- WITH FULL MODEM CONTROL AND WITH DATA LEADS ONLY
- WITH STATION ADDRESS SEARCH
- FOR ALL FRAME PASSING UPON RECEPTION

ALL THE SUPPORTED MODEM SIGNALS ARE TESTED (ACCORDING TO THE LOOP BACK FACILITY USED), EXCEPT RING INDICATOR (CCITT 125) WHICH IS LOOPED ON TERMINAL IN SERVICE, THE LATTER BEING NOT SUPPORTED BY THE HDLC FRAMING FIRMWARE.

THIS DIAGNOSTIC DOESN'T REQUIRE ANY MANUAL INTERVENTION DURING TEST PERIOD.

THIS PROGRAM WILL BE IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR AND A STRUCTURED PROGRAMMING APPROACH. BECAUSE THE DESIGN CONFORMS 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 Q BUS ADDRESS.

157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
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

VECTOR ADDRESS.

CAUTION:

THIS DIAGNOSTIC IS DESIGNED TO RUN WITH A SPECIFIC FIRMWARE WHICH IS LINKED AND LISTED AS THE LAST PART OF THE PRESENT DIAGNOSTIC. ANY PATCHES INSIDE THE APPLICATION CODE BEFORE OR AFTER IT WAS LOADED BY THE DIAGNOSTIC IN THE KMV RAM, WILL CAUSE FALSE TEST CONDITIONS (PARTICULARLY, THIS CONCERNS CCB'S ROB'S LENGTH AND NUMBER).

1.2 HARDWARE INTRODUCTION

TO RUN THE DIAGNOSTIC, EXTERNAL LOOP BACK CONNECTOR MUST BE INSTALLED.

EXTERNAL LOOP BACK CONNECTORS:

KMV11-A CAN OPERATE UNDER THE RS422 OR RS423 RECOMMENDATIONS

RS422 LOOP BACK:

- USE M3255 PLUG TO LOOP DIRECTLY AT THE OUTPUT OF THE MODULE
- USE M3251 PLUG AT THE END OF BC55U MODEM CABLE CONNECTOR ASSY.

RS423 LOOP BACK:

- USE M3255 PLUG TO LOOP DIRECTLY AT THE OUTPUT OF THE MODULE
- USE M3251 PLUG AT THE END OF BC55M CABLE CONNECTOR ASSY.

RS232 LOOP BACK:

- SAME AS FOR RS423

CAUTION:

USE OF M325 LOOP BACK CONNECTOR WILL CAUSE MESSAGE ERROR IN TESTS

CAUTION:

AT THE BEGINNING OF THE DIAG. THE OPERATOR MUST ANSWER BY '1' FOR YES OR '0' FOR NO TO THE QUESTION:

"IS EXTERNAL LOOP BACK CONNECTOR INSERTED?"

IF HE ANSWERS NO OR IF HE ANSWERS YES (THE PLUG BEING NOT INSERTED), IN BOTH CASES, THE DIAGNOSTIC WILL REPORT AN ERROR AND EXIT CORRESPONDING TESTS.

202
203
204
205
206
207
208
209
210
211
212
213
214

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE KHV11-A FUNCTIONAL
DIAGNOSTIC:

- SBC-11/21+
- 16K MEMORY
- CONSOLE TERMINAL
- THE M7500 MODULE WITH EPROM'S AT ECO LEVEL 1

216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236

3.0 PRELIMINARY PROGRAM REQUIREMENTS

THE PROCESSOR AND MEMORY SHOULD BE THOROUGHLY TESTED PRIOR TO RUNNING THIS DIAGNOSTIC.

MOREOVER, IN ORDER TO BE SURE OF THE MODULE FROM A HARDWARE POINT OF VIEW, IT IS GREATLY RECOMMENDED TO RUN THE KVV11-A LINE CONTROLLER STATIC TESTS NKMB AND THE DEC/X11 MODULE XKMD.

```

*****
*
*       NOTE: THE KVV11 DIAGNOSTICS NKMDA AND NKMBA SHOULD BE
*       BEFORE RUNNING NKMCA.
*
*****

```

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
286
287
288
289
290
291
292
293
294

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

THE TOTAL TIME REQUIRED TO RUN THE M7500 FUNCTIONAL TESTS IS ABOUT 260 SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP.

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

4.4 ACT

THIS PROGRAM MAY BE LOADED UNDER ACT 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

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM. IF IT IS INSTALLED, IT IS DISABLED BY THE PROGRAM.

4.7 MEMORY PARITY OPTION

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

4.8 ERROR LOGGING

THE NUMBER OF ERRORS WHICH HAVE OCCURRED ON EACH DEVICE

295
296
297

UNDER TEST SINCE THE LAST START OR RESTART COMMAND IS KEPT
IN AN ERROR LOG. THIS LOG MAY BE PRINTED BY USING THE
'PRINT' COMMAND (SEE SECTION 6.3.8).

299
300
301
302
303
304
305
306
307
308
309
310
311
312
313

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER OR FROM ACT, 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.

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
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

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 PROMPT (DR>)
- C) ENTER START COMMAND(SEE 6.3.1)
- 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

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

```
DIAG. RUN-TIME SERVICES -A-0
NKMCAO
KMV11-A FUNCTIONAL DIAGNOSTIC
UNIT IS M7500
50 HZ (L) N? Y
LSI (L) N? Y
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).

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
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

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:

- 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

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
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

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 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

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
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

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 (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>)

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
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

<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.

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

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
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656

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 PRIN COMMAND

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

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

657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713

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- INITIAL DIALOGUE (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 CAUSES ALL TELETYPE OUTPUT TO BE SUPPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 4 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. KMV11 A CSR ADDRESS : () ?77000 ?

714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE ON THE QBUS. THE DEFAULT IS 177000 (OCTAL).

2. DEVICE VECTOR ADDRESS : (0) 300 ?

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

3. DEVICE PRIORITY LEVEL : (0) 4 ?

THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF THIS DEVICE WILL BE EXECUTED. THE DEFAULT VALUE IS 4.

4. WILL LOOP CONNECTOR BE USED 0=NO,1=YES : (0) 1 ?

THIS TELLS THE PROGRAM IF THE LOOP-BACK CONNECTOR IS PLUGGED IN OR NOT. THE ALLOWABLE VALUES ARE 0 AND 1, AND THE DEFAULT VALUE IS 1 (PLUGGED IN).

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETERS ARE REQUESTED BY THE LOGIC DIAGNOSTIC

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.

AFTER THE INITIAL DIALOGUE, THE SUPERVISOR ISSUES THE QUESTION BELOW:

CHANGE HW (L) ?

IF YOUR RESPONSE IS Y (YES), THE SUPERVISOR ASKS FOR THE NUMBER OF UNITS UNDER TEST:

* UNITS (D) ?

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

771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827

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 (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 1

<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

UNIT 21

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

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 16 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR 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 A 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.

828
829
830

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

832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888

7.0 TEST DESCRIPTIONS

*****TEST01*****

** - VERIFY THAT THE KMV11-A CAN BE INITIALIZED

MODE: APPLICATION MODE

REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR

*****TEST02*****

** - SELF-TEST RUNNING TEST

KMV ON BOARD SWITCHES MUST BE:
E13-SW8 ON
E29-SW10 OFF

MODE: APPLICATION MODE

REPORTS: ERROR 1 SELF TEST IS NOT CORRECTLY RUN
ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR

*****TEST03*****

** - TEST IF APPLICATION PROGRAM CAN BE LOADED AND STARTED

MODE: APPLICATION MODE

REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR
ERROR 10001 RUN FUNCTION NOT CORRECTLY PERFORMED
ERROR 10002 WRITE FUNCTION NOT CORRECTLY PERFORMED
ERROR 10003 READ FUNCTION NOT CORRECTLY PERFORMED
ERROR 10004 DATA COMPARE ERROR
ERROR 10008 UNEXPECTED EPROM'S ON KMV

889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945

*****TEST04*****

** - TEST OF THE CSR HANDSHAKING WITHOUT INTERRUPT

COMMAND LOADED: READ MODEM (F14)

RESPONSE EXPECTED: READ MODEM RESPONSE WITH :
STATUS = 371
MODEM = ALL OFF

REPORTS:	ERROR 10000	KMV11 FAILS TO RESET MASTER CLEAR
	ERROR 10001	RUN FUNCTION NOT CORRECTLY PERFORMED
	ERROR 10002	WRITE FUNCTION NOT CORRECTLY PERFORMED
	ERROR 10003	READ FUNCTION NOT CORRECTLY PERFORMED
	ERROR 10004	DATA COMPARE ERROR
	ERROR 10005	UNEXPECTED INTERRUPT IN
	ERROR 10006	UNEXPECTED INTERRUPT OUT
	ERROR 10007	NO MORE INTERRUPT WHILE QIO PENDING
	ERROR 10008	UNEXPECTED EPROM'S ON KMV
	ERROR 10009	UNEXPECTED QIO RESPONSE
	ERROR 00002	RDYI NOT ASSERTED BY THE KMV AFTER AN INPUT REQ
	ERROR 00003	RDYO NOT ASSERTED BY THE KMV IN RESPONSE
	ERROR 00004	UNEXPECTED INTERRUPT RECEIVED IEI&IEO DISABLED

*****TEST05*****

** - TEST OF THE CSR HANDSHAKING WITH INTERRUPTS

COMMAND LOADED: READ MODEM (F14)

RESPONSE EXPECTED: READ MODEM RESPONSE WITH :
STATUS = 371
MODEM = ALL OFF

REPORTS:	ERROR 10000	KMV11 FAILS TO RESET MASTER CLEAR
	ERROR 10001	RUN FUNCTION NOT CORRECTLY PERFORMED
	ERROR 10002	WRITE FUNCTION NOT CORRECTLY PERFORMED
	ERROR 10003	READ FUNCTION NOT CORRECTLY PERFORMED
	ERROR 10004	DATA COMPARE ERROR
	ERROR 10005	UNEXPECTED INTERRUPT IN
	ERROR 10006	UNEXPECTED INTERRUPT OUT
	ERROR 10007	NO MORE INTERRUPT WHILE QIO

946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002

PENDING
ERROR 10008 UNEXPECTED EPROM'S ON KMV
ERROR 10009 UNEXPECTED QIO RESPONSE

*****TEST06*****

** - TEST OF QIO PROCESSING IN CASE OF KMV RESOURCE ERROR

TO DO THIS TEST, THE KMV POOL IS ARTIFICIALLY EMPTIED BEFORE ANY TRANSACTIONS.

COMMAND LOADED: ALL COMMANDS

RESPONSE EXPECTED:

STATUS = 357 (FOR ALL EXCEPT F16,F17)
STATE = S1 (READ COMMAND ONLY)
MODEM = ALL OFF(F16,F17 AND F14 ONLY)

REPORTS:

ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR
ERROR 10001 RUN FUNCTION NOT CORRECTLY PERFORMED
ERROR 10002 WRITE FUNCTION NOT CORRECTLY PERFORMED
ERROR 10003 READ FUNCTION NOT CORRECTLY PERFORMED
ERROR 10004 DATA COMPARE ERROR
ERROR 10005 UNEXPECTED INTERRUPT IN
ERROR 10006 UNEXPECTED INTERRUPT OUT
ERROR 10007 NO MORE INTERRUPT WHILE QIO PENDING
ERROR 10008 UNEXPECTED EPROM'S ON KMV
ERROR 10009 UNEXPECTED QIO RESPONSE

*****TEST07*****

** - TEST OF QIO PROCESSING AS BELOW:

SUBTEST1 - STATE = S1

ALL THE COMMANDS EXCEPT F1 ARE PASSED WHILE THE LINE IS NOT CONFIGURATED. EXPECTED STATUS ARE:

STATUS = 371 FOR F2,F5,F6,F7,F10,F14
= 1 FOR F16,F17
= NONE FOR DUMMY COMMANDS

SUBTEST2 - ENTER COMMAND F16 TWICE

ENTER COMMAND F1 TWICE FOR:
SDLC PROTOCOL
FULL MODEM CONTROL
CLOCK SOURCE INTERNAL

1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059

RATE = 2.4K

EXPECTED RESPONSES ARE:

STATUS = 363 FOR SECOND F16
= 363 FOR SECOND F1
= 1 FOR F16 WITH MODEM OFF
= 1 FOR F4 WITH S109 CHANGE
= 1 FOR F4 WITH S106 CHANGE
= 1 FOR F4 WITH S107 CHANGE
= 1 FOR F1

CURRENT STATE = S6
MODEM = ALL ON(EXCEPT S125 WHICH IS LOOPED ON
THE TERMINAL IN SFRVICE SIGNAL =OFF)

SUBTEST3 - DECONFIGURATE THE LINE

ENTER COMMAND F17 TWICE
ENTER COMMAND F14 TWICE
ENTER COMMAND F16 TWICE
ENTER COMMAND F1
" " F2 TWICE
" " F1

EXPECTED RESPONSES ARE:

STATUS = 363 FOR F17
= 363 FOR F16
= 371 FOR F1
= 1 FOR F17
= 363 FOR F2
= 356 FOR F1
= 1 FOR F14 WITH STATE=S6 AND
S142+S112+S109+S106+S107 ON
= 1 FOR F16 WITH THE SAME PARAMETERS
= 1 FOR F4 WITH S106 & S109 CHANGE
= 1 FOR F4 WITH S107 CHANGE
= 1 FOR F2 COMPLETED

THEN CURRENT STATE = S1
MODEM = ALL OFF

SUBTEST4 - TEST THAT WE ARE OFF LINE AGAIN

ENTER COMMAND F17 FOR STATUS = 1
F14 FOR STATUS = 371

REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER
CLEAR
ERROR 10001 RUN FUNCTION NOT CORRECTLY
PERFORMED
ERROR 10002 WRITE FUNCTION NOT CORRECTLY
PERFORMED
ERROR 10003 READ FUNCTION NOT CORRECTLY
PERFORMED
ERROR 10004 DATA COMPARE ERROR
ERROR 10005 UNEXPECTED INTERRUPT IN
ERROR 10006 UNEXPECTED INTERRUPT OUT
ERROR 10007 NO MORE INTERRUPT WHILE QIO
PENDING

1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116

ERROR 10008 UNEXPECTED EPROM'S ON KMV
ERROR 10009 UNEXPECTED QIO RESPONSE
ERROR 10010 UNEXPECTED NUMBER OF RESP-
ONSES
ERROR 10012 LOOP BACK NOT USED

*****TEST08*****

TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 2.4K
WITH FULL MODEM CONTROL

CONFIGURATE: SDLC PROTOCOL
FULL MODEM CONTROL
WITHOUT ADDRESS SEARCH
CLOCK SOURCE INTERNAL
RATE = 2.4K

BEFORE SUBTESTS, A COMMAND F14 IS GIVEN FIRST
TO CONTROL MODEMS AND THE LINE STATE, SUCH AS:
STATE = S6
MODEM = S142.S112.S109.S106.S107 ON

SUBTEST1 - TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
OF 2 BUFFERS.
THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
XMIT BUFFERS ARE GIVEN ONE OF WHICH WITH 1 BYTE LONG
EXPECTED RESPONSES ARE:
STATUS = 360 FOR F6 THIRD BUFFER PASSED
 = 372 FOR F5 1 BYTE LONG BUFFER
 = 360 FOR F5 THIRD BUFFER PASSED
 = 1 FOR F5 FIRST BUFFER XMITTED
 = 1 FOR F6 FIRST BUFFER FELT
 = 1 FOR F5 SECOND BUFFER XMITTED
 = 1 FOR F6 SECOND BUFFER FELT

SUBTEST2 - TEST OF BUFFER OVERFLOW STATUS 373 FOR RECEPTION
A 32 BYTE LONG BUFFER IS AFFECTED FOR RECEIVE WHILE
A 100 BYTE LONG BUFFER IS PASSED TO XMIT.
EXPECTED RESPONSES ARE:
STATUS = 1 FOR F5 BUFFER XMITTED
 = 373 FOR F6 BUFFER OVERFLOW

SUBTEST3 - TEST OF NON EXISTENT MEMORY STATUS 374
TWO RECEIVE BUFFER ARE AFFECTED ONE OF WHICH WITH
NON EXISTENT ADDRESS
TWO XMIT BUFFERS ARE PASSED ONE OF WHICH WITH NON
EXISTENT ADDRESS
EXPECTED RESPONSES ARE:
STATUS = 374 FOR F6 NON EXISTENT MEMORY BUFFER
 = 1 FOR F5 COORRECT BUFFER
 = 374 FOR F5 NON EXISTENT MEMORY BUFFER
 = 364 FOR F6 RECEIVE ABORT

SUBTEST4 - TEST OF ABORT RECEIVED STATUS 364 AND ACTION OF XMIT

1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173

ABORT COMMAND.
ONE RECEIVE BUFFER IS AFFECTED THEN, TWO XMIT BUFFERS
ARE PASSED. AFTER A SHORT WAIT, XMIT ABORT IS GIVEN
EXPECTED STATUS ARE:
STATUS = 1 FOR F7 XMIT ABORTED
= 364 FOR F6 ABORT RECEIVED

SUBTEST5 - TEST OF ACTION OF RECEIVE ABORT COMMAND.
TWO RECEIVE BUFFERS ARE AFFECTED THEN, TWO XMIT ONES
ARE PASSED. AFTER A SHORT WAIT, RECEIVE ABORT IS GIVEN
EXPECTED STATUS ARE:
STATUS = 1 FOR F10 RECEIVE ABORTED
= 1 FOR F5 FIRST BUFFER SENT
= 1 FOR F5 SECOND BUFFER SENT

SUBTEST6 - TEST OF KMV RDB OVERFLOW DURING XMIT/RECEIVE BUFFER
ONE RECEIVE BUFFER IS AFFECTED,
THEN TWO XMIT BUFFER ARE PASSED THE FIRST ONE TOO
LONG.
EXPECTED STATUS ARE:
STATUS = 373 FOR FIRST F5
= 1 FOR SECOND F5
= 1 FOR F6

SUBTEST7 - TEST OF A DECONFIGURATE DURING XMIT/RECEIVE BUFFER,
TWO RECEIVE BUFFERS ARE AFFECTED THEN, TWO XMIT ONES
ARE PASSED. AFTER A SHORT WAIT, DECONFIGURATE IS PASSED
EXPECTED STATUS IS:
STATUS = 1 FOR F2 DECONFIGURATE DONE
AND ALL BUFFERS ARE KILLED

PATTERN: INCREMENTAL
FOR EACH TEST WHERE A CORRECT BUFFER IS OBTAINED, THIS BUFFER
IS COMPARED TO THE CORRESPONDING XMIT BUFFER

REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER
CLEAR
ERROR 10001 RUN FUNCTION NOT CORRECTLY
PERFORMED
ERROR 10002 WRITE FUNCTION NOT CORRECTLY
PERFORMED
ERROR 10003 READ FUNCTION NOT CORRECTLY
PERFORMED
ERROR 10004 DATA COMPARE ERROR
ERROR 10005 UNEXPECTED INTERRUPT IN
ERROR 10006 UNEXPECTED INTERRUPT OUT
ERROR 10007 NO MORE INTERRUPT WHILE QIO
PENDING
ERROR 10008 UNEXPECTED EPROM'S ON KMV
ERROR 10009 UNEXPECTED QIO RESPONSE
ERROR 10010 UNEXPECTED NUMBER OF RESP-
ONSES RECEIVED
ERROR 10011 UNEXPECTED DATA RECEIVED
ERROR 10012 LOOP BACK NOT USED

1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230

*****TEST09*****

TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 2.4K
WITH DATA LEADS ONLY

CONFIGURATE: HDLC PROTOCOL
DATA LEADS ONLY
WITHOUT ADDRESS SEARCH
CLOCK SOURCE INTERNAL
RATE = 2.4K

BEFORE SUBTESTS A COMMAND F14 IS GIVEN FIRST TO CONTROL
THE LINE STATE AND MODEM SUCH AS:

STATE = S6
MODEM = S142 ONLY

SUBTEST1 - TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
OF 2 BUFFERS.
THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
XMIT BUFFERS ARE GIVEN ONE OF WHICH WITH 1 BYTE LONG
EXPECTED RESPONSES ARE:
STATUS = 360 FOR F6 THIRD BUFFER PASSED
= 372 FOR F5 1 BYTE LONG BUFFER
= 360 FOR F5 THIRD BUFFER PASSED
= 1 FOR F5 FIRST BUFFER XMITTED
= 1 FOR F6 FIRST BUFFER FELT
= 1 FOR F5 SECOND BUFFER XMITTED
= 1 FOR F6 SECOND BUFFER FELT

SUBTEST2 - TEST OF BUFFER OVERFLOW STATUS 373 FOR RECEPTION
A 32 BYTE LONG BUFFER IS AFFECTED FOR RECEPTION WHILE
A 100 BYTE LONG BUFFER IS PASSED TO XMIT.
EXPECTED RESPONSES ARE:
STATUS = 1 FOR F5 BUFFER XMITTED
STATUS = 373 FOR F6 BUFFER OVERFLOW

SUBTEST3 - TEST OF NON EXISTENT MEMORY STATUS 374
TWO RECEIVE BUFFER ARE AFFECTED ONE OF WHICH WITH
NON EXISTENT ADDRESS
TWO XMIT BUFFERS ARE PASSED ONE OF WHICH WITH NON
EXISTENT ADDRESS
EXPECTED RESPONSES ARE:
STATUS = 374 FOR F6 NON EXISTENT MEMORY BUFFER
= 1 FOR F5 CORRECT BUFFER
= 374 FOR F5 NON EXISTENT MEMORY BUFFER
= 364 FOR F6 RECEIVE ABORT

SUBTEST4 - TEST OF ABORT RECEIVED STATUS 364 AND ACTION OF XMIT
ABORT COMMAND.
ONE RECEIVE BUFFER IS AFFECTED THEN, TWO XMIT BUFFERS
ARE PASSED. AFTER A SHORT WAIT, XMIT ABORT IS GIVEN
TWICE.
EXPECTED STATUS ARE:
STATUS = 1 FOR F5 XMIT ABORTED
= 364 FOR F6 ABORT RECEIVED

1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287

SUBTEST5 - TEST OF ACTION OF RECEIVE ABORT COMMAND.
TWO RECEIVE BUFFERS ARE AFFECTED THEN, TWO XMIT ONES
ARE PASSED. AFTER A SHORT WAIT, RECEIVE ABORT IS GIVEN
EXPECTED STATUS ARE:
STATUS = 1 FOR F10 RECEIVE ABORTED
 = 1 FOR F5 FIRST BUFFER SENT
 = 1 FOR F5 SECOND BUFFER SENT

SUBTEST6 - TEST OF KMV RDB OVERFLOW DURING XMIT/RECEIVE BUFFER
ONE RECEIVE BUFFER IS AFFECTED,
THEN TWO XMIT BUFFER ARE PASSED THE FIRST ONE TOO LONG
EXPECTED STATUS ARE:
STATUS = 373 FOR FIRST F5
 = 1 FOR SECOND F5
 = 1 FOR F6

SUBTEST7 - TEST OF A DECONFIGURATE DURING XMIT/RECEIVE BUFFER,
TWO RECEIVE BUFFERS ARE AFFECTED THEN, TWO XMIT ONES
ARE PASSED. AFTER A SHORT WAIT, DECONFIGURATE IS PASSED
EXPECTED STATUS IS:
STATUS = 1 FOR F2 DECONFIGURATE DONE
 AND ALL BUFFERS ARE KILLED

PATTERN: INCREMENTAL

REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR
 ERROR 10001 RUN FUNCTION NOT CORRECTLY PERFORMED
 ERROR 10002 WRITE FUNCTION NOT CORRECTLY PERFORMED
 ERROR 10003 READ FUNCTION NOT CORRECTLY PERFORMED
 ERROR 10004 DATA COMPARE ERROR
 ERROR 10005 UNEXPECTED INTERRUPT IN EXPECTED INTERRUPT OUT
 ERROR 10006 EXPECTED INTERRUPT OUT
 ERROR 10007 NO MORE INTERRUPT WHILE QIO PENDING
 ERROR 10008 UNEXPECTED EPROM'S ON KMV
 ERROR 10009 UNEXPECTED QIO RESPONSE
 ERROR 10010 UNEXPECTED NUMBER OF RESPONSES RECEIVED
 ERROR 10011 UNEXPECTED DATA RECEIVED
 ERROR 10012 LOOP BACK NOT USED

.....

.....TEST10.....

TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 64K
WITH FULL MODEM CONTROL

CONFIGURATE: SOLC PROTOCOL
 FULL MODEM CONTROL
 WITHOUT ADDRESS SEARCH

1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344

CLOCK SOURCE INTERNAL
RATE = 64K

BEFORE TEST, A COMMAND F14 IS GIVEN FIRST
TO CONTROL MODEMS AND THE LINE STATE, SUCH AS:
STATE = S6
MODEM = S142.S112.S109.S106.S107 ON

TEST - TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
OF 2 BUFFERS.
THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
XMIT BUFFERS ARE GIVEN ONE OF WHICH WITH 1 BYTE LONG
EXPECTED RESPONSES ARE:
STATUS = 360 FOR F6 THIRD BUFFER PASSED
 = 372 FOR F5 1 BYTE LONG BUFFER
 = 360 FOR F5 THIRD BUFFER PASSED
 = 1 FOR F5 FIRST BUFFER XMITTED
 = 1 FOR F6 FIRST BUFFER FELT
 = 1 FOR F5 SECOND BUFFER XMITTED
 = 1 FOR F6 SECOND BUFFER FELT

PATTERN: INCREMENTAL

REPORTS: ERROR 10000 KMV11 FAILS TO RESET MAST_R
 CLEAR
 ERROR 10001 RUN FUNCTION NOT CORRECTLY
 PERFORMED
 ERROR 10002 WRITE FUNCTION NOT CORRECTLY
 PERFORMED
 ERROR 10003 READ FUNCTION NOT CORRECTLY
 PERFORMED
 ERROR 10004 DATA COMPARE ERROR
 ERROR 10005 UNEXPECTED INTERRUPT IN
 ERROR 10006 UNEXPECTED INTERRUPT OUT
 ERROR 10007 NO MORE INTERRUPT WHILE QIO
 PENDING
 ERROR 10008 UNEXPECTED EPROM'S ON KMV
 ERROR 10009 UNEXPECTED QIO RESPONSE
 ERROR 10010 UNEXPECTED NUMBER OF RESP-
 ONSES RECEIVED
 ERROR 10011 UNEXPECTED DATA RECEIVED
 ERROR 10012 LOOP BACK NOT USED

.....

.....TEST11.....

TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 64K
WITH DATA LEADS ONLY

CONFIGURATE: HDLC PROTOCOL
DATA LEADS ONLY
WITHOUT ADDRESS SEARCH
CLOCK SOURCE INTERNAL
RATE = 64K

1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401

BEFORE TEST, A COMMAND F14 IS GIVEN FIRST
TO CONTROL MODEMS AND THE LINE STATE, SUCH AS:
STATE = S6
MODEM = S142 ON

SUBTEST1 - TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
OF 2 BUFFERS.
THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
XMIT BUFFERS ARE GIVEN ONE OF WHICH WITH 1 BYTE LONG
EXPECTED RESPONSES ARE:
STATUS = 360 FOR F6 THIRD BUFFER PASSED
 = 372 FOR F5 1 BYTE LONG BUFFER
 = 360 FOR F5 THIRD BUFFER PASSED
 = 1 FOR F5 FIRST BUFFER XMITTED
 = 1 FOR F6 FIRST BUFFER FELT
 = 1 FOR F5 SECOND BUFFER XMITTED
 = 1 FOR F6 SECOND BUFFER FELT

PATTERN: INCREMENTAL

REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER
 CLEAR
 ERROR 10001 RUN FUNCTION NOT CORRECTLY
 PERFORMED
 ERROR 10002 WRITE FUNCTION NOT CORRECTLY
 PERFORMED
 ERROR 10003 READ FUNCTION NOT CORRECTLY
 PERFORMED
 ERROR 10004 DATA COMPARE ERROR
 ERROR 10005 UNEXPECTED INTERRUPT IN
 ERROR 10006 UNEXPECTED INTERRUPT OUT
 ERROR 10007 NO MORE INTERRUPT WHILE QIO
 PENDING
 ERROR 10008 UNEXPECTED EPROM'S ON KMV
 ERROR 10009 UNEXPECTED QIO RESPONSE
 ERROR 10010 UNEXPECTED NUMBER OF RESP-
 ONSES RECEIVED
 ERROR 10011 UNEXPECTED DATA RECEIVED
 ERROR 10012 LOOP BACK NOT USED

*****TEST12*****

TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 48K
WITH FULL MODEM CONTROL AND ADDRESS SEARCH

CONFIGURATE: SOLC PROTOCOL
 FULL MODEM CONTROL
 WITH ADDRESS SEARCH(252)
 CLOCK SOURCE INTERNAL
 RATE = 48K

BEFORE TEST, A COMMAND F14 IS GIVEN FIRST

1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445

TO CONTROL MODEMS AND THE LINE STATE, SUCH AS:
STATE = S6
MODEM = S142+S109+S106+S107 ON

TEST:

- TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
OF 1 OF 2 BUFFERS.
THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
XMIT BUFFERS ARE GIVEN ONE OF WHICH WITH 1 BYTE LONG
AND ONE OF WHICH WITH BAD STATION ADDRESS
EXPECTED RESPONSES ARE:
STATUS = 360 FOR F6 THIRD BUFFER PASSED
 = 372 FOR F5 1 BYTE LONG BUFFER
 = 360 FOR F5 THIRD BUFFER PASSED
 = 1 FOR F5 FIRST BUFFER XMITTED
 = 1 FOR F5 SECOND BUFFER XMITTED
 = 1 FOR F6 SECOND BUFFER RECEIVED

PATTERN:

INCREMENTAL

REPORTS:

ERROR 10000	KMV11 FAILS TO RESET MASTER CLEAR
ERROR 10001	RUN FUNCTION NOT CORRECTLY PERFORMED
ERROR 10002	WRITE FUNCTION NOT CORRECTLY PERFORMED
ERROR 10003	READ FUNCTION NOT CORRECTLY PERFORMED
ERROR 10004	DATA COMPARE ERROR
ERROR 10005	UNEXPECTED INTERRUPT IN
ERROR 10006	UNEXPECTED INTERRUPT OUT
ERROR 10007	NO MORE INTERRUPT WHILE QIO PENDING
ERROR 10008	UNEXPECTED EPROM'S ON KMV
ERROR 10009	UNEXPECTED QIO RESPONSE
ERROR 10010	UNEXPECTED NUMBER OF RESPONSES RECEIVED
ERROR 10011	UNEXPECTED DATA RECEIVED
ERROR 10012	LOOP BACK NOT USED

1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503

8.0 ERROR INFORMATION

8.1 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 ERROR INFORMATION (EXTENDED ERROR INFORMATION CAN BE SUPPRESSED BY SETTING THE FLAG SWITCH IXE).

THE FOLLOWING EXAMPLE PROVIDE TYPICAL ERROR REPORT:

NKMCAO WRD ERR 10007 ON UNIT 00 TST 005 SUB 000 PC: 032164

NO-MORE INPUT INTERRUPTS WHILE QIO PENDING
NUMBER OF PENDING INPUTS = 3
NUMBER OF RESPONSES = 1

- ERROR REPORT LIST

8.1.1 ERROR REPORTS NOTICED INSIDE SUB ROUTINES

** ERROR REPORT 10000
KMV11 FAILS TO RESET MASTER CLEAR

	READ	EXPECTED
SELO	040000	000000

** - ERROR REPORT 10001
RUN FUNCTION NOT CORRECTLY PERFORMED

	READ	EXPECTED
SELO	040000	000000

** - ERROR REPORT 10002
WRITE FUNCTION NOT CORRECTLY PERFORMED

	READ	EXPECTED
SELO	040000	000000

** - ERROR REPORT 10003
READ FUNCTION NOT CORRECTLY PERFORMED

	READ	EXPECTED
SELO	040000	000000

** ERROR REPORT 10004
DATA COMPARE ERROR DURING APPLICATION CODE LOADING

	ADDRESS	READ	EXPECTED
1504	34000	127000	002737
1505			
1506			
1507	** ERROR REPORT 10005		
1508	UNEXPECTED INTERRUPT IN		
1509			
1510	SEL0	READ = 000000	
1511	SEL2	READ = 000000	
1512	SEL4	READ = 000000	
1513	SEL6	READ = 000000	
1514			
1515	NUMBER OF PENDING INPUTS = 06		
1516	NUMBER OF RESPONSES = 00		
1517			
1518	** - ERROR REPORT 10006		
1519	UNEXPECTED INTERRUPT OUT		
1520			
1521	SEL0	READ = 000000	
1522	SEL2	READ = 000000	
1523	SEL4	READ = 000000	
1524	SEL6	READ = 000000	
1525			
1526	NUMBER OF PENDING INPUTS = 06		
1527	NUMBER OF RESPONSES = 00		
1528			
1529	** - ERROR REPORT 10007		
1530	NO MORE INTERRUPT IN WHILE INPUTS ARE PENDING		
1531			
1532	NUMBER OF PENDING INPUTS = 06		
1533	NUMBER OF RESPONSES = 00		
1534			
1535	** - ERROR REPORT 10008		
1536	UNEXPECTED EPROM'S ECO LEVEL		
1537			
1538	OBTAINED ECO LEVEL = 000001		
1539	EXPECTED ONE = 000002		
1540			
1541	** - ERROR REPORT 10009		
1542	UNEXPECTED QIO RESPONSE		
1543			
1544	BAD QIO RESPONSE NUMBER = 2 (SPECIFY THE SEQUENCING NUMBER OF IT)		
1545			
1546	SEL2	READ = 000201	
1547	SEL4	READ = 000000	
1548	SEL4	READ = 000400	
1549			
1550	** - ERROR REPORT 10010		
1551	UNEXPECTED NUMBER OF RESPONSES RECEIVED		
1552			
1553	EXPECTED NUMBER OF RESPONSES = 3		
1554	NUMBER OF RESPONSES RECEIVED = 1		
1555			
1556	** ERROR REPORT 10011		
1557	UNEXPECTED DATA RECEIVED		
1558			
1559	TRANSMIT BUFFER ADDRESS : 005512		
1560	RECEIVE BUFFER ADDRESS : 002512		

1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590

RECORD SIZE : 100 (DECIMAL)
 BYTES IN ERROR : 88 (DECIMAL)

	ADDRESS	DATA
XMIT BUFFER:	005552	040
RCV BUFFER:	002552	000
XMIT BUFFER:	005553	041
RCV BUFFER:	002553	000

.... UP TO 8 ERROR ARE REPORTED ONLY

** - ERROR REPORT 10012
 THIS TEST IS SKIPPED BECAUSE THERE IS NO EXTERNAL LOOP

8.1.2 ERROR REPORT NOTICED INSIDE THE TEST:

** - ERROR REPORT 00001
 SELF TEST IS NOT CORRECTLY RUN

** - ERROR REPORT 00002
 RDYI NOT ASSERTED BY THE KMV AFTER AN RQI

SEL02 = 000000 EXPECTED VALUE = 000400

** - ERROR REPORT 00003
 RDYO NOT ASSERTED BY THE KMV FOR A VALID RESPONSE

SEL02 = 0000C0 EXPECTED VALUE = 000204

1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610

9.0 HISTORY

- VERSION 01, REVISION 00 NOVEMBER 1982

- CVKMCA => CNKMCA APRIL 9, 1984 JAKI BERG
CHANGES WERE MADE TO CVKMCA TO PRODUCE CNOMCA FOR THE FALCON-PLUS PROJECT
(SBC-11/21+). CHANGES, MARKED BY "JOB REV A-0", ARE:

- SET THE ODT BREAK VECTOR (LOCATION 140) TO THE STARTING ADDRESS OF FALCON'S ODT ROM (170000-OCTAL).
- REMOVE THE "IS THE HOST COMPUTER AN 11/23+ SYSTEM?" QUESTION.
- ADJUSTED WAIT MACRO FOR FALCON+ TIMING.
- CHANGED PRIORITY LEVEL 7 TO LEVEL 6 TO ALLOW BREAK KEY TO INTERRUPT.

Ⓢ

```
1620          .NLIST ME
1621          002000          . =2000
1622
1623
1624
1625
1626
1627          .MCALL SVC
1628 002000          SVC          ; INITIALIZE SUPERVISOR MACROS
1629
1630
1631
1632
1633
1634 002000          BGNMOD NKMCAO
1635
1636
1637          177777          $LSTIN= -1
1638          177777          $LSTTAG= -1
1639          177777          SVCINS= -1          ; LIST INSTRUCTIONS, SHIFTED RIGHT
1640          177777          SVCTST= -1          ; LIST TEST TAGS, SHIFTED RIGHT
1641          177777          SVCSUB= -1          ; LIST SUBTEST TAGS, SHIFTED RIGHT
1642          177777          SVCGBL= -1          ; LIST GLOBAL TAGS, SHIFTED RIGHT
1643          177777          SVCTAG= -1          ; LIST OTHER TAGS, SHIFTED RIGHT
1644
1645          ;          CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1646          ;          TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
1647          ;          SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
1648          ;          CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1649
1650
```



```
1652      .SBTTL PROGRAM HEADER
1653      ;**
1654      ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1655      ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1656      ;
1657
1658 002000          POINTER BGNDU,BGNSETUP
1659
1660
1661
1662
1679
1680 002000          HEADER NKMCA0,A,0.180.,0
1681
1692
```

1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1720
1721
1722
1723

002122

002122 000000
002124 177777
002126 177777

002130

; THIS TABLE IS USED BY THE RUNTIME SERVICES
; TO PROTECT THE LOAD MEDIA.

BGNPROT

0 ;OFFSET INTO P-TABLE FOR CSR ADDRESS
-1 ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
-1 ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

ENDPROT

1725
1726
1727
1728
1729
1730
1731
1732 002130
1733
1740
1741
1742
1743
1744

.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 12

```
1746      .SBTTL DEFAULT HARDWARE P-TABLE
1747
1748      ;////////////////////////////////////
1749      ;/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1750      ;/ THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
1751      ;/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P TABLE.
1752      ;/ AND IS USED AS A " TEMPLATE" FOR BUILDING THE P TABLE
1753      ;////////////////////////////////////
1754
1755      .ENABL  AMA      DFPTBL
1756 002162    BGNHW
1757
1758      .WORD 177000      ;KMV11 CSR ADDRESS
1759 002166    300         ;KMV11 VECTOR ADDRESS IN
1770 002170    4000      ;INTERRUPT PRIORITY LEVEL
1771 002172    1         ;TEST CONNECTOR INSTALLED FLAG
1772          :          ;WORD=1 FOR PDP11/23.      ;JB REV A 0
1773 002174    0         ;WORD=0 FOR NOT A PDP11/23.  ;JB REV A-0
1774
1775
1776 002176    ENDPHW
```

1778
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1807
1808
1823
1824 002176

.SBTTL GLOBAL EQUATES SECTION

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

.LIST MC
EQUALS

;
; BIT DEFINITIONS

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

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

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

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

;

1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853

000340
000300
000240
000200
000140
000100
000040
000000

000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000

000000
000002
000004
000006
000010
000012
000014
000016

000000
000001
000002
000003
000004
000005
000006
000007

000010

;
; PRIORITY LEVEL DEFINITIONS

;
PRI07== 340
PRI06== 300
PRI05== 240
PRI04== 200
PRI03== 140
PRI02== 100
PRI01== 40
PRI00== 0

;
; OPERATOR FLAG BITS

;
EVL== 4
LOT== 10
ADR== 20
IDU== 40
ISR== 100
UAM== 200
BOE== 400
PNT== 1000
PRI== 2000
IXE== 4000
IBE== 10000
IER== 20000
LOE== 40000
MOE== 100000

.NLIST ME

;;
;* PROGRAM DIRECT ASSIGNMENTS
;;

;
; CSR OFFSETS DEFINITIONS

;
SEL0 = 0
SEL2 = SEL0+2
SEL4 = SEL2+2
SEL6 = SEL4+2
SEL10 = SEL6+2
SEL12 = SEL10+2
SEL14 = SEL12+2
SEL16 = SEL14+2

BSEL0 = 0
BSEL1 = BSEL0+1
BSEL2 = BSEL1+1
BSEL3 = BSEL2+1
BSEL4 = BSEL3+1
BSEL5 = BSEL4+1
BSEL6 = BSEL5+1
BSEL7 = BSEL6+1

CSRLEN = 8.

;LENGTH OF CSR IN WORDS

```

1854
1855
1856      ;
1857      ; M.M.U. REGISTERS
1858      ;
1859      172340      PAR0      = 172340      ;PAGE ADDRESS REGISTER 0
1860      172356      PAR7      = 172356      ; " " " " 7
1861      172300      PDRO      = PAR0 40      ;PAGE DESCRIPTOR REGISTER 0
1862      077406      PDDEF     = 77406       ;PLF-128 ED-UP ALP-R/W
1863
1864      177572      SRO       = 177572      ;STATUS REGISTER 0
1865      172516      SR3       = 172516      ; " " " 3
1866
1867      ;
1868      ; SELO BIT DEFINITIONS
1869      ;
1870
1871      100000      RUN       = BIT15      ;TO RUN APPLICATION FIRMWARE OR, IF
1872                                     ;MCLR SET, TO RUN SELF-TEST
1873      040000      MCLR      = BIT14      ;MASTER CLEAR
1874      020000      WRITE    = BIT13      ;TO LOAD APPLICATION CODE
1875      010000      MODE     = BIT12      ;RESERVED
1876      002000      READ     = BIT10      ;TO UNLOAD APPLICATION CODE
1877      001000      LOOP     = BIT09      ;TO SELECT INTERNAL LOOP
1878      000400      ERR      = BIT08      ;LOAD/UNLOAD ILLEGAL ADDRESS ERROR
1879      000200      RQI      = BIT07      ;TO REQUEST CSR TRANSACTION
1880      000020      IE0      = BIT04      ;INTERRUPT ENABLE OUT
1881      000001      IEI      = BIT00      ;INTERRUPT ENABLE IN
1882
1883      ;
1884      ; BSEL2 BIT DEFINITIONS
1885      ;
1886
1887      000200      RDYO      = BIT07      ;READY OUT - SET BY KMV11
1888      000020      RDYI      = BIT04      ;READY IN   SET BY KMV11
1889
1890      ;
1891      ; KMV11 A COMMAND AND RESPONSE CODES
1892      ;
1893
1894      000000      F0        = 0          ;DUMMY COMMAND
1895      000001      F1        = 1          ;CONFIGURATE
1896      000002      F2        = 2          ;DECONFIGURATE
1897      000003      F3        = 3          ;DUMMY COMMAND
1898      000004      F4        = 4          ;DUMMY COMMAND
1899      000005      F5        = 5          ;TRANSMIT
1900      000006      F6        = 6          ;RECEIVE
1901      000007      F7        = 7          ;KILL TRANSMIT
1902      000010      F10       = 10         ;KILL RECEIVE
1903      000011      F11       = 11         ;DUMMY COMMAND
1904      000012      F12       = 12         ;DUMMY COMMAND
1905      000013      F13       = 13         ;MODEM CHANGE RESPONSE
1906      000014      F14       = 14         ;READ MODEM LEADS
1907      000015      F15       = 15         ;DUMMY
1908      000016      F16       = 16         ;START MODEM SURVEY
1909      000017      F17       = 17         ;STOP MODEM SURVEY
1910

```

```

1911
1912
1913
1914
1915      000001      SUCCES      = 1      ;COMMAND CORRECTLY PERFORMED
1916      000374      NXMEM      = 374      ;NON EXISTENT BUFFER MEMORY
1917      000373      BUFOVF     = 373      ;BUFFER OVERFLOW
1918      000372      FRMLER     = 372      ;FRAME LENGTH ERROR
1919      000371      OUTSEQ     = 371      ;OUT OF SEQUENCE
1920      000370      LTCYER     = 370      ;LATENCY ERROR
1921      000367      FCSER      = 367      ;FCS ERROR
1922      000365      MODDOWN    = 365      ;MODEM DOWN
1923      000364      RXABO      = 364      ;ABORT RECEIVED
1924      000363      DBLCMD     = 363      ;DOUBLE COMMAND
1925      000360      TOOBUF     = 360      ;MORE THAN TWO BUFFERS
1926      000357      KMVER      = 357      ;KMV RESSOURCE ERROR
1927      000356      DCONFP     = 356      ;DECONFIGURATE PENDING
1928
1929
1930      ; DEC STD 52 STATE OF THE LINE (FROM KMV)
1931
1932
1933      000000      S1          = 0      ;OFF LINE
1934      000001      S3          = 1      ;LINE ENABLE
1935      000002      S5          = 2      ;PENDING START
1936      000003      S6          = 3      ;DATA EXCHANGE
1937      000004      S6A         = 4      ;LINE FAILURE
1938      000005      S7A         = 5      ;LINE DISCONNECT
1939      000006      S7B         = 6      ;DECONFIGURATE
1940
1941
1942      READ MODEM LEADS
1943
1944
1945      000001      ..09        = BIT00    ;(CD)DATA CHANNEL RECEIVED LINE
1946
1947      000002      S125        = BIT01    ;SIGNAL DETECTOR
1948      000004      S107        = BIT02    ;(RI)CALLING INDICATOR
1949      000010      S106        = BIT03    ;(DSR)DATA SET READY
1950      000020      S112        = BIT04    ;(CTS)READY FOR SENDING
1951      000040      S142        = BIT05    ;DATA SIGNAL RATE SELECTOR
1952
1953
1954      ; DEFINITIONS FOR APPLICATION FIRMWARE LOADING
1955
1956
1957      030306      $START      = COMEXE    ;START ADDRESS TO RUN FIRMWARE
1958      030306      $RAM        = $BUFF    ;FIRST RAM ADDRESS TO LOAD FIRMWARE
1959      160000      $$STR       = 160000   ;FIRST PROM ADDRESS IN KMV
1960      160002      ECONB      = $$STR+2  ;KMV FIRMWARE ECO LEVEL ADDRESS
1961      160004      PARTNB     = $$STR+4  ;KMV PROM NUMBER
1962      000400      $RDBS      = 256.     ;LENGTH OF RDB'S
1963      000017      $RDBC      = 15.      ;NUMBER OF RDB'S
1964
1965
1966      ; CONFIGURATE PARAMETERS
1967      ; LOADED IN SEL4

```


1968				
1969				
1970	000001	SDLC	= BIT00	;BOP SDLC SELECTED
1971				
1972	000010	S140	= BIT03	;REMOTE LOOP BACK REQUEST
1973	000020	S141	= BIT04	;LOCAL LOOP REQUEST
1974	000004	S111	= BIT02	;DATA SIGNALING RATE SELECTOR
1975				
1976	000040	DLO	= BIT05	;DATA LEADS ONLY
1977				
1978	000200	ADSRCH	= BIT07	;ONLY FRAMES WITH GLOBAL ADDRESS OR
1979				;IF ADDRESS MATCH FOUND WILL BE PASSED
1980				
1981	100000	CLKDTE	= BIT15	;CLOCK SOURCE INTERNAL
1982				
1983	000000	C1200	= 0	;1.2K
1984	000400	C2400	= BIT08	;2.4K
1985	001000	C4800	= BIT09	;4.8K
1986	001400	C9600	= BIT08!BIT09	;9.6K
1987	002000	C19K	= BIT10	;19.2K
1988	002400	C48K	= BIT10!BIT08	;48K
1989	003000	C56K	= BIT10!BIT09	;56K
1990	003400	C64K	= BIT10!BIT09!BIT08	;72K
1991				

1993
1994
1995
1996
1997
1998
1999
2000
2006
2007
2008
2009
2010
2011 002176
2012
2013
2014
2027
2028 002236
002236 000000
002240 000000
002242 000000
002244 000000
2029
2030
2031
2032
2033
2034 002246 000000
2035 002250 000000
2036
2037 002252 000000
2038 002254 000000
2039 002256 000000
2040
2041 002260 000000
2042 002262 000005
2043 002264 000000
2044 002266 000000
2045 002270 000015
2046 002272 000000
2047
2048
2049
2050
2051
2052 002274 000000
2053 002276 000000
2054 002300 000000
2055 002302 000000
2056 002304 000000
2057 002306 000000
2058 002310 000000
2059 002312 000000
2060 002314 000000
2061 002316 000000
2062 002320 000000

.SBTTL GLOBAL DATA SECTION

```

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

```

```

;*****
;* STORAGE FOR DEVICE REGISTERS
;*****
DESCRIPT <KMV11-A FUNCTIONAL DIAGNOSTIC>

```

```

ERRTBL
ERRTYP: .WORD 0
ERRNBR: .WORD 0
ERRMSG: .WORD 0
ERRBLK: .WORD 0

```

```

;*****
;* PROGRAM CONTROL DEFINITIONS
;*****

```

```

LOGDEV: 0 ;LOGICAL DEVICE NUMBER
PSTACK: 0 ;BASE LEVEL PROGRAM STACK POINTER

```

```

SAVE4: 0
SAVE6: 0
FTIME: 0 ;FIRST PASS FLAG

```

```

LOCK: 0 ;ADDRESS FOR LOCK CURRENT DATA
MAXERR: 5 ;MAX ERROR BEFORE DROPPING THE UNIT
ERCNTR: 0 ;UNIT ERROR COUNTER
L$W: 0
L$UIT: 15 ;MAX NUMBER OF UNIT
UNIT: 0

```

```

;*****
;* MISCELLANEOUS STORAGE
;*****

```

```

GOOD: 0 ;POINT TO GOOD DATA
GOOD0: 0
GOOD2: 0
GOOD4: 0
GOOD6: 0
GOOD10: 0
GOOD12: 0
GOOD14: 0
GOOD16: 0
BAD: 0
BADO: 0 ;POINT TO BAD DATA

```

2063	002322	000000	BAD2:	0	
2064	002324	000000	BAD4:	0	
2065	002326	000000	BAD6:	0	
2066	002330	000000	BAD10:	0	
2067	002332	000000	BAD12:	0	
2068	002334	000000	BAD14:	0	
2069	002336	000000	BAD16:	0	
2070	002340	000000	DATA:	0	; POINTER TO CURRENT PATTERN
2071	002342	000000	RATE:	0	; POINT TO THE CURRENT LINE CLOCK RATE
2072	002344	000000	LLOOP:	0	; LOCAL LOOP FLAG
2073	002346	000000	LENGTH:	0	; LENGTH OF BUFFER FOR DMA
2074	002350	000000	REG0:	0	; FOR GENERAL REGISTER SAVING
2075	002352	000000	REG1:	0	
2076	002354	000000	REG2:	0	
2077	002356	000000	REG3:	0	
2078	002360	000000	REG4:	0	
2079	002362	000000	REG5:	0	
2080	002364	000000	REG6:	0	
2081	002366	000000	REG7:	0	
2082	002370	000000	ERRFLG:	0	; ERROR FLAGGED UNDER IN/OUT INTERRUPTS
2083	002372	000000	REQCNT:	0	; INPUT COMMAND COUNT
2084	002374	000000	TEMP:	0.0	; TEMPORARY STORAGE
2085	002400	000000	LSTLGH:	0	; EXPECTED RESPONSE LIST LENGTH
2086	002402	000000	RSPCNT:	0	; OUTPUT RESPONSE COUNT
2087	002404	000000	TMOUT:	0	; CURRENT TIME-OUT VALUE
2088	002406	000000	INLST:	0	; POINT TO THE CURRENT INPUT LIST
2089	002410	000000	OUTLST:	0	; POINT TO THE CURRENT OUTPUT LIST
2090	002412	000000	S.LOAD:	0	; STATUS FOR APPLICATION CODE LOADING
2091	002414	000000	APPFLG:	0	; LOADING STATE OF THE APPLICATION CODE
2092	002416	000000	SELNUM:	0	; LOAD WITH CSR OFFSET
2093	002420	000000	XMTADD:	0	; XMIT BUFFER ADDRESS UNDER TEST
2094	002422	000000	RCVADD:	0	; RECEIVE BUFFER ADDRESS UNDER TEST
2095	002424	000000	XMTBUF:	0	; XMIT BUFFER
2096	002426	000000	RCVBUF:	0	; RECEIVE BUFFER
2097	002430	000000	ERRCNT:	0	; BYTES IN ERROR
2098	002432	000000	BADLOC:	.BLKW 18.	; LOAD WITH ADDRESS OF UN-CORRECT DATA
2099	002476	000000	OUTNUM:	0	; CURRENT BAD QIO RESPONSE
2100					
2101	002500	000000	LSTAD:	0	; LAST MEMORY ADDRESS(15-0)
2102	002502	000000	LSTBK:	0	; LAST MEMORY ADDRESS(21-16)
2103					
2104					;*****
2105					;* POINTERS TO KHV11 VECTORS AND REGISTERS
2106					;*****
2107					
2108	002504	000000	KMVCSR:	0	; CSR ADDRESS
2109	002506	000000	INTIN:	0	; POINTER TO INTERRUPT VECTOR XX0
2110	002510	000000	INTOUT:	0	; POINTER TO INTERRUPT VECTOR XX4
2111	002512	000000	PRILEV:	0	; INTERRUPT SERVICING RELATIVE LEVEL
2112	002514	000000	PDPTYP:	0	; PDP11/23+ FLAG
2113					
2114					;*****
2115					;* PROGRAM CONTROL FLAGS
2116					;*****
2117					
2118	002516	000	INIFLG:	.BYTE 0	
2119				.EVEN	

```

2120 002520      000      LOKFLG: .BYTE  0
2121 002521      000      QV.FLG:  .BYTE  0
2122                                     .EVEN
2123 002522  000000      UUT:      .WORD  0
2124
2125                                     ;*****
2126                                     ;* DATA STORAGE
2127                                     ;*****
2128
2129 002524      045      116      045  MPROM:  .ASCII  /#N#A  EPROM PART NUMBER:  /
      002527      101      040      040
      002532      105      120      122
      002535      117      115      040
      002540      120      101      122
      002543      124      040      116
      002546      125      115      102
      002551      105      122      072
      002554      040      040
2130 002556                                     PROMNB: .BLKB  11.      ;KMV PROM NUMBER TO PRINT
2131 002571      045      116      000      .ASCIZ  /#N/
2132                                     .EVEN
2133
2134 002574                                     RXBUF0: .BLKB  256.      ;RX BUFFERS FOR DMA
2135 003174                                     RXBUF1: .BLKB  256.
2136 003574                                     RXBUF2: .BLKB  512.
2137
2138 004574                                     TXBUF0: .BLKB  256.      ;TX BUFFERS FOR DMA
2139 005174                                     TXBUF1: .BLKB  256.
2140 005574                                     TXBUF2: .BLKB  512.
2141
2142 006574                                     RSPOKE: .BLKW  10.      ;UP TO 20. RESPONSES ARE POSSIBLE. RSPOKE
2143                                     ;IS USED TO FLAG EXPECTED RESPONSES DURING
2144                                     ;TEST
2145
2146 006620                                     OUTBUF: .BLKW  3*20.      ;TO SAVE QIO RESPONSES
2147
2148                                     ;*****
2149                                     ;* PATTERN TABLE
2150                                     ;*****
2151
2152 007010  007012      PATTAB: PATTRN      ;TABLE OF DATA FOR TEST
2153 007012  000000      PATTRN: 000000
2154 007014  177777      177777
2155 007016  052525      052525
2156 007020  125252      125252
2157 007022  177776      177776
2158 007024  177775      177775
2159 007026  177773      177773
2160 007030  177767      177767
2161 007032  177757      177757
2162 007034  177737      177737
2163 007036  177677      177677
2164 007040  177577      177577
2165 007042  177377      177377
2166 007044  176777      176777
2167 007046  175777      175777
2168 007050  173777      173777

```

2169	007052	167777	167777
2170	007054	157777	157777
2171	007056	137777	137777
2172	007060	077777	077777
2173	007062	100000	100000
2174	007064	040000	040000
2175	007066	020000	020000
2176	007070	010000	010000
2177	007072	004000	004000
2178	007074	002000	002000
2179	007076	001000	001000
2180	007100	000400	000400
2181	007102	000200	000200
2182	007104	000100	000100
2183	007106	000040	000040
2184	007110	000020	000020
2185	007112	000010	000010
2186	007114	000004	000004
2187	007116	000002	000002
2188	007120	000001	000001
2189	007122	000000	000000
2190	007124	000112	

PATLGH: .-PATRN

```

;*****
;* LOCATION 'GDREV' MUST BE LOADED WITH THE PROM VERSION NUMBER THAT IS
;* COMPATIBLE WITH THIS DIAG.
;*****

```

2191			
2192			
2193			
2194			
2195			
2196			
2197	007126	000001	GDREV: .WORD 1

2199
2200
2201 007130
2202
2203
2204
2205
2206
2207 007130
2208 007330

REGADR:

;* STACK USED FOR SUBROUTINE LINKAGE

.BLKW 100

SSTACK:

2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2233
2234
2235
2236
2237

007330

.SBTTL GLOBAL TEXT SECTION

```

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

;*****
;# NAMES OF DEVICES SUPPORTED BY PROGRAM
;*****
DEV TYP <M7500>

```

```

2239      .SBTTL GLOBAL SUBROUTINES
2240
2241
2242
2243
2244      ;////////////////////////////////////
2245      ;/      THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
2246      ;////////////////////////////////////
2278
2320
2321
2322
2323
2324
2325
2326
2327      ;*****
2328      ;* LIST OF THE MACRO CALLS AND ROUTINES
2329      ;*
2330      ;* -TO SAVE GENERAL REGISTERS           : SAVE  012...
2331      ;* -TO RESTORE GENERAL REGISTERS      : GET   012...
2332      ;* -TO SHIFT LEFT N LOCATIONS         : ASHL  A,N
2333      ;* -TO SHIFT RIGHT N LOCATIONS        : ASHR  A,N
2334      ;* -TO WAIT N TIMES 100 MICROS        : WAIT  N
2335      ;*
2336      ;* -TO WRITE DATA IN KMV RAM          : CALL  WDATA
2337      ;* -TO READ DATA FROM KMV RAM         : CALL  RDATA
2338      ;* -TO CLEAR A BUFFER                  : CALL  CLEAR
2339      ;* -TO SIZE MEMORY                     : CALL  ..SIZE
2340      ;* -TO COMPARE TWO BUFFERS             : CALL  COMPAR
2341      ;* -TO CHECK QIO RESPONSES            : CALL  CHKRSP
2342      ;* -TO COPY FROM ONE BUFFER TO ANO YER : CALL  COPY
2343      ;* -TO ENTER APPLICATION MODE         : CALL  MODE0
2344      ;* -TO LOAD A BUFFER WITH INCREMENTAL PATTERNS : CALL  BUFLD
2345      ;* -TO INIT QIO                       : CALL  INIQIO
2346      ;* -TO PROCESS QIO                    : CALL  QIOP
2347      ;* -TO CHECK KMV PROM NUMBER          : CALL  REVCHK
2348      ;* -TO LOAD AND CHECK APPLICATION CODE : CALL  LDAPPL
2349      ;* -TO START APPLICATION              : CALL  RUNAPP
2350      ;*
2351      ;* -TO SERVE IN INTERRUPT             : ITIN
2352      ;* -TO SERVE OUT INTERRUPT           : ITOUT
2353      ;*
2354      ;*****

```


2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379

```

; **
; MACRO TO SAVE GENERAL REGISTERS
;
; DESCRIPTION: PERMITS TO SAVE GENERAL REGITERS R0 TO R7
;
; CALLING SEQUENCE: SAVE      123...
;
; INPUTS: REGISTER NUMEROS LIST
;
; OUTPUTS: REG(N)=R(N)
;
; CAUTIONS: NONE
; --
```

```

.MACRO SAVE      ARLST
      .IRPC      N,<ARLST>
      MOV       R'N,REG'N
      .ENDR
.ENDM
```

2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402

```

;
; **
; MACRO TO RESTORE GENERAL REGISTERS
;
; DESCRIPTION: PERMITS TO RESTORE GENERAL REGISTERS R0 TO R7
;
; CALLING SEQUENCE: GET      123....
;
; INPUTS: REGISTER NUMEROUS LIST
;
; OUTPUTS: NONE
;
; CAUTIONS: NONE
; **

```

```

.MACRO GET      ARGV1 ARGV2
      .IRPC N, <ARGV1>
      MOV     REG'N,R'N
      .ENDR
.ENDM

```

2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423

```
***  
; MACRO TO SHIFT N BITS ON THE LEFT  
;  
; CALLING SEQUENCE: ASHL A,N  
;  
; INPUT PARAMETERS:  
; N: COUNT OF BITS TO BE SHIFTED  
; A: ARGUMENT TO BE SHIFTED  
;  
;--
```

```
.MACRO ASHL A,N,?B  
MOV @N,RO  
B: ASL A  
SOB RO,B  
.ENDM
```

2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444

```

; **
; MACRO TO SHIFT N BITS ON THE RIGHT
;
; CALLING SEQUENCE: ASHR  A,N
;
; INPUT PARAMETERS:
; N: COUNT OF BITS TO BE SHIFTED
; A: ARGUMENT TO BE SHIFTED
;
; --

```

```

.MACRO ASHR  A,N,?B
MOV  @N,RO
B:   ASR  A
SOB  RO,B
.ENDM

```

2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463

```
***  
; MACRO TO FILL SPACE WITH NOP  
;  
; CALL ING SEQUENCE : CHECK  
;  
      .MACRO CHECK  
      .REPT 30000  
$$$ADD * -0  
      .IF GT <30000-$$$ADD>  
      NOP  
      .IFF  
      .MEXIT  
      .ENDC  
      .E OR  
      .ENDM
```

```

2465
2466
2467
2468 ;**
2469 ; MACRO FOR SIMPLE WAIT
2470 ;
2471 ; CALLING SEQUENCE : WAIT N.
2472 ; WHERE N INDICATES THE TIME LENGTH
2473 ; (TIME UNIT IS 100 MICROS)
2474 ;
2475 ; INPUTS: NONE
2476 ;
2477 ; OUTPUTS: RO DESTROYED
2478 ;
2479 ; -
2480
2481 .MACRO WAIT N
2482 CALL .WAIT
2483 .WORD N
2484
2485 .ENDM
2486 007336 .WAIT: SAVE 12
2487 007346 011601 MOV (SP),R1 ;GET LENGTH
2488 007350 012102 MOV (R1)+,R2
2489 007352 010116 MOV R1,(SP) ;UP DATE RETURN PC
2490 007354
2491 007354
2492
2493 007356 012701 000044 ; BREAK
2494 007362 077101 ; MOV #60.,R1 ;WAIT 100 MICROS ;JB REV A-0
2495 ; MOV #36.,R1 ;WAIT 100 MICROS ;JB REV A 0
2496 007364 077205 ; SOB R1,2#
2497 ; SOB R2,1# ;WAIT FOR N*100
2498 007366
2499 007376 000207 GET 12
2500 RETURN

```

```

2502      ;**
2503      ; ROUTINE TO READ A DATA FROM KMV RAM
2504      ;
2505      ; INPUTS: R3 = RAM ADDRESS
2506      ;          R5 = CSR BASE ADDRESS
2507      ;
2508      ; OUTPUTS: SEL6 = DATA
2509      ;           CARRY SET IF ANY ERROR
2510      ;
2511      ; REPORT: ERROR 10003          KMV FAILS TO READ
2512      ;
2513      ;--
2514
2515
2516      RDATA:
2517      007400      SAVE      1
2518      007400      MOV      R3,SEL4(R5)          ;LOAD RAM ADDRESS
2519      007410      010365  000004  000000      BIS      #READ,SELO(R5)      ;READ WORD
2520      007416      052765  002000  000000      MOV      #10.,R1          ;INIT TIME-OUT
2521      007422      012701  000012
2522      007422      11$:
2523      007430      032765  002400  000000      WAIT     10.          ;WAIT FOR 1 MS
2524      007436      001417      BIT      #READ!ERR,SELO(R5)      ;READ CORRECTLY DONE?
2525      007440      077110      BEQ     12$          ;YES, COMPARE
2526      007442      005037  002276  000000      SOB     R1,11$      ;NOT, LOOP TILL TIME OUT
2527      007446      016537  000000  002320      CLR     GOOD0          ;SET REPORT
2528      007454      MOV     SELO(R5),BADO
2529      007464      ERRHRD  10003,E10003,PRSELO      ;IF TIME-OUT
2530      007472      WAIT     10000.
2531      007474      SEC
2532      007476      BR      13$
2533
2534      007476      000241      12$:
2535      007500      000207      13$:
2536      007504      CLC
2537      007504      GET      1
2537      007504      RETURN

```

```

2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555 007506
2556 007506
2557 007512 010365 000004
2558 007516 011465 000006
2559 007522 052765 020000 000000
2560 007530 012701 000012
2561 007534
2562 007534
2563 007542 032765 020400 000000
2564 007550 001417
2565 007552 077110
2566
2567 007554 005037 002276
2568 007560 016537 000000 002320
2569 007566
2570 007576
2571 007604 000261
2572 007606 000401
2573
2574 007610 000241
2575 007612
2576 007616 000207

; **
; ROUTINE TO WRITE DATA IN KMY RAM
;
; INPUTS: R3 = RAM ADDRESS
;         R4 POINTS TO THE DATA
;         R5 = CSR BASE ADDRESS
;
; OUTPUTS: CARRY SET IF ANY ERROR
;
; REPORT: ERROR 10002          KMY FAILS TO WRITE
;
; --

WDATA:
        SAVE      1
        MOV       R3,SEL4(R5)          ;LOAD RAM ADDRESS
        MOV       (R4),SEL6(R5)        ;LOAD WORD
        BIS       @WRITE,SELO(R5)      ;WRITE IT
        MOV       @10.,R1              ;INIT TIME OUT
3$:
        WAIT     10.                    ;WAIT FOR 1 MS
        BIT       @WRITE!ERR,SELO(R5)  ;IS IT CORRECTLY WRITE?
        BEQ      20$                     ;YES,NEXT WORD
        SOB      R1,3$                   ;NOT, LOOP TILL TIME OUT
        CLR      GOODO                   ;SET REPORT
        MOV      SELO(R5),BADO
        ERHRD    10002,E10002,PRSELO    ;IF TIME OUT
        WAIT    10000.
        SEC
        BR       21$
20$:
        CLC
21$:
        GET      1
        RETURN
    
```



```

2578
2579
2580
2581
2582 ; **
2583 ; ROUTINE TO SIZE MEMORY
2584 ;
2585 ; NOTE: THIS ROUTINE MUST RESIDE WITHIN THE
2586 ; FIRST 24K OF MEMORY.
2587 ;
2588 ; OUTPUTS:
2589 ; LSTAD WILL CONTAIN THE LAST AVAILABLE MEMORY LOCATION (BITS 15-0)
2590 ; LSTBK WILL CONTAIN THE LAST AVAILABLE MEMORY LOCATION (BITS 21-16)
2591 ;
2592 ; --
2593
2594
2595
2596 007620 ..SIZE: SAVE 0123
2597 007640 010637 002364 MOV SP,REG6 ;SAVE CURRENT SP
2598 007644 012737 010004 000004 MOV #6$,4 ;SET TIME-OUT TRAP
2599 007652 106737 000006 MFPS 6 ;SAVE THE CURRENT PSW
2600 007656 012701 003776 MOV #3776,R1 ;INIT MEMORY ADDRESS POINTER
2601
2602 ; CHECK IF M.M.U. INSTALLED
2603
2604 007662 005737 177572 TST SRO ;IS IT?
2605
2606 ; YES, SIZE THE MEMORY BY USING THE M.M.
2607
2608 007666 005046 CLR -(SP) ;INITIALIZE FOR "PAR" LOADING
2609 007670 012702 172340 MOV #PAR0,R2 ;ADDRESS OF FIRST "PAR"
2610 007674 012703 000010 MOV #+D8,R3 ;LOAD EIGHT "PAR.'S AND EIGHT "PDR.'S"
2611 007700 012762 077406 177740 1$: MOV #PDDEF,-40(R2) ;SET PDR'S
2612 007706 011622 MOV (SP),(R2) ;SET PAR'S
2613 007710 062716 000200 ADD #200,(SP) ;UPDATE FOR NEXT "PAR"
2614 007714 077307 SOB R3,1$ ;LOOP UNTIL ALL EIGHT ARE LOADED
2615 007716 012742 177600 MOV #177600,(R2) ;SETUP PAR7 FOR I/O
2616 007722 005042 CLR -(R2) ;SETUP PAR6 FOR TESTING
2617 007724 012737 007742 000004 MOV #2$,4 ;CATCH TIMEOUT IF NO SR3
2618 007732 012737 000020 172516 MOV #20,SR3 ;ENABLE 22 BIT MODE
2619 007740 000401 BR 3$ ;THIS PDP-11 HAS A SR3 REGISTER
2620
2621 007742 022626 2$: CMP (SP),.(SP) ;CLEAN OFF THE STACK -NO SR3
2622
2623 007744 005237 177572 3$: INC SRO ;TURN ON MEMORY MANAGEMENT
2624 007750 012737 007774 000004 MOV #5$,4 ;SET FOR TIME OUT
2625 007756 005737 143776 4$: TST 143776 ;TRAP ON NON-EX-MEM
2626 007762 062712 000040 ADD #40,(R2) ;MAKE A 1K STEP
2627 007766 023712 172356 CMP PAR7,(R2) ;LAST ONE?
2628 007772 101371 BHI 4$ ;NO--TRY AGAIN
2629
2630 007774 011202 5$: MOV (R2),R2 ;GET LAST BANK*1
2631 007776 005037 177572 CLR SRO ;TURN OFF MEMORY MANAGEMENT
2632 010002 000416 BR 9$
2633
2634 ; SIZE MEMORY WITHOUT M.M.

```

```

2635
2636 010004 012737 010034 000004 6$:   MOV    #8$,4           ;SET FOR TIMEOUT
2637 010012 005002                CLR    R2             ;SET UP BANK
2638 010014 062701 004000          7$:   ADD    #4000,R1      ;INCREMENT BY 1K
2639 010020 062702 000040                ADD    #40,R2        ;1K STEP
2640 010024 005711                TST    (R1)          ;TRAP ON TIME OUT
2641 010026 022701 177776          CMP    #177776,R1    ;LAST ONE
2642 010032 001370                BNE    7$            ;NO TRY AGAIN
2643
2644
2645                ; TERMINATE SIZING
2646 010034 162701 004000          8$:   SUB    #4000,R1
2647
2648 010040 162702 000040          9$:   SUB    #40,R2           ;DROP BACK
2649 010044 013706 002364                MOV    REG6,SP       ;RESTORE SP
2650 010050 013737 002252 000004    MOV    SAVE4,4       ;RESTORE ERROR VECTOR
2651 010056 013737 002254 000006    MOV    SAVE6,6
2652 010064 010137 002500                MOV    R1,LSTAD     ;LAST ADDRESS (BITS 10-0)
2653 010070 000241                CLC
2654 010072 006002                ROR    R2            ;ROTATE BANK
2655 010074 006002                ROR    R2
2656 010076 150237 002501          BISB  R2,LSTAD+1     ;LAST ADDRESS (BITS 15 11)
2657 010102 105002                CLRB  R2             ;CLEAR BITS 15 11
2658 010104 000302                SWAB  R2            ;SWAP R2
2659 010106 010237 002502          MOV    R2,LSTBK     ;LAST ADDRESS (BITS 21-16)
2660 010112                GET    0123         ;RESTORE REGISTERS
2661 010132 000207                RETURN
2662
2663

```

```

2665
2666
2667 ; **
2668 ; ROUTINE TO CHECK QIO RESPONSES
2669 ;
2670 ; DESCRIPTION: SUBSEQUENT RESPONSES(SEL2,SEL4,SEL6) ARE
2671 ; COMPARED TO THE EXPECTED ONES
2672 ;
2673 ; INPUTS: R1 EXPECTED OUT LIST LENGTH
2674 ; R1+2 EXPECTED OUT LIST BASE ADDRESS
2675 ; OUTBUF RECEIVED OUT LIST ADDRESS
2676 ; RSPCNT FOR THE NUMBER OF RESPONSES
2677 ;
2678 ; OUTPUTS: R0 TO R4 ARE DESTROYED
2679 ; CARRY BIT CLEARED IF TEST OKE
2680 ; CARRY SET IF NOT, IN THIS CASE,
2681 ; BAD2,4,6 GIVES UN-EXPECTED RESPONSES
2682 ; OUTNUM = THE SEQUENTIAL NUMBER OF UN-EXPECTED RESPONSE
2683 ;
2684 ; REPORTS: ERROR 10009 NOT EXPECTED RESPONSE RECEIVED
2685 ; ERROR 10010 NOT EXPECTED NUMBER OF RESPONSES RECEIVED
2686 ; --
2687
2688 010134
2689 010134 012137 002400
2690 010140
2691 010144 012703 006620
2692 010150 005037 002476
2693 010154 005002
2694
2695 010156 023737 002402 002400
2696 010164 001407
2697
2698 010166 013737 002400 002476
2699 010174
2700
2701 010204 011304
2702 010206 116300 000005
2703
2704 010212 012337 002322
2705 010216 012337 002324
2706 010222 012337 002326
2707 010226
2708
2709 010232 042704 177760
2710 010236 000241
2711 010240 006304
2712 010242 016404 010554
2713
2714 010246 122700 000001
2715 010252 001444
2716 010254 122700 000374
2717 010260 001440
2718 010262 122700 000373
2719 010266 001434
2720 010270 122700 000372
2721 010274 001430

```

```

CHKRSP:
MOV (R1)+,LSTLGH ;GET OUT LIST LENGTH
SAVE 1 ;SAVE LIST POINTER
MOV #OUTBUF,R3 ;POINT TO OUTLIST
CLR OUTNUM ;CLEAR BAD RESPONSE NUMBER POINTER
CLR R2 ;INIT RESPONSE COUNTER

CMP RSPCNT,LSTLGH ;EXPECTED NUMBER OF RESPONSES?
BEQ 1$ ;IF YES

MOV LSTLGH,OUTNUM ;IF NOT, SET REPORT
ERRHRD 10010,E10010,PRBNUM

1$:
MOV (R3),R4 ;GET RCV RESPONSE ID
MOVB 5(R3),R0 ;GET RCV STATUS

MOV (R3)+,BAD2 ;GET COMPLETE RESPONSE BEFORE
MOV (R3)+,BAD2+2 ;ANY REPORT
MOV (R3)+,BAD2+4
SAVE 3 ;SAVE POINTER

BIC #177760,R4 ;KEEP FXX ONLY
CLC
ASL R4 ;ADJUST FOR WORD BOUNDARY
MOV RSPLST(R4),R4 ;POINT TO THE RIGHT FUNCTION TEST

CMPB #SUCCES,R0 ;THEN LOOK FOR THE RIGHT TEST
BEQ 10$ ;ACCORDING TO THE EXPECTED STATUS
CMPB #NXMEM,R0
BEQ 9$
CMPB #BUFOVF,R0
BEQ 8$
CMPB #FRMLER,R0
BEQ 7$

```

```

2722 010276 122700 000371      CMPB    #OUTSEQ,RO
2723 010302 001424      BEQ     6#
2724 010304 122700 000364      CMPB    #RXABO,RO
2725 010310 001420      BEQ     5#
2726 010312 122700 000363      CMPB    #DBLCMD,RO
2727 010316 001414      BEQ     4#
2728 010320 122700 000360      CMPB    #TOOBUF,RO
2729 010324 001410      BEQ     3#
2730 010326 122700 000357      CMPB    #KMVER,RO
2731 010332 001404      BEQ     2#
2732 010334 122700 000356      CMPB    #DCONFP,RO
2733 010340 001055      BNE     21#
2734
2735 010342 005724      TST     (R4),
2736 010344 005724      2#:    TST     (R4),
2737 010346 005724      3#:    TST     (R4),
2738 010350 005724      4#:    TST     (R4),
2739 010352 005724      5#:    TST     (R4),
2740 010354 005724      6#:    TST     (R4),
2741 010356 005724      7#:    TST     (R4),
2742 010360 005724      8#:    TST     (R4),
2743 010362 005724      9#:    TST     (R4),
2744 010364 005714      10#:   TST     (R4)
2745 010366 001442      BEQ     21#
2746
2747 010370 011404      MOV     (R4),R4
2748 010372      SAVE   4
2749 010376      GET    1
2750 010402 005003      CLR    R3
2751 010404 005000      CLR    R0
2752
2753 010406 012137 002300      11#:   MOV     (R1)+,GOOD2
2754 010412 012137 002302      MOV     (R1)+,GOOD2+2
2755 010416 012137 002304      MOV     (R1)+,GOOD2+4
2756 010422 122760 000077 006574      CMPB    #77,RSPOKE(R0)
2757
2758 010430 001413      BEQ     14#
2759
2760 010432 105714      12#:   TSTB   (R4)
2761 010434 100004      BPL    13#
2762 010436 112760 000077 006574      MOVB   #77,RSPOKE(R0)
2763 010444 000423      BR     22#
2764
2765 010446 112403      13#:   MOVB   (R4)+,R3
2766 010450 126363 002300 002322      CMPB   GOOD2(R3),BAD2(R3)
2767 010456 001765      BEQ    12#
2768
2769 010460      14#:   GET    4
2770 010464 005200      INC    R0
2771 010466 020037 002400      CMP    R0,LSTLGH
2772 010472 001345      BNE    11#
2773
2774      ; UNEXPECTED RESPONSES, REPORT
2775
2776 010474      21#:
2777 010474 010237 002476      MOV    R2,OUTNUM
2778 010500 005237 002476      INC    OUTNUM

```

; IF NOT EXPECTED RESPONSE

; CHECK FOR ANY TEST TYPE?
; IF NOT, REPORT ERROR

; GET TEST TYPE ADDRESS IN R4
; SAVE IT
; RESTORE EXPECTED LIST POINTER

; GET EXPECTED RESPONSE PARAMETERS

; SEE IF THIS RESPONSE
; NOT RECEIVED YET, IS IT?
; IF NOT, GET IT

; IF YES, LOOK FOR BYTE TEST TERMINATOR?
; BRANCH IF IT IS NOT
; RESPONSE CORRECT, SET FLAG
; THEN SELECT NEXT RESPONSE

; SET INDEX FOR BYTE TEST
; IF YES, TEST IF THE SAME ?
; IF YES, LOOP

; IF NOT, RESTORE TEST ADDRESS TYPE
; UP-DATE CORRECT RESPONSE POINTER
; ANY-MORE EXPECTED RESPONSES?
; IF YES

; GET RESPONSE NUMBER

```

2779 010504          ERRHRD 10009,E10009,PBRSP      ;REPORT
2780
2781 010514          22$:  GET      3              ;RESTORE RCV RESPONSE POINTER
2782 010520 005202   INC      R2              ;INC RESPONSE COUNTER
2783 010522 005337 002402 DEC     RSPCNT      ;ANY MORE RESPONSE?
2784 010526 001226   BNE     1$              ;IF YES, LOOP
2785
2786 010530 005737 002476 TST     OUTNUM      ;TEST COMPLETELY OKE?
2787 010534 001002   BNE     40$          ;NOT, SET CARRY
2788 010536 000241   CLC                    ;YES, CLEAR CARRY
2789 010540 000207   RETURN
2790
2791 010542          40$:  WAIT     10000.
2792 010550 000261   SEC
2793 010552 000207   RETURN
2794
2795

```

2797
 2798
 2799
 2800
 2801
 2802 010554 010614
 2803 010556 010640
 2804 010560 010664
 2805 010562 010710
 2806 010564 010734
 2807 010566 010760
 2808 010570 011004
 2809 010572 011030
 2810 010574 011124
 2811 010576 011150
 2812 010600 011174
 2813 010602 011220
 2814 010604 011244
 2815 010606 011270
 2816 010610 011314
 2817 010612 011340

; POSSIBLE RESPONSE POINTER LIST

RSPLST: TSTF0	; DUMMY RESPONSE
TSTF1	; CONFIGURATE RESPONSE
TSTF2	; DECONFIGURATE RESPONSE
TSTF3	; DUMMY RESPONSE
TSTF4	; MODEM CHANGE
TSTF5	; XMIT BUFFER
TSTF6	; RCV BUFFER
TSTF7	; XMIT KILL
TSTF10	; RCV KILL
TSTF11	; DUMMY RESPONSE
TSTF12	; DUMMY RESPONSE
TSTF13	; DUMMY RESPONSE
TSTF14	; READ MODEM RESPONSE
TSTF15	; DUMMY RESPONSE
TSTF16	; ENABLE
TSTF17	; DISABLE " "

```

2819
2820      .NLIST BEX
2821
2822      ; RESPONSE TEST TABLE IN FUNCTION OF STATUS RECEIVED
2823
2824      ;STAT.      1   374   373   372   371   364   363   360   357   356
2825
2826 010614 000000 000000 000000 000000 000000,00000,00000,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2827 010640 011364 000000 000000 000000 TSTF1: TYP00,00000,00000,00000,00000,TYP00,00000,TYP00,00000,TYP00,TYP00
2828 010664 011364 000000 000000 000000 TSTF2: TYP00,00000,00000,00000,00000,TYP00,00000,TYP00,00000,TYP00,00000
2829 010710 000000 000000 000000 000000 TSTF3: 00000,00000,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2830 010734 011367 000000 000000 000000 TSTF4: TYP01,00000,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2831 010760 011364 011364 011364 011364 TSTF5: TYP00,TYP00,TYP00,TYP00,TYP00,00000,00000,TYP00,TYP00,00000
2832 011004 011367 011364 011364 011367 TSTF6: TYP01,TYP00,TYP01,00000,TYP00,TYP00,00000,TYP00,TYP00,00000
2833 011030 011364 000000 000000 000000 TSTF7: TYP00,00000,00000,00000,00000,TYP00,00000,00000,00000,TYP00,00000
2834 011054 000000 000000 000000 000000 TSTF8: 00000,00000,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2835 011100 000000 000000 000000 000000 TSTF9: 00000,00000,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2836 011124 011364 000000 000000 000000 TSTF10: TYP00,00000,00000,00000,00000,TYP00,00000,00000,00000,TYP00,00000
2837 011150 000000 000000 000000 000000 TSTF11: 00000,00000,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2838 011174 000000 000000 000000 000000 TSTF12: 00000,00000,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2839 011220 000000 000000 000000 000000 TSTF13: 00000,00000,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2840 011244 011367 000000 000000 000000 TSTF14: TYP01,00000,00000,00000,00000,TYP00,00000,00000,00000,TYP00,00000
2841 011270 000000 000000 000000 000000 TSTF15: 00000,00000,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2842 011314 011367 000000 000000 000000 TSTF16: TYP01,00000,00000,00000,00000,00000,00000,00000,TYP00,00000
2843 011340 011364 000000 000000 000000 TSTF17: TYP00,00000,00000,00000,00000,00000,00000,TYP00,00000,TYP00,00000
2844
2845      ; POSSIBLE TYPE OF TESTS
2846
2847 011364      000      005      TYP00: .BYTE 0,5      ;TEST OF BSEL2 AND BSEL7
2848 011366      200
2849
2850 011367      000      002      003 TYP01: .BYTE 0,2,3,5      ;TEST OF BSEL2,SEL4 AND BSEL7
2851 011373      200
2852
2853      .EVEN

```

2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911

011374
011404 005037 002430
011410 010137 002424
011414 010237 002426
011420 012704 002432
011424 013703 002346

011430 121112
011432 001010

011434 005201
011436 005202
011440 077305

011442 005737 002430
011446 001013
011450 000241
011452 000421

011454 005237 002430
011460 022737 000010 002430
011466 103762

011470 010124
011472 010224
011474 000757

011476
011506
011514 000261
011516

```

; **
; ROUTINE TO COMPARE BLOCKS OF DATA
; DESCRIPTION: TWO BUFFERS ARE COMPARED BYTE BY BYTE
; INPUTS: R1 SOURCE BUFFER 1 ADDRESS
;         R2 SOURCE BUFFER 2 ADDRESS
;         LENGTH = BYTE COUNT
; OUTPUTS: R1,R2 POINT TO CURRENT LOCATIONS
;         XMTBUF = XMIT BUFFER ADDRESS
;         RCVBUF = RECV BUFFER ADDRESS
;         ERRCNT = TOTAL ERROR COUNT
;         BADLOC = ERROR TABLE BASE ADDRESS
;         CARRY SET IF A MISMATCH OCCURS
;         LENGTH = INITIAL BYTE COUNT
; REPORT:  ERROR 10011          NOT EXPECTED DATA RECEIVED
; --

```

```

COMPAR:  SAVE      34
        CLR      ERRCNT          ;INIT ERROR COUNTER
        MOV      R1,XMTBUF      ;SAVE ADDRESSES
        MOV      R2,RCVBUF
        MOV      @BADLOC,R4    ;POINT TO ERROR TABLE
        MOV      LENGTH,R3     ;GET NUMBER OF BYTE
10:     CMPB     (R1),(R2)      ;THE SAME DATA?
        BNE     30             ;IF NOT
20:     INC      R1             ;NEXT LOCATION
        INC      R2
        SOB     R3,10         ;DEC R3 AND CONTINUE IF NOT NULL
        TST     ERRCNT        ;ANY ERROR?
        BNE     40             ;YES, SET CARRY
        CLC
        BR      50             ;NO, CLEAR CARRY
        ;TERMINATE
30:     INC      ERRCNT        ;INC ERROR COUNT
        CMP     @B.,ERRCNT    ;8 ERRORS ?
        BLO     20             ;YES, JUST LOOP
        MOV     R1,(R4).      ;IF NOT, SAVE BAD ADDRESS
        MOV     R2,(R4).
        BR      20             ;THEN LOOP BACK
40:     ERMRD   10011,E10011,PRBCOM ;REPORT
        WAIT   10000.
        SEC
50:     GET      34

```


ENKPCAO KMYLLA ECTNL DIAG
GLOBAL SUBROUTINES

MACRO M1200 05-APR-84 11:29 PAGE 35-1

D6

SEQ 68

2912 011526 000207

RETURN

2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927 011530
2928 011534 013703 002346
2929 011540 005021
2930 011542 077302
2931 011544
2932 011550 000207

; ROUTINE TO CLEAR A SPECIFIED BUFFER
;
;
; INPUTS: R1 BUFFER ADDRESS
; LENGTH = NUMBER OF WORDS
;
; OUTPUTS: R1 DESTROYED
;
;--

BUFCLR: SAVE 3
 MOV LENGTH,R3
1\$: CLR (R1). ;GET NUMBER OF WORDS
 SOB R3,1\$;CLEAR BUFFER
 GET 3 ;DEC COUNTER AND CONTINUE IF NOT NULL
 RETURN

2934
 2935
 2936
 2937
 2938
 2939
 2940
 2941
 2942
 2943
 2944
 2945
 2946
 2947 011552
 2948 011556 013703 002346
 2949 011562 112122
 2950 011564 077302
 2951 011566
 2952 011572 000207
 2953

```

; **
; ROUTINE TO COPY ONE BUFFER IN A SECOND ONE
;
; INPUTS: R1 SOURCE BUFFER ADDRESS
;         R2 DESTINATION BUFFER ADDRESS
;         LENGTH = BYTE COUNT
;
; OUTPUTS: R1,R2 DESTROYED
;
; --

```

```

COPY:  SAVE      3
      MOV       LENGTH,R3
1$:   MOVB      (R1),.(R2),
      SOB       R3,1$
      GET       3
      RETURN
      ;GET NUMBER OF BYTE
      ;COPY FROM SOURCE TO DESTINATION BUFFER
      ;DEC COUNTER AND CONTINUE IF NOT NULL

```

```

2955
2956
2957
2958 ;**
2959 ; ROUTINE TO ENTER THE APPLICATION MODE
2960 ;
2961 ; DESCRIPTION: RESET ON Q-BUS THEN WAIT FOR 5 MS
2962 ; SET APPLICATION MODE AND MCLR BITS. THEN, CHECK
2963 ; IF MCLR CLEARED BY THE KMV. IF NOT, REPORT ERROR 2
2964 ;
2965 ; INPUTS: NONE
2966 ;
2967 ; OUTPUTS: R5 = CSR BASE ADDRESS
2968 ; CARRY BIT SET IN CASE OF ERROR
2969 ;
2970 ; ERROR REPORT: ERROR 10000 KMV11 FAILS TO RESET MCLR
2971 ;
2972 ;--
2973
2974 011574 MODE0: SAVE 1
2975
2976 011600 013705 002504 MOV KMVCSR,R5 ;GET CSR ADDRESS IN R5
2977
2978 011604 005065 000000 CLR SELO(R5) ;RESET SELO
2979 011610 012765 040000 000000 MOV #MCLR,SELO(R5) ;SET MODE 0 AND MCLR
2980 011616 012701 000012 MOV #10.,R1 ;SET WAIT OF 10 MS
2981 011622 1$:
2982 011622 WAIT 10. ;WAIT 1 MS
2983 011630 032765 040000 000000 BIT #MCLR,SELO(R5) ;MCLR CLEARED?
2984 011636 001417 BEQ 2$ ;IF YES
2985 011640 077110 SOB R1,1$ ;IF NOT, LOOP TILL TIME-OUT
2986
2987 011642 005037 002276 CLR G0000 ;SET REPORT FOR TIME OUT
2988 011646 016537 000000 002320 MOV SELO(R5),BADO
2989 011654 ERRMRD 10000,E10000,PRSELO ;REPORT
2990 011664 WAIT 10000.
2991 011672 000261 SEC
2992 011674 000401 BR 3$
2993
2994 011676 000241 2$: CLC
2995 011700 3$: GET 1
2996
2997 011704 000207 RETURN
2998

```

```

3000
3001
3002      ;**
3003      ; ROUTINE TO TEST IF LOOP BACK CONNECTOR PLUGGED
3004      ;
3005      ; INPUT: LCLOOP = LOOP FLAG, LOADED FROM P.TABLE
3006      ;
3007      ; OUTPUT: CARRY SET IF NOT INSERTED
3008      ;          CARRY CLEARED IF YES
3009      ;
3010      ; REPORT: ERROR 10012          LOOP CONNECTOR NOT INSTALLED
3011      ;
3012      ;
3013
3014 011706 000241      LPBACK: CLC
3015 011710 005737 002344      TST      LCLOOP      ;TEST IF PLUGGED?
3016 011714 001010      BNE      1$          ;YES
3017
3018 011716      ERRHRD 10012,E10012 ;NO. REPORT
3019 011726      WAIT   10000.
3020 011734 000261      SEC
3021 011736      1$:      ;SET CARRY IF NOT
3022 011736 000207      RETURN
3023

```

```

3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037 011740
3038 011750 013702 002346
3039 011754 005003
3040 011756
3041 011756 110321
3042 011760 005203
3043 011762 077203
3044
3045 011764
3046 011774 000207
3047
3048

```

```

; **
; ROUTINE TO LOAD BUFFER WITH AND INCREMENTAL PATTERN
;
; INPUTS: R1 = BUFFER ADDRESS
;         LENGTH = BYTE COUNT
;
; OUTPUT: R1 DESTROYED
;
; -
;
BUFLD:  SAVE    23
        MOV     LENGTH,R2
        CLR    R3
        ;READ LENGTH
        ;INIT PATTERN
1$:
        MOVB   R3,(R1)+
        INC    R3
        ;LOAD BYTE
        ;NEXT
        SOB    R2,1$
        ;FOR ALL
        GET    23
        RETURN

```

```

3050
3051
3052 ;**
3053 ; ROUTINE TO INIT IN/OUT HANDSHAKING
3054 ;
3055 ; INPUTS: INTIN = POINTS TO THE FIRST VECTOR ADDRESS
3056 ;          PRILEV = POINT TO THE KMV DEVICE PRIORITY LEVEL
3057 ;
3058 ; OUTPUTS: RSPCNT,REQCNT AND ERRFLG ARE RESET
3059 ;          XX0 AND XX4 VECTORS ARE SET
3060 ;          MOST PRIORITY LEVEL = (PRILEV)-1
3061 ;          OUTLST POINTS TO TOP OF OUT BUFFER
3062 ;          IN/OUT INTERRUPT ENABLE BITS SET
3063 ;          ALL SAVE BUFFERS ARE CLEARED
3064 ;          XMIT BUFFERS ARE FELT WITH INCREMENTAL BYTES
3065 ;--
3066
3067 011776 005037 002372      INIQIO: CLR      REQCNT      ;CLEAR INPUT COUNT
3068 012002 005037 002402      CLR      RSPCNT      ;CLEAR OUTPUT COUNT
3069 012006 005037 002370      CLR      ERRFLG     ;CLEAR ERROR FLAG REG
3070 012012 012737 006620 002410  MOV      #OUTBUF,OUTLST ;POINT TO TOP OF OUT BUFFER
3071
3072 012020 012700 000010      MOV      #CSRLLEN,R0 ;CLEAR ALL CSR'S
3073 012024 013702 002504      MOV      KMVCSR,R2
3074 012030 005022          1$: CLR      (R2)+
3075 012032 077002          SOB      R0,1$
3076
3077 012034 012737 002106 002346  MOV      #3*20.+10.+1024.,LENGTH ;CLEAR BUFFERS
3078 012042 012701 002574      MOV      #RXBUF0,R1
3079 012046 004737 011530      CALL     BUFCLR
3080
3081 012052 012701 004574          MOV      #TXBUF0,R1 ;POINT TO XMIT BUFFERS
3082 012056 012737 002000 002346  MOV      #1024.,LENGTH ;SET BUFFER LENGTH
3083 012064 004737 011740      CALL     BUFLD ;LOAD INCREMENTAL BYTES
3084
3085 012070          SAVE     1
3086 012074 013701 002506      MOV      INTIN,R1 ;POINT TO XX0
3087 012100 012721 012710      MOV      #ITIN,(R1)+ ;LOAD IN SERVICE ROUTINE
3088 ;          MOV      #PRI07,(R1)+ ;THEN HIGHEST PRIORITY ;JB REV A 0
3089 012104 012721 000300      MOV      #PRI06,(R1)+ ;THEN HIGH PRIORITY ;JB REV A-0
3090 012110 012721 013044      MOV      #ITOUT,(R1)+ ;LOAD OUT SERVICE ROUTINE IN XX4
3091 ;          MOV      #PRI07,(R1) ;THEN HIGHEST PRIORITY ;JB REV A-0
3092 012114 012711 000300      MOV      #PRI06,(R1) ;THEN HIGH PRIORITY ;JB REV A-0
3093
3094 012120 013701 002512      MOV      PRILEV,R1 ;GET DEVICE PRIORITY
3095 012124          ASHR     R1,4 ;ADJUST
3096 012134 162701 000040      SUB      #40,R1 ;ONE LESS
3097 012140 106401          MTPS    R1 ;MODIFY MOST PRIORITY LEVEL
3098
3099 012142          GET      1
3100 012146 000207          RETURN
3101

```

```

3103
3104
3105
3106      ; ROUTINE TO LOAD APPLICATION FIRMWARE
3107      ;
3108      ; DESCRIPTION: LOAD ALL THE FIRMWARE
3109      ;                TEST IF CORRECTLY LOADED
3110      ;                THEN SET FLAG
3111      ;
3112      ; INPUTS:  $BUFF = APPLICATION CODE FILE
3113      ;          BUFLGH = APPLICATION FIRMWARE LENGTH
3114      ;          $RAM = KMV RAM LOAD ADDRESS
3115      ;          APPFLG = SET WHEN CODE IS CORRECTLY LOADED
3116      ;          S.LOAD = FLAG TO INDICATE LOAD(0) OR COMPARE(1)
3117      ;          R5 = CSR BASE ADDRESS
3118      ;
3119      ;           USING OF R1 TO R4.
3120      ;           R1 = TIME-OUT COUNTER
3121      ;           R2 = WORD LOADING COUNTER
3122      ;           R3 = RAM LOAD ADDRESS
3123      ;           R4 = BUFFER ADDRESS
3124      ;
3125      ; OUTPUTS: APPFLG = 1 IF CORRECTLY LOADED
3126      ;           CARRY SET IF ANY ERROR
3127      ;
3128      ; REPORT:  ERROR 10002           WRITE FAILURE
3129      ;           ERROR 10003           READ FAILURE
3130      ;           ERROR 10004           DATA COMPARE ERROR
3131      ;
3132      ;
3133      ;
3134      ;
3135      ;
3136 012150   LDAPPL: SAVE      1234
3137
3138      ; WRITE PHASE
3139
3140 012170   005037   002412      CLR      S.LOAD           ;0 FOR LOAD
3141 012174
3142 012174   012704   030306      MOV      # $BUFF,R4       ;POINT TO SOURCE BUFFER
3143 012200   013702   043660      MOV      BUFLGH,R2       ;SET LENGTH
3144 012204   012703   030306      MOV      # $RAM,R3       ;POINT TO DESTINATION
3145 012210
3146 012210   005737   002412      TST      S.LOAD           ;IS IT A LOAD?
3147 012214   001004              BNE      10$             ;BR IF COMPARE
3148
3149 012216   004737   007506      CALL     WDATA           ;WRITE DATA
3150 012222   103436              BCS     30$             ;IF ERROR
3151 012224   000422              BR      20$             ;IF OKE, WRITE NEXT DATA
3152
3153      ; TO COMPARE
3154
3155 012226
3156 012226   004737   007400      CALL     RDATA           ;READ DATA
3157 012232   103432              BCS     30$
3158
3159 012234   021465   000006      CMP      (R4),SEL6(R5)  ;THE SAME?

```



```

3160 012240 001414          BEQ      20$          ;YES, NEXT WORD
3161
3162 012242 011437 002304          MOV      (R4),GOOD6          ;SET REPORT
3163 012246 016537 000006 002326  MOV      SEL6(R5),BAD6
3164 012254 010337 002432          MOV      R3,BADLOC
3165 012260          ERRHRD  10004,E10004,PRDAT          ;IF NOT, REPORT
3166 012270 000413          BR       30$
3167
3168 012272 062703 000002          20$:   ADD      #2,R3          ;NEXT RAM ADDRESS
3169 012276 005724          TST      (R4)+          ;NEXT SOURCE ADDRESS
3170 012300 005302          DEC      R2          ;NEXT
3171 012302 001342          BNE      2$          ;LOOP TILL BUFFER END
3172
3173          ; BUFFER END - TEST IF COMPARE DONE
3174
3175 012304 005737 002412          TST      S.LOAD          ;IS IT?
3176 012310 001010          BNE      40$          ;YES, LOAD COMPLETED
3177 012312 005237 002412          INC      S.LOAD          ;NOT, COMPARE RAM AND SOURCE BUFFER
3178 012316 000726          BR       1$
3179
3180 012320          30$:   WAIT     10000.
3181 012326 000261          SEC
3182 012330 000403          BR       50$          ;SET CARRY IF ERROR
3183          ;THEN EXIT
3184 012332 000241          40$:   CLC
3185 012334 005237 002414          INC      APPFLG          ;IF OKE, SET FLAG FOR LOADED
3186 012340          50$:
3187 012340          GET      1234
3188 012360 000207          RETURN
3189

```

```

3191
3192
3193 ; **
3194 ; ROUTINE TO CHECK KMV PROM ECO LEVEL AND PRINT PART NUMBER
3195 ;
3196 ; DESCRIPTION: BEGIN
3197 ; READ ACTUAL ECO LEVEL AT ADDRESS=ECONB
3198 ; COMPARE TO GOLVL
3199 ; IF NOT EQUAL THEN REPORT ERROR
3200 ; FOR KMV ADDRESS=PARTNB TO LAST CHR DO
3201 ; READ ASCII CHR
3202 ;
3203 ; NEXT
3204 ; PRINT PART NUMBER
3205 ; END
3206 ;
3207 ; INPUTS: R5 = CSR BASE ADDRESS
3208 ; PARTNB IS THE KMV PROM PART NUMBER ADDRESS
3209 ; PROMNB IS THE ADDRESS OF GOOD ONE
3210 ; ECONB IS THE ADDRESS OF ACTUAL PROM ECO LEVEL
3211 ; GOLVL IS THE LEVEL EXPECTED BY THE DIAG.
3212 ;
3213 ; USING OF R1 TO R4
3214 ; R1=TIME-OUT COUNTER
3215 ; R3=ROM LOAD ADDRESS
3216 ; R4=GOOD CHR ADDRESS
3217 ;
3218 ; OUTPUTS: CARRY SET IF ANY ERROR
3219 ;
3220 ; REPORTS: ERROR 10003 READ FAILURE
3221 ; ERROR 10008 BAD EPROM ECO LEVEL
3222 ;
3223 ; -
3224 REVCHK: SAVE 1234
3225
3226 012402 012703 160002 MOV #ECONB,R3 ;INIT R3
3227
3228 012406 004737 007400 CALL RDATA ;READ DATA
3229 012412 103441 BCS 30#
3230
3231 012414 016537 000006 002326 MOV SEL6(R5),BAD6 ;GET SEL6
3232 012422 023737 002326 007126 CMP BAD6,GDREV ;THE SAME LEVEL?
3233 012430 001437 BEQ 40# ;IF YES
3234
3235 012432 ERRHRD 10008,E10008,PRBECO ;IF NOT, REPORT
3236
3237 ; PRINT PART NUMBER
3238
3239 012442 012703 160004 1#: MOV #PARTNB,R3 ;INIT R3
3240 012446 012704 002556 MOV #PROMNB,R4 ;INIT R4
3241 012452
3242 012452 004737 007400 2#: CALL RDATA ;READ CHR
3243 012456 103417 BCS 30#
3244
3245 012460 016524 000006 MOV SEL6(R5),(R4). ;SAVE IT
3246 012464 062703 000002 ADD #2,R3 ;NEXT ADDRESS
3247 012470 020427 002572 CMP R4,#PROMNB+12. ;READ COMPLETE?

```

```
3248 012474 001366      BNE      2$           ;LOOP JF NOT
3249
3250 012476             PRINTF  #MPROM       ;PRINT PART NUMBER
3251
3252 012516             30$:  WAIT    10000.      ;SET CARRY IF ERROR
3253 012524 000261      SEC
3254 012526 000401      BR      50$           ;THEN EXIT
3255
3256 012530 000241      40$:  CLC
3257
3258 012532             50$:  GET     1234
3259 012552 000207      RETURN
3260
```

3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316

```

***
; ROUTINE TO START APPLICATION FIRMWARE
;
; DESCRIPTION: FIRST, TEST IS DONE TO SEE IF APPLICATION CODE
;              CAN BE LOADED. IF YES, THE CODE IS LOADED.
;              THEN, THE PROGRAM IS STARTED.
;
; INPUTS: APPFLG INDICATES IF APPLICATION CODE LOADED(1) OR NOT(0)
;         $START POINT TO THE FIRMWARE START ADDRESS
;         R5 = KHV CSR ADDRESS
;
; OUTPUTS: APPFLG CLEARED IF ANY ERROR DURING LOAD OR START
;         CARRY SET IF ANY ERROR
;
; REPORTS: ERROR 10001      RUN FAILURE
;           ERROR 10002      WRITE FAILURE
;           ERROR 10003      READ FAILURE
;           ERROR 10004      COMPARE FAILURE
;           ERROR 10008      BAD PROM NUMBER
;
;--

```

```

RUNAPP: TST      APPFLG      ;CODE ALREADY LOADED?
        BNE      10         ;YES, START IT
        CALL     REVCHK      ;CHECK KHV PROM NUMBER
        BCS      200        ;EXIT IF ERROR
        CALL     LDAPPL      ;NOT, LOAD IT
        BCS      200        ;EXIT IF LOAD ERROR
10:     MOV      @1START,SEL4(R5) ;OKE, LOAD START ADDRESS
        MOV      @RUN,SELO(R5)   ;START
        MOV      @10.,R1        ;SET TIME-OUT
20:     WAIT     10.           ;WAIT FOR 1 MS
        BIT      @RUN!ERR,SELO(R5) ;IS CORRECTLY STARTED?
        BEQ      300          ;YES, EXIT
        SOB     R1,20         ;NOT, LOOP TILL TIME OUT
300:    CLR      GOOD0        ;SET REPORT
        MOV      SELO(R5),BADO
        ERHRD   10001,E10001,PRSELO ;IF TIME-OUT
200:    CLR      APPFLG      ;CLEAR FLAG
        WAIT    10000.
        SEC
        RETURN
300:    WAIT     100.         ;WAIT FOR FIRWARE INIT.
        CLC
        RETURN

```

3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374

```

; **
; INPUT INTERRUPT SERVICE ROUTINE
;
; DESCRIPTION: BEGIN
;               IF RQI NOT SET
;                 THEN REPORT ERROR
;               IF RDYI NOT SET
;                 THEN REPORT ERROR
;               IF NO QIO
;                 THEN INPUT DUMMY COMMAND
;               POINT TO CURRENT COMMAND IN LIST
;               INPUT PARAMETERS
;               IF NO MORE COMMAND IN LIST
;                 THEN CLEAR RQI
;               INPUT COMMAND NUMBER
;               CLEAR RDYI
;               END
;
; INPUTS: R5 = KMY CSR ADDRESS
;         INLIST POINT TO THE CURRENT COMMAND TO INPUT
;         REQCNT INDICATES THE NUMBER OF INPUT STILL TO DO
;         TMOUT INDICATES INTERRUPT TIME-OUT LENGTH
;
; OUTPUTS: ERRFLG SET SUCH AS,
;          BIT2 SET IF NO RQI OR RDYI
;
; --

```

```

ITIN:  MOV    R1, -(SP)           ;SAVE R1,R2
        MOV    R2, -(SP)
;
        BIT    @IEI,SEL0(R5)     ;INT. ENABLE?
        BEQ    1$               ;NOT, SET ERROR
        BIT    @RQI,SEL0(R5)    ;IS RQI SET?
        BEQ    1$               ;NOT, SET ERROR
        BIT    @RDYI,SEL2(R5)   ;IS RDYI SET?
        BNE    2$               ;YES, SERVE INT.
1$:
        BIS    @BIT2,ERRFLG     ;SET ERROR FLAG
        BR     30$              ;THEN EXIT
;
2$:
        TST    REQCNT           ;ANY-MORE COMMAND?
        BNE    3$               ;YES, PROCESS
        MOV    #F0,R2          ;NOT, SET DUMMY COMMAND
        BR     20$              ;THEN CLEAR RDYI AND EXIT
;
3$:
        MOV    INLIST,R1        ;POINT TO COMMAND PARAMETERS
        MOV    (R1)+,R2        ;READ COMMAND NUMBER
;
        MOV    (R1)+,SEL4(R5)   ;SET SEL4
        MOV    (R1)+,SEL6(R5)   ;SET SEL6
;
        MOV    R1,INLIST       ;SAVE NEW POINTER
;
        DEC    REQCNT           ;NEXT COMMAND?

```

```

3375 013016 001003          BNE      20$          ;ANY-MORE COMMAND, DON'T CLEAR RQI
3376 013020 042765 000200 000000      BIC      @RQI,SEL0(R5) ;NO-MORE COMMAND, CLEAR RQI
3377
3378 013026          20$:          MOV      R2,SEL2(R5)      ;SET SEL2 WITH COMMAND NUMBER
3379 013026 010265 000002          ;
3380
3381 013032          30$:          MOV      TMOUT,R4          ;INIT AGAIN TIME-OUT
3382 013032 013704 002404          MOV      (SP)+,R2      ;RSTORE R1,R2
3383 013036 012602          MOV      (SP)+,R1
3384 013040 012601          RTI
3385 013042 000002
3386

```

```

3388
3389
3390      ; OUTPUT INTERRUPT SERVICE ROUTINE
3391      ;
3392      ; DESCRIPTION: BEGIN
3393      ;                   IF RDYO NOT SET
3394      ;                       THEN REPORT ERROR
3395      ;                   IF NO RESPONSE AWAITED
3396      ;                       THEN REPORT ERROR
3397      ;                   SAVE RESPONSE IN BUFFER
3398      ;                   UP-DATE RESPONSE COUNT
3399      ;                   CLEAR RDYO
3400      ;                   END
3401      ;
3402      ; INPUTS: R5 = KMV CSR ADDRESS
3403      ;          TMOUT INDICATES THE TIME-OUT VALUE FOR INT.
3404      ;          OUTLST POINTS TO THE CURRENT FREE BUFFER
3405      ;          RSPCNT CURRENT RESPONSE COUNT
3406      ;
3407      ; OUTPUTS: R4 = TMOUT
3408      ;          ERRFLG SET WITH BIT0 IF ANY ERROR
3409      ;
3410      ; --
3411
3412
3413      ITOUT:  MOV     R1, -(SP)                ;SAVE R1
3414
3415      013044 010146      BIT     #IE0,SEL0(R5)        ;INT. ENABLE?
3416      013046 032765 000020 000000      BEQ     1#                          ;NOT, SET ERROR
3417      013054 001404      BIT     #RDYO,SEL2(R5)       ;IS RDYO SET?
3418      013056 032765 000200 000002      BNE     2#                          ;YES, SAVE
3419      013064 001004      BIS     #BIT0,ERRFLG        ;NOT, SET ERROR
3420      013066 052737 000001 002370 1# :   BR      30#                          ;THEN EXIT
3421      013074 000423
3422      013076 022737 000024 002402 2# :   CMP     #20.,RSPCNT          ;OUTBUF FULL?
3423      013104 001417      BEQ     30#                          ;YES, TERMINATE
3424      013106 013701 002410      MOV     OUTLST,R1          ;POINT TO FREE BUFFER
3425      013112 016521 000002      MOV     SEL2(R5),(R1)      ;SAVE SEL2 TILL SEL6
3426      013116 016521 000004      MOV     SEL4(R5),(R1)
3427      013122 016521 000006      MOV     SEL6(R5),(R1)
3428
3429      013126 010137 002410      MOV     R1,OUTLST        ;POINT TO NEXT FREE BUFFER
3430      013132 005237 002402      INC     RSPCNT          ;ONE RESPONSE MORE
3431      013136 042765 000200 000002      BIC     #RDYO,SEL2(R5)  ;CLEAR RDYO
3432
3433      013144 013704 002404      30# :   MOV     TMOUT,R4          ;INIT AGAIN TIME-OUT
3434      013150 012601      MOV     (SP),R1        ;RESTORE R1
3435      013152 000002      RTI
3436

```

3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494

```

; **
; ROUTINE TO PROCESS QIO WITH INTERRUPTS
;
; DESCRIPTION: BEGIN
;               SET POINTERS
;               SET RQI
;               INIT INTERRUPT WATCH DOG
;               REPEAT
;                   WAIT FOR 500MICROS
;                   IF ERRFLG SET
;                       THEN REPORT ERROR
;                   IF REQCNT & RSPCNT = 0
;                       THEN END
;               UNTIL WATCH DOG DONE
;               IF WATCH DOG DONE
;                   THEN REPORT ERROR
;               END
;
; INPUTS: R5 = KMV CSR ADDRESS
;         ERRFLG INDICATES IN IT ERROR(BIT2) OR/AND OUT IT ERROR(BIT0)
;         R1 POINTS TO A TABLE WHICH CONTENTS:
;         REQCNT COMMAND BASE ADDRESS
;         TIME-OUT LENGTH
;
; OUTPUTS: CARRY SET IF ANY ERROR
;
; REPORTS: ERROR 10005      UN-EXPECTED IN IT
;          ERROR 10006      UN-EXPECTED OUT IT
;          ERROR 10007      NO MORE IN/OUT IT WHILE QIO ARE PENDING
;
; --

```

```

QIOP:  MOV      (R1)+,R2          ;GET BASE ADDRESS
      MOV      (R1),TMOUT        ;GET TIME-OUT LENGTH
      MOV      (R2)+,REQCNT      ;GET IN LIST LENGTH
      MOV      R2,INLST         ;POINT TO IN LIST
      MOV      TMOUT,R4          ;SET TIME OUT
      BIS      @RQI!IEI!IEO,SEL0(R5) ;SET REQUEST AND ENABLE INT.
1$:
      WAIT     10.              ;WAIT FOR 1 MS
      TST     ERRFLG            ;ANY ERROR?
      BNE     10$              ;YES, REPORT
      SOB     R4,1$            ;NOT, LOOP TILL TIME OUT
      TST     REQCNT            ;NO-MORE REQUEST?
      BEQ     30$              ;YES, END OF QIO
      ERRHRD  10007,E10007,PRQION ;REPORT
      BR      20$              ;SET CARRY AND EXTT
20$:
10$:  MOV      SEL0(R5),BADO      ;SET BADO
      MOV      SEL2(R5),BAD2     ;SET BAD2
      MOV      SEL4(R5),BAD4     ;SET BAD4
      MOV      SEL6(R5),BAD6     ;SET BAD6

```

```

013154 012102
013156 011137 002404
013162 012237 002372
013166 010237 002406
013172 013704 002404
013176 052765 000221 000000
013204
013204
013212 005737 002370
013216 001011
013220 077407
013222 005737 002372
013226 001451
013230
013240 000437
013242 016537 000000 002320
013250 016537 000002 002322
013256 016537 000004 002324
013264 016537 000006 002326

```



```

3512
3513
3514 ; ROUTINE TO DROP UNIT AFTER 5 ERRORS
3515 ;
3516 ; BEFORE DROPPING, FLAGS ARE TESTED TO SEE IF 'LOOP ON ERROR OR
3517 ; 'DROPPING INHIBITTED' ARE SET.
3518 ;
3519 ; INPUT: ERCNTR = NUMBER OF ERRORS
3520 ; MAXERR = MAXIMUM NUMBER OF ERROR
3521 ;
3522 ; OUTPUT: NONE
3523 ;
3524 ; -
3525
3526
3527 013356          CHKMAX: INLOOP          ;LOOP ON ERROR?
3528 013360          BCOMPLETE          2$      ;IF YES, EXIT
3529
3530 013362 012700 100000          MOV      #100000,RO      ;TAKE A BREAK
3531 013366 077001          1$: SOB      RO,1$
3532
3533 013370          RFLAGS RO          ;READ OPERATOR FLAG
3534 013372 032700 000040          BIT      #IDU,RO      ;DROPPING INHIBITTED?
3535 013376 001026          BNE      2$      ;IF YES, EXIT
3536
3537 013400 005237 002264          INC      ERCNTR          ;UPDATE ERROR COUNTER
3538 013404 023737 002262 002264          CMP      MAXERR,ERCNTR      ;TOO MANY ERRORS?
3539 013412 003020          BGT      2$      ;IF NOT, EXIT
3540
3541 013414          PRINTF #MERR,MAXERR,LOGDEV      ;IF YES, REPORT
3542 013444          DODU LOGDEV          ;THEN DROP UNIT
3543
3544 013452          DOCLN          ;END THE SUBPASS
3545
3546 013454 000207          2$: RETURN
3547
3548
3549 013456          045 116 045 MERR: .ASCIZ /#N#A MORE THAN #D3#A ERRORS ON UNIT #D2#N/
3550 .EVEN
3551

```

3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586

```

; **
; ROUTINE TO PRINT TEST NUMBER
;
; DESCRIPTION: NONE
;
; CALLING SEQUENCE: BADHEAD
;
; INPUTS: L$TEST=TEST NUMBER
;
; OUTPUTS: NONE
;
; CAUTION: NONE
; --

      .MACRO ED$CALL XY
      .LIST
;          ** TEST'XY' **
      .NLIST
      .ENDM

      .MACRO BADHEAD
      .RADIX 10
      ED$CALL \T$TESTNUM+1
      .RADIX 8
      .ENDM

```

```
3588 .SBTTL GLOBAL ERROR REPORT SECTION
3589
3590 ;////////////////////////////////////
3591 ;/ ERROR MESSAGE SECTION
3592 ;////////////////////////////////////
3593 .NLIST BEX
3594
3595 013532 045 116 045 MRDEXP: .ASCIZ /#N#A READ EXPECTED/
3596 013572 045 116 045 MSEL0: .ASCIZ /#N#A SEL0 #06#A #06#N/
3597 013626 045 116 045 MSEL2: .ASCIZ /#N#A SEL2 #06#A #06#N/
3598 013660 045 116 045 MSEL4: .ASCIZ /#N#A SEL4 #06#A #06#N/
3599 013712 045 116 045 MSEL6: .ASCIZ /#N#A SEL6 #06#A #06#N/
3600 013746 045 116 045 MSEL10: .ASCIZ /#N#A SEL10 #06#A #06#N/
3601 014002 045 116 045 MSEL12: .ASCIZ /#N#A SEL12 #06#A #06#N/
3602 014036 045 116 045 MSEL14: .ASCIZ /#N#A SEL14 #06#A #06#N/
3603 014072 045 116 045 MSEL16: .ASCIZ /#N#A SEL16 #06#A #06#N/
3604
3605 014126 045 116 045 MECO: .ASCII /#N#A OBTAINED ECO LEVEL = #06/
3606 014165 045 116 045 .ASCIZ /#N#A EXPECTED ONE = #06#N/
3607
3608 014227 045 116 045 MSEL: .ASCIZ /#N#A SEL#02#A = #06#A EXPECTED VALUE = #06#N/
3609
3610 014315 045 116 045 MQIO: .ASCII /#N#A NUMBER OF PENDING INPUTS = #02/
3611 014362 045 116 045 .ASCIZ /#N#A NUMBER OF RESPONSES = #02#N/
3612
3613 014425 045 116 045 MBNUM: .ASCII /#N#A EXPECTED NUMBER OF RESPONSES = #02/
3614 014474 045 116 045 .ASCIZ /#N#A NUMBER OF RESPONSES RECEIVED = #02#N/
3615
3616 014546 045 116 045 MCSRO: .ASCII /#N#A SEL0 READ = #06/
3617 014575 045 116 045 MCSR2: .ASCIZ /#N#A SEL2 READ = #06/
3618 014625 045 116 045 MCSR1: .ASCII /#N#A SEL4 READ = #06/
3619 014674 045 116 045 .ASCIZ /#N#A SEL6 READ = #06#N/
3620
3621 014706 045 116 045 MQION: .ASCIZ /#N#A UNEXPECTED QIO RESPONSE NUMBER = #02#N/
3622
3623 014764 045 116 045 MDATO: .ASCIZ /#N#A ADDRESS READ EXPECTED/
3624 015044 045 116 045 MDAT1: .ASCIZ /#N#A #06#A #06#A #06#N/
3625
3626 015117 045 116 045 MBBUFO: .ASCII /#N#A TRANSMIT BUFFER ADDRESS : #06/
3627 015161 045 116 045 .ASCIZ /#N#A RECEIVE BUFFER ADDRESS : #06/
3628 015224 045 116 045 MBBUF1: .ASCII /#N#A RECORD SIZE : #03/
3629 015266 045 116 045 .ASCIZ /#N#A BYTES IN ERROR : #03#N/
3630 015333 045 116 045 CDBFER: .ASCIZ /#N#A ADDRESS DATA#N/
3631
3632 015411 045 116 045 MXMTER: .ASCIZ /#N#A TRANSMIT BUF: #06#A #03/
3633 015463 045 116 045 MRCVER: .ASCIZ /#N#A RECEIVE BUF: #06#A #03#N/
3634
3635 015537 045 116 045 TFM36: .ASCIZ /#N#A ADDRESS = #06#A UNIT = #02#N/
3636
3637 015610 113 115 126 E10000: .ASCIZ /KMV11 FAILS TO RESET MASTER CLEAR/
3638 015652 122 125 116 E10001: .ASCIZ /RUN FUNCTION NOT CORRECTLY PERFORMED IN APPL. MODE/
3639 015735 127 122 111 E10002: .ASCIZ /WRITE FUNCTION NOT CORRECTLY PERFORMED IN APPL. MODE/
3640 016022 122 105 101 E10003: .ASCIZ /READ FUNCTION NOT CORRECTLY PERFORMED IN APPIC.MODE/
3641 016106 104 101 124 E10004: .ASCIZ /DATA COMPARE ERROR DURING APPLICATION CODE LOADING/
3642 016171 125 116 105 E10005: .ASCIZ /UNEXPECTED INTERRUPT IN/
3643 016221 125 116 105 E10006: .ASCIZ /UNEXPECTED INTERRUPT OUT/
3644 016252 116 117 040 E10007: .ASCIZ /NO MORE INPUT INTERRUPTS WHILE INPUTS ARE PENDING/
```

3645	016334	125	116	105	E10008: .ASCIZ	/UNEXPECTED EPROM'S ECO LEVEL/
3646	016371	125	116	105	E10009: .ASCIZ	/UNEXPECTED QIO RESPONSE/
3647	016421	125	116	105	E10010: .ASCIZ	/UNEXPECTED NUMBER OF RESPONSES RECEIVED/
3648	016471	125	116	105	E10011: .ASCIZ	/UNEXPECTED DATA RECEIVED/
3649	016522	124	110	111	E10012: .ASCIZ	/THIS TEST IS SKIPPED BECAUSE THERE IS NO EXTERNAL LOOP/
3650						
3651	016611	124	111	115	E00000: .ASCIZ	/TIME-OUT TRAP ON REFERENCING Q-BUS DEVICE REGISTER/
3652	016674	124	110	105	E00001: .ASCIZ	/THE KMV11-A FAILS TO RUN THE SELF TEST/
3653	016743	122	104	131	E00002: .ASCIZ	/RDYI NOT ASSERTED BY THE KMV AFTER AN INPUT REQUEST/
3654	017027	122	104	131	E00003: .ASCIZ	/RDYO NOT ASSERTED BY THE KMV IN RESPONSE TO A VALID COMMAND/
3655	017123	125	116	105	E00004: .ASCIZ	/UNEXPECTED INTERRUPT WHEN IEI AND IEO CLEARED/
3656					.EVEN	

3658
3659
3660
3661
3662
3663
3664
3665
3666
3667 017202
3668 017202
3669 017232 004737 013356
3670 017236
3671
3672 017240
3673 017240
3674 017260
3675 017310 004737 013356
3676 017314
3677
3678 017316
3679 017316
3680 017346 004737 013356
3681 017352
3682
3683 017354
3684 017354
3685 017374
3686 017430 004737 013356
3687 017434
3688
3689 017436
3690 017436
3691 017466
3692 017516
3693 017546 004737 013356
3694 017552
3695
3696 017554
3697 017554
3698 017610 004737 013356
3699 017614
3700
3701 017616
3702 017616
3703 017646 004737 013356
3704 017652
3705
3706 017654
3707 017654
3708 017700
3709 017720
3710 017750
3711 020000
3712 020030 004737 013356
3713 020034
3714

```
;/
; ERROR REPORT SECTION
;
;***
; ERROR REPORT FOR KMV11 CSR TEST
;
BGNMSG  BADAD
PRINTB  #TFM36,BADLOC,UNIT
CALL    CHKMAX
ENDMSG
BGNMSG  PRSELO
PRINTB  #MRDEXP
PRINTB  #MSELO,BAD0,GOOD0
CALL    CHKMAX
ENDMSG
BGNMSG  PRBECO
PRINTB  #MECO,BAD6,GDREV
CALL    CHKMAX
ENDMSG
BGNMSG  PRDAT
PRINTB  #MDATO
PRINTB  #MDAT1,BADLOC,BAD6,GOOD6
CALL    CHKMAX
ENDMSG
BGNMSG  PRSTAT
PRINTB  #MCSR0,BAD0,BAD2
PRINTB  #MCSR1,BAD4,BAD6
PRINTB  #MQIO,REQCNT,RSPCNT
CALL    CHKMAX
ENDMSG
BGNMSG  PRSEL
PRINTB  #MSEL,SELNUM,BAD,GOOD
CALL    CHKMAX
ENDMSG
BGNMSG  PRQION
PRINTB  #MQIO,REQCNT,RSPCNT
CALL    CHKMAX
ENDMSG
BGNMSG  PRBQIO
PRINTB  #MQION,OUTNUM
PRINTB  #MRDEXP
PRINTB  #MSEL2,BAD2,GOOD2
PRINTB  #MSEL4,BAD4,GOOD4
PRINTB  #MSEL6,BAD6,GOOD6
CALL    CHKMAX
ENDMSG
```

3715	020036			BGNMSG	PBRSP	
3716	020036				PRINTB	#MQION,OUTNUM
3717	020062				PRINTB	#MCSR2,BAD2
3718	020106				PRINTB	#MCSR1,BAD4,BAD6
3719	020136	004737	013356		CALL	CHKMAX
3720	020142			ENDMSG		
3721						
3722	020144			BGNMSG	PRBCOM	
3723	020144	005037	002274		CLR	GOOD
3724	020150	005037	002316		CLR	BAD
3725	020154	013704	002430		MOV	ERRCNT,R4
3726	020160	012703	002432		MOV	#BADLOC,R3
3727	020164				PRINTB	#MBBUF0,XMTBUF,RCVBUF
3728	020214				PRINTB	#MBBUF1,LENGTH,ERRCNT
3729	020244				PRINTB	#CDBFER
3730						
3731	020264	012702	000010		MOV	#8.,R2
3732	020270			1\$:		
3733	020270	012337	002420		MOV	(R3)+,XMTADD
3734	020274	012337	002422		MOV	(R3)+,RCVADD
3735	020300	117737	162114	002274	MOVB	#XMTADD,GOOD
3736	020306	117737	162110	002316	MOVB	#RCVADD,BAD
3737	020314				PRINTB	#MXMTER,XMTADD,GOOD
3738	020344				PRINTB	#MRCVER,RCVADD,BAD
3739	020374	005304			DEC	R4
3740	020376	001401			BEQ	2\$
3741	020400	077245			SOB	R2,1\$
3742	020402	004737	013356	2\$:	CALL	CHKMAX
3743	020406			ENDMSG		
3744						
3745	020410			BGNMSG	PRBNUM	
3746	020410				PRINTB	#MBNUM,OUTNUM,RSPCNT
3747	020440	004737	013356		CALL	CHKMAX
3748	020444			ENDMSG		

3750
3751
3752
3753
3754
3755
3756
3757 020446
3758
3759 020446
3760
3761 020452
3762

```
.SBTTL REPORT CODING SECTION  
; **  
; THE REPORT CODING SECTION CONTAINS THE  
; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.  
;  
BGNRPT  
EXIT RPT  
ENDRPT
```



```

3764          SBTTL  INITIALIZE SECTION
3765
3766
3767          ;
3768          ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
3769          ; AT THE BEGINNING OF EACH PASS.
3770          ;
3771 020454          BGNINIT
3772
3773
3808          .EVEN
3809
3810
3811 020454          SETVEC  #140,#170000,#340          ;ODT ROM ADDRESS          ;JB REV A 0
3812
3813 020502 012705 007330          MOV  #SSTACK,R5          ;INITIALIZE STACK
3814 020506 010637 002250          MOV  SP,PSTACK          ;STORE STACK POINTER
3815
3816 020512 005037 002414          CLR  APPFLG          ;CLEAR FLAG TO REQUEST LOAD OF THE
3817                                     ;APPLICATION FIRMWARE
3818
3819 020516 005737 002256          TST  FTIME          ;IS THE FIRST PASS?
3820 020522 001011                                     BNE  18          ;IF NOT
3821 020524 013737 000004 002252          MOV  @#4,SAVE4          ;SAVE TIME-OUT TRAP VECTOR IF YES
3822 020532 013737 000006 002254          MOV  @#6,SAVE6
3823 020540 012737 000001 002256          MOV  #1,FTIME          ;THEN SET FIRST PASS FLAG
3824
3825 020546 013737 002252 000004 18:          MOV  SAVE4,@#4          ;RESTORE TRAP VECTOR
3826 020554 013737 002254 000006          MOV  SAVE6,@#6
3827
3828          ;SEE IF PROGRAM JUST STARTED, BR IF YES
3829 020562          READEF #EF.START
3830 020570          BCOMPLETE      SETUP
3831
3832          ;SEE IF PROGRAM WAS JUST CONTINUED
3833 020572          READEF #EF.CONTINUE
3834 020600          BCOMPLETE      END
3835
3836          ;SEE IF THIS IS A NEW PASS, BR IF NOT
3837 020602          READEF #EF.NEW
3838 020610          BNCOMPLETE     NEXT
3839
3840 020612          SETUP:
3841 020612 012737 177777 002522          MOV  #1,UUT          ;SET LOGICAL DEVICE TO -1

```

```

3843
3844
3845 020620          NEXT:
3846
3847          ;TEST IF ALL UNITS TESTED
3848
3849 020620 005237 002522          INC      UUT
3850 020624 023737 002522 002270  CMP      UUT,L#UIT
3851 020632 001443          BEQ      ABORT          ;YES ABORT THE PASS
3852
3853 020634 013701 002522          MOV      UUT,R1
3854 020640          PRINTF  @RUNNING,R1
3855          .EVEN
3856          ; PRINTF  @MARK0          ;JB REV A-0
3857          ; .EVEN          ;JB REV A-0
3858          ; PRINTF  @MARK1          ;JB REV A-0
3859          ; .EVEN          ;JB REV A-0
3860
3861          ;GET P-TABLE IF AVAILABLE FOR THIS UNIT
3862 020662          GPHARD  UUT,R1
3863 020672          BNCOMPLETE  NEXT          ;IF NOT, TRY THE NEXT ONE
3864
3865          ;GET KVV11-C CSR ADDRESS
3866 020674 012137 002504          MOV      (R1)+,KVVCSR          ;LOAD CSR ADDRESS
3867
3868          ;GET KVV11 INTERRUPT VECTORS
3869 020700 011137 002506          MOV      (R1),INTIN
3870 020704 012137 002510          MOV      (R1)+,INTOUT
3871 020710 062737 000004 002510  ADD      @4,INTOUT
3872
3873          ;GET KVV11 PRIORITY
3874 020716 012137 002512          MOV      (R1)+,PRILEV
3875
3876          ;GET LOOP INDICATOR
3877 020722 012137 002344          MOV      (R1)+,LCLOOP
3878
3879          ;GET PDP11/23 TYPE
3880 020726 012137 002514          MOV      (R1)+,PDPTYP
3881
3882          ;RESET ERROR COUNTER
3883 020732 005037 002264          CLR      ERCNTR
3884
3885 020736          EXIT      INIT
3886
3887 020742          ABORT:  DOCLN          ;CLEAN UP THEN ABORT PASS
3888 020744          EXIT      INII
3889
3890 020750 000472          END:    BR      END1
3891 020752          045      116      045  RUNNING:  .ASCIZ  /#N#A RUNNING ON UNIT #D2#A /
3892 021006          045      116      045  RMARK0:   .ASCIZ  /#N#A SUBTEST 3 OF TEST 8 AND 9 ARE SKIPPED IF /
3893 021064          045      116      045  RMARK1:   .ASCIZ  /#N#A THE SYSTEM IS A PDP11 23 WITH 128K#N/
3894          .EVEN
3895
3896
3897 021136          END1:  ENDINIT
3898

```

```

3900          .SBTTL  AUTODROP SECTION
3901
3902
3903          ;**
3904          ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
3905          ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
3906          ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
3907          ; DROPPED FROM TESTING.
3908          ;-
3909 021140    .EVEN          BGNAUTO
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920          ;DEVICE DOES NOT HAVE A "READY"
3921
3922 021140    013701  002504          MOV      KMVCSR,R1      ;R1 CONTAINS BASE KMV11 ADDRESS
3923 021144    012705  000010          MOV      @CSRLEN,R5     ;NUMBER OF REGISTERS TO BE TESTED
3924 021150    012737  021200  000004  MOV      @28.4         ;SET OUT TIMEOUT TRAP
3925          ;
3926 021156    012737  000300  000006  MOV      @340.6        ;LEVEL 7                      ;JB REV A-0
3927 021164    005711          MOV      @300.6        ;LEVEL 6                      ;JB REV A 0
3928 021166    000240          1$:  TST      (R1)      ;REFERENCE DEVICE REGISTERS
3929 021170    062701  000002          NOP
3930 021174    077505          ADD      @2.R1         ;NEXT REGISTER
3931 021176    000405          SOB     R5.18         ;LOOP TILL ALL ADDRESSED
3932          BR      3$
3933 021200    062706  000004          2$:  ADD      @4.SP
3934 021204          DDDU   LOGDEV
3935
3936 021212    013737  002252  000004  3$:  MOV      SAVE4.4
3937 021220    013737  002254  000006  MOV      SAVE6.6
3938 021226          ENDAUTO
3939
3940
3941
3942

```

3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983

```
.SBTTL CLEANUP CODING SECTION  
;/////////////////////////////////////////////////////////////////  
; / THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED  
; / AT THE END OF EACH PASS.  
;/////////////////////////////////////////////////////////////////  
                BGNCLN  
  
                BRESET  
  
                ENDCLN
```

3985
3986
3987
3988
3989
3990
3991
3992 021234
3993
3994
3995
4004 021234
4005 021256
4006
4018 021262
4019
4020
4021
4022 021314
4023
4024
4025
4026
4027

.SBTTL DROP UNIT SECTION
;////////////////////////////////////
;// THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
;// TO NO LONGER BE TESTED.
;////////////////////////////////////

BGNDU

PRINTF #MDROP,RO ;UNIT DROPPED
EXIT DU

045 116 045 MDROP: .ASCIZ /#N#A UNIT #D2#A DROPPED#N/
.EVEN

ENDDU

4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4047
4048 021316
4049 021316
4050
4051

.SBTTL ADD UNIT SECTION

```
;///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////  
;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE  
;/ TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF  
;/ "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.  
;///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

BGNAU
ENDAU

4053
4054
4058
4067
4101
4102
4103 021320

BADHEAD

** TEST1 **

; VERIFY THAT THE KMV11-A CAN BE RESTARTED

4104
4105
4106
4107
4108
4109
4110
4111

; FIRST, VERIFY THAT REFERENCING Q-BUS DEVICE DOESN'T
; CAUSE A TIME OUT TRAP
; THEN, SELO IS LOADED FOR APPLICATION MODE

; MODE: APPLICATION MODE

4112
4113 021320

; REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR

BADHEAD

** TEST1 **

4114

BGNTST

4115 021320
4116 021320 013701 002504
4117 021324 012737 021346 000004
4118
4119 021332 012737 000300 000006
4120 021340 005711
4121 021342 000240
4122 021344 000423
4123

MOV KMVCSR,R1 ;R1 CONTAINS BASE KMV11 ADDRESS
MOV #2,4 ;SET OUT TIMEOUT TRAP
; MOV #340,6 ;LEVEL 7 ;JB REV A-0
MOV #300,6 ;LEVEL 6 ;JB REV A-0
1\$: TST (R1) ;REFERENCE DEVICE REGISTERS
NOP
BR 3\$;IF ADDRESS EXISTS

4124 021346 062706 000004
4125 021352 010137 002432
4126 021356 013737 002522 002272
4127 021364
4128 021374 013737 002252 000004
4129 021402 013737 002254 000006
4130 021410
4131

2\$: ADD #4,SP ;ELSE, REPORT
MOV R1,BADLOC ;ADDRESS LOCATION
MOV UUT,UNIT ;UNIT NUMBER
ERRMRD 0,E00000,BADAD ;BUS TIMEOUT,ADDRESS PROBLEM
MOV SAVE4,4 ;THEN RESTORE VECTOR AND EXIT
MOV SAVE6,6
EXIT TST

4132 021414 013737 002252 000004
4133 021422 013737 002254 000006
4134
4135

3\$: MOV SAVE4,4
MOV SAVE6,6

4136 021430 004737 011574

CALL MODE0 ;SET MASTER CLEAR AND APPLICATION MODE
;AND WAIT FOR SELO CLEARED BY THE KMV

4137
4138

4139 021434
4140 021442

WAIT 10000.
ENDTST

4142 021444
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155 021444
4156
4157 021444
4158 021444 004737 011574
4159 021450 103002
4160
4161 021452
4162
4163
4164
4165 021456 005065 000000
4166 021462 052765 140000 000000
4167 021470 012701 000014
4168 021474
4169 021474
4170 021502 032765 140000 000000
4171 021510 001412
4172 021512 077110
4173
4174 021514 005037 002276
4175 021520 016537 000000 002320
4176 021526
4177
4178 021536
4179 021536
4180 021544

```

BADHEAD
;                               ** TEST2 **
; SELF-TEST RUNNING TEST
;
; WHEN SELF TEST IS SELECTED, THIS TEST PERMITS TO SEE
; IF THE KMV11-A CAN RUN IT BY ASSERTING RUN BIT IN THE SAME
; TIME THAN MASTER CLEAR IN BSEL1(IN THIS CASE THE SELF TEST
; RUNS ONE PASS ONLY.THEN, IF SELF TEST CORRECTLY RUNS, APPLICATION
; MODE IS ACCESSED AND BSEL1 IS CLEARED.
;
; MODE:           APPLICATION MODE
;
; REPORTS:        ERROR 1           SELF TEST DIDNOT RUN CORRECTLY
;                 ERROR 10000      KMV11 FAILS TO RESET MASTER CLEAR
BADHEAD
;                               ** TEST2 **
;
BGNTST
CALL   MODE0           ;SET APPLICATION MODE
BCC    1#              ;IF NO ERROR, JUMP
EXIT   TST             ;ELSE, REPORT ERROR
; RUN SELF TEST
1#:   CLR   SELO(R5)   ;RESET SELO
      BIS   #MCLR!RUN,SELO(R5) ;RUN IT
      MOV   #12.,R1    ;INIT TIME OUT(20 S)
2#:   WAIT  10000.     ;WAIT FOR 1S
      BIT   #MCLR!RUN,SELO(R5) ;TEST FOR SELO CLEARED?
      BEQ   3#         ;IF YES
      SOB   R1,2#     ;IF NOT, LOOP TILL TIME OUT
      CLR   G0000
      MOV   SELO(R5),BADO
      ERHRD 1,E00001,PRSELO ;IF TIME OUT
3#:   WAIT  10000.
ENDTST

```


4182 021546

BADHEAD

** TEST3 **

4183

; TEST IF APPLICATION PROGRAM CAN BE LOADED AND STARTED

4184

4185

; MODE: APPLICATION MODE

4186

4187

; REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR
ERROR 10001 RUN FUNCTION NOT CORRECTLY PERFORMED
ERROR 10002 WRITE FUNCTION NOT CORRECTLY PERFORMED
ERROR 10003 READ FUNCTION NOT CORRECTLY PERFORMED
ERROR 10004 DATA COMPARE ERROR
ERROR 10008 UNEXPECTED EPROM'S ON KMV11 A(ECO LEVEL
OF EPROM'S IS BAD)

4188

4189

4190

4191

4192

4193

4194 021546

BADHEAD

** TEST3 **

4195

4196

4197 021546

BGNTST

4198 021546 004737 011574

CALL MODE0 ,SET APPLICATION MODE
BCC 1\$;IF CORRECTLY DONE

4199 021552 103002

4200

EXIT TST ;IF NOT, REPORT

4201 021554

4202

4203 021560 004737 012554

1\$: CALL RUNAPP ;LOAD AND RUN APPLICATION

4204

4205 021564

ENDTST

```

4207 021566      BADHEAD
                  ;
                  ; ** TEST4 **
4208             ; TEST THE CSR HANDSHAKING WITHOUT INTERRUPT
4209             ;
4210             ; COMMAND LOADED:          READ MODEM (F14)
4211             ;
4212             ; RESPONSE EXPECTED:      READ MODEM RESPONSE WITH :
4213             ;                               STATUS = 371
4214             ;                               MODEM = ALL OFF
4215             ;
4216             ; REPORTS:                ERROR 10000   KMV11 FAILS TO RESET MASTER CLEAR
4217             ;                               ERROR 10001   RUN FUNCTION NOT CORRECTLY PERFORMED
4218             ;                               ERROR 10002   WRITE FUNCTION NOT CORRECTLY PERFORMED
4219             ;                               ERROR 10003   READ FUNCTION NOT CORRECTLY PERFORMED
4220             ;                               ERROR 10004   DATA COMPARE ERROR
4221             ;                               ERROR 10005   UNEXPECTED INTERRUPT IN
4222             ;                               ERROR 10006   UNEXPECTED INTERRUPT OUT
4223             ;                               ERROR 10007   NO MORE INTERRUPT WHILE QIO PENDING
4224             ;                               ERROR 10008   UNEXPECTED EPROM'S ECO LEVEL ON KMV11-A
4225             ;                               ERROR 10009   UNEXPECTED QIO RESPONSE
4226             ;                               ERROR 00002   RDYI NOT ASSERTED BY THE KMV AFTER AN INPUT REQ
4227             ;                               ERROR 00003   RDYO NOT ASSERTED BY THE KMV IN RESPONSE
4228             ;                               ERROR 00004   UNEXPECTED INTERRUPT RECEIVED WHEN IEI&IEO
4229             ;                               ARE DISABLED
4230 021566      BADHEAD
                  ;
                  ; ** TEST4 **
4231
4232
4233 021566      BGNTST
4234 021566      004737 011574      CALL      MODE0      ;SET APPLICATION MODE
4235 021572      103002              BCC        1$          ;IF CORRECTLY DONE
4236
4237 021574              EXIT      TST
4238
4239 021600      004737 012554      1$:      CALL      RUNAPP      ;LOAD AND RUN APPLICATION
4240 021604      103002              BCC        2$          ;IF CORRECTLY DONE
4241
4242 021606              EXIT      TST
4243
4244 021612      004737 011776      2$:      CALL      INIQIO      ;INIT QIO PROCESSING
4245 021616      012704 000005      MOV        #5,R4      ;SET WATCH DOG
4246 021622      012765 000200 000000 MOV        #RQI,SEL0(R5) ;REQUEST CSR TRANSACTION
4247 021630      3$:
4248 021630              WAIT      100.          ;WAIT FOR 1MS
4249 021636      032765 000020 000002 BIT        #RDYI,SEL2(R5) ;IS RDYI SET BY KMV?
4250 021644      001020              BNE        4$          ;IF YES
4251 021646      077410              SOB        R4,3$      ;NOT, LOOP TILL TIME-OUT
4252
4253 021650      012737 000002 002416 MOV        #2,SELNUM   ;RDYI NEVER SET, REPORT
4254 021656      016537 000002 002316 MOV        SEL2(R5),BAD
4255 021664      012737 000020 002274 MOV        #RDYI,GOOD
4256 021672              ERRHRD  2,E00002,PRSEL
4257
4258 021702              EXIT      TST
4259
4260 021706      012701 022066      4$:      MOV        #7#+2,R1   ;RDYI SET, POINT TO COMMAND
4261 021712      012102              MOV        (R1)+,R2   ;SAVE COMMAND NUMBER

```

```

4262 021714 012165 000004          MOV      (R1)+,SEL4(R5)          ;SET SEL4
4263 021720 012165 000006          MOV      (R1)+,SEL6(R5)          ;SET SEL6
4264 021724 042765 000200 000000    BIC      @RQI,SELO(R5)          ;THEN CLEAR RQI
4265 021732 010265 000002          MOV      R2,SEL2(R5)           ;AND SET SEL2
4266
4267 021736 012704 000005          MOV      @5,R4                  ;SET WATCH DOG
4268 021742
4269 021742
4270 021750 032765 000200 000002    5$:     WAIT      100.            ;WAIT 1MS
4271 021756 001020                    BIT      @RDY0,SEL2(R5)        ;IS RDY0 SET BY KMV?
4272 021760 077410                    BNE      6$                    ;IF YES
4273
4274 021762 012737 000002 002416    SOB      R4,5$                 ;NOT, LOOP TILL TIME OUT
4275 021770 016537 000002 002316    MOV      @2,SELNUM             ;RDY0 NEVER SET, REPORT
4276 021776 012737 000200 002274    MOV      SEL2(R5),BAD
4277 022004
4278 022014
4279
4280 022020 016537 000002 006620    6$:     MOV      SEL2(R5),OUTBUF
4281 022026 016537 000004 006622    MOV      SEL4(R5),OUTBUF+2
4282 022034 016537 000006 006624    MOV      SEL6(R5),OUTBUF+4
4283 022042 005237 002402          INC      RSPCNT
4284
4285 022046 012701 022074          MOV      @8$,R1                ;LOAD CORRECT RESPONSE FOR TEST
4286 022052 004737 010134          CALL     CHKRSP                 ;THEN CHECK RESPONSE
4287 022056 103012                    BCC      9$                     ;IF RESPONSE CORRECT
4288
4289 022060          EXIT      TST
4290
4291          ; COMMAND LIST
4292
4293 022064
4294 022064 000001
4295 022066 000014 000000 000000    7$:     1
14,0,0          ;READ MODEM
4296
4297          ; RESPONSE LIST
4298
4299 022074
4300 022074 000001
4301 022076 000214 000000 174400    8$:     1
14+RDY0,0,371+400  ;READ MODEM RESPONSE
4302
4303 022104
4304 022104
9$:     ENDTST

```

4306 022106

BADHEAD

** TESTS **

4307

; TEST THE CSR HANDSHAKING WITH INTERRUPTS

4308

; COMMAND LOADED: READ MODEM (F14)

4309

4310

; RESPONSE EXPECTED: READ MODEM RESPONSE WITH :

4311

4312

STATUS = 371

4313

MODEM = ALL OFF

4314

; REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR

4315

4316

4317

4318

4319

4320

4321

4322

4323

4324

BADHEAD

** TESTS **

4325 022106

4326

4327

4328 022106

BGNTST

4329 022106 004737 011574

CALL MODEM ;SET APPLICATION MODE
BCC 1; ;IF CORRECTLY DONE

4330 022112 103002

4331

4332 022114

EXIT TST

4333

4334 022120 004737 012554

1;: CALL RUNAPP ;LOAD AND RUN APPLICATION
BCC 2; ;IF CORRECTLY DONE

4335 022124 103002

4336

4337 022126

EXIT TST

4338

4339 022132 004737 011776

2;: CALL INIQIO ;INIT QIO PROCESSING

4340

4341 022136 012701 022154

MOV #60\$,R1 ;TABLE ADDRESS IN R1

4342 022142 004737 013154

CALL QIOP ;PROCESS QIO

4343 022146 103014

BCC 5; ;IF CORRECTLY DONE

4344

4345 022150

EXIT TST

4346

4347

; PARAMETERS FOR QIO PROCESSING

4348

4349 022154 022160

60\$: 3; ;IN LIST TABLE BASE ADDRESS

4350 022156 000062

50. ;TIME OUT LENGTH(N*10 MS)

4351

4352

; COMMAND LIST

4353

4354 022160

3;:

4355 022160 000001

1

4356 022162 000014 000000 000000

14,0,0 ;READ MODEM

4357

4358

; RESPONSE LIST

4359

4360 022170

4;:

```
4361 022170 000001          1
4362 022172 000214 000000 174400 14*RDY0.0.371*400          ;READ MODEM RESPONSE
4363
4364 022200          5$:
4365 022200 012701 022170      MOV    04$,R1          ;SAVE RESPONSE LIST ADDRESS
4366 022204 004737 010134      CALL   CHKRSP         ;THEN CHECK RESPONSES
4367
4368 022210          6$:
4369 022210          ENDTST
```

4371 022212

4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393

4394 022212

4395
4396

4397 022212

4398 022212 004737 011574

4399 022216 103430

4400
4401

4402

4403 022220 013737 030742 002374

4404 022226 013737 030704 002376

4405 022234 012737 000207 030742

4406 022242 012737 000207 030704

4407

4408 022250 005037 002414

4409 022254 004737 012554

4410 022260 103407

4411

4412 022262 004737 011776

4413

4414 022266 012701 022302

4415 022272 004737 013154

4416 022276 103153

4417

4418 022300 000556

4419

4420

4421

4422 022302 022306

4423 022304 001274

4424

4425

BADHEAD

```

;
; ** TEST6 **
; TEST OF QIO PROCESSING IN CASE OF KMV RESOURCE ERROR
;
; TO DO THIS TEST, THE KMV POOL IS EMPTIED BY MAKING ITS LENGTH=0
; THEN THE APPLICATION CODE IS LOADED IN THE KMV11-A
;
; COMMAND LOADED:      ALL COMMANDS
;
; RESPONSE EXPECTED:  FOR EACH COMMAND,
;                       STATUS = 357
;                       STATE = S1      (READ COMMAND ONLY)
;                       MODEM = ALL OFF(MODEM SURVEY & READ CMD ONLY)
;
; REPORTS:      ERROR 10000  KMV11 FAILS TO RESET MASTER CLEAR
;                ERROR 10001  RUN FUNCTION NOT CORRECTLY PERFORMED
;                ERROR 10002  WRITE FUNCTION NOT CORRECTLY PERFORMED
;                ERROR 10003  READ FUNCTION NOT CORRECTLY PERFORMED
;                ERROR 10004  DATA COMPARE ERROR
;                ERROR 10005  UNEXPECTED INTERRUPT IN
;                ERROR 10006  UNEXPECTED INTERRUPT OUT
;                ERROR 10007  NO MORE INTERRUPT WHILE QIO PENDING
;                ERROR 10008  UNEXPECTED EPROM'S ON KMV11-A
;                ERROR 10009  UNEXPECTED QIO RESPONSE
;
BADHEAD

```

BADHEAD

; ** TEST6 **

BGNTST

```

; SET APPLICATION MODE
; IF NOT CORRECTLY DONE.
;
; EMPTIED KMV POOL
;
; MOV      %RDBIN,TEMP      ;SAVE THIS LOCATION
; MOV      %CCBIN,TEMP*2    ;SAVE IT TOO
; MOV      %207,%RDBIN     ;SET RETURN INSTEAD OF IT
; MOV      %207,%CCBIN
;
; CLR      APPFLG          ;CLEAR FLAG
; CALL    RUNAPP          ;LOAD APPLI. CODE
; BCS     %4              ;EXIT IF ANY ERROR
;
30:  CALL  INIQIO          ;INIT QIO PROCESSING
;
; MOV      %60,%R1        ;TABLE ADDRESS IN R1
; CALL    QIOP            ;PROCESS QIO
; BCC     %7              ;IF CORRECTLY DONE
;
40:  BR    %8              ;EXIT
;
; PARAMETERS FOR QIO PROCESSING
;
600:  %5              ;IN LIST TABLE BASE ADDRESS
;      %700           ;TIME OUT LENGTH(N*10 MS)
;
; COMMAND LIST

```

```

4426
4427 022306
4428 022306 000020
4429 022310 000000 000000 000000
4430 022316 000001 000000 000000
4431 022324 000002 000000 000000
4432 022332 000003 000000 000000
4433 022340 000004 000000 000000
4434 022346 000005 000000 000000
4435 022354 000006 000000 000000
4436 022362 000007 000000 000000
4437 022370 000010 000000 000000
4438 022376 000011 000000 000000
4439 022404 000012 000000 000000
4440 022412 000013 000000 000000
4441 022420 000014 000000 000000
4442 022426 000015 000000 000000
4443 022434 000016 000000 000000
4444 022442 000017 000000 000000
4445
4446
4447
4448 022450
4449 022450 000020
4450 022452 000200 000000 167400
4451 022460 000201 000000 167400
4452 022466 000202 000000 167400
4453 022474 000203 000000 167400
4454 022502 000204 000000 167400
4455 022510 000205 000000 167400
4456 022516 000206 000000 167400
4457 022524 000207 000000 167400
4458 022532 000210 000000 167400
4459 022540 000211 000000 167400
4460 022546 000212 000000 167400
4461 022554 000213 000000 167400
4462 022562 000214 000000 167400
4463 022570 000215 000000 167400
4464 022576 000216 000000 167400
4465 022604 000217 000000 167400
4466 022612 000000 000000 000000
4467
4468 022626
4469 022626 012701 022450
4470 022632 004737 010134
4471
4472
4473 022636 005037 002414
4474 022642 013737 002374 030742
4475 022650 013737 002376 030704
4476
4477 022656

```

54:

```

16.
0.0.0
1.0.0
2.0.0
3.0.0
4.0.0
5.0.0
6.0.0
7.0.0
10.0.0
11.0.0
12.0.0
13.0.0
14.0.0
15.0.0
16.0.0
17.0.0

```

```

; DUMMY
; CONFIGURATE
; DECONFIGURATE
; DUMMY
; DUMMY
; XMIT BUFFER
; RECEIVE BUFFER
; XMIT KILL
; RECEIVE KILL
; DUMMY
; DUMMY
; DUMMY
; READ MODEM
; DUMMY
; ENABLE MODEM SURVEY
; DISABLE MODEM SURVEY

```

64:

```

16.
0.RDY0,0.357*400
1.RDY0,0.357*400
2.RDY0,0.357*400
3.RDY0,0.357*400
4.RDY0,0.357*400
5.RDY0,0.357*400
6.RDY0,0.357*400
7.RDY0,0.357*400
10.RDY0,0.357*400
11.RDY0,0.357*400
12.RDY0,0.357*400
13.RDY0,0.357*400
14.RDY0,0.357*400
15.RDY0,0.357*400
16.RDY0,0.357*400
17.RDY0,0.357*400
.WORD 0.0.0.0.0.0

```

```

; DUMMY
; KMV RESOURCE ERROR FOR F1
; KMV RESOURCE ERROR FOR F2
; DUMMY
; DUMMY
; KMV RESOURCE ERROR FOR F5
; KMV RESOURCE ERROR FOR F6
; KMV RESOURCE ERROR FOR F7
; KMV RESOURCE ERROR FOR F10
; DUMMY
; DUMMY
; DUMMY
; KMV RESOURCE ERROR FOR F14
; DUMMY
; KMV RESOURCE ERROR FOR F16
; KMV RESOURCE ERROR FOR F17

```

74:

```

MOV #64,R1
CALL CHKRSP

```

```

; SAVE RESPONSE LIST ADDRESS
; THEN CHECK RESPONSES

```

84:

```

CLR APPFLG
MOV TEMP,8RDBIN
MOV TEMP,2,8CCBIN

```

```

; TO LOAD AGAIN CODE
; RESTORE INITIAL CONTENTS

```

ENDTST

4479 022660

4480
 4481
 4482
 4483
 4484
 4485
 4486
 4487
 4488
 4489
 4490
 4491
 4492
 4493
 4494
 4495
 4496
 4497
 4498
 4499
 4500
 4501
 4502
 4503
 4504
 4505
 4506
 4507
 4508
 4509
 4510
 4511
 4512
 4513
 4514
 4515
 4516
 4517
 4518
 4519
 4520
 4521
 4522
 4523
 4524
 4525
 4526
 4527
 4528
 4529
 4530
 4531
 4532
 4533
 4534

```

BADHEAD
;
;          ** TEST7 **
; TEST OF QIO PROCESSING AS BELOW:
;
; SUBTEST1 - STATE = S1          ALL THE COMMANDS EXCEPT F1 ARE PASSED
;                               WHILE THE LINE IS NOT CONFIGURED.
;                               EXPECTED STATUS ARE:
;                               STATUS = 371 FOR F2,F5,F6,F7,F10,F14
;                               * 1   FOR F16,F17
;                               * NONE FOR DUMMY COMMANDS
;
; SUBTEST2 - ENTER COMMAND F16 TWICE
;            ENTER COMMAND F1 TWICE FOR:
;            SDLC PROTOCOL
;            FULL MODEM CONTROL
;            CLOCK SOURCE INTERNAL
;            RATE = 2.4K
;
;            EXPECTED RESPONSES ARE:
;            STATUS = 363 FOR SECOND F16
;            * 363 FOR SECOND F1
;            * 1   FOR F16 WITH MODEM OFF
;            * 1   FOR F4 WITH S109 CHANGE
;            * 1   FOR F4 WITH S106 CHANGE
;            * 1   FOR F4 WITH S107 CHANGE
;            * 1   FOR F1
;
;            CURRENT STATE = S6
;            MODEM = ALL ON(EXCEPT S125 WHICH IS LOOPED ON
;                               THE TERMINAL IN SERVICE SIGNAL =OFF)
;
; SUBTEST3 - DECONFIGURATE THE LINE
;            ENTER COMMAND F17 TWICE
;            ENTER COMMAND F14 TWICE
;            ENTER COMMAND F16 TWICE
;            ENTER COMMAND F1
;            "   "   F2 TWICE
;            "   "   F1
;
;            EXPECTED RESPONSES ARE:
;            STATUS = 363 FOR F17
;            * 363 FOR F16
;            * 371 FOR F1
;            * 1 FOR F17
;            * 363 FOR F2
;            * 356 FOR F1
;            * 1 FOR F14 WITH STATE=S6 AND S142,S112,S109,S106
;                               *S107 ON
;            * 1 FOR F16 WITH THE SAME PARAMETERS
;            * 1 FOR F4 WITH S106 & S109 CHANGE
;            * 1 FOR F4 WITH S107 CHANGE
;            * 1 FOR F2 COMPLETED
;
;            THEN CURRENT STATE = S1
;            MODEM = ALL OFF
;
; SUBTEST4 - TEST THAT WE ARE OFF LINE AGAIN
  
```



```

4591 023020 000000 000000 000000      0,0,0      ;DUMMY COMMANDS
4592 023026 000003 000000 000000      3,0,0
4593 023034 000004 000000 000000      4,0,0
4594 023042 000011 000000 000000     11,0,0
4595 023050 000012 000000 000000     12,0,0
4596 023056 000015 000000 000000     15,0,0
4597
4598      ; RESPONSE LIST
4599
4600 023064      41:
4601 023064 000010      8.
4602 023066 000214 000000 174400     14*RDY0,0,371*400      ;F14 OUT OF SEQUENCE
4603 023074 000202 000000 174400     2*RDY0,0,371*400      ;F2 OUT OF SEQUENCE
4604 023102 000205 000000 174400     5*RDY0,0,371*400      ;F5 OUT OF SEQUENCE
4605 023110 000206 000000 174400     6*RDY0,0,371*400      ;F6 OUT OF SEQUENCE
4606 023116 000207 000000 174400     7*RDY0,0,371*400      ;F7 OUT OF SEQUENCE
4607 023124 000210 000000 174400     10*RDY0,0,371*400     ;F10 OUT OF SEQUENCE
4608 023132 000216 000000 000400     16*RDY0,0,1*400      ;CORRECT RESPONSE FOR F16
4609 023140 000217 000000 000400     17*RDY0,0,1*400      ;CORRECT RESPONSE FOR F17
4610 023146 000000 000000 000000     .WORD 0,0,0,0,0,0
4611
4612 023162      51:
4613 023162 012701 023064      MOV  #41,R1      ;SAVE RESPONSE LIST ADDRESS
4614 023166 004737 010134      CALL CMKRSP      ;THEN CHECK RESPONSES
4615
4616 023172      ENDSUB
4617
4618 023174      BGNSUB
4619 023176 004737 011776      CALL INIQIO      ;INIT QIO PROCESSING
4620
4621 023202 012701 023220      MOV  #60,R1      ;TABLE ADDRESS IN R1
4622 023206 004737 013154      CALL QIOP        ;PROCESS QIO
4623 023212 103055      BCC  51          ;IF CORRECTLY DONE
4624
4625 023214      EXIT  SUB
4626
4627      ; PARAMETERS FOR QIO PROCESSING
4628
4629 023220 023224      601:  31          ;IN LIST TABLE BASE ADDRESS
4630 023222 001274      700.          ;TIME OUT LENGTH(N*10 MS)
4631
4632      ; COMMAND LIST
4633
4634 023224      31:
4635 023224 000004      4
4636 023226 007416 000000 000000     16*««S106*S109*S125*S107»*400»,0,0      ;ENABLE MODEM SURVEY
4637 023234 007416 000000 000000     16*««S106*S109*S125*S107»*400»,0,0      ;ENABLE MODEM SURVEY
4638 023242 000001 100425 000000     1,SDLC*S111*S141*CLKDTE*C2400,0      ;CONFIGURATE
4639 023250 000001 100401 000000     1,SDLC*CLKDTE*C2400,0      ;CONFIGURATE
4640
4641      ; RESPONSE LIST
4642
4643 023256      41:
4644 023256 000007      7
4645 023260 000216 000000 171400     16*RDY0,0,363*400      ;DOUBLE COMMAND FOR F16
4646 023266 000201 000000 171400     1*RDY0,0,363*400      ;DOUBLE COMMAND F1
4647 023274 000216 000000 000400     16*RDY0,0,1*400      ;CORRECT RESPONSE FOR F16

```

```

4648 023302 000204 000401 000400      4.RDY0,S109*400,S109,1*400      ;S109 MODEM CHANGE
4649 023310 000204 004011 000400      4.RDY0,S106*400,S109,S106,1*400 ;S106 MODEM CHANGE
4650 023316 000204 002015 000400      4.RDY0,S107*400,S109,S107,S106,1*400 ;S107 MODEM CHANGE
4651 023324 000201 000000 000400      1.RDY0,0,1*400                  ;CORRECT RESPONSE FOR F1
4652
4653 023332 000000 000000 000000      .WORD 0,0,0,0,0,0
4654
4655 023346
4656 023346 012701 023256      5$:      MOV      #4$,R1                ;SAVE RESPONSE LIST ADDRESS
4657 023352 004737 010134      CALL     CKRSP                    ;THEN CHECK RESPONSES
4658
4659 023356      ENDSUB
4660
4661 023360      BGNSUB
4662 023362 004737 011776      CALL     INIQIO                   ;INIT QIO PROCESSING
4663
4664 023366 012701 023404      MOV      #60$,R1                 ;TABLE ADDRESS IN R1
4665 023372 004737 013154      CALL     QIOP                     ;PROCESS QIO
4666 023376 103110      BCC     5$                       ;IF OKE
4667
4668 023400      EXIT     SUB
4669
4670      ; PARAMETERS FOR QIO PROCESSING
4671
4672 023404 023410      60$:      3$                       ;IN LIST TABLE BASE ADDRESS
4673 023406 000764      500.                          ;TIME-OUT LENGTH(N*10 MS)
4674
4675      ; COMMAND LIST
4676
4677 023410      3$:
4678 023410 000011      9.
4679 023412 000017 000000 000000      17,0,0                          ;DISABLE MODEM SURVEY
4680 023420 000017 000000 000000      17,0,0                          ;DISABLE MODEM SURVEY
4681 023426 000014 000000 000000      14,0,0                          ;READ MODEM
4682 023434 006016 000000 000000      16,<<<S107,S106>*400>,0,0        ;ENABLE MODEM SURVEY
4683 023442 006016 000000 000000      16,<<<S107,S106>*400>,0,0        ;ENABLE MODEM SURVEY
4684 023450 000001 101440 000377      1,CLKDTE,C9600,DLO,377
4685 023456 000002 000000 000000      2,0,0                            ;CONFIGURATE
4686 023464 000002 000000 000000      2,0,0                            ;DECONFIGURATE
4687 023472 000001 101400 000000      1,CLKDTE,C9600,0                 ;DECONFIGURATE
4688
4689      ; CONFIGURATE
4690
4691 023500      ; RESPONSE LIST
4692 023500 000013      4$:      11.
4693 023502 000217 000000 171400      17.RDY0,0,363*400                ;DOUBLE COMMAND FOR F17
4694 023510 000216 000000 171400      16.RDY0,0,363*400                ;DOUBLE COMMAND FOR F16
4695 023516 000201 000000 174400      1.RDY0,0,371*400                 ;F1 OUT OF SEQ
4696 023524 000217 000000 000400      17.RDY0,0,1*400                  ;OKE FOR F17
4697 023532 000202 000000 171400      2.RDY0,0,363*400                 ;DOUBLE COMMAND FOR F2
4698 023540 000201 000000 167000      1.RDY0,0,356*400                 ;CONFIGURATE PENDING
4699 023546 000214 001475 000400      14.RDY0,S6*400,S142,S112,S106,S107,S109,1*400 ;READ MODEM RESPONSE
4700 023554 000216 001475 000400      16.RDY0,S6*400,S142,S112,S106,S107,S109,1*400 ;F16 CORRECT RESPONSE
4701 023562 000204 004004 000400      4.RDY0,S106*400,S107,1*400       ;S106 MODEM CHANGE
4702 023570 000204 002000 000400      4.RDY0,S107*400,1*400           ;S107 " "
4703 023576 000202 000000 000400      2.RDY0,0,1*400                  ;CORRECT RESPONSE FOR F2
4704

```

```

4705 023604 000000 000000 000000      .WORD  0.0,0.0,0.0,0
4706
4707 023620      5$:
4708 023620 012701 023500      MOV    #4$,R1      ;SAVE RESPONSE LIST ADDRESS
4709 023624 004737 010134      CALL   CHKRSP      ;CHECK QIO
4710
4711 023630      ENDSUB
4712
4713 023632      BGNSUB
4714 023634 004737 011776      CALL   INIQIO      ;INIT QIO PROCESSING
4715
4716 023640 012701 023656      MOV    #60$,R1     ;TABLE ADDRESS IN R1
4717 023644 004737 013154      CALL   QIOP        ;PROCESS QIO
4718 023650 103030      BCC    5$         ;IF OKE
4719
4720 023652      EXIT   SUB
4721
4722      ; PARAMETERS FOR QIO PROCESSING
4723
4724 023656 023662      60$:   3$        ;IN LIST TABLE BASE ADDRESS
4725 023660 000310      200.      ;TIME OUT LENGTH(N*10 MS)
4726
4727      ; COMMAND LIST
4728
4729 023662      3$:
4730 023662 000002      2
4731 023664 000014 000000 000000      14.0,0      ;READ MODEM
4732 023672 000017 000000 000000      17.0,0      ;DISABLE MODEM SURVEY
4733
4734      ; RESPONSE LIST
4735
4736 023700      4$:
4737 023700 000002      2
4738 023702 000214 000000 174400      14.RDY0,0.371*400      ;OUT OF SEQ. FOR F14
4739 023710 000217 000000 171400      17.RDY0,0.363*400      ;DOUBLE COMMAND FOR F17
4740
4741 023716 000000 000000 000000      .WORD  0.0,0.0,0.0,0
4742
4743 023732      5$:
4744 023732 012701 023700      MOV    #4$,R1      ;SAVE RESPONSE LIST ADDRESS
4745 023736 004737 010134      CALL   CHKRSP      ;THEN CHECK RESPONSES
4746
4747 023742      ENDSUB
4748
4749 023744      ENDTST

```

4751 023746

BADHEAD

4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806

```

;
;          ** TEST8 **
; TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 2.4K
; WITH FULL MODEM CONTROL
;
; CONFIGURATE:  SDLC PROTOCOL
;                FULL MODEM CONTROL
;                WITHOUT ADDRESS SEARCH
;                CLOCK SOURCE INTERNAL
;                RATE = 2.4K
;
;          FOR EACH SUBTEST, A COMMAND F14 IS GIVEN FIRST TO CONTROL
;          MODEMS AND THE LINE STATE, SUCH AS:
;                STATE = S6
;                MODEM = S!42.S112.S109.S106.S107 ON
;
; SUBTEST1  - TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
;            OF 2 BUFFERS.
;            THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
;            XMIT BUFFERS ARE GIVEN ONE OF WHICH IS 1 BYTE LONG
;            EXPECTED RESPONSES ARE:
;            STATUS = 360 FOR F6 THIRD BUFFER PASSED
;                   = 372 FOR F5 1 BYTE LONG BUFFER
;                   = 360 FOR F5 THIRD BUFFER PASSED
;                   = 1   FOR F5 FIRST BUFFER XMITTED
;                   = 1   FOR F6 FIRST BUFFER FELT
;                   = 1   FOR F5 SECOND BUFFER XMITTED
;                   = 1   FOR F6 SECOND BUFFER FELT
;
; SUBTEST2  TEST OF BUFFER OVERFLOW STATUS 373 FOR RECEPTION
;            A 32 BYTE LONG BUFFER IS AFFECTED FOR RECEIVE WHILE
;            A 100 BYTE LONG BUFFER IS PASSED TO XMIT.
;            EXPECTED RESPONSES ARE:
;            STATUS = 1   FOR F5 BUFFER XMITTED
;                   = 373 FOR F6 BUFFER OVERFLOW
;
; SUBTEST3  - TEST OF NON EXISTENT MEMORY STATUS 374
;            TWO RECEIVE BUFFER ARE AFFECTED ONE OF WHICH CONTAINS
;            A NON EXISTENT ADDRESS
;            TWO XMIT BUFFERS ARE PASSED ONE OF WHICH CONTAINS A NON
;            EXISTENT ADDRESS
;            EXPECTED RESPONSES ARE:
;            STATUS = 374 FOR F6 NON EXISTENT MEMORY BUFFER
;                   = 1   FOR F5 COORRECT BUFFER
;                   = 374 FOR F5 NON EXISTENT MEMORY BUFFER
;                   = 364 FOR F6 RECEIVE ABORT
;
; SUBTEST4  TEST OF ABORT RECEIVED STATUS 364 AND ACTION OF XMIT
;            ABORT COMMAND.
;            ONE RECEIVE BUFFER IS USED THEN, TWO XMIT BUFFERS
;            ARE PASSED. AFTER A SHORT WAIT, XMIT ABORT IS GIVEN
;            EXPECTED STATUS ARE:
;            STATUS = 1   FOR F7 XMIT ABORTED
;                   = 364 FOR F6 ABORT RECEIVED
;
; SUBTEST5  - TEST THE ACTION OF RECEIVE ABORT COMMAND.
;            TWO RECEIVE BUFFERS ARE USED THEN, TWO XMIT BUFFERS

```

```

4807 ; ARE USED. AFTER A SHORT WAIT, RECEIVE ABORT IS GIVEN
4808 ; EXPECTED STATUS ARE:
4809 ; STATUS = 1 FOR F10 RECEIVE ABORTED
4810 ; = 1 FOR F5 FIRST BUFFER SENT
4811 ; = 1 FOR F5 SECOND BUFFER SENT
4812 ;
4813 ; SUBTEST6 - TEST OF KMV OVERFLOW DURING XMIT/RECEIVE BUFFER
4814 ; ONE RECEIVE BUFFER IS USED,
4815 ; THEN TWO XMIT BUFFER ARE SENT, THE FIRST ONE TOO
4816 ; LONG.
4817 ; EXPECTED STATUS ARE:
4818 ; STATUS = 373 FOR FIRST F5
4819 ; = 1 FOR SECOND F5
4820 ; = 1 FOR F6
4821 ;
4822 ; SUBTEST7 TEST OF A DECONFIGURATE DURING XMIT/RECEIVE BUFFER,
4823 ; TWO RECEIVE BUFFERS ARE USED THEN, TWO XMIT BUFFERS
4824 ; ARE PASSED. AFTER A SHORT WAIT, DECONFIGURATE COMMAND
4825 ; IS ISSUED. EXPECTED STATUS IS:
4826 ; STATUS = 1 FOR F2 DECONFIGURATE DONE
4827 ; AND ALL BUFFERS ARE KILLED
4828 ;
4829 ; PATTERN: INCREMENTAL
4830 ; FOR EACH TEST WHERE A CORRECT BUFFER IS OBTAINED, THIS BUFFER
4831 ; IS COMPARED TO THE CORRESPONDING XMIT BUFFER
4832 ;
4833 ; REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR
4834 ; ERROR 10001 RUN FUNCTION NOT CORRECTLY PERFORMED
4835 ; ERROR 10002 WRITE FUNCTION NOT CORRECTLY PERFORMED
4836 ; ERROR 10003 READ FUNCTION NOT CORRECTLY PERFORMED
4837 ; ERROR 10004 DATA COMPARE ERROR
4838 ; ERROR 10005 UNEXPECTED INTERRUPT IN
4839 ; ERROR 10006 UNEXPECTED INTERRUPT OUT
4840 ; ERROR 10007 NO MORE INTERRUPT WHILE QIO PENDING
4841 ; ERROR 10008 UNEXPECTED EPROM'S ON KMV11 A
4842 ; ERROR 10009 UNEXPECTED QIO RESPONSE
4843 ; ERROR 10010 UNEXPECTED NUMBER OF RESPONSES RECEIVED
4844 ; ERROR 10011 UNEXPECTED DATA RECEIVED
4845 023746 BADHEAD
4846 ; ** TEST8 **
4847 ;
4848 023746 BGNTST
4849 023746 004737 011574 CALL MODE0 ;SET APPLICATION MODE
4850 023752 103406 BCS 1$ ;IF NOT CORRECTLY DONE
4851 ;
4852 023754 004737 011706 CALL LPBACK ;TEST OF LOOP
4853 023760 103403 BCS 1$ ;IF NOT
4854 ;
4855 023762 004737 012554 CALL RUNAPP ;LOAD AND RUN APPLICATION
4856 023766 103002 BCC 2$ ;IF CORRECTLY DONE
4857 ;
4858 023770 1$: EXIT TST
4859 ;
4860 023774 004737 011776 2$: CALL INIQIO ;INIT QIO PROCESSING
4861 ;
4862 024000 012701 024016 MOV #60$,R1 ;TABLE ADDRESS IN R1

```

```

4863 024004 004737 013154          CALL   QIOP          ;PROCESS QIO
4864 024010 103022                BCC    5#           ;IF CORRECTLY DONE
4865
4866 024012                EXIT   TST
4867
4868                ; PARAMETERS FOR QIO PROCESSING
4869
4870 024016 024022                60#:   3#           ;IN LIST TABLE BASE ADDRESS
4871 024020 000764                500.   ;TIME-OUT LENGTH(N*10 MS)
4872
4873                ; COMMAND LIST
4874
4875 024022                3#:
4876 024022 000001                1
4877 024024 000001 100421 000000    1,SDLC+C2400+CLKDTE+S141.0 ;CONFIGURATE
4878
4879                ; RESPONSE LIST
4880
4881 024032                4#:
4882 024032 000001                1
4883 024034 000201 000000 000400    1.RDY0.0,1*400 ;CONFIGURATE DONE
4884 024042 000000 000000 000000    .WORD 0,0,0,0,0,0
4885
4886 024056                5#:
4887 024056 012701 024032          MOV    #4#,R1      ;SAVE RESPONSE LIST ADDRESS
4888 024062 004737 010134          CALL   CHKRSP     ;THEN CHECK RESPONSES
4889 024066 103002                BCC    6#         ;IF TEST IS CORRECT
4890
4891 024070                EXIT   TST
4892
4893 024074 004737 011776          6#:   CALL   INIQIO   ;INIT QIO PROCESSING
4894
4895 024100 012701 024116          MOV    #61#,R1    ;TABLE ADDRESS IN R1
4896 024104 004737 013154          CALL   QIOP       ;PROCESS QIO
4897 024110 103022                BCC    9#         ;IF CORRECTLY DONE
4898
4899                EXIT   TST
4900
4901                ; PARAMETERS FOR QIO PROCESSING
4902
4903 024116 024122                61#:   7#           ;IN LIST TABLE BASE ADDRESS
4904 024120 000144                100.   ;TIME-OUT LENGTH(N*10 MS)
4905
4906                ; COMMAND LIST
4907
4908 024122                7#:
4909 024122 000001                1
4910 024124 000014 000000 000000    14,0,0 ;READ MODEM
4911
4912                ; RESPONSE LIST
4913
4914 024132                8#:
4915 024132 000001                1
4916 024134 000214 001455 000400    14.RDY0,S6*400+S142+S106+S107+S109,1*400 ;RESPONSE FOR F14
4917 024142 000000 000000 000000    .WORD 0,0,0,0,0,0
4918
4919 024156                9#:

```

```

4920 024156 012701 024132      MOV    08,R1      ;SAVE RESPONSE LIST ADDRESS
4921 024162 004737 010134      CALL   CHKRSP    ;THEN CHECK RESPONSES
4922 024166 103002                BCC    10        ;IF TEST CORRECTLY DONE
4923
4924 024170                EXIT   TST
4925
4926 024174                10:
4927 024174                BGNSUB
4928 024176 004737 024606      CALL   SUB1      ;SUBTEST1
4929 024202                ENDSUB
4930
4931 024204                BGNSUB
4932 024206 004737 025064      CALL   SUB2      ;SUBTEST2
4933 024212                ENDSUB
4934
4935 024214                BGNSUB
4936 024216 004737 025220      CALL   SUB3      ;SUBTEST3
4937 024222                ENDSUB
4938
4939 024224                BGNSUB
4940 024226 004737 025514      CALL   SUB4      ;SUBTEST4
4941 024232                ENDSUB
4942
4943 024234                BGNSUB
4944 024236 004737 025654      CALL   SUB5      ;SUBTEST5
4945 024242                ENDSUB
4946
4947 024244                BGNSUB
4948 024246 004737 026030      CALL   SUB6      ;SUBTEST6
4949 024252                ENDSUB
4950
4951 024254                BGNSUB
4952 024256 004737 026166      CALL   SUB7      ;SUBTEST7
4953 024262                ENDSUB
4954
4955 024264                ENDTST

```


4957 024266

4958
 4959
 4960
 4961
 4962
 4963
 4964
 4965
 4966
 4967
 4968
 4969
 4970
 4971
 4972
 4973
 4974
 4975
 4976
 4977
 4978
 4979
 4980
 4981
 4982
 4983
 4984
 4985
 4986
 4987
 4988
 4989
 4990
 4991
 4992
 4993
 4994
 4995
 4996
 4997
 4998
 4999
 5000
 5001
 5002
 5003
 5004
 5005
 5006
 5007
 5008
 5009
 5010
 5011
 5012

BADHEAD

```

:                                     ** TEST9 **
: TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 2.4K
: WITH DATA LEADS ONLY
:
: CONFIGURATE:  SDLC PROTOCOL
:                DATA LEADS ONLY
:                WITHOUT ADDRESS SEARCH
:                CLOCK SOURCE INTERNAL
:                RATE = 2.4K
:
:                FOR EACH SUBTEST A COMMAND F14 IS GIVEN FIRST TO CONTROL
:                THE LINE STATE AND MODEM SUCH AS:
:                STATE = S6
:                MODEM = S142 ONLY
:
: SUBTEST1  - TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
:            OF 2 BUFFERS.
:            THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
:            XMIT BUFFERS ARE GIVEN ONE OF WHICH WITH 1 BYTE LONG
:            EXPECTED RESPONSES ARE:
:            STATUS = 360 FOR F6 THIRD BUFFER PASSED
:                   = 372 FOR F5 1 BYTE LONG BUFFER
:                   = 360 FOR F5 THIRD BUFFER PASSED
:                   = 1   FOR F5 FIRST BUFFER XMITTED
:                   = 1   FOR F6 FIRST BUFFER RECEIVED
:                   = 1   FOR F5 SECOND BUFFER XMITTED
:                   = 1   FOR F6 SECOND BUFFER RECEIVED
:
: SUBTEST2  - TEST OF BUFFER OVERFLOW STATUS 373 FOR RECEPTION
:            A 32 BYTE LONG BUFFER IS AFFECTED FOR RECEIVE WHILE
:            A 100 BYTE LONG BUFFER IS PASSED TO XMIT.
:            EXPECTED RESPONSES ARE:
:            STATUS = 1   FOR F5 BUFFER XMITTED
:            STATUS = 373 FOR F6 BUFFER OVERFLOW
:
: SUBTEST3  - TEST OF NON EXISTENT MEMORY STATUS 374
:            TWO RECEIVE BUFFER ARE AFFECTED ONE OF WHICH CONTAINS
:            A NON EXISTENT ADDRESS
:            TWO XMIT BUFFERS ARE PASSED ONE OF WHICH CONTAINS A NON
:            EXISTENT ADDRESS
:            EXPECTED RESPONSES ARE:
:            STATUS = 374 FOR F6 NON EXISTENT MEMORY BUFFER
:                   = 1   FOR F5 CORRECT BUFFER
:                   = 374 FOR F5 NON EXISTENT MEMORY BUFFER
:                   = 364 FOR F6 RECEIVE ABORT
:
: SUBTEST4  - TEST OF ABORT RECEIVED STATUS 364 AND ACTION OF XMIT
:            ABORT COMMAND.
:            ONE RECEIVE BUFFER IS USED THEN, TWO XMIT BUFFERS
:            ARE PASSED. AFTER A SHORT WAIT, XMIT ABORT IS GIVEN
:            TWICE.
:            EXPECTED STATUS ARE:
:            STATUS = 1   FOR F7 XMIT ABORTED
:                   = 364 FOR F6 ABORT RECEIVED
:
: SUBTEST5  - TEST OF ACTION OF RECEIVE ABORT COMMAND.

```

```

5013      ;
5014      ; TWO RECEIVE BUFFERS ARE USED THEN, TWO XMIT BUFFERS
5015      ; ARE PASSED. AFTER A SHORT WAIT, RECEIVE ABORT IS GIVEN
5016      ; EXPECTED STATUS ARE:
5017      ;   STATUS = 1 FOR F10 RECEIVE ABORTED
5018      ;           = 1 FOR F5 FIRST BUFFER SENT
5019      ;           = 1 FOR F5 SECOND BUFFER SENT
5020      ;
5021      ; SUBTEST6
5022      ; - TEST OF KMV RDB OVERFLOW DURING XMIT/RECEIVE BUFFER
5023      ; ONE RECEIVE BUFFER IS USED
5024      ; THEN TWO XMIT BUFFER ARE PASSED THE FIRST ONE TOO LONG
5025      ; EXPECTED STATUS ARE:
5026      ;   STATUS = 373 FOR FIRST F5
5027      ;           = 1 FOR SECOND F5
5028      ;           = 1 FOR F6
5029      ;
5030      ; SUBTEST7
5031      ; - TEST OF A DECONFIGURATE DURING XMIT/RECEIVE BUFFER,
5032      ; TWO RECEIVE BUFFERS ARE USED THEN, TWO XMIT BUFFERS
5033      ; ARE PASSED. AFTER A SHORT WAIT, DECONFIGURATE COMMAND
5034      ; IS USED. EXPECTED STATUS IS:
5035      ;   STATUS = 1 FOR F2 DECONFIGURATE DONE
5036      ;           AND ALL BUFFERS ARE KILLED
5037      ;
5038      ; PATTERN: INCREMENTAL
5039      ;
5040      ; REPORTS:
5041      ; ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR
5042      ; ERROR 10001 RUN FUNCTION NOT CORRECTLY PERFORMED
5043      ; ERROR 10002 WRITE FUNCTION NOT CORRECTLY PERFORMED
5044      ; ERROR 10003 READ FUNCTION NOT CORRECTLY PERFORMED
5045      ; ERROR 10004 DATA COMPARE ERROR
5046      ; ERROR 10005 UNEXPECTED INTERRUPT IN
5047      ; ERROR 10006 UNEXPECTED INTERRUPT OUT
5048      ; ERROR 10007 NO MORE INTERRUPT WHILE QIO PENDING
5049      ; ERROR 10008 UNEXPECTED EPROM'S ON KMV11 A
5050      ; ERROR 10009 UNEXPECTED QIO RESPONSE
5051      ; ERROR 10010 UNEXPECTED NUMBER OF RESPONSES RECEIVED
5052      ; ERROR 10011 UNEXPECTED DATA RECEIVED
5053      ;
5054      ; BADHEAD
5055      ;
5056      ; ** TEST9 **
5057      ;
5058      ;
5059      ;
5060      ;
5061      ;
5062      ;
5063      ;
5064      ;
5065      ;
5066      ;
5067      ;
5068      ;

```

```

5050
5051
5052 024266      BGNTST
5053 024266 004737 011574      CALL MODEO      ;SET APPLICATION MODE
5054 024272 103406      BCS 1$      ;IF NOT CORRECTLY DONE
5055
5056 024274 004737 011706      CALL LPBACK     ;TEST OF LOOP
5057 024300 103403      BCS 1$      ;IF NOT
5058
5059 024302 004737 012554      CALL RUNAPP     ;LOAD AND RUN APPLICATION
5060 024306 103002      BCC 2$      ;IF CORRECTLY DONE
5061
5062 024310      1$: EXIT TST
5063
5064 024314 004737 011776      2$: CALL INIQIO ;INIT QIO PROCESSING
5065
5066 024320 012701 024336      MOV 060$,R1    ;TABLE ADDRESS IN R1
5067 024324 004737 013154      CALL QIOP      ;PROCESS QIO
5068 024330 103022      BCC 5$      ;IF CORRECTLY DONE

```

```

5069
5070 024332          EXIT   TST
5071
5072          ; PARAMETERS FOR QIO PROCESSING
5073
5074 024336 024542   60:    3:          ; IN LIST TABLE BASE ADDRESS
5075 024340 000764   500.          ; TIME-OUT LENGTH(N*10 MS)
5076
5077          ; COMMAND LIST
5078
5079 024342   3:
5080 024342 000001          1
5081 024344 000001 100461 000000          1,DLO*SDLC*C2400*CLKDTE*S141,0          ; CONFIGURATE
5082
5083          ; RESPONSE LIST
5084
5085 024352   4:
5086 024352 000001          1
5087 024354 000201 000000 000400          1*RDY0,0,1*400          ; CONFIGURATE DONE
5088 024362 000000 000000 000000          .WORD 0,0,0,0,0,0
5089
5090 024376   5:
5091 024376 012701 024352          MOV    #4: ,R1          ; SAVE RESPONSE LIST ADDRESS
5092 024402 004737 010134          CALL   CHKRSP          ; THEN CHECK RESPONSES
5093 024406 103002          BCC   6:          ; IF TEST IS CORRECT
5094
5095 024410          EXIT   TST
5096
5097 024414 004737 011776   6:    CALL   INIQIO          ; INIT QIO PROCESSING
5098
5099 024420 012701 024436          MOV    #61: ,R1          ; TABLE ADDRESS IN R1
5100 024424 004737 013154          CALL   QIOP           ; PROCESS QIO
5101 024430 103022          BCC   9:          ; IF CORRECTLY DONE
5102
5103          EXIT   TST
5104
5105          ; PARAMETERS FOR QIO PROCESSING
5106
5107 024436 024442   61:    7:          ; IN LIST TABLE BASE ADDRESS
5108 024440 000144   100.          ; TIME-OUT LENGTH(N*10 MS)
5109
5110          ; COMMAND LIST
5111
5112 024442   7:
5113 024442 000001          1
5114 024444 000014 000000 000000          14,0,0          ; READ MODEM
5115
5116          ; RESPONSE LIST
5117
5118 024452   8:
5119 024452 000001          1
5120 024454 000214 001440 000400          14*RDY0,56*400*S142,1*400          ; RESPONSE FOR F14
5121 024462 000000 000000 000000          .WORD 0,0,0,0,0,0
5122
5123 024476   9:
5124 024476 012701 024452          MOV    #8: ,R1          ; SAVE RESPONSE LIST ADDRESS
5125 024502 004737 010134          CALL   CHKRSP          ; THEN CHECK RESPONSES

```

```

5126 024506 103002          BCC      100          ;IF TEST IS CORRECT
5127
5128 024510          EXIT     TST
5129
5130 024514          100:
5131 024514          BGNSUB
5132 024516 004737 024606          CALL     SUB1          ;SUBTEST1
5133 024522          ENDSUB
5134
5135 024524          BGNSUB
5136 024526 004737 025064          CALL     SUB2          ;SUBTEST2
5137 024532          ENDSUB
5138
5139 024534          BGNSUB
5140 024536 004737 025220          CALL     SUB3          ;SUBTEST3
5141 024542          ENDSUB
5142
5143 024544          BGNSUB
5144 024546 004737 025514          CALL     SUB4          ;SUBTEST4
5145 024552          ENDSUB
5146
5147 024554          BGNSUB
5148 024556 004737 025654          CALL     SUB5          ;SUBTEST5
5149 024562          ENDSUB
5150
5151 024564          BGNSUB
5152 024566 004737 026030          CALL     SUB6          ;SUBTEST6
5153 024572          ENDSUB
5154
5155 024574          BGNSUB
5156 024576 004737 026166          CALL     SUB7          ;SUBTEST7
5157 024602          ENDSUB
5158
5159 024604          ENDTST

```

5161
5162
5163
5164
5165
5166 024606
5167 024606 004737 011776
5168
5169 024612 012701 024626
5170 024616 004737 013154
5171 024622 103065
5172
5173 024624 000207
5174
5175
5176
5177 024626 024632
5178 024630 007640
5179
5180
5181
5182 024632
5183 024632 000007
5184 024634 000006 002574 000376
5185 024642 000006 003174 000367
5186 024650 000006 003574 000066
5187 024656 000005 004574 000001
5188 024664 000005 004574 000366
5189 024672 000005 005174 000367
5190 024700 000005 005574 000400
5191
5192
5193
5194 024706
5195 024706 000007
5196 024710 000206 000000 170000
5197 024716 000205 000000 175000
5198 024724 000205 000000 170000
5199 024732 000205 000000 000400
5200 024740 000206 000366 000400
5201 024746 000205 000000 000400
5202 024754 000206 000367 000400
5203 024762 000000 000000 000000
5204
5205 024776
5206 024776 012701 024706
5207 025002 004737 010134
5208 025006 103001
5209
5210 025010 000207
5211
5212 025012 012701 004574
5213 025016 012702 002574
5214 025022 012737 000366 002346
5215 025030 004737 011374
5216 025034 103001
5217

; COMPLETE XMIT/RECEIVE BUFFER TEST COMMON TO TEST08 AND TEST09

SUB1:

CALL INIQIO ; INIT QIO PROCESSING
MOV #601,R1 ; TABLE ADDRESS IN R1
CALL QIOP ; PROCESS QIO
BCC 51 ; IF OKE
RETURN

; PARAMETERS FOR QIO PROCESSING

601: 31 ; IN LIST TABLE BASE ADDRESS
4000. ; TIME-OUT LENGTH (N*10 MS)

; COMMAND LIST

31:
7.
6,RXBUF0,#RDBS-2 ; FIRST RECEIVE BUFFER
6,RXBUF1,#RDBS-9. ; SECOND
6,RXBUF2,54. ; THIRD "
5,TXBUF0,1 ; FIRST XMIT BUFFER
5,TXBUF0,#RDBS-10. ; SECOND "
5,TXBUF1,#RDBS-9. ; THIRD "
5,TXBUF2,#RDBS ; FORTH "

; RESPONSE LIST

41:
7.
6*RDY0,0,360*400 ; TOO MANY BUFFERS FOR F6
5*RDY0,0,372*400 ; BUFFER TOO SHORT FOR F5
5*RDY0,0,360*400 ; TOO MANY BUFFERS FOR F5
5*RDY0,0,1*400 ; FIRST BUFFER SENT
6*RDY0,#RDBS-10.,1*400 ; FIRST RECEIVE BUFFER
5*RDY0,0,1*400 ; SECOND BUFFER SENT
6*RDY0,#RDBS-9.,1*400 ; SECOND BUFFER RECEIVED
.WORD 0,0,0,0,0,0

51:

MOV #41,R1 ; SAVE RESPONSE LIST ADDRESS
CALL CHKRSP ; THEN CHECK RESPONSES
BCC 61 ; IF TEST IS CORRECT

RETURN

61:

MOV #TXBUF0,R1 ; CHECK FIRST RECEIVE BUFFER
MOV #RXBUF0,R2
MOV #RDBS-10.,LENGTH
CALL COMPAR
BCC 71 ; IF CORRECT

```

5218 025036 000207          RETURN
5219
5220 025040 012701 005174      7$:  MOV    #TXBUF1,R1          ;CHECK SECOND RECEIVE BUFFER
5221 025044 012702 003174      MOV    #RXBUF1,R2
5222 025050 012737 000367      002346  MOV    #RDBS-9.,LENGTH
5223 025056 004737 011374      CALL   COMPAR
5224
5225 025062 000207          RETURN
5226
5227 025064          SUB2:
5228 025064 004737 011776      CALL   INIQIO          ;INIT QIO PROCFSING
5229
5230 025070 012701 025104      MOV    #60$,R1        ;TABLE ADDRESS IN R1
5231 025074 004737 013154      CALL   QIOP           ;PROCESS QIO
5232 025100 103027          BCC    5$             ;IF CORRECT
5233
5234 025102 000207          RETURN
5235
5236          ; PARAMETERS FOR QIO PROCESSING
5237
5238 025104 025110      60$:  3$             ;IN LIST TABLE BASE ADDRESS
5239 025106 000620      400.           ;TIME-OUT LENGTH(N*10 MS)
5240
5241          ; COMMAND LIST
5242
5243 025110      3$:
5244 025110 000002          2
5245 025112 000006 002574 000040      6,RXBUF0,32.          ;SET RECEIVE BUFFER
5246 025120 000005 005574 000144      5,TXBUF2,100.        ;SET XMIT BUFFER
5247
5248          ; RESPONSE LIST
5249
5250 025126      4$:
5251 025126 000002          2
5252 025130 000205 000000 000400      5*RDY0,0,1*400      ;BUFFER SENT FOR F5
5253 025136 000206 000144 175400      6*RDY0,100.,373*400 ;OVERFLOW ERROR FOR F6
5254 025144 000000 000000 000000      .WORD 0,0,0,0,0,0
5255
5256 025160      5$:
5257 025160 012701 025126      MOV    #4$,R1          ;SAVE RESPONSE LIST ADDRESS
5258 025164 004737 010134      CALL   CHKRSP         ;THEN CHECK RESPONSES
5259 025170 103001          BCC    6$             ;IF TEST CORRECT
5260
5261 025172 000207          RETURN
5262
5263 025174 012701 005574      6$:  MOV    #TXBUF2,R1          ;CHECK RECEIVE BUFFER
5264 025200 012702 002574      MOV    #RXBUF0,R2
5265 025204 012737 000040      002346  MOV    #32.,LENGTH
5266 025212 004737 011374      CALL   COMPAR
5267
5268 025216 000207          RETURN
5269
5270 025220          SUB3:
5271 025220 004737 011776      CALL   INIQIO          ;INIT QIO PROCESSING
5272
5273 025224 004737 0C7620      CALL   ..SIZE         ;SIZE MEMORY
5274 025230 005737 002514      TST   PDPTYP         ;CHECK IF A PDP11/23.?

```

```

5275 025234 001024          BNE      1#          ;IF YES
5276
5277 025236 022737 157776 002500  CMP      #157776,LSTAD ;NO, CHECK LAST ADDRESS FOR IO PAGE
5278 025244 001020          BNE      1#          ;IF NOT
5279 025246 013701 002502  MOV      LSTBK,R1
5280 025252 042701 177774  BIC      #177774,R1
5281 025256 022701 000003  CMP      #3,R1
5282 025262 001011          BNE      1#          ;IF NOT
5283
5284 025264          PRINTF  #MNONEX ;IF YES, REPORT THAT SUBTEST IS DROPPED
5285          .EVEN
5286 025304 000207          RETURN ;THEN EXIT
5287
5288 025306 013701 002500 1#:  MOV      LSTAD,R1 ;GET LAST MEMORY ADDRESS
5289 025312 162701 000012  SUB      #10.,R1 ;POSITON 10 BYTES BEFORE
5290 025316 010137 025366  MOV      R1,3#*4 ;SET BAD ADDRESS IN TABLE FOR F6
5291 025322 010137 025402  MOV      R1,3#*20 ;" " " " F5
5292 025326 013701 002502  MOV      LSTBK,R1 ;GET LAST MEMORY ADDRESS(21 16)
5293 025332 110137 025365  MOV      R1,3#*3 ;SET BAD ADDRESS IN TABLE FOR F6
5294 025336 110137 025401  MOV      R1,3#*17 ;" " " " F5
5295
5296 025342 012701 025356  MOV      #60#,R1 ;TABLE ADDRESS IN R1
5297 025346 004737 013154  CALL     QIOP ;PROCESS QIO
5298 025352 103035          BCC      5#          ;IF CORRECT
5299
5300 025354 000207          RETURN
5301
5302          ; PARAMETERS FOR QIO PROCESSING
5303
5304 025356 025362 60#:  3#          ;IN LIST TABLE BASE ADDRESS
5305 025360 001750          1000. ;TIME-OUT LENGTH(N*10 MS)
5306
5307          ; COMMAND LIST
5308
5309 025362 3#:
5310 025362 000003          3
5311 025364 000006 000000 000400  6,0,#RDBS ;FIRST RECEIVE BUFFER WITH BAD ADDRESS
5312 025372 000005 004574 000371  5,TXBUFO,#RDBS 7 ;FIRST XMIT BUFFER
5313 025400 000005 000000 000400  5,0,#RDBS ;SECOND XMIT BUFFER WITH BAD ADDRESS
5314
5315          ; RESPONSE LIST
5316
5317 025406 4#:
5318 025406 000003          3
5319 025410 000205 000000 000400  5*RDY0,0,1*400 ;FIRST BUFFER SENT
5320 025416 000205 000000 176000  5*RDY0,0,374*400 ;NON EXISTENT ADDRESS FOR F5
5321 025424 000206 000000 176000  6*RDY0,0,374*400 ;NON EXISTENT ADDRESS FOR F6
5322 025432 000000 000000 000000  .WORD   0,0,0,0,0,0
5323
5324 025446 5#:
5325 025446 012701 025406  MOV      #4#,R1 ;SAVE RESPONSE LIST ADDRESS
5326 025452 004737 010134  CALL     CHKRSP ;THEN CHECK RESPONSES
5327
5328 025456 000207          RETURN
5329
5330 025460 045 116 045 MNONEX: .ASCIZ /#N#A SUBTEST 3 IS SKIPPED/
5331          .EVEN

```

```

5332
5333 025514          SUB4:
5334 025514 004737 011776      CALL    INIQIO          ;INIT QIO PROCESSING
5335
5336 025520 012701 025544      MOV     #60$,R1        ;TABLE ADDRESS IN R1
5337 025524 004737 013154      CALL    QIOP          ;PROCESS QIO
5338
5339 025530 012701 025550      MOV     #61$,R1        ;TABLE ADDRESS IN R1
5340 025534 004737 013154      CALL    QIOP          ;PROCESS QIO
5341 025540 103040          BCC     5$            ;IF OKE
5342
5343 025542 000207          RETURN
5344
5345          ; PARAMETERS FOR QIO PROCESSING
5346
5347 025544 025554          60$:   3$            ;IN LIST TABLE BASE ADDRESS
5348 025546 000150          104.          ;TIME OUT LENGTH(N*10 MS)
5349
5350 025550 025600          61$:   33$           ;
5351 025552 000620          400.          ;
5352
5353          ; COMMAND LIST
5354
5355 025554          3$:
5356 025554 000003          3            ;SECOND RX BUFFER
5357 025556 000006 003174 000144 6,RXBUF1,100. ;FIRST XMIT BUFFER
5358 025564 000005 005174 000376 5, TXBUF1,$RDBS-2 ;SECOND XMIT BUFFER
5359 025572 000005 005574 000144 5, TXBUF2,100.
5360
5361 025600 000001          33$:  1            ;XMIT ABORT
5362 025602 000007 000000 000000 7,0,0
5363
5364          ; RESPONSE LIST
5365
5366 025610          4$:
5367 025610 000002          2            ;OKE FOR F7
5368 025612 000207 000000 000400 7*RDY0,0,1*400 ;ABORT RECEIVED FOR F6
5369 025620 000206 000030 172000 6*RDY0,24.,364*400
5370 025626 000000 000000 000000 .WORD 0,0,0,0,0,0
5371
5372 025642          5$:
5373 025642 012701 025610      MOV     #4$,R1        ;SAVE RESPONSE LIST ADDRESS
5374 025646 004737 010134      CALL    CHKRSP       ;THEN CHECK RESPONSES
5375
5376 025652 000207          RETURN
5377
5378 025654          SUB5:
5379 025654 004737 011776      CALL    INIQIO          ;INIT QIO PROCESSING
5380
5381 025660 012701 025704      MOV     #60$,R1        ;TABLE ADDRESS IN R1
5382 025664 004737 013154      CALL    QIOP          ;PROCESS QIO
5383
5384 025670 012701 025710      MOV     #61$,R1        ;TABLE ADDRESS IN R1
5385 025674 004737 013154      CALL    QIOP          ;PROCESS QIO
5386 025700 103046          BCC     5$            ;IF OKE
5387
5388 025702 000207          RETURN

```



```

5389
5390           ; PARAMETERS FOR QIO PROCESSING
5391
5392 025704 025714 60$: 3$           ; IN LIST TABLE BASE ADDRESS
5393 025706 000170          120.       ; TIME OUT LENGTH(N*10 MS)
5394
5395 025710 025746 61$: 33$          ;
5396 025712 001130          600.       ;
5397
5398           ; COMMAND LIST
5399
5400 025714 3$: 4
5401 025714 000004          4
5402 025716 000006 002574 000400      6,RXBUF0,#RDBS      ; FIRST RX BUFFER
5403 025724 000006 003174 000144      6,RXBUF1,100.      ; SECOND RX BUFFER
5404 025732 000005 005174 000400      5,TXBUF1,#RDBS      ; FIRST XMIT BUFFER
5405 025740 000005 005574 000144      5,TXBUF2,100.      ; SECOND XMIT BUFFER
5406
5407 025746 000001 33$: 1
5408 025750 000010 000000 000000      10,0,0             ; RECEIVE ABORT
5409
5410           ; RESPONSE LIST
5411
5412 025756 4$: 3
5413 025756 000003          3
5414 025760 000210 000000 000400      10*RDY0,0,1*400    ; OKE FOR F10
5415 025766 000205 000000 000400      5*RDY0,0,1*400    ; OKE FOR F5
5416 025774 000205 000000 000400      5*RDY0,0,1*400    ; OKE FOR F5
5417 026002 000000 000000 000000      .WORD 0,0,0,0,0,0
5418
5419 026016 5$:
5420 026016 012701 025756      MOV #4$,R1          ; SAVE RESPONSE LIST ADDRESS
5421 026022 004737 010134      CALL CHKRSP        ; THEN CHECK RESPONSES
5422
5423 026026 000207          RETURN
5424
5425 026030 SUB6:
5426 026030 004737 011776      CALL INIQIO        ; INIT QIO PROCESSING
5427
5428 026034 012737 000424 026076      MOV #RDBS*20.,3$*6 ; OVERFLOW THE RIB BY 20.
5429 026042 012737 000424 026104      MOV #RDBS*20.,3$*14
5430
5431 026050 012701 026064      MOV #60$,R1        ; TABLE ADDRESS IN R1
5432 026054 004737 013154      CALL QIOP          ; PROCESS QIO
5433 026060 103035          BCC 5$             ; IF OKE
5434
5435 026062 000207          RETURN
5436
5437           ; PARAMETERS FOR QIO PROCESSING
5438
5439 026064 026070 60$: 3$           ; IN LIST TABLE BASE ADDRESS
5440 026066 001750          1000.      ; TIME OUT LENGTH(N*10 MS)
5441
5442           ; COMMAND LIST
5443
5444 026070 3$: 3
5445 026070 000003          3

```

```

5446 026072 000006 003174 000000      6,RXBUF1,0      ;SECOND RX BUFFER
5447 026100 000005 005174 000000      5,TXBUF1,0      ;FIRST XMIT BUFFER
5448 026106 000005 005574 000144      5,TXBUF2,100.   ;SECOND XMIT BUFFER
5449
5450      ; RESPONSE LIST
5451
5452 026114      4$:
5453 026114 000003      3
5454 026116 000205 000000 175400      5,RDY0,0,373*400
5455 026124 000205 000000 000400      5,RDY0,0,1*400
5456 026132 000206 000144 000400      6,RDY0,100.,1*400
5457 026140 000000 000000 000000      .WORD 0,0,0,0,0,0
5458
5459 026154      5$:
5460 026154 012701 026114      MOV #4$,R1      ;SAVE RESPONSE LIST ADDRESS
5461 026160 004737 010134      CALL CHKRSP     ;THEN CHECK RESPONSES
5462
5463 026164 000207      RETURN
5464
5465 026166      SUB7:
5466 026166 004737 011776      CALL INIQIO     ;INIT QIO PROCESSING
5467
5468 026172 012701 026216      MOV #60$,R1    ;TABLE ADDRESS IN R1
5469 026176 004737 013154      CALL QIOP      ;PROCESS QIO
5470
5471 026202 012701 026222      MOV #61$,R1
5472 026206 004737 013154      CALL QIOP
5473 026212 103040      BCC 5$        ;IF OKE
5474
5475 026214 000207      RETURN
5476
5477      ; PARAMETERS FOR QIO PROCESSING
5478
5479 026216 026226      60$: 3$      ;IN LIST TABLE BASE ADDRESS
5480 026220 000310      200.        ;TIME OUT LENGTH(N*10 MS)
5481
5482 026222 026260      61$: 33$     ;
5483 026224 001274      700.        ;
5484
5485      ; COMMAND LIST
5486
5487 026226      3$:
5488 026226 000004      4
5489 026230 000006 002574 000400      6,RXBUF0,$RDBS ;FIRST RX BUFFER
5490 026236 000006 003174 000144      6,RXBUF1,100.  ;SECOND RX BUFFER
5491 026244 000005 005174 000400      5,TXBUF1,$RDBS ;FIRST XMIT BUFFER
5492 026252 000005 005574 000144      5,TXBUF2,100.  ;SECOND XMIT BUFFER
5493
5494 026260 000001      33$: 1
5495 026262 000002 000000 000000      2,0,0        ;DECONFIGURATE
5496
5497      ; RESPONSE LIST
5498
5499 026270      4$:
5500 026270 000001      1
5501 026272 000202 000000 000400      2,RDY0,0,1*400 ;DECONFIGURATE DONE
5502 026300 000000 000000 000000      .WORD 0,0,0,0,0,0

```

```
5503  
5504 026314          54:  
5505 026314 012701 026270      MOV    #44,R1          ;SAVE RESPONSE LIST ADDRESS  
5506 026320 004737 010134      CALL   CHKRSP          ;THEN CHECK RESPONSES  
5507  
5508 026324 000207          RETURN  
5509
```

5511
5512 026326

BADHEAD

** TEST10 **

5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550 026326

```

; TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 64K
; WITH FULL MODEM CONTROL
;
; CONFIGURATE:  SDLC PROTOCOL
;                FULL MODEM CONTROL
;                WITHOUT ADDRESS SEARCH
;                CLOCK SOURCE INTERNAL
;                RATE = 64K
;
; TEST          TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
;                OF 2 BUFFERS.
;                THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
;                XMIT BUFFERS ARE GIVEN ONE OF WHICH WITH 1 BYTE LONG
;                EXPECTED RESPONSES ARE:
;                STATUS = 360 FOR F6 THIRD BUFFER PASSED
;                = 372 FOR F5 1 BYTE LONG BUFFER
;                = 360 FOR F5 THIRD BUFFER PASSED
;                = 1   FOR F5 FIRST BUFFER XMITTED
;                = 1   FOR F6 FIRST BUFFER RECEIVED
;                = 1   FOR F5 SECOND BUFFER XMITTED
;                = 1   FOR F6 SECOND BUFFER RECEIVED
;
; PATTERN:      INCREMENTAL
;
; REPORTS:      ERROR 10000  KMV11 FAILS TO RESET MASTER CLEAR
;                ERROR 10001  RUN FUNCTION NOT CORRECTLY PERFORMED
;                ERROR 10002  WRITE FUNCTION NOT CORRECTLY PERFORMED
;                ERROR 10003  READ FUNCTION NOT CORRECTLY PERFORMED
;                ERROR 10004  DATA COMPARE ERROR
;                ERROR 10005  UNEXPECTED INTERRUPT IN
;                ERROR 10006  UNEXPECTED INTERRUPT OUT
;                ERROR 10007  NO MORE INTERRUPT WHILE QIO PENDING
;                ERROR 10008  UNEXPECTED EPROM'S ON KMV11 A
;                ERROR 10009  UNEXPECTED QIO RESPONSE
;                ERROR 10010  UNEXPECTED NUMBER OF RESPONSES RECEIVED
;                ERROR 10011  UNEXPECTED DATA RECEIVED
;                ERROR 10012  NO LOOP BACK

```

BADHEAD

** TEST10 **

5551
5552
5553 026326
5554 026326 004737 011574
5555 026332 103406
5556
5557 026334 004737 011706
5558 026340 103403
5559
5560 026342 004737 012554
5561 026346 103002
5562
5563 026350
5564
5565 026354 004737 011776

BGNTST

```

CALL MODE0 ;SET APPLICATION MODE
BCS 1$ ;IF NOT CORRECT
;
CALL LPBACK ;ANY LOOP?
BCS 1$ ;IF NOT
;
CALL RUNAPP ;LOAD AND RUN APPLICATION
BCC 2$ ;IF CORRECT
;
1$: EXIT TST
;
2$: CALL INIQIO ;INIT QIO PROCESSING

```

```

5566
5567 026360 012701 026376      MOV    #60$,R1      ;TABLE ADDRESS IN R1
5568 026364 004737 013154      CALL   QIOP         ;PROCESS QIO
5569 026370 103022                BCC    5$          ;IF CORRECT
5570
5571 026372                EXIT   TST
5572
5573                ; PARAMETERS FOR QIO PROCESSING
5574
5575 026376 026402      60$:    3$          ;IN LIST TABLE BASE ADDRESS
5576 026400 000764                500.             ;TIME-OUT LENGTH(N*10 MS)
5577
5578                ; COMMAND LIST
5579
5580 026402      3$:
5581 026402 000001                1
5582 026404 000001 103421 000000      1,SDLC,C64K,CLKDTE,S141,0      ;CONFIGURATE
5583
5584                ; RESPONSE LIST
5585
5586 026412      4$:
5587 026412 000001                1
5588 026414 000201 000000 000400      1,RDY0,0,1*400             ;CONFIGURATE DONE
5589 026422 000000 000000 000000      .WORD 0,0,0,0,0,0
5590
5591 026436      5$:
5592 026436 012701 026412      MOV    #4$,R1      ;SAVE RESPONSE LIST ADDRESS
5593 026442 004737 010134      CALL   CHKRSP     ;THEN CHECK RESPONSES
5594 026446 103002                BCC    6$          ;IF TEST CORRECT
5595
5596 026450                EXIT   TST
5597
5598
5599 026454 004737 011776      6$:    CALL   INIQIO      ;INIT QIO PROCESSING
5600
5601 026460 012701 026476      MOV    #61$,R1      ;TABLE ADDRESS IN R1
5602 026464 004737 013154      CALL   QIOP         ;PROCESS QIO
5603 026470 103022                BCC    9$          ;IF CORRECT
5604
5605 026472                EXIT   TST
5606
5607                ; PARAMETERS FOR QIO PROCESSING
5608
5609 026476 026502      61$:    7$          ;IN LIST TABLE BASE ADDRESS
5610 026500 000144                100.             ;TIME-OUT LENGTH(N*10 MS)
5611
5612                ; COMMAND LIST
5613
5614 026502      7$:
5615 026502 000001                1
5616 026504 000014 000000 000000      14,0,0             ;READ MODEM
5617
5618                ; RESPONSE LIST
5619
5620 026512      8$:
5621 026512 000001                1
5622 026514 000214 001455 000400      14,RDY0,S6*400,S142,S106,S107,S109,1*400      ;RESPONSE FOR F14

```

```
5623 026522 000000 000000 000000 .WORD 0.0.0.0.0.0
5624
5625 026536          9$:
5626 026536 012701 026512      MOV    #8$,R1      ;SAVE RESPONSE LIST ADDRESS
5627 026542 004737 010134      CALL   CHKRSP      ;THEN CHECK RESPONSES
5628 026546 103002          BCC    10$         ;IF TEST CORRECT
5629
5630 026550          EXIT   TST
5631
5632 026554          10$:
5633 026554 004737 027016      CALL   SHORTST     ;FOR SHORT XMIT/RECEIVE BUFFER TEST
5634
5635 026560          ENDTST
```

5637 026562

5638
5639
5640
5641
5642
5643
5644
5645
5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674

BADHEAD

```

;
; ** TEST11 **
; TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 64K
; WITH DATA LEADS ONLY
;
; CONFIGURATE:  HDLC PROTOCOL
;                DATA LEADS ONLY
;                WITHOUT ADDRESS SEARCH
;                CLOCK SOURCE INTERNAL
;                RATE = 64K
;
; SUBTEST1      - TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
;                OF 2 BUFFERS.
;                THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
;                XMIT BUFFERS ARE GIVEN ONE OF WHICH WITH 1 BYTE LONG
;                EXPECTED RESPONSES ARE:
;                STATUS = 360 FOR F6 THIRD BUFFER PASSED
;                = 372 FOR F5 1 BYTE LONG BUFFER
;                = 360 FOR F5 THIRD BUFFER PASSED
;                = 1   FOR F5 FIRST BUFFER XMITTED
;                = 1   FOR F6 FIRST BUFFER RECEIVED
;                = 1   FOR F5 SECOND BUFFER XMITTED
;                = 1   FOR F6 SECOND BUFFER RECEIVED
;
; PATTERN:      INCREMENTAL
;
; REPORTS:      ERROR 10000  KMV11 FAILS TO RESET MASTER CLEAR
;                ERROR 10001  RUN FUNCTION NOT CORRECTLY PERFORMED
;                ERROR 10002  WRITE FUNCTION NOT CORRECTLY PERFORMED
;                ERROR 10003  READ FUNCTION NOT CORRECTLY PERFORMED
;                ERROR 10004  DATA COMPARE ERROR
;                ERROR 10005  UNEXPECTED INTERRUPT IN
;                ERROR 10006  UNEXPECTED INTERRUPT OUT
;                ERROR 10007  NO MORE INTERRUPT WHILE QIO PENDING
;                ERROR 10008  UNEXPECTED EPROM'S ON KMV11-A
;                ERROR 10009  UNEXPECTED QIO RESPONSE
;                ERROR 10010  UNEXPECTED NUMBER OF RESPONSES RECEIVED
;                ERROR 10011  UNEXPECTED DATA RECEIVED
;                ERROR 10012  NO LOOP BACK

```

5675 026562

BADHEAD

** TEST11 **

5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691

BGNTST

```

CALL MODE0          ;SET APPLICATION MODE
BCS 1$             ;IF NOT CORRECT

CALL LPBACK        ;ANY LOOP?
BCS 1$             ;IF NOT

CALL RUNAPP        ;LOAD AND RUN APPLICATION
BCC 2$             ;IF CORRECT

1$: EXIT TST

2$: CALL INIQIO    ;INIT QIO PROCESSING

```

```

5692 026614 012701 026632      MOV      #601,R1      ;TABLE ADDRESS IN R1
5693 026620 004737 013154      CALL     QIOP        ;PROCESS QIO
5694 026624 103022                BCC      51          ;IF CORRECT
5695
5696 026626                EXIT     TST
5697
5698                ; PARAMETERS FOR QIO PROCESSING
5699
5700 026632 026636      601:    31          ;IN LIST TABLE BASE ADDRESS
5701 026634 000764                500.          ;TIME-OUT LENGTH(N*10 MS)
5702
5703                ; COMMAND LIST
5704
5705 026636      31:
5706 026636 000001                1
5707 026640 000001 103460 000000        1,DLO.C64K.CLKDTE.S141.0      ;CONFIGURATE
5708
5709                ; RESPONSE LIST
5710
5711 026645      41:
5712 026646 000001                1
5713 026650 000201 000000 000400        1.RDY0.0.1*400          ;CONFIGURATE DONE
5714 026656 000000 000000 000000        .WORD 0.0.0.0.0.0
5715
5716 026672      51:
5717 026672 012701 026646      MOV      #41,R1      ;SAVE RESPONSE LIST ADDRESS
5718 026676 004737 010134      CALL     CHKRSR     ;THEN CHECK RESPONSES
5719 026702 103002                BCC      61          ;IF TEST CORRECT
5720
5721 026704                EXIT     TST
5722
5723 026710 004737 011776      61:    CALL     INIQIO      ;INIT QIO PROCESSING
5724
5725 026714 012701 026732      MOV      #611,R1    ;TABLE ADDRESS IN R1
5726 026720 004737 013154      CALL     QIOP        ;PROCESS QIO
5727 026724 103022                BCC      91          ;IF CORRECT
5728
5729 026726                EXIT     TST
5730
5731                ; PARAMETERS FOR QIO PROCESSING
5732
5733 026732 026736      611:   71          ;IN LIST TABLE BASE ADDRESS
5734 026734 000144                100.          ;TIME-OUT LENGTH(N*10 MS)
5735
5736                ; COMMAND LIST
5737
5738 026736      71:
5739 026736 000001                1
5740 026740 000014 000000 000000        14.0.0          ;READ MODEM
5741
5742                ; RESPONSE LIST
5743
5744 026746      81:
5745 026746 000001                1
5746 026750 000214 001440 000400        14.RDY0.S6*400.S142.1*400    ;RESPONSE FOR F14
5747 026756 000000 000000 000000        .WORD 0.0.0.0.0.0
5748

```


5749 026772
 5750 026772 012701 026746
 5751 026776 004737 010134
 5752 027002 103002
 5753
 5754 027004
 5755
 5756 027010
 5757 027010 004737 027016
 5758
 5759 027014

91:

MOV #81,R1
 CALL CHKRSP
 BCC 101

;SAVE RESPONSE LIST ADDRESS
 ;THEN CHECK RESPONSES
 ;IF TEST CORRECT

EXIT TST

101:

CALL SHORTST

;FOR SHORT XMIT/RECEIVE BUFFER TEST

ENDTST

```

5761
5762
5763           ; SHORT XMIT/RECEIVE BUFFER TEST COMMON TO TEST10 AND TEST11
5764
5765 027016     SHORTST:
5766
5767 027016     004737 011776           CALL    INIQIO           ; INIT QIO PROCESSING
5768
5769 027022     012701 027036           MOV     #60,R1          ; TABLE ADDRESS IN R1
5770 027026     004737 013154           CALL    QIOP           ; PROCESS QIO
5771 027032     103065                    BCC    5:             ; IF CORRECT
5772
5773 027034     000207                    RETURN
5774
5775           ; PARAMETERS FOR QIO PROCESSING
5776
5777 027036     027042     60:      3:             ; IN LIST TABLE BASE ADDRESS
5778 027040     000620                    400.                ; TIME-OUT LENGTH(N*10 MS)
5779
5780           ; COMMAND LIST
5781
5782 027042     3:
5783 027042     000007                    7.
5784 027044     000006 002574 000376     6,RXBUF0,#RDBS-2      ; FIRST RECEIVE BUFFER
5785 027052     000006 003174 000367     6,RXBUF1,#RDBS-9.    ; SECOND "
5786 027060     000006 003574 000066     6,RXBUF2,54.         ; THIRD "
5787 027066     000005 004574 000001     5,TXBUF0,1           ; FIRST XMIT BUFFER
5788 027074     000005 004574 000366     5,TXBUF0,#RDBS-10.  ; SECOND "
5789 027102     000005 005174 000367     5,TXBUF1,#RDBS-9.   ; THIRD "
5790 027110     000005 005574 000400     5,TXBUF2,#RDBS      ; FORTH "
5791
5792           ; RESPONSE LIST
5793
5794 027116     4:
5795 027116     000007                    7.
5796 027120     000206 000000 170000     6,RDY0,0,360*400    ; TOO MANY BUFFERS FOR F6
5797 027126     000205 000000 175000     5,RDY0,0,372*400    ; TOO SHORT BUFFER FOR F5
5798 027134     000205 000000 170000     5,RDY0,0,360*400    ; TOO MANY BUFFERS FOR F5
5799 027142     000206 000366 000400     6,RDY0,#RDBS-10..1*400 ; FIRST RECEIVE BUFFER
5800 027150     000205 000000 000400     5,RDY0,0,1*400      ; FIRST BUFFER SENT
5801 027156     000206 000367 000400     6,RDY0,#RDBS-9..1*400 ; SECOND BUFFER RECEIVED
5802 027164     000205 000000 000400     5,RDY0,0,1*400      ; SECOND BUFFER SENT
5803 027172     000000 000000 000000     .WORD 0,0,0,0,0,0
5804
5805 027206     5:
5806 027206     012701 027116           MOV     #4,R1          ; SAVE RESPONSE LIST ADDRESS
5807 027212     004737 010134           CALL    CHKRSR        ; THEN CHECK RESPONSES
5808 027216     103001                    BCC    6:             ; IF TEST CORRECT
5809
5810 027220     000207                    RETURN
5811
5812 027222     012701 004574     6:      MOV     #TXBUF0,R1     ; CHECK FIRST RECEIVE BUFFER
5813 027226     012702 002574           MOV     #RXBUF0,R2
5814 027232     012737 000366 002346     MOV     #RDBS-10..LENGTH
5815 027240     004737 011374           CALL    COMPAR
5816 027244     103001                    BCC    7:             ; IF CORRECT
5817

```

5818	027246	000207			RETURN	
5819						
5820	027250	012701	005174	7#:	MOV	#TXBUF1,R1 ;CHECK SECOND RECEIVE BUFFER
5821	027254	012702	003174		MOV	#RXBUF1,R2
5822	027260	012737	000367	002346	MOV	#RDBS-9.,LENGTH
5823	027266	004737	011374		CALL	COMPAR
5824						
5825	027272	000207			RETURN	

5827 027274

BADHEAD

** TEST12 **

5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864

; TEST OF XMIT/RECEIVE BUFFER PROCESSING AT 48K
; WITH FULL MODEM CONTROL AND ADDRESS SEARCH
;
; CONFIGURATE: SOLC PROTOCOL
; FULL MODEM CONTROL
; WITH ADDRESS SEARCH(252)
; CLOCK SOURCE INTERNAL
; RATE = 48K
;
; TEST: - TEST OF STATUS 1,360,372 AND CORRECT XMIT/RECEIVE
; OF 1 OF 2 BUFFERS.
; THREE BUFFERS ARE PASSED FOR RECEPTION WHILE FOUR
; XMIT BUFFERS ARE GIVEN ONE OF WHICH WITH 1 BYTE LONG
; AND ONE OF WHICH WITH BAD STATION ADDRESS
; EXPECTED RESPONSES ARE:
; STATUS = 360 FOR F6 THIRD BUFFER PASSED
; = 372 FOR F5 1 BYTE LONG BUFFER
; = 360 FOR F5 THIRD BUFFER PASSED
; = 1 FOR F5 FIRST BUFFER XMITTED
; = 1 FOR F5 SECOND BUFFER XMITTED
; = 1 FOR F6 SECOND BUFFER RECEIVED
;
; PATTERN: INCREMENTAL
;
; REPORTS: ERROR 10000 KMV11 FAILS TO RESET MASTER CLEAR
; ERROR 10001 RUN FUNCTION NOT CORRECTLY PERFORMED
; ERROR 10002 WRITE FUNCTION NOT CORRECTLY PERFORMED
; ERROR 10003 READ FUNCTION NOT CORRECTLY PERFORMED
; ERROR 10004 DATA COMPARE ERROR
; ERROR 10005 UNEXPECTED INTERRUPT IN
; ERROR 10006 UNEXPECTED INTERRUPT OUT
; ERROR 10007 NO MORE INTERRUPT WHILE QIO PENDING
; ERROR 10008 UNEXPECTED EPROM'S ON KMV11 A
; ERROR 10009 UNEXPECTED QIO RESPONSE
; ERROR 10010 UNEXPECTED NUMBER OF RESPONSES RECEIVED
; ERROR 10011 UNEXPECTED DATA RECEIVED
; ERROR 10012 NO LOOP BACK

5865 027274

BADHEAD

** TEST12 **

5866
5867
5868 027274
5869 027274 004737 011574
5870 027300 103406
5871
5872 027302 004737 011706
5873 027306 103403
5874
5875 027310 004737 012554
5876 027314 103002
5877
5878 027316
5879
5880 027322 004737 011776
5881

BGNTST

CALL MODEO ;SET APPLICATION MODE
BCS 1# ;IF NOT CORRECT
;
CALL LPBACK ;ANY LOOP?
BCS 1# ;IF NOT
;
CALL RUNAPP ;LOAD AND RUN APPLICATION
BCC 2# ;IF CORRECT
;
1#: EXIT TST
;
2#: CALL INIQIO ;INIT QIO PROCESSING

```

5882 027326 012701 027344      MOV      #60,R1      ;TABLE ADDRESS IN R1
5883 027332 004737 013154      CALL     QIOP        ;PROCESS QIO
5884 027336 103022              BCC      5#         ;IF CORRECT
5885
5886 027340              EXIT     TST
5887
5888      ; PARAMETERS FOR QIO PROCESSING
5889
5890 027344 027350      60#:      3#         ;IN LIST TABLE BASE ADDRESS
5891 027346 000764              500.             ;TIME-OUT LENGTH(N*10 MS)
5892
5893      ; COMMAND LIST
5894
5895 027350      3#:
5896 027350 000001              1
5897 027352 000001 102621 000252      1,SDLC+C48K+ADSRCH+CLKDTE+S141,252 ;CONFIGURATE
5898
5899      ; RESPONSE LIST
5900
5901 027360      4#:
5902 027360 000001              1
5903 027362 000201 000000 000400      1,RDY0,0,1*400 ;CONFIGURATE DONE
5904 027370 000000 000000 000000      .WORD 0,0,0,0,0,0
5905
5906 027404      5#:
5907 027404 012701 027360      MOV      #4,R1      ;SAVE RESPONSE LIST ADDRESS
5908 027410 004737 010134      CALL     CHKRSR     ;THEN CHECK RESPONSES
5909 027414 103002              BCC      6#         ;IF TEST CORRECT
5910
5911 027416              EXIT     TST
5912
5913 027422 004737 011776      6#:      CALL     INIQIO ;INIT QIO PROCESSING
5914
5915 027426 012701 027444      MOV      #61,R1     ;TABLE ADDRESS IN R1
5916 027432 004737 013154      CALL     QIOP        ;PROCESS QIO
5917 027436 103022              BCC      9#         ;IF CORRECT
5918
5919 027440              EXIT     TST
5920
5921      ; PARAMETERS FOR QIO PROCESSING
5922
5923 027444 027450      61#:      7#         ;IN LIST TABLE BASE ADDRESS
5924 027446 000144              100.             ;TIME OUT LENGTH(N*10 MS)
5925
5926      ; COMMAND LIST
5927
5928 027450      7#:
5929 027450 000001              1
5930 027452 000014 000000 000000      14,0,0 ;READ MODEM
5931
5932      ; RESPONSE LIST
5933
5934 027460      8#:
5935 027460 000001              1
5936 027462 000214 001455 000400      14,RDY0,S6*400+S142+S106+S107+S109,1*400 ;RESPONSE FOR F14
5937 027470 000000 000000 000000      .WORD 0,0,0,0,0,0
5938

```

```

5939 027504          98:      MOV      #88,R1          ;SAVE RESPONSE LIST ADDRESS
5940 027504 012701 027460      CALL     CHKRSP        ;THEN CHECK RESPONSES
5941 027510 004737 010134      BCC      108          ;IF TEST CORRECT
5942 027514 103002
5943
5944 027516          EXIT     TST
5945 027522          108:     CALL     INIQIO        ;INIT QIO PROCESSING
5946 027522 004737 011776      CALL     INIQIO
5947
5948 027526 112737 000252 005174  MOVB     #252, TXBUF1  ;SET GOOD STATION ADDRESS IN BUFFER 1
5949
5950 027534 012701 027552      MOV      #628,R1
5951 027540 004737 013154      CALL     QIOP          ;PROCESS QIO
5952 027544 103063      BCC      138          ;IF CORRECT
5953
5954 027546          EXIT     TST
5955
5956          ; PARAMETERS FOR QIO PROCESSING
5957
5958 027552 027556      628:     118          ;IN LIST TABLE BASE ADDRESS
5959 027554 001750      1000.    ;TIME-OUT LENGTH(N*10 MS)
5960
5961          ; COMMAND LIST
5962
5963 027556          118:     7.
5964 027556 000007      6, RXBUF0, #RDBS-2      ;FIRST RECEIVE BUFFER
5965 027560 000006 002574 000376      6, RXBUF1, #RDBS-9.    ;SECOND "
5966 027566 000006 003174 000367      6, RXBUF2, 54.         ;THIRD "
5967 027574 000006 003574 000066      5, TXBUF0, 1          ;FIRST XMIT BUFFER
5968 027602 000005 004574 000001      5, TXBUF0, #RDBS-10.  ;SECOND "
5969 027610 000005 004574 000366      5, TXBUF1, #RDBS-9.   ;THIRD "
5970 027616 000005 005174 000367      5, TXBUF2, #RDBS      ;FORTH "
5971 027624 000005 005574 000400
5972
5973          ; RESPONSE LIST
5974
5975 027632          128:     6
5976 027632 000006      6*RDY0, 0, 360*400    ;TOO MANY BUFFERS FOR F6
5977 027634 000206 000000 170000      5*RDY0, 0, 372*400    ;TOO SHORT BUFFER FOR F5
5978 027642 000205 000000 175000      5*RDY0, 0, 360*400    ;TOO MANY BUFFERS FOR F5
5979 027650 000205 000000 170000      5*RDY0, 0, 1*400     ;FIRST BUFFER SENT
5980 027656 000205 000000 000400      6*RDY0, #RDBS-9..1*400 ;SECOND BUFFER RECEIVED
5981 027664 000206 000367 000400      5*RDY0, 0, 1*400     ;SECOND BUFFER SENT
5982 027672 000205 000000 000400
5983 027700 000000 000000 000000      .WORD 0,0,0,0,0,0
5984
5985 027714          138:     MOV      #128,R1        ;SAVE RESPONSE LIST ADDRESS
5986 027714 012701 027632      CALL     CHKRSP        ;THEN CHECK RESPONSES
5987 027720 004737 010134      BCC      148          ;IF TEST CORRECT
5988 027724 103002
5989
5990 027726          EXIT     TST
5991
5992 027732 012701 005174      148:     MOV      #TXBUF1,R1    ;CHECK RECEIVE BUFFER
5993 027736 012702 002574      MOV      #RXBUF0,R2
5994 027742 012737 000367 002346      MOV      #RDBS 9.,LENGTH
5995 027750 004737 011374      CALL     COMPAR

```

CNKMCAO KMV11 A FCTNL DIAG
HARDWARE TESTS

MACRO M1200 05-APR-84 11:29 PAGE 72-3

SEQ 138

5996
5997 027754

ENDTST

5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015 027756
6016
6017 027760
6018 027770
6019 030000
6020 030012
6021
6022
6023 030024
6024
6025
6026
6033
6034
6035 030024
6036 030051
6037 030107
6038 030137
6039 030226
6040
6041
6042
6043
6044
6045
6046
6047
6048
6057
6058
6065
6066
6067

.SBTTL HARDWARE PARAMETER CODING SECTION

```
;/;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;/
;/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
;/ WITH THE OPERATOR.
;/;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;/
```

BGNHRD

```
GPRMA  ADDRES,0,0,160000,177776,YES
GPRMA  VECTOR,2,0,0,774,YES
GPRMD  PRIRTY,4,0,7000,4,7,YES
GPRMD  LOOPBK,6,0,1,0,1,YES
GPRMD  PDPTST,10,0,1,0,1,YES ;JB REV A 0
```

ENDHRD

.EVEN

```
104 105 126  ADDRESS: .ASCIZ /DEVICE CSR ADDRESS: /
104 105 126  VECTOR: .ASCIZ /DEVICE FIRST VECTOR ADDRESS: /
104 105 126  PRIRTY: .ASCIZ /DEVICE PRIORITY LEVEL: /
111 123 040  LOOPBK: .ASCIZ /IS EXTERNAL LOOP BACK CONNECTOR INSERTED? 0=NO,1=YES: /
111 123 040  PDPTST: .ASCIZ \IS THE HOST COMPUTER A PDP11/23.? 0=NO,1=YES: \
```

.EVEN


```

2
3      043660
4 043660 013352      BUFLGH: .=$FWEND      ;FIRMWARE CODE LENGTH
5 043662 000000      .WORD .-$BUFF
6 043664 000000      .WORD 0
7 043666 000000      .WORD 0
8
9 043670      $PATCH::
10 043670      .BLKW 50
11
12 044010      LASTAD
13 044014      L$LAST::
14 044014      ENDMOD
15

```

17
18
19 044014
20 044014
21 044020 177000
22 044022 000300
23 044024 004000
24 044026 000000
25 044030
26 044030
27
28 000001

BGNSETUP 1
BGNPTAB
.WORD 177000
.WORD 300
.WORD 4000
.WORD 0
ENDPTAB
ENDSETUP
.END

ABORT	020742	BSEL7	= 000007	CG.STR	= 000002	CSRPDV	031120 G	C#GPLO	= 000030
ABOTX	= 000010	BUFCLR	011530	CHARES	= 000030	CS RTE	032042	C#GPRI	= 000040
ADDRESS	030024	BUFLD	011740	CHG	043154	CSTART	035602	C#INIT	= 000011
ADR	= 000020 G	BUFLGH	043660	CHGMDM	033546	CSTOP	035652	C#INLP	= 000020
ADSRCH	= 000200	BUFOVF	= 000373	CHKCO	= 000020	CS.ABO	= 000100	C#MANI	= 000050
APPFLG	002414	BW105	= 000002	CHKCP	= 000040	CS.BUF	= 000200	C#MEM	= 000031
ASSEMB	= 000010	BW108	= 000001	CHKDBL	032320	CS.DCR	= 000400	C#MSG	= 000023
AUXPDV	031070 G	BW111	= 000040	CHKDCP	032414	CS.DEV	= 000002	C#OPEN	= 000034
BAD	002316	BW140	= 000100	CHKDP	= 000100	CS.DIS	= 000040	C#PNTB	= 000014
BADAD	017202 G	BW141	= 000270	CHKEM	= 000200	CS.ENB	= 000020	C#PNTF	= 000017
BADLOC	002432	CB.CCB	= 000002	CHKINV	032246	CS.EOF	= 000001	C#PNTS	= 000016
BAD0	002320	CB.RDB	= 000004	CHKKMV	032332	CS.ERR	= 100000	C#PNTX	= 000015
BAD10	002330	CCONDP	035552	CHKLGT	032370	CS.HFE	= 002000	C#QIO	= 000377
BAD12	002332	CCONF	033434	CHKMAX	013356	CS.LST	= 040000	C#QUE	= 000002 G
BAD14	002334	CCON1	033364	CHKNEX	032306	CS.MTL	= 004000	C#RDBU	= 000007
BAD16	002336	CCON2	035526	CHKOVR	032344	CS.RNG	= 000010	C#REFG	= 000047
BAD2	002322	CDBFER	015333	CHKPDV	031102 G	CS.ROV	= 000004	C#RESE	= 000033
BAD4	002324	CDDO	036252	CHKRET	032254	CS.RSN	= 010000	C#REVI	= 000003
BAD6	002326	CDDS6	036416	CHKRP	= 000004	CS.SUC	= 000001	C#RFLA	= 000021
BIT0	= 000001 G	CDECON	033502	CHKRSP	010134	CS.TMO	= 020000	C#RPT	= 000025
BIT00	= 000001 G	CDIMDM	033546	CHKRW	= 000010	CTUP	036324	C#SEFG	= 000046
BIT01	= 000002 G	CDISAB	035734	CHKSEQ	032274	CTUSS	036540	C#SPRI	= 000041
BIT02	= 000004 G	CDUP	036234	CHKSUC	032356	CTXKI1	033762	C#SVEC	= 000037
BIT03	= 000010 G	CDUS5	036342	CHKTMB	032402	CTXKI2	035466	C#TIM	= 000000 G
BIT04	= 000020 G	CDUS6A	036402	CHKTP	= 000001	CXMT1	033704	C#TPRI	= 000013
BIT05	= 000040 G	CENAB	035700	CHKTW	= 000002	CXMT2	035440	C.BID	= 000003
BIT06	= 000100 G	CENMDM	033546	CHK.AH	= 000014	C#AU	= 000052	C.BL10	= 100010 G
BIT07	= 000200 G	CE.ABO	= 100364	CHK.AL	= 000012	C#AUTO	= 000061	C.BL11	= 100011 G
BIT08	= 000400 G	CE.DBL	= 100363	CHK.FL	= 000002	C#BRK	= 000022	C.BL12	= 100012 G
BIT09	= 001000 G	CE.DCP	= 100356	CHK.PR	= 000010	C#BSEG	= 000004	C.BL13	= 100013 G
BIT1	= 000002 G	CE.DIS	= 100366	CHK.SS	= 000016	C#BSUB	= 000002	C.BL14	= 100014 G
BIT10	= 002000 G	CE.ERR	= 100370	CHK.WR	= 000006	C#CEFG	= 000045	C.BL15	= 100015 G
BIT11	= 004000 G	CE.FCS	= 100367	CHK.WW	= 000004	C#CLCK	= 000062	C.BL16	= 100016 G
BIT12	= 010000 G	CE.INV	= 100362	CLKDTE	= 100000	C#CLEA	= 000012	C.BL17	= 100017 G
BIT13	= 020000 G	CE.KMV	= 100357	CLKRT	040000	C#CLOS	= 000035	C.BSLO	= 100000 G
BIT14	= 040000 G	CE.LAT	= 100370	CMODCH	036022	C#CLP1	= 000006	C.BSL1	= 100001 G
BIT15	= 100000 G	CE.MDM	= 100365	CMODWR	036600	C#CVEC	= 000036	C.BSL2	= 100002 G
BIT2	= 000004 G	CE.MLN	= 100372	COMEXE	030306 G	C#DCLN	= 000044	C.BSL3	= 100003 G
BIT3	= 000010 G	CE.MOP	= 100372	COMPAR	011374	C#DODU	= 000051	C.BSL4	= 100004 G
BIT4	= 000020 G	CE.NTE	= 100361	CONDIS	041246	C#DRPT	= 000024	C.BSL5	= 100005 G
BIT5	= 000040 G	CE.NXM	= 100374	CONDP2	035172	C#DU	= 000053	C.BSL6	= 100006 G
BIT6	= 000100 G	CE.ODD	= 100361	CONDP3	040206	C#EDIT	= 000003	C.BSL7	= 100007 G
BIT7	= 000200 G	CE.OOS	= 100371	CONENB	040704	C#ERDF	= 000055	C.BUF	= 000014
BIT8	= 000400 G	CE.OVF	= 100373	CONF1	032434	C#ERHR	= 000056	C.BUF1	= 000014
BIT9	= 001000 G	CE.RSE	= 000357	CONF2	035232	C#ERRO	= 000060	C.BUF2	= 000024
BI106	= 000010	CE.RTE	= 100376	CONMSN	041274	C#ERSF	= 000054	C.CNT	= 000020
BI107	= 000004	CE.SRC	= 100364	CONSTP	040466	C#ERSO	= 000057	C.CNT1	= 000020
BI109	= 000001	CE.TMB	= 100360	CONSTR	040220	C#ESCA	= 000010	C.CNT2	= 000030
BI125	= 000002	CE.TMO	= 100374	CON2	035162	C#ESUB	= 000003	C.DAI	= 100024 G
BOE	= 000400 G	CF.EOM	= 000004	CON3	040176	C#ETST	= 000001	C.DAO	= 100026 G
BSELO	= 000000	CF.HDR	= 000020	COPY	011552	C#EXIT	= 000032	C.DDI	= 100020 G
BSEL1	= 000001	CF.LB	= 100000	CRCV1	033552	C#FLG	= 000001 G	C.DDO	= 100022 G
BSEL2	= 000002	CF.SOM	= 000010	CRCV2	035500	C#GETB	= 000026	C.DXI	= 100034 G
BSEL3	= 000003	CF.SYN	= 000040	CRDMDM	033546	C#GETW	= 000027	C.DX0	= 100036 G
BSEL4	= 000004	CF.TRN	= 000100	CSRCSC	= 100000	C#GMAN	= 000043	C.FLG	= 000022
BSEL5	= 000005	CG.DVM	= 000004	CSRDP	032154	C#GPHR	= 000042	C.FLG1	= 000022
BSEL6	= 000006	CG.ENB	= 000001	CSRLN	= 000010			C.FLG2	= 000032

C.FNC = 000010	DFPTBL 002164 G	E00002 016743	FS.TX = 005000	G#OFFS= 000400
C.LIN = 000006	DIAGMC = 000000	E00003 017027	FS.XKL = 002000	G#OF SI= 000376
C.LNK = 000000	DIMDM1 033336	E00004 017123	FTIME = 002256	G#PRMA= 000001
C.MDS = 100030 G	DIMDM2 035426	E10000 015610	F#AU = 000015	G#PRMD= 000002
C.MOD = 000011	DISINT= 000050	E10001 015652	F#AUTO= 000020	G#PRML= 000000
C.PBIM= 100032 G	DLO = 000040	E10002 015735	F#BGN = 000040	G#RADA= 000140
C.PRI = 000200	DMAOUT 034464	E10003 016022	F#CLEA= 000007	G#RADB= 000000
C.RSV = 000002	DMAIN 034170	E10004 016106	F#DU = 000016	G#RADD= 000040
C.SELO= 100000 G	DMAOUT 034464	E10005 016171	F#END = 000041	G#RADL= 000120
C.SEL2= 100002 G	DMODCH 036226	E10006 016221	F#HARD= 000004	G#RADO= 000020
C.SEL4= 100004 G	DMODWR 036702	E10007 016252	F#HW = 000013	G#XFER= 000004
C.SEL6= 100006 G	DMODWX 036710	E10008 016334	F#INIT= 000006	G#YES = 000010
C.SL10= 100010 G	DSDO 036306	E10009 016371	F#JMP = 000050	HDLSDL= 000001
C.SL12= 100012 G	DSDS6 036502	E10010 016421	F#MOD = 000000	HELP = 000000
C.SL14= 100014 G	DSTART 035620	E10011 016471	F#MSG = 000011	MOE = 100000 G
C.SL16= 100016 G	DSTOP 035672	E10012 016522	F#PROT= 000021	IBE = 010000 G
C.STA = 000007	DSUP 036270	FC SER = 000367	F#PWR = 000017	IDLE = 030700 G
C.STS = 000012	DSUS3 036430	FC.CCP= 000020	F#RPT = 000012	IDU = 000040 G
C.VECO= 000060	DUMMY1 032426	FC.CTL= 000006	F#SEG = 000003	IEI = 000001
C.VEC2= 000070	DUMMY2 035106	FC.KCP= 000016	F#SOFT= 000005	IEO = 000020
C.WRK = 000004	DUMMY3 037710	FC.KIL= 000004	F#SRV = 000010	IER = 020000 G
C.XXXX= 100000	ECONB = 160002	FC.MAX= 000020	F#SUB = 000002	INIFLG 002516
CO.IEI= 000001	EF.CON= 000036 G	FC.RCE= 000002	F#SW = 000014	INIQIO 011776
CO.IEO= 000020	EF.NEW= 000035 G	FC.RCP= 000014	F#TEST= 000001	INLST 002406
CO.RQI= 000200	EF.PWR= 000034 G	FC.TIM= 000010	F0 = 000000	INTIN 002506
C1.ERR= 000001	EF.RES= 000037 G	FC.XCP= 000012	F1 = 000001	INTOUT 002510
C1.MCL= 000100	EF.STA= 000040 G	FC.XME= 000000	F10 = 000010	ISR = 000100 G
C1.MDE= 000020	ENATX = 000151	FIRST = 000002	F11 = 000011	ITIN 012710
C1.MT1= 000010	END 020750	FRMLER= 000372	F12 = 000012	ITOUT 013044
C1.MT2= 000002	ENDINT= 000070	FS.AST= 000000	F13 = 000013	IXE = 004000 G
C1.RD = 000004	ENDMA = 000040 G	FS.CIB= 002000	F14 = 000014	I#AU = 000041
C1.RUN= 000200	ENDSEI= 000005	FS.CON= 001000	F15 = 000015	I#AUTO= 000041
C1.WRT= 000040	ENDXI1= 000020	FS.CRA= 001000	F16 = 000016	I#CLN = 000041
C111 = 000004	ENDXI2= 000060	FS.DCN= 002000	F17 = 000017	I#DU = 000041
C112 = 000040	ENDXI3= 000070	FS.DIS= 003000	F2 = 000002	I#HRD = 000041
C1200 = 000000	END1 021136	FS.DMS= 017000	F3 = 000003	I#INIT= 000041
C140 = 000010	ENPDM1 033310	FS.DVC= 001000	F4 = 000004	I#MOD = 000041
C141 = 000020	ENPDM2 035406	FS.EMS= 016000	F5 = 000005	I#MSG = 000041
C142 = 000100	ENNIRX= 000040	FS.ENB= 002000	F6 = 000006	I#PROT= 000040
C19K = 002000	ENRX = 000331	FS.KIL= 000000	F7 = 000007	I#PTAB= 000041
C2.RYI= 000020	ENRXAD= 000335	FS.KRX= 010000	GDREV 007126	I#PWR = 000041
C2.RYO= 000200	ENTHUN= 000020	FS.KTX= 007000	GOOD 002274	I#RPT = 000041
C2400 = 000400	ERCNTR 002264	FS.LTM= 001000	GOOD0 002276	I#SEG = 000041
C48K = 002400	ERR = 000400	FS.MCG= 005000	GOOD10 002306	I#SETU= 000041
C4800 = 001000	ERRBLK 002244 G	FS.MCH= 004000	GOOD12 002310	I#SRV = 000041
C56K = 003000	ERRCNT 002430	FS.MCR= 013000	GOOD14 002312	I#SUB = 000041
C64K = 003400	ERRFLG 002370	FS.MOR= 013000	GOOD16 002314	I#TST = 000041
C9600 = 001400	ERRMSG 002242 G	FS.MOW= 014000	GOOD2 002300	J#JMP = 000167
DALON = 000040	ERRNBR 002240 G	FS.MRD= 014000	GOOD4 002302	KINVAL 031344
DATA 002340	ERRTYP 002236 G	FS.MSN= 004000	GOOD6 002304	KIMVA1 031362
DBLCMD= 000363	EVL = 000004 G	FS.RTN= 001000	G#CNT0= 000200	KMVCSR 002504
DCONF= 000356	EXIT0 031370	FS.RX = 006000	G#DELM= 000372	KMVER = 000357
DCONF1 032476	EXIT1 032432	FS.SIA= 011000	G#DISP= 000003	LAST = 000001
DCONF2 035270	EXIT3 037714	FS.SHM= 000000	G#EXCP= 000400	LCLoop 002344
DDISAB 035754	E#END = 002100	FS.STO= 012000	G#HILI= 000002	LC.SM = 000274
DECONS 032662	E#LOAD= 000035	FS.STP= 001000	G#LQI= 000001	LC.S1 = 000076
DENAB 035722	E00000 016611	FS.STR= 000000	G#NO = 000000	LDAPPL 012150
	E00001 016674			

LDLDB	031616	LN.OFF	= 000001	L#INIT	020454	G	L10014	020444	MDMS3	= 000001
LENGTH	002346	LN.ON	= 000000	L#LADP	002026	G	L10015	020452	MDMS5	= 000002
LF.ACT	100000	LN.OOP	= 000004	L#LAST	044014	G	L10016	021136	MDMS6	= 000003
LF.BWT	000007	LN.OPE	= 000001	L#LOAD	002100	G	L10017	021226	MDMS6A	= 000004
LF.DLO	000046	LN.REF	= 000002	L#LUN	002074	G	L10020	021232	MDMS7A	= 000005
LF.ENA	002000	LN.SER	= 000002	L#MREV	002050	G	L10021	021314	MDMS7B	= 000006
LF.LPB	001000	LN.STA	= 000017	L#NAME	002000	C	L10022	021316	MDM.FL	= 000002
LF.MDC	000100	LN.SUB	= 000360	L#PRIO	002042	G	L10023	021442	MDM.LD	= 000006
LF.MFL	004000	LN.TRI	= 000006	L#PROT	002122	G	L10024	021544	MDM.MA	= 000010
LF.MTP	000020	LNOPDV	031140	L#PRT	002112	G	L10025	021564	MDM.NH	= 000004
LF.PAC	000200	LOCK	002260	L#REPP	002062	G	L10026	022104	MDM.OH	= 000011
LF.RDY	040000	LOE	= 040000	L#REV	002010	G	L10027	022210	MDM.ST	= 000005
LF.REA	010000	LOGDEV	002246	L#RPT	020446	G	L10030	022656	MDROP	021262
LF.TIM	000010	LOKFLG	02520	L#SPC	002056	G	L10031	023744	MD105	= 000002
LF.UNL	020000	LOOP	= 001000	L#SPCP	002020	G	L10032	023172	MD108	= 000200
LGT	= 000002	LOOPBK	030137	L#SPTP	002024	G	L10033	023356	MD111	= 000001
LINAR	= 004000	LOT	= 000010	L#STA	002030	G	L10034	023630	MD140	= 000002
LINCP	= 002000	LPBACK	011706	L#SW	002266	G	L10035	023742	MD141	= 000004
LINCR	= 100000	LSTAD	002500	L#TEST	002114	G	L10036	024264	MECO	014126
LINCW	= 000100	LSTBK	002502	L#TIML	002014	G	L10037	024202	MERR	013456
LINCX	= 003500	LSTLGM	002400	L#UIT	002270	G	L10040	024212	MINONEX	025460
LINC1	= 000400	LTCYER	= 000370	L#UNIT	002012	G	L10041	024222	MODDP	033374
LINC2	= 001000	LTIS5	037174	L#CMR	= 120016	G	L10042	024232	MODOWN	= 000365
LINDI	= 010000	LTIS6A	037236	L#COST	= 000013	G	L10043	024242	MODE	= 010000
LIND	= 000002	LTIS7	037264	L#CTL	= 000010	G	L10044	024252	MODE0	011574
LINOV	= 020000	L#ACP	002110	L#C1R	= 120000	G	L10045	024262	MODMDL	= 000026
LINPW	= 000200	L#APT	002036	L#C1W	= 120002	G	L10046	024604	MODPDV	031130
LINRP	= 000004	L#AU	021316	L#C2R	= 120004	G	L10047	024522	MPROM	002524
LINRW	= 000010	L#AUT	002070	L#C2W	= 120006	G	L10050	024532	MPSTR	= 000017
LINTU	= 000020	L#AUTO	021140	L#DDM	= 000002	G	L10051	024542	MQIO	014315
LIN.AD	000330	L#CCP	002106	L#DDS	= 000004	G	L10052	024552	MQION	014706
LIN.A3	000332	L#CLEA	021230	L#DLC	= 000003	G	L10053	024562	MRCVER	015463
LIN.A5	000333	L#CO	002032	L#DLS	= 000006	G	L10054	024572	MRDEXP	015532
LIN.B1	000334	L#DEPO	002011	L#FLG	= 000000	G	L10055	024602	MSEL	014227
LIN.B2	000335	L#DESC	002176	L#KRBA	= 000014	G	L10056	026560	MSEL0	013572
LIN.CP	000342	L#DESP	002076	L#LEN	= 000020	G	L10057	027014	MSEL10	013746
LIN.FL	000304	L#DEVP	002060	L#MPF	= 000020	G	L10060	027754	MSEL12	014002
LIN.LP	000312	L#DISP	002132	L#NMST	= 000016	G	L10061	030024	MSEL14	014036
LIN.NH	000314	L#DLY	002116	L#NSTA	= 000012	G	L10062	044020	MSEL16	014072
LIN.OA	000316	L#DTP	002040	L#ONR	= 000017	G	L10064	044030	MSEL2	013626
LIN.OB	000317	L#DTYP	002034	L#PRI	= 000240	G	MAXERR	002262	MSEL4	013660
LIN.OM	000315	L#DU	021234	L#RTR	= 120010	G	MBBUF0	015117	MSEL6	013712
LIN.PB	000331	L#DUT	002072	L#RTW	= 120012	G	MBBUF1	015224	HXMTER	015411
LIN.RB	000320	L#DVTY	007330	L#UNT	= 000011	G	MBNUM	014425	NEXT	020620
LIN.RP	000306	L#EF	002052	L#VEC	= 000130	G	MCLR	= 040000	NKMCAL	002000
LIN.RS	000322	L#ENVI	002044	L#XXX	= 120000	G	MCSR0	014546	NOCHAN	036224
LIN.RW	000310	L#ERRT	002236	L10001	002176	G	MCSR1	014625	NOTIM	037166
LIN.R3	000340	L#ETP	002102	L10002	017236	G	MCSR2	014575	NXMEM	= 000374
LIN.S0	000336	L#EXP1	002046	L10003	017314	G	MDATO	014764	OUTBUF	006620
LIN.S1	000337	L#EXP4	002064	L10004	017352	G	MDAT1	015044	OUTLST	002410
LIN.TB	000324	L#EXP5	002066	L10005	017434	G	MDMA	= 000040	OUTNUM	002476
LIN.TS	000326	L#HARD	027760	L10006	017552	G	MDM	035374	OUTSEQ	= 000371
LN.CLO	= 000000	L#HIME	002120	L10007	017614	G	MDMCP	= 000002	O#APTS	= 000000
LN.DUM	= 000005	L#HPCP	002016	L10010	017652	G	MDMDL	= 000001	O#AU	= 000000
LN.LOA	= 000004	L#HPTP	002022	L10011	020034	G	MDMDP	= 000004	O#BGNR	= 000000
LN.LOO	= 000003	L#HW	002164	L10012	020142	G	MDMSE	= 000010	O#BGNS	= 000000
LN.OAU	= 000003	L#ICP	002104	L10013	020406	G	MDMS1	= 000000	O#DU	= 000001

O\$ERRT= 000000
O\$GNSW= 000000
O\$POIN= 000001
O\$SETU= 000001
PARTNB= 160004
PAR0 = 172340
PAR7 = 172356
PATLGM 007124
PATTAB 007010
PATTRN 007012
PA.DIP= 000020
PA.TMO= 000200
PA.112= 000040
PA.142= 000100
PBRSP 020036 G
PB.DMA= 000040
PB.108= 000200
PB.111= 000001
PB.140= 000002
PB.141= 000004
PC.DC = 000000
PC.EC = 000001
PC.EC0= 000100
PC.EC2= 000200
PC.GF = 000002
PC.GN = 000003
PC.GRN= 000002
PC.LF = 000010
PC.LN = 000011
PC.RED= 000010
PC.RF = 000006
PC.RN = 000007
PC.RTC= 000001
PC.SCH= 000040
PC.SE = 000012
PC.SI = 000013
PC.SLM= 000020
PC.SM = 000220
PC.YFL= 000004
PC.YF = 000004
PC.YN = 000005
PC.OF = 000014
PC.ON = 000015
PC.2F = 000016
PC.2N = 000017
PDDEF = 077406
PDPTST 030226
PDPTYP 002514
PDRO = 172300
PDVNM = 000005
PNT = 001000 G
PRBCOM 020144 G
PRBECO 017316 G
PRBNUM 020410 G
PRBQIO 017654 G
PRDAT 017354 G
PRI = 002000 G

PRILEV 002512
PRIRTY 030107
PRI00 = 000000 G
PRI01 = 000040 G
PRI02 = 000100 G
PRI03 = 000140 G
PRI04 = 000200 G
PRI05 = 000240 G
PRI06 = 000300 G
PRI07 = 000340 G
PRMNB 002556
PRQION 017616 G
PRSEL 017554 G
PRSELO 017240 G
PRSTAT 017436 G
PRO = 000000
PR1 = 000040
PR2 = 000100
PR3 = 000140
PR4 = 000200
PR5 = 000240
PR6 = 000300
PR7 = 000340
PSTACK 002250
P.CMR = 130016 G
P.RPA = 130000 G
P.WPB = 130012 G
P.WPC = 130006 G
P.XXX= 130000
QC.CTO= 000200
QC.DCO= 000001
QC.HLT= 000002
QC.VCO= 000040
QC.VC4= 000100
QIOP 013154
QV.FLG 002521
Q.CTL = 140000 G
Q.XXX= 140000
RATE 002342
RCDATA= 000001
RCRCRX= 000100
RCRCTX= 000200
RCVADD 002422
RCVBUF 002426
RCVKIL 033132
RCV1 032712
RDATA 007400
RDBOK 037716
RDMDM1 033260
RDMDM2 035350
RDYI = 000020
RDYIDN 031756
RDYIST 031376
RDYD = 000200
RDYODN 031710
RDYOST 031456
READ = 002000

REANEX 034040
REASEN 034050
REASHO 034030
REASN 034056
REAWI1 03114
REAWI3 043436
RECENA 032132
REGADR 007130
REGO 002350
REG1 002352
REG2 002354
REG3 002356
REG4 002360
REG5 002362
REG6 002364
REG7 002366
REQCNT 002372
REVCHK 012362
REXTST= 000020
RMARK0 021006
RMARK1 021064
RQI = 000200
RSPCNT 002402
RSPLST 010554
RSPOKE 006574
RTXINT= 000050
RUN = 100000
RUNAPP 012554
RUNDTX= 000300
RUNNIN 020752
RXABO = 000364
RXBUF0 002574
RXBUF1 003174
RXBUF2 003574
RXENA = 000001
R9901 042054
SAVE4 002252
SAVE6 002254
SCABOR= 000200
SCCRC = 000100
SCENDR= 000200
SCIP = 000002
SCOVER= 000040
SCTBE = 000004
SC.DTR= 000200
SC.MOF= 000005
SC.RTS= 000002
SC106 = 000040
SC107 = 000010
SC109 = 000010
SC125 = 000040
SDLC = 000001
SECON = 000001
SELNUM 002416
SELO = 000000
SEL10 = 000010
SEL12 = 000012

SEL14 = 000014
SEL16 = 000016
SEL2 = 000002
SEL4 = 000004
SEL6 = 000006
SEND 032222
SENDCO 032214
SENDTX 032236
SETUP 020612
SF.ACT= 000200
SF.ENA= 000100
SF.LPB= 000004
SF.PAC= 000020
SF.REA= 000010
SF.UNL= 000040
SHORTS 027016
SLINM = 00002
SLT0 031044 G
SLT1 031056 G
SRO = 177572
SR3 = 172516
SSTACK 007330
STAT1 037440
STAT3 037472
STAT5 037524
STAT6 037556
STAT6A 037566
STAT7A 037622
STAT7B 037632
STA6 037576
STA7 037642
STIS7 036776
SUB1 024606
SUB2 025064
SUB3 025220
SUB4 025514
SUB5 025654
SUB6 026030
SUB7 026166
SUCCES= 000001
SVCGBL = 000000
SVCINS= 177777
SVCSUB= 177777
SVCTAG= 177777
SVCTST= 177777
SYNMO= 000040
SILSYM= 010000
S.CMDA= 110006 G
S.CMDB= 110016 G
S.COST= 000001
S.FLG = 000000
S.LEN = 000004
S.LOAD 002412
S.NMST= 000002
S.OWNR= 000003
S.PRI= 000340
S.PRI5= 000240

S.PRIX= 000300
S.RCVA= 110000 G
S.RCVB= 110010 G
S.STSA= 110004 G
S.STSB= 110014 G
S.VCRA= 000140
S.VCRB= 000150
S.VCSP= 000120
S.VCXA= 000100
S.VCXB= 000110
S.XMTA= 110002 G
S.XMTB= 110012 G
S.XXX= 110000
S1 = 000000
S106 = 000010
S107 = 000004
S109 = 000001
S111 = 000004
S112 = 000020
S125 = 000002
S140 = 000010
S141 = 000020
S142 = 000040
S3 = 000001
S5 = 000002
S6 = 000003
S6A = 000004
S7A = 000005
S7B = 000006
TEMP 002374
TFM36 015537
TIC = 000001
TIMDP 036742
TIMOUT 036716
TMOUT 002404
TOOBUF= 000360
TSTF0 010614
TSTF1 010640
TSTF10 011124
TSTF11 011150
TSTF12 011174
TSTF13 011220
TSTF14 011244
TSTF15 011270
TSTF16 011314
TSTF17 011340
TSTF2 010664
TSTF3 010710
TSTF4 010734
TSTF5 010760
TSTF6 011004
TSTF7 011030
TSTF8 011054
TSTF9 011100
TXBUF0 004574
TXBUF1 005174
TXBUF2 005574

TXENA = 000010	T\$\$TES= 010060	XMT3 040030	\$CMPDV 000236 G	\$LNOLG= 000021 G
TYPOO 011364	T1 021320 G	XSTART 035644	\$CMQIN= 173610	\$LNOTB 000302 G
TYPO1 011367	T10 026326 G	X\$ALWA= 000000	\$CMQRM= 173624	\$LSTIN= 177777
T\$ARGC= 000001	T11 026562 G	X\$FALS= 000040	\$CRCT 043620	\$LSTTA= 177777
T\$CODE= 003032	T12 027274 G	X\$OFFS= 000400	\$CRCTC 043614	\$LTrNC 000270 G
T\$ERRN= 000003	T2 021444 G	X\$TRUE= 000020	\$CRCTX 043512	\$MODD= 035064 G
T\$EXCP= 000000	T3 021546 G	X990\$ 041762	\$CSLO 031172 G	\$MODLG= 000005 G
T\$FLAG= 000040	T4 021566 G	X999\$ 041772	\$CSL2 031232 G	\$MODTB 000372 G
T\$FREE= 044030	T5 022106 G	ZF.COU= 001000	\$CSL2\$ 031264	\$OVERR 043352
T\$GMAN= 000000	T6 022212 G	ZF.DDM= 000001	\$CSRDP 031150 G	\$PATCH 043670 G
T\$HILI= 000001	T7 022560 G	ZF.DLC= 000002	\$CSRLG= 000003 G	\$PDVNM 000206 G
T\$LAST= 000001	T7.1 022706	ZF.LLC= 000004	\$CSRTB 000274 G	\$PDVTA 000200 G
T\$LOLI= 000000	T7.2 023174	ZF.LMC= 000100	\$CTCMP= 173006	\$PDVTB 031022 G
T\$LSYM= 010000	T7.3 023360	ZF.MFL= 000010	\$DBG = 000001	\$RAM = 030306
T\$LTNO= 000014	T7.4 023632	ZF.MUX= 000040	\$DDCCP= 172632	\$RCCMP= 173026
T\$NEST= 177777	T8 023746 G	ZF.TIM= 000200	\$DDDIS= 172474	\$RDBAF 000230 G
T\$NSO = 000000	T8.1 024174	Z.DAT = 000010	\$DDENB= 172466	\$RDBC = 000017
T\$NS1 = 000004	T8.2 024204	Z.DSP = 000000	\$DDKCP= 172622	\$RDBCT 000224 G
T\$NS2 = 000002	T8.3 024214	Z.FLG = 000006	\$DDKIE= 172422	\$RDBGT= 172074
T\$PCNT= 000000	T8.4 024224	Z.LEN = 000010	\$DDMSN= 172502	\$RDBIN 030742 G
T\$PTAB= 010063	T8.5 024234	Z.LLN = 000004	\$DDRCP= 172612	\$RDBLH 000250 G
T\$PTHV= 000001	T8.6 024244	Z.MAP = 000012	\$DDSTP= 172460	\$RDBNM 000216 G
T\$PTNU= 000001	T8.7 024254	Z.NAM = 000002	\$DDSTR= 172452	\$RDBQP= 172162
T\$SAVL= 177777	T9 024266 G	Z.SCH = 000005	\$DDXME= 172406	\$RDBRT= 172216
T\$SEGL= 177777	T9.1 024514	\$ABORE 043270	\$DDXMP= 172602	\$RDBS = 000400
T\$SIZE= 000006	T9.2 024524	\$AUXDP= 174372	\$DLCRQ= 173034	\$RDBSZ 000220 G
T\$SUBN= 000000	T9.3 024534	\$BREAK 030702 G	\$ENDRC 042426	\$RDBTH 000234 G
T\$TAGL= 177777	T9.4 024544	\$BUFF 030306 G	\$ENDRE 043412	\$RDBWT= 172140
T\$TAGN= 010065	T9.5 024554	\$CCBAF 000226 G	\$FRKHD 000242 G	\$RDQCT 000254 G
T\$TEMP= 000000	T9.6 024564	\$CCBC = 000005	\$FWEND= 043660	\$RDQSL 000252 G
T\$TEST= 000014	T9.7 024574	\$CCBCT 000222 G	\$HIGH 000272 G	\$REC 043276
T\$TSTM= 177777	UAM = 000200 G	\$CCBIN 030704 G	\$INTSX= 173246	\$SLTMA 000202 G
T\$TSTS= 000001	UNIT 002272	\$CCBLH 000246 G	\$INTX7= 173346	\$SLTMB 031034 G
T\$\$AU = 010022	UNLDB 031516	\$CCBNM 000212 G	\$KCE.E= 000102	\$SLTMM 000210 G
T\$\$AUT= 010017	UUT 002522	\$CCBRT= 172352	\$KCE.I= 000200	\$START= 030306
T\$\$CLE= 010020	VECTOR 030051	\$CCBS = 000034	\$KLCMP= 173016	\$STKDP 000240 G
T\$\$CAT= 010064	WDATA 007506	\$CCBSZ 000214 G	\$LDBAF 000232 G	\$STMDA 000262 G
T\$\$DU = 010021	WRIRET 034156	\$CHKDP 032110 G	\$LDBGT= 172044	\$STMDB 000264 G
T\$\$HAR= 010061	WRITE = 020000	\$CHKLG= 000013 G	\$LDBRT= 172216	\$STMFC 000266 G
T\$\$HW = 010001	XDP 042560	\$CHKTB 000344 G	\$LINDP 037666 G	\$XMCMP= 172776
T\$\$INI= 010016	XMTADD 002420	\$CKINT= 174042	\$LLCRQ= 173116	\$XXX 041734
T\$\$MSG= 010014	XMTBUF 002424	\$CLKCT 000256 G	\$LLCTA 000204 G	\$\$STR = 160000
T\$\$PC = 000001	XMTKIL 033232	\$CLKDC 000260 G	\$LLCTB 031040 G	\$\$\$ADD= 030306
T\$\$PRO= 010000	XMTKI2 035150	\$CLKTK= 000024	\$LNAXA 042002 G	\$\$\$STR 030306 G
T\$\$PTA= 010063	XMTKI3 040122	\$CLKVL= 005216	\$LNAXA 042516 G	.WAIT 007336
T\$\$RPT= 010015	XMT1 033002	\$CLMEM 031004 G	\$LNTXA 041572 G	..SIZE 007620
T\$\$SUB= 010055	XMT2 035114			

. ABS. 044030 000
000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 31368 WORDS (123 PAGES)
DYNAMIC MEMORY: 19748 WORDS (75 PAGES)
ELAPSED TIME: 00:16:48

MMCAO.BIN,MMCAO.SEQ/CRF/-SP=LIBA/ML,MMCAO,KEXDEF,KASTOR.TMP,PGEND.MAC