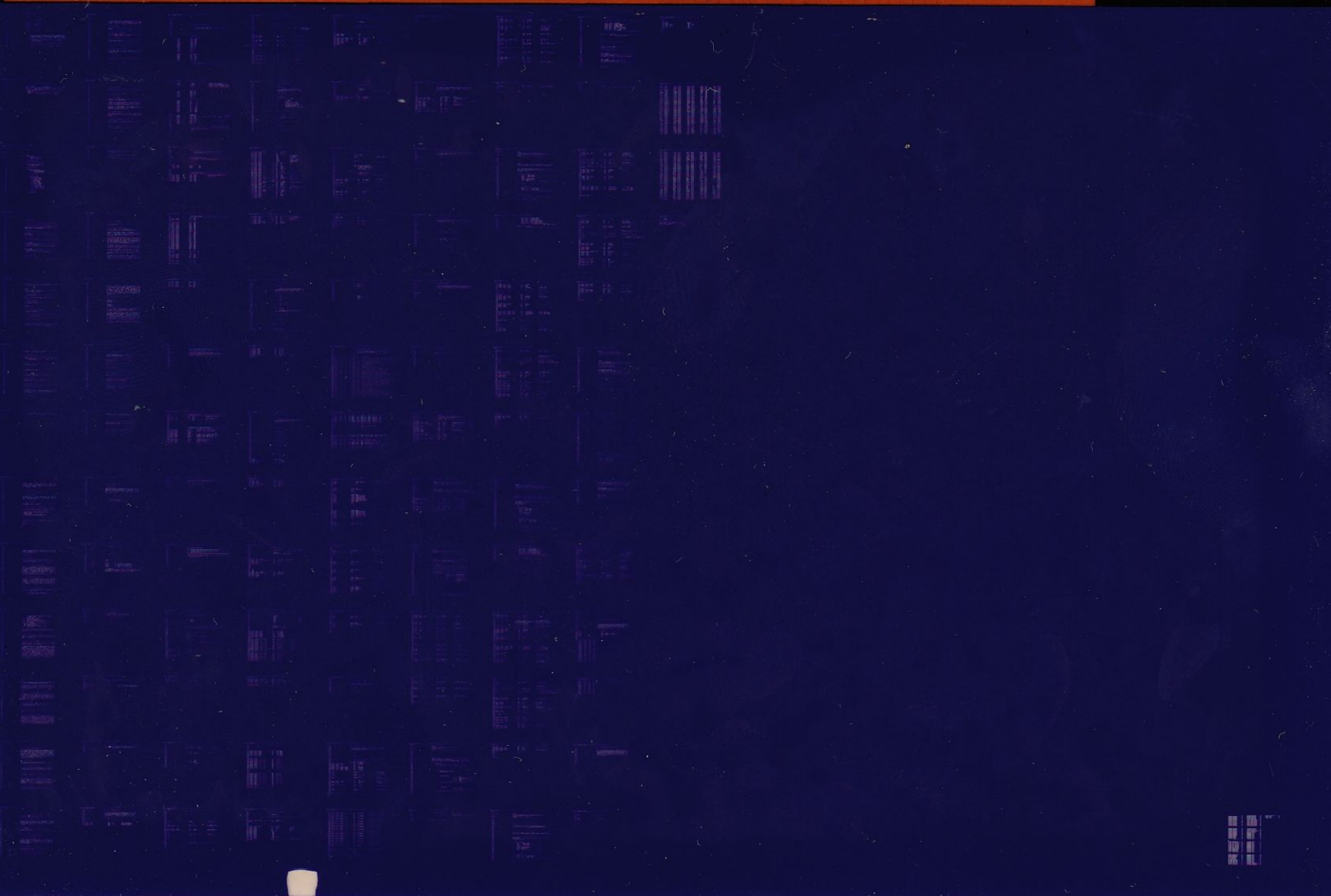


11/21+
KMV 11A

KMV 11A LINE CTRL DIAG
CNKMBRO

COPYRIGHT (c) 1982-84
AH-T845A-MC
FICHE 01 OF 01

JUL 1984
digital
Made In USA



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

.NLIST TOC
.REM @

IDENTIFICATION

PRODUCT CODE: AC T844A MC
PRODUCT NAME: CNKMBAO KMV11A LINE CNT DIAG
PRODUCT DATE: APRIL 1984
MAINTAINER: ISS DIAGNOSTICS
AUTHOR: MICHELET GU
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) 1979,1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	VNIBUS	MASBUS
DEC	DECUS	DECTAPE	

43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

***** MODIFICATION HISTORY *****

REV A: ORIGINAL RELEASE: GUY MICHELET 14 JAN 81

CVKMBA -> CNKMBA JAKI BERG 9-APR 84
CHANGES WERE MADE TO CVKMBA TO PRODUCE CNKMBA FOR THE FALCON-PLUS PROJECT
(SBC 11/21*). CHANGES, MARKED BY ";JB REV A 0", ARE:
SET THE ODT BREAK VECTOR (LOCATION 140) TO THE STARTING ADDRESS OF
FALCON'S ODT ROM (170000 OCTAL).
CHANGE PRIORITY FROM LEVEL 7 TO LEVEL 6 TO ALLOW THE BREAK KEY TO
INTERRUPT.
LOWERED RATE FROM 72KB TO 64KB IN TEST 06. IT IS HOPED THAT THIS
CHANGE WILL BE REVERSED IN A FUTURE RELEASE. SEE TEST 6.

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

TABLE OF CONTENTS

- 1.0 INTRODUCTION
 - 1.1 PROGRAM ABSTRACT
 - 1.2 HARDWARE INTRODUCTION
 - 1.3 DIAGNOSTIC DESCRIPTION
- 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

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

1.0 INTRODUCTION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST OUT THE KMV11 MODULE
THE PROGRAM WAS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR.
THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM WILL ALLOW
MODIFICATION OF DEVICE PARAMETERS, SUCH AS UNIBUS ADDRESS,
VECTOR ADDRESS, AND PROCESSOR TYPE.

1.2 HARDWARE INTRODUCTION

THIS DIAGNOSTIC WILL TEST ALL THE HARDWARE PART OF THE KMV11 A
MODULE (M7500).
TO TEST COMPLETELY THIS PART ,EXTERNAL LOOP BACK CONNECTOR
MUST BE INSTALLED.

EXTERNAL LOOP BACK CONNECTOR:

KMV11 A CAN OPERATE EITHER IN RS422 OR RS 423 LEVEL CONVERTERS

RS422 LOOP BACK:

TO TEST COMPLETELY A KMV11 B IN RS422 MODE ,RUN THIS DIAGNOSTIC
WITH LOOP BACK CONNECTOR PLUG .
USE M3255 TO LOOP DIRECTLY AT THE OUTPUT OF THE MODULE
USE M3251 PLUG AT THE END OF BC55U MODEM CABLE CONNECTOR ASSY.

RS423 LOOP BACK:

TO TEST COMPLETELY A KMV11 A IN RS423 MODE ,RUN THIS DIAGNOSTIC
WITH LOOP BACK CONNECTOR PLUG ;
USE M3255 TO LOOP AT THE OUTPUT OF THE MODULE
USE M3251 PLUG AT THE END OF BC55M MODEM CABLE CONNECTOR ASSY.

RS232 LOOP BACK:

SAME AS FOR RS423.

CAUTION:

USE OF M3251 LOOP BACK CONNECTOR WILL CAUSE MESSAGES ERROR IN TEST 8.

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
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222

DIAGNOSTIC WILL TEST KMV11 CLOCKS, LINE INTERRUPTS, TX AND RX FUNCTION
IN INTERNAL AND EXTERNAL LOOP BACK AND MODEM SIGNALS.

CAUTION:

AT THE BEGINNING OF THE DIAGNOSTIC THE OPERATOR WILL ANSWER
BY "YES " OR "NO " AT THE QUESTION:
IS EXTERNAL CONNECTOR PLUGGED?

IF CONNECTOR NO PLUGGED THE DIAGNOSTIC WILL REPORT AN ERROR
AND EXIT CORRESPONDING TEST.

KMV11 A IS FULLY TESTED ONLY WHEN DIAGNOSTIC HAS BEEN RUN
SUCCESSFULLY IN BOTH RS422 AND RS423 LOOP BACK.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE KMV11 A
LINE CONTROLLER STATIC TESTS:

- SBC-11/21.
- 16K MEMORY
- CONSOLE TERMINAL

3.0 PRELIMINARY PROGRAM REQUIREMENTS

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

*
* NOTE: THE KMV11 DIAGNOSTICS NKMDA AND NKMBA SHOULD BE *
* BEFORE RUNNING NKMCA. *
*

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

224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280

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 KMV11 LINE CNT DIAGNOSTIC IS ABOUT :

- 160 SECONDS FROM TEST 1 TO TEST 6 (TEST IN INTERNAL LOOP).
- 260 SECONDS FROM TEST 1 TO TEST 8 (COMPLETE TEST, WITH EXTERNAL CONNECTOR).

4.3 XXDP.

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

4.4 ACT/SLIDE

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

4.5 APT

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

4.6 MEMORY MANAGEMENT

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 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).

5.0 PROGRAM LOAD MEDIA

HJ

KMV11A LINE CNT DIAG
PROGRAM DOCUMENT

MACRO M1.00 09 APR 84 17:38 PAGE 6 1

SEQ 7

281
282
283

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE
ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM

285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339

ANY MEDIA SUPPORTED BY XXDP*. WHEN USING THE PAPER TAPE ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP*, THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

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

6.1.2 STARTING PROCEDURES

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

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

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

6.2 INITIAL DIALOGUE

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

DRS LOADED
DIAG. RUN TIME SERVICES
NKMBAO
KMV11 A LINE CONTROLLER DIAGNOSTIC
DR>

11

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
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

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

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

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

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

<TEST LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR
RANGES OF DECIMAL NUMBER, (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 OCCURRENCE 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

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
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

CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR

IER INHIBIT ERROR REPORTING
IBE INHIBIT BASIC ERROR REPORTS
IXE INHIBIT EXTENDED ERROR REPORTS
PRI DIRECT ALL MESSAGES TO A LINE PRINTER
PNT PRINT NUMBER OF TEST BEING EXECUTED
BOE BELL ON ERROR
UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR INHIBIT STATISTICAL REPORTS
IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
LOT LOOP ON TEST

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

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

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

6.3.1.5 EFFECT OF START COMMAND

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

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION "N 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.

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
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509

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

EXAMPLE:

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

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

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

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

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
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564

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>)

<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>

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
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620

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

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

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
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676

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

6.3.7 PRINT COMMAND

PRI(NT)

6.3.7.1 EFFECT OF PRINT COMMAND

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE 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

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.

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
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734

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 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.

2. MICRO CPU CSR ADDRESS: (O) 177000?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000 177776 (OCTAL), AND THE DEFAULT IS 177000.

3. MICRO CPU VECTOR ADDRESS: (O) 300?

THE ALLOWABLE RANGE IS 300 770, AND DEFAULT VALUE IS 300

4. MICRO CPU PRIORITY LEVEL: (4) ??

35
36
37
38
39
40
41
42
43
44
45
46
47
48

DEFAULT VALUE IS 4

NOTE:

M7500 AND M7501 MODULE MOUNTED WITH DC003 CHIPS CAN ONLY
INTERRUPT ON LEVEL 4

5. IS LOOP BACK CONNECTOR PLUGGED? 0=NO,1=YES (0) 1 ?

DEFAULT VALUE IS 1 (YES)

NOTE :

REFER TO CHAPTER 1.2 FOR LOOP BACK CONNECTOR DESCRIPTION.

750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
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

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 2 OF THE STATIC LOGIC TESTS.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

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

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

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

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

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

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

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

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
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850

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:

0 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.

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

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
889
890
891
892
893
894
895
896
897
898
899
900
901

7.0 TEST DESCRIPTIONS

***** TEST 1 *****
*VERIFY THAT REFERENCED UNIBUS DEVICE REGISTERS
*DOES NOT CAUSE TIME OUT TRAP

***** TEST 2 *****
*
*PROM REVISION TEST
*

***** TEST 3 *****
*
*REAL TIME CLOCK TEST
*

***** TEST 4 *****
*
*BAUD RATE GENERATOR TEST
*

***** TEST 5 *****
*
*TRANSMIT FRAMES AT LOW SPEED IN INTERNAL LOOP
*ON CHANNEL A WITHOUT ANY INTERRUPT
*

***** TEST 6 *****
*
*TRANSMIT AND RECEIVE FRAMES IN INTERNAL LOOP AT
*DIFFERENT SPEED WITH INTERRUPT
*

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

CAUTION:
TEST NUMBER 7 AND 8 LOOP BACK CONNECTOR MUST BE INSTALLED.
REFER TO CHAPTER 1.2 FOR LOOP BACK DESCRIPTION

***** TEST 7 *****
*
*TRANSMIT AND RECEIVE FRAMES IN EXTERNAL LOOP BACK
*(WITH EXTERNAL LOOP BACK)
*

***** TEST 8 *****
*
*TEST ALL MODEM SIGNAL IN EXTERNAL LOOP BACK
*

930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961

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 AND EXTENDED ERROR INFORMATION.

9.0 HISTORY

DESIGN STARTED ON MAY 82
REVIEW ON DECEMBER 82

a

```

963          .TITLE KMV11 A LINE CNT DIAGNOSTIC
971          002000          .*2000
972
973
974
975
976
977
978          .MCALL  SVC
979 002000          SVC          ; INITIALIZE SUPERVISOR MACRO
980
981
982
983
984
985 002000          GNMOD  KMV11A
986
987
988          000000          $LSTIN= 0
989          000000          $LSTTAG= 0
990          177777          SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
991          177777          SVCTS= 1      ; LIST TEST TAGS, SHIFTED RIGHT
992          177777          SVCSUB= 1     ; LIST SUBTEST TAGS, SHIFTED RIGHT
993          177777          SVCGBL= 1    ; LIST GLOBAL TAGS, SHIFTED RIGHT
994          177777          SVCTAG= 1    ; LIST OTHER TAGS, SHIFTED RIGHT
995
996          ;          CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
997          ;          TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
998          ;          SYMBOLS TO BE MINUS ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
999          ;          CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1000
1001

```

K?

1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1030
1031
1032

002000

002000

.SBTTL PROGRAM HEADER
: *
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:

POINTER BGNSW,BGNDU,BGNSETUP

HEADER NKMBA0,A,0.240..0

12

1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1070
1071
1072
1073

002122

000000

177777

177777

002130

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

BGNPROT

0

1

1

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

ENDPROT

1075
1076
1077
1078
1079
1080
1081
1082 002130
1083
1090
1091

.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 8

1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1114
1115
1116
1117
1118
1119
1120

002152

002154 177000
002156 000300
002160 004000
002162 000001
002164

```

.SBTTL DEFAULT HARDWARE P TABLE
;/////////////////////////////////////////////////////////////////
;/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
;/ THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
;/ IS IDENTICAL TO THE STRUCTURE OF THE RUN TIME P-TABLE.
;/ AND IS USED AS A "TEMPLATE" FOR BUILDING THE P TABLE
;/////////////////////////////////////////////////////////////////
.ENABL  AMA      DFPTBL
      BGNHW

      .WORD 177000 ;KMV11.CSRS ADDRESS
      .WORD 300   ;KMV11. VECTOR ADDRESS
      .WORD 4000 ;INTERRUPT PRIORITY LEVEL
      .WORD 1    ;LOOP BACK CONNECTOR?
      ENDDHW

```

1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1149
1150
1165
1166 002164

.SBTTL GLOBAL EQUATES SECTION

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

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

; BIT POSITION IN SECOND STATUS WORD

```

000040 EF.START.. 32. ; (100000) START COMMAND WAS ISSUED
000037 EF.RESTART.. 31. ; (040000) RESTART COMMAND WAS ISSUED
000036 EF.CONTINUE.. 30. ; (020000) CONTINUE COMMAND WAS ISSUED
000035 EF.NEW.. 29. ; (010000) A NEW PASS HAS BEEN STARTED
000034 EF.PWR.. 28. ; (004000) A POWER FAIL/POWER UP OCCURRED

```

```

;
; PRIORITY LEVEL DEFINITIONS
;

```

```

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

```

```

; OPERATOR FLAG BITS
;

```

```

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

```

1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189

```

000300 ;MAXPRI..340
054000 MAINT0..54000
044000 MAINT1..44000
040000 MCLR..40000
052525 DATA1.. 052525
125252 DATA2.. 125252
013224 KB1.2.. 5780.
000154 KB64.. 108.
000174 KB56.. 124.
000146 KB68.. 102.
000141 KB72.. 97.

```

```

;JB REV A.0
;JB REV A.0
;MASTER CLEAR = 1,MODE = 1,MAINT 1 = 1,T11=HOLD
;MASTER CLEAR = 1,MODE = 0,MAINT 1 = 0,T11=NOT HOLD

```

```

;OCTAL VALUE OF 1.2 KBAUDS
;      . 64
;      . 56
;      . 68
;      . 72

```

```

;DIVIDER CALCULATION
;DECIMAL VALUE = 6912.000 KBAUDS

```

```

;*****
; PROGRAM EVENT FLAG DEFINITIONS
;*****

```

1191
1192
1193
1194
1195
1196
1197
1203
1204
1205
1206
1207
1208 002164
1209
1210
1211
1224
1225 002220
002220 000000
002222 000000
002224 000000
002226 000000

1226
1227
1228
1229
1230
1231
1232
1233 002230 000000
1234 002232 000005
1235 002234 000000
1236 002236 000000
1237 002240 000015
1238 002242 000000
1239 002244 000000
1240 002246 000000
1241 002250 000000

```
.SBTTL GLOBAL DATA SECTION  
;/////////////////////////////////////////////////////////////////  
; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED  
; IN MORE THAN ONE TEST.  
;/////////////////////////////////////////////////////////////////  
  
;*****  
; * STORAGE FOR DEVICE REGISTERS  
;*****  
DESCRIPT <KMV11A LINE CNT DIAGNOSTIC>  
  
ERRTBL  
ERRTYP: .WORD 0  
ERRNBR: .WORD 0  
ERRMSG: .WORD 0  
ERRBLK: .WORD 0  
  
;*****  
; * PROGRAM CONTROL PARAMETERS  
;*****  
LOCK: .WORD 0 ;ADDRESS FOR LOCK CURRENT DATA  
MAXERR: .WORD 5 ;MAX ERROR BEFORE DROPPING THE UNIT  
ERRCNT: .WORD 0 ;ERROR COUNT  
L$SW: .WORD 0  
L$UIT: .WORD 15 ;MAX LINE UNIT  
LOGDEV: .WORD 0  
SAVPC: .WORD 0  
PSTACK: .WORD 0  
FTIME: .WORD 0
```

```

1243
1244
1245
1246 002252 000000
1247 002254 000000
1248 002256 000000
1249
1250 002260 000000
1251 002262 000000
1252 002264 000000
1253 002266 000000
1254 002270 000000
1255 002272 000000
1256 002274 000000
1257 002276 000000
1258 002300 000000
1259 002302 000000
1260 002304 000000
1261 002306 000000
1262 002310 000000
1263 002312 000000
1264 002314 000000
1265 002316 000000
1266 002320 000000
1267 002322 000000
1268 002324 000000
1269 002326 000000
1270 002330 000000
1271 002332 000000
1272 002334 000000
1273 002336 000000
1274 002340 000000
1275 002342 000000
1276 002344 000000
1277 002346 000000
1278 002350 000000
1279 002352 000000
1280 002354 000000
1281 002356 000000
1282 002360 000000
1283
1284 002362
1285 006362
1286
1287 012362 000000
1288 012364 000000
1289 012366 000000
1290 012370 000000
1291 012372 000000
1292 012374 000000
1293
1294
1295 012376 000000
1296 012400 000000
1297
1298 012402 000000
1299 012404 000000

```

```

;*****
;* MISCELLANEOUS STORAGE
;*****
SAVE4: .WORD 0
SAVE6: .WORD 0
FLAG: .WORD 0

DELCT1: .WORD 0
DELCT2: .WORD 0
GOOD: .WORD 0
GOOD0: .WORD 0
GOOD1: .WORD 0
GOOD2: .WORD 0
GOOD4: .WORD 0
GOOD6: .WORD 0
GOOD10: .WORD 0
GOOD12: .WORD 0
GOOD14: .WORD 0
GOOD16: .WORD 0
SELO: .WORD 0
SEL1: .WORD 0
SEL2: .WORD 0
SEL4: .WORD 0
SEL6: .WORD 0
SEL10: .WORD 0
SEL12: .WORD 0
SEL14: .WORD 0
SEL16: .WORD 0
BSEL1: .WORD 0
RANST: .WORD 0
RANSEL: .WORD 0
RANMTA: .WORD 0
RANDN: .WORD 0
SAVPC1: .WORD 0
SAVSTA: .WORD 0
COUNT: .WORD 0
NUMBER: .WORD 0
ADDR: .WORD 0
GDDAT: .WORD 0
BDDAT: .WORD 0

TTABLE: .BLKW 2000
RTABLE: .BLKW 2000

EXADDR: .WORD 0
INTFLG: .WORD 0
BAD: .WORD 0
BSELO: .WORD 0
DATA: .WORD 0
VECT: .WORD 0

KIND: .WORD 0
CHANEL: .WORD 0

TXDATA: .WORD 0
RXDATA: .WORD 0

```

;=0 IF KMV11A .=1 IF KMV11B

1300 012406 000000
1301 012410 000000
1302 012412 000000
1303 012414 000000
1304 012416 000000
1305 012420 000000
1306 012422 000000

TSPEED: .WORD 0
LENGTH: .WORD 0
NUB: .WORD 0
RXCNT: .WORD 0
STAERR: .WORD 0
WRDCNT: .WORD 0
UNIT: .WORD 0

1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330

012424 000001

```
*****  
;LOAD IN LOCATION "GDREV" THE PROM VERSION NUMBER THAT IS *  
;COMPATIBLE WITH THIS DIAGNOSTIC *  
; *  
;EACH PROM CONTAIN A REV LEVEL AND A ECO LEVEL: *  
;THE REV LEVEL IS MODIFIED EACH TIME A MODIFICATION IS DONE *  
;THE ECO LEVEL IS MODIFIED WHEN THE PROM MODIFICATION NEED *  
;A DIAGNOSTIC MODIFICATION *  
*****
```

GDREV: .WORD 1

```

1332 ;*****
1333 ;* PROGRAM CONTROL FLAGS
1334 ;*****
1335 012426 000 INIFLG: .BYTE 0 ;PROGRAM INITIALIZING FLAG
1336 .EVEN
1337 012430 000 LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
1338 012431 000 QV.FLG: .BYTE 0 ;QUICK VERIFY FLAG
1339 .EVEN
1340 012432 000000 UUT: .WORD 0 ;CURRENT UNIT UNDER TEST
1341
1342
1343
1344
1345
1346
1347 ;*****
1348 ;* POINTERS TO KMV11 VECTORS AND REGISTERS
1349 ;*****
1350 012434 000000 KMVV00: 0 ;POINTER TO KMV11 INTRPT VECTOR 0
1351 012436 000000 KMVLVL: 0 ;POINTER TO KMV11 INTRPT SERVICE
1352 012440 000000 KMVV04: 0 ;POINTER TO KMV11 INTRPT VECTOR 04
1353 012442 000000 KMVV02: 0 ; " " " " 02
1354 012444 000000 KMVV06: 0 ; " " " " 06
1355 012446 000000 KMTLVL: 0 ;POINTER TO KMV11 TX INTRPT SERVICE PS
1356 012450 000000 KMVCSR: 0 ;POINTER TO KMV11 CONTROL STATUS REGISTER
1357 012452 000000 KMVP02: 0 ;POINTER TO KMV11 PORT REGISTER SEL2
1358 012454 000000 KMVP04: 0 ;POINTER TO KMV11 PORT REGISTER - SEL4
1359 012456 000000 KMVP06: 0 ;POINTER TO KMV11 PORT REGISTER SEL6
1360
1361 012460 000000 KMVP10: 0 ;POINTER TO KMV11 PORT REG SEL10
1362 012462 000000 KMVP12: 0 ;POINTER TO PORT REG SEL 14
1363 012464 000000 KMVP14: 0 ;POINTER TO PORT REG SEL14
1364 012466 000000 KMVP16: 0 ;POINTER TO PORT REG 16
1365
1366 012470 000000 LOOP: 0 ;POINTER TO LOOP BACK CONNECTOR

```

1368
1369
1370 012472
1371
1372
1373 012472
1374 012672

***** PRIMARY REG ADRS STORAGE FOR THIS UNIT *****
;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
REGADR:

;***** STACK USED FOR SUBROUTINE LINKAGE *****
.BLKW 100
SSTACK:

1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388 012672
1389
1390
1391
1392
1393
1394
1401
1402
1403
1404
1405

```

.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 <KMV11A>

;
; FORMAT STATEMENTS USED IN PRINT CALLS
;

```

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
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463

```

.SBTTL GLOBAL SUBROUTINES

;MACRO'S NEEDED TO CALL SUBROUTINES
;-----

.MACRO CLRMAR
      ROMCLK
      004000
.ENDM CLRMAR

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

;ROUTINE TO WAIT FOR EVENT OR TIMEOUT

;CALLING SEQUENCE:      JSR PC,WAIT1
;                       JSR PC,WAIT2

;INPUTS PARAMETERS:    DELCT1,DELCT2

;                       INC DELCT1 UNTIL 0
;                       DEC DELCT2 UNTIL 0      DELCT2= NUMB OF WAIT1 PASSES

WAIT2: INC DELCT1
      BNE WAIT2

      BREAK

      DEC DELCT2
      BNE WAIT2

      RTS PC

```

```

012702 005237 002260
012706 001375
012710
012712 005337 002262
012716 001371
012720 000207

```

1464
1465
1466 012722 005237 002260
1467 012726 001375
1468
1469 012730 000207

WAIT1: INC DELCT1
BNE WAIT1
RTS PC

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

;MACRO TO WAIT A FEW MS

;CALLING SEQUENCE: WAITA X 0<X<177777
; WAITB X,Y 0<X OR Y<177777

.MACRO WAITA X
 MOV @X,DELCT1 ;LOAD COUNT
 JSR PC,WAIT1 ;WAIT
.ENDM

.MACRO WAITB X,Y
 MOV @X,DELCT1
 MOV @Y,DELCT2
 JSR PC,WAIT2
.ENDM


```

1546 ;ROUTINE TO CHECK REGISTER BSELO AND TO REPORT ERROR
1547
1548
1549
1550
1551
1552
1553 ;CALLING SEQUENCE: JSR PC,TSTERR
1554
1555
1556 ;OUTPUT PARAMETERS: RETURN TO PC IF TEST IS OK
1557 ; IF TIMEOUT DURING TEST
1558 ; PC.2 IF NO KMV11 ANSWER
1559 ; PC.4 IF DATA CMP ERROR
1560 ; PC.6
1561
1562
1563
1564
1565
1566
1567 013074 004537 013644 TSTERR: JSR R5,CBSELO ;LOOK IF BSELO=0
1568 013100 000000 .WORD 0 ;TEST IS OK ,RTS PC
1569 013102 000411 BR 1#
1570
1571
1572 013104 004537 013644 JSR R5,CBSELO ;LOOK IF BSELO=200
1573 013110 000200 .WORD 200 ;TIMEOUT DURING TEST,RTS PC.2
1574 013112 000406 BR 2#
1575
1576
1577 013114 004537 013644 JSR R5,CBSELO ;LOOK IF BSELO=100
1578 013120 000100 .WORD 100 ;DATA CMP ERROR,RTS PC.6
1579 013122 000405 BR 3#
1580
1581
1582
1583 013124 000407 BR 4# ;NO KMV11 ANSWER ,RTS PC.4
1584
1585
1586
1587 013126 000207 1#: RTS PC ;TEST OK
1588
1589
1590 013130 062716 000002 2#: ADD #2,(SP)
1591 013134 000207 RTS PC ;TIMEOUT ERROR
1592
1593
1594 013136 062716 000006 3#: ADD #4,(SP)
1595 013142 000207 RTS PC ;DATA CMP ERROR
1596
1597
1598 013144 062716 000004 4#: ADD #4,(SP)
1599 013150 000207 RTS PC ;NO KMV11 ANSWER
1600

```



```

1659
1660
1661 013152 042703 177770
1662 013156 004737 013452
1663 013162 006303
1664 013164 000173 013170
1665 013170 013210
1666 013172 013214
1667 013174 013222
1668 013176 013230
1669 013200 013236
1670 013202 013246
1671 013204 013304
1672 013206 013424
1673 013210 005000
1674 013212 000507
1675 013214 005000
1676 013216 005100
1677 013220 000504
1678 013222 012700 052525
1679 013226 000501
1680 013230 012700 125252
1681 013234 000476
1682 013236 000241
1683 013240 004737 013260
1684 013244 000472
1685 013246 000241
1686 013250 004737 013260
1687 013254 005100
1688 013256 000465
1689 013260 006037 013302
1690 013264 001003
1691 013266 012737 100000 013302
1692 013274 013700 013302
1693 013300 000207
1694 013302 000001
1695 013304 012737 000005 002336
1696 013312 004737 013324
1697 013316 013700 002342
1698 013322 000443
1699 013324 013702 002342
1700 013330 001002
1701 013332 013702 002334
1702 013336 032737 000777 002336
1703 013344 001003
1704 013346 012737 000001 002336
1705 013354 013703 002336
1706 013360 013702 002342
1707 013364 033702 002340
1708 013370 001405
1709 013372 005102
1710 013374 033702 002340
1711 013400 001401
1712 013402 000402
1713 013404 000241
1714 013406 000401
1715 013410 000261

;
; GENER: BIC #177770,R3
; JSR PC,SAVREG
; ASL R3
; JMP @GENSEL(R3)
; GENSEL: GEN0 ;ALL ZERO WORD
; GEN1 ;ALL ONE WORD
; GEN52 ;52 PATTERN
; GEN25 ;25 PATTERN
; GENR1 ;ROTATE '1' EACH CALL
; GENRO ;ROTATE '0' EACH CALL
; GENRAN ;RANDOM NUMBER
; GENINC ;INCREMENTING COUNT
; GENC: CLR RO ;0>RO
; BR GENEX
; GEN1: CLR RO ;NOT0>RO
; COM RO
; BR GENEX
; GEN52: MOV #52525,RO ;5252>RO
; BR GENEX
; GEN25: MOV #125252,RO ;125252>RO
; BR GENEX
; GENR1: CLC
; JSR PC,GENROT ;SHIFT 1 > RO
; BR GENEX
; GENRO: CLC
; JSR PC,GENROT ;
; COM RO ;SHIFT 0 > RO
; BR GENEX
; GENROT: ROR GENISH ;ROTATE 1 PATTERN
; BNE GENER1 ;= 0?
; MOV #100000,GENISH ;YES, SET MSB
; MOV GENISH,RO ;PUT 1 IN RO
; RTS PC ;AND EXIT
; GENISH: 1
; GENRAN: MOV #5,RANSEL ;SET SELECT VALUE TO 5
; JSR PC,RANGEN ;GENERATE RANDOM NUMBER IN RO
; MOV RANDN,RO ;
; BR GENEX ;
; RANGEN: MOV RANDN,R2 ;
; BNE RAN1 ;IS RANDOM = 0
; MOV RANST,R2 ;YES, PUT RANDOM START VALUE IN
; BIT #777,RANSEL ;NO; IS RANSEL SELECT VALUE = 0
; BNE RAN2 ;NO
; MOV #1,RANSEL ;YES: SET RANSEL = 1
; RAN2: MOV RANSEL,R3 ;
; MOV RANDN,R2 ;
; BIT RANMTA,R2 ;GET R2 <0 AND 1>
; BEQ RANCLC ;
; COM R2
; BIT RANMTA,R2
; BEQ RANCLC
; BR RANSEC
; RANCLC: CLC
; BR RAN4
; RANSEC: SEC

```

1716	013412	006037	002342		RAN4:	ROR	RANDN	;ROTATE C TO B15
1717	013416	005303				DEC	R3	;IS THIS NUMBER REQUIRED?
1718	013420	001357				BNE	RAN2+4	;NO, GET ANOTHER
1719	013422	000207			RANEX:	RTS	PC	;YES, EXIT
1720	013424	013700	002264		GENINC:	MOV	GOOD,RO	;INCREMENTS LOC. GOOD
1721	013430	005200				INC	RO	
1722	013432	010037	002264		GENEX:	MOV	RO,GOOD	
1723	013436	004737	013532			JSR	PC,RSTREG	
1724	013442	013737	002264	012372		MOV	GOOD,DATA	
1725	013450	000207				RTS	PC	
1726								

1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782 013452
1783 013460
1784 013466 012637 002244

```
.SBTTL SAVE REGISTERS  
:  
:  
DESCRIPTION:  
:  
: ROUTINE TO SAVE ALL THE GENERAL PURPOSE  
: REGISTERS ON THE STACK, AND LEAVE THE ADDRESS OF THE  
: CALLING ROUTINE ON THE STACK. THE ROUTINE WILL RUN AT  
: PRIORITY 6 TO AVOID MOST INTERRUPTS  
:  
CAUTION:REGISTER R0 IS NOT SAVED  
:  
:  
CALLING SEQUENCE:  
: JSR PC, SAVREG  
:  
INPUT PARAMETERS:  
: NONE  
:  
IMPLICIT INPUT PARAMETERS:  
: NONE  
:  
OUTPUT PARAMETERS:  
: REGISTERS 0 THRU 5 ARE SAVED ON THE STACK  
: AND THE RETURN ADDRESS OF THE CALLING ROUTINE IS  
: SET AS THE LAST ENTRY ON THE STACK  
:  
IMPLICIT OUTPUT PARAMETERS:  
: NONE  
:  
COMPLETION CODES:  
: NONE  
:  
POSSIBLE ERROR CODES:  
:  
NONE  
:  
SAVREG: GETPRI SAVSTA  
: SETPRI MAXPRI  
: MOV (SP), SAVPC ;SAVE PC FOR RETURN FROM THIS ROUTINE
```

1785	013472	012637	002344	MOV	(SP), SAVPC1	
1786	013476	010546		MOV	R5, -(SP)	
1787	013500	010446		MOV	R4, -(SP)	
1788	013502	010346		MOV	R3, (SP)	
1789	013504	010246		MOV	R2, (SP)	
1790	013506	010146		MOV	R1, (SP)	
1791	013510	010046		MOV	R0, (SP)	
1792	013512	013746	002344	MOV	SAVPC1, (SP)	
1793	013516	013746	002244	MOV	SAVPC, -(SP)	;PUT PC READY FOR
1794	013522			SETPRI	SAVSTA	
1795	013530	000207		RTS	PC	;RETURN
1796						
1797						
1798						

1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
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
1854
1855
1856

013532
013540
013546 012637 002244
013552 012637 002344
013556 012600
013560 012601

.SBTTL RESTORE REGISTERS
:
:
:
DESCRIPTION:
:
RESTORE TO RESTORE THE GENERAL PURPOSE
REGISTERS. THE STACK IS LEFT IN THE SAME STATE AS IT
WAS WHEN SAVREG WAS CALLED.
:
CAUTION: REGISTER R0 IS NOT SAVED
:
:
:
CALLING SEQUENCE:
:
JSR PC,RSTREG
:
:
INPUT PARAMETERS:
:
NONE
:
:
IMPLICIT INPUT PARAMETERS:
:
NONE
:
:
OUTPUT PARAMETERS:
:
R1 THRU R5 RESTORED
:
:
IMPLICIT OUTPUT PARAMETERS:
:
NONE
:
:
COMPLETION CODES:
:
NONE
:
:
POSSIBLE ERROR CODES:
:
NONE
:
:
RSTREG: GETPRI SAVSTA
SETPRI MAXPRI
MOV (SP),SAVPC
MOV (SP),SAVPC1
MOV (SP),R0
MOV (SP),R1

KMV11 A LINE CNT DIAGNOSTIC
RESTORE REGISTERS

MACRO M1200 09-APR 84 17:38 PAGE 38 1

SEQ 47

1857	013562	012602		MOV	(SP)+,R2	
1858	013564	012603		MOV	(SP)+,R3	
1859	013566	012604		MOV	(SP)+,R4	
1860	013570	012605		MOV	(SP)+,R5	
1861	013572	013746	002344	MOV	SAVPC1,-(SP)	
1862	013576	013746	002244	MOV	SAVPC,(SP)	;PUT PC READY FOR
1863	013602			SETPRI	SAVSTA	
1864	013610	000207		RTS	PC	


```

1866 ;CHECK CONTENT OF ONE OF THE 8 REGISTERS
1867
1868 ; CALLING SEQUENCE
1869 ; JSR R5,CKSELN ; N = REGISTER NUMBER
1870 ; .WORD A A=EXPECTED CONTENT OF REGISTER N
1871
1872 ;OUTPUT PARAMETER:
1873 ; BRANCH IN PC+2 IF ERROR DETECTED
1874 ; BRANCH IN PC IF NO ERROR DETECTED
1875
1876
1877
1878
1879
1880 013612 012537 002264 CKSELO: MOV (R5)+,GOOD ;WRITE GOOD
1881 013616 017737 176626 002310 MOV @KMVCSR,SELO ;READ SEL 0
1882 013624 023737 002310 002264 CMP SELO,GOOD ;CMP ?
1883 013632 001001 BNE 1$
1884 013634 000402 BR 2$
1885 013636 062705 000002 1$: ADD @2,R5
1886 013642 000205 2$: RTS R5
1887
1888
1889
1890
1891
1892
1893
1894 013644 005037 002264 CBSELO: CLR GOOD
1895 013650 012537 002264 MOV (R5)+,GOOD
1896 013654 117737 176570 012370 MOVB @KMVCSR,BSELO
1897 013662 123737 012370 002264 CMPB BSELO,GOOD
1898 013670 001001 BNE 1$
1899 013672 000402 BR 2$
1900 013674 062705 000002 1$: ADD @2,R5
1901 013700 000205 2$: RTS R5

```

1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959

;ROUTINE TO CHECK ALL REGISTER FROM SEL0 TO SEL16

;CALLING SEQUENCE:

:	JSR R5,CKALL		
:	.WORD A	A	= EXPECTED VALUE FOR SEL0
:	.WORD B	B	" " SEL2
:	.WORD C	C	" " SEL4
:	.WORD D	D	" " SEL6
:	.WORD E	E	" " SEL10
:	.WORD F	F	" " SEL12
:	.WORD G	G	" " SEL14
:	.WORD H	H	" " SEL16

;OUTPUT PARAMETER:

; BRANCH IN PC+2 IF ERROR
; BRANCH IN PC IF NO ERROR

```

CKALL:  MOV    (R5)+,GOOD0
        MOV    (R5)+,GOOD2
        MOV    (R5)+,GOOD4
        MOV    (R5)+,GOOD6
        MOV    (R5)+,GOOD10
        MOV    (R5)+,GOOD12
        MOV    (R5)+,GOOD14
        MOV    (R5)+,GOOD16

        MOV    @KMVCSR,SEL0      ;READ SEL0
        NOP
        MOV    @KMVP02,SEL2     ;READ SEL2
        NOP
        MOV    @KMVP04,SEL4     ;READ SEL4
        NOP
        MOV    @KMVP06,SEL6     ;READ SEL6
        NOP
        MOV    @KMVP10,SEL10    ;READ SEL10
        NOP
        MOV    @KMVP12,SEL12    ;READ SEL12
        NOP
        MOV    @KMVP14,SEL14    ;READ SEL14
        NOP
        MOV    @KMVP16,SEL16    ;READ SEL16

        CMP    SEL0,GOOD0
        BNE    1$
        CMP    SEL2,GOOD2
        BNE    1$
        CMP    SEL4,GOOD4
        BNE    1$
        CMP    SEL6,GOOD6
        BNE    1$
        CMP    SEL10,GOOD10
        BNE    1$
        CMP    SEL12,GOOD12

```

013702	012537	002266		
013706	012537	002272		
013712	012537	002274		
013716	012537	002276		
013722	012537	002300		
013726	012537	002302		
013732	012537	002304		
013736	012537	002306		
013742	017737	176502	002310	
013750	000240			
013752	017737	176474	002314	
013760	000240			
013762	017737	176466	002316	
013770	000240			
013772	017737	176460	002320	
C14000	000240			
014002	017737	176452	002322	
014010	000240			
014012	017737	176444	002324	
014020	000240			
014022	017737	176436	002326	
014030	000240			
014032	017737	176430	002330	
014040	023737	002310	002266	
014046	001035			
014050	023737	002314	002272	
014056	001031			
014060	023737	002316	002274	
014066	001025			
014070	023737	002320	002276	
014076	001021			
014100	023737	002322	002300	
014106	001015			
014110	023737	002324	002302	

1960	014116	001011			BNE	1\$
1961	014120	023737	002326	002304	CMP	SEL14,GOOD14
1962	014126	001005			BNE	1\$
1963	014130	023737	002330	002306	CMP	SEL16,GOOD16
1964	014136	001001			BNE	1\$
1965						
1966	014140	000402			BR	2\$
1967	014142	062705	000002		ADD	42,R5
1968	014146	000205			RTS	R5

```

1970                               ;ROUTINE TO CHECK SEL2 TO SEL16
1971
1972
1973
1974
1975
1976 014150 012537 002272          CKREG:  MOV      (R5)+,GOOD2
1977 014154 012537 002274          MOV      (R5)+,GOOD4
1978 014160 012537 002276          MOV      (R5)+,GOOD6
1979 014164 012537 002300          MOV      (R5)+,GOOD10
1980 014170 012537 002302          MOV      (R5)+,GOOD12
1981 014174 012537 002304          MOV      (R5)+,GOOD14
1982 014200 012537 002306          MOV      (R5)+,GOOD16
1983
1984
1985 014204 017737 176242 002314    MOV      @KMVP02,SEL2
1986 014212 000240                   NOP
1987 014214 017737 176234 002316    MOV      @KMVP04,SEL4
1988 014222 000240                   NOP
1989 014224 017737 176226 002320    MOV      @KMVP06,SEL6
1990 014232 000240                   NOP
1991 014234 017737 176220 002322    MOV      @KMVP10,SEL10
1992 014242 000240                   NOP
1993 014244 017737 176212 002324    MOV      @KMVP12,SEL12
1994 014252 000240                   NOP
1995 014254 017737 176204 002326    MOV      @KMVP14,SEL14
1996 014262 000240                   NOP
1997 014264 017737 176176 002330    MOV      @KMVP16,SEL16
1998
1999
2000
2001
2002 014272 023737 002314 002272    CMP      SEL2,GOOD2
2003 014300 001031                   BNE      1$
2004 014302 023737 002316 002274    CMP      SEL4,GOOD4
2005 014310 001025                   BNE      1$
2006 014312 023737 002320 002276    CMP      SEL6,GOOD6
2007 014320 001021                   BNE      1$
2008 014322 023737 002322 002300    CMP      SEL10,GOOD10
2009 014330 001015                   BNE      1$
2010 014332 023737 002324 002302    CMP      SEL12,GOOD12
2011 014340 001011                   BNE      1$
2012 014342 023737 002326 002304    CMP      SEL14,GOOD14
2013 014350 001005                   BNE      1$
2014 014352 023737 002330 002306    CMP      SEL16,GOOD16
2015 014360 001001                   BNE      1$
2016 014362 000402                   BR       1$
2017
2018 014364 062705 000002          1$:     ADD      @2,R5
2019 014370 000205                   2$:     RTS      R5

```

```

2021 ;ROUTINE TO CLEAR KMV11 MODULE
2022
2023
2024 ;CALLING SEQUENCE:
2025 ; JSR PC,CLRKMV
2026
2027 ;ROUTINE DESCRIPTION: CLEAR ALL CSR'S REGISTERS AND CHECK IF - 0
2028
2029
2030
2031 014372 005077 176052 CLRKMV: CLR @KMVCSR
2032 014376 012777 054000 176044 MOV #MAINTO,@KMVCSR
2033 014404 WAITA 0
2034
2035
2036
2037
2038 014416 012702 000010 MOV #10,R2
2039 014422 013701 012450 MOV KMVCSR,R1 ;LOAD ADDRESS
2040 014426 005021 1$: CLR (R1)+ ;CLEAR
2041 014430 005302 DEC R2 ;ALL DONE
2042 014432 001375 BNE 1$ ;NO
2043 014434 004537 013702 JSR R5,CKALL ;CHECK ALL REG - 0
2044 014440 000000 .WORD 0
2045 014442 000000 .WORD 0
2046 014444 000000 .WORD 0
2047 014446 000000 .WORD 0
2048 014450 000000 .WORD 0
2049 014452 000000 .WORD 0
2050 014454 000000 .WORD 0
2051 014456 000000 .WORD 0
2052 014460 000404 BR 2$ ;OK BRANCH AT END
2053 014462 ERRHRD 1,EM0002,PRALL ;CSR S REGISTERS CAN T BE CLEARED
2054 014472 000207 2$: RTS PC
2055

```

{ } ,

2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093

;ROUTINE TO SET MAINT MODE 1 AND CHECK DCT11 CLEAR SELO AFTER HAVING DECODED

;CALLING SEQUENCE:
; JSR PC,MAINM1

;GIVE AN ERROR IF MASTER CLEAR IS NOT CLEAR BY DCT11
;MAINT1= MASTER CLEAR=1 . MAINT 1 =0 . MODE = 1 : T11=HOLD

```

MAINM1: CLR      @KMVCSR
        NOP
        NOP
        NOP
        MOV      @MAINT1,@KMVCSR      ;LOAD ADDRESS
        MOV      #0,DELCT1
        MOV      #1,DELCT2
        JSR      PC,WAIT2
        JSR      R5,CKSELO           ;CHECK SELO=0 BUT MODE BIT =1
        .WORD    4000
        BR       1$                 ;OK BRANCH
        ERRHRD  2,EM0001,PRSELO
1$:     RTS      PC
    
```

```

2075 014474 005077 175750
2076 014500 000240
2077 014502 000240
2078 014504 000240
2079
2080 014506 012777 044000 175734
2081 014514 012737 000000 002260
2082 014522 012737 000001 002262
2083 014530 004737 012702
2084 014534 004537 013612
2085 014540 004000
2086 014542 000404
2087 014544
2088 014554 000207
    
```

2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113

014556 012537 012412
014562 053777 012412 175660
014570 012737 000000 002260
014576 004737 012722
014602 000205

;ROUTINE TO SET TEST NUMBER ON BSELO

;CALLING SEQUENCE:
; JSR R5,TSTNUB
; .WORD ,A

A=TEST MICRO PROGRAM NUMBER

TSTNUB: MOV (R5),NUB
BIS NUB,@KMVCSR
MOV #0000,DELCT1
JSR PC,WAIT1
RTS R5

;LOAD TEST NUMBER
;WAIT

2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171

```

;ROUTINE TO WRITE OR READ ONE OF THE KMV11 REGISTERS

;CALLING SEQUENCE:
;JSR   R5,WRITE
;.WORD A           A=ADDRESS TO WRITE
;.WORD B           B=DATA TO WRITE
;
;
;JSR   R5,READ
;.WORD A           A=ADDRESS TO READ
;
;MICRO DIAG NB 47 DESCRIPTION:
;WRITE: PUT ADDRESS, TO WRITE IN SEL2
;       PUT DATA TO WRITE IN SEL4
;       SET BIT 0 OF SEL6(WRITE BIT)
;       SET TEST NB 44
;       KMV11 CLEAR BSEL0 WHEN DONE
;
;READ:  PUT ADDRESS TO READ IN SEL2
;       CLEAR BIT 0 IN SEL6
;       SET TEST 47
;       KMV11 READ ADDRESS IN SEL2 AND CLEAR BSEL0 WHEN DONE

WRITE:  MOV   (R5),@KMVP02      ;WRITE ADDRESS
        MOV   (R5),@KMVP04      ;DATA
        MOV   @1,@KMVP06        ;BIT WRITE
        JSR   R5,TSTNIB        ;SEND TEST NB 44
        .WORD 47
        RTS   R5               ;RETURN

READ:   MOV   (R5),@KMVP02      ;SET ADDRESS TO READ
        CLR   @KMVP04
        CLR   @KMVP06
        JSR   R5,TSTNIB        ;SEND TEST NB 44
        .WORD 47
        JSR   PC,TSTERR        ;CHECK BSEL 0
        BR    1$              ;OK

```

```

014604 012577 175642
014610 012577 175640
014614 012777 000001 175634
014622 004537 014556
014626 000047
014630 000205
014632 012577 175614
014636 005077 175612
014642 005077 175610
014646 004537 014556
014652 000047
014654 004737 013074
014660 000412

```


KMV11 A LINE CNT DIAGNOSTIC
RESTORE REGISTERS

MACRO M1200 09 APR 84 17:38 PAGE 45 1

SEQ 56

2172	014662	000402				BR	28	
2173	014664	000401				BR	28	
2174	014666	000400				BR	28	
2175								
2176	014670				28:	ERRHRD	4,EM0004	;NO KMV ANSWER
2177	014700	004737	012732			JSR	PC,CHKMAX	
2178	014704	000205				RTS	R5	
2179								
2180	014706	017737	175542	012366	18:	MOV	@KMVP04,BAD	;READ DATA IN BAD
2181	014714	000205				RTS	R5	
2182								
2183								
2184								
2185								
2186								

2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209

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

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

```

2211          .SBTTL  GLOBAL ERROR REPORT SECTION
2212
2213          ;//////////
2214          ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
2215          ;/      THAT ARE USED IN MORE THAN ONE TEST.
2216          ;//////////
2217
2218          .NLIST BEX
2219
2220 014716      040      102      125  TIM:      .ASCIZ  / BUS TIMEOUT/
2221
2222 014733      045      116      045  TFM36:   .ASCIZ  /#N#AREGISTER ADDRESS ERROR,ADDRESS = #06#A,UNIT = #02/
2223
2224 015021      115      101      123  EM0001: .ASCIZ  /MASTER CLEAR FAIL TO RESET: DCT11 CAN'T CLEAR MASTER CLEAR /
2225
2226 015115      040      113      115  EM0002: .ASCIZ  / KMV11 REGISTERS CAN'T BE CLEARED /
2227
2228 015160      040      104      101  EM0003: .ASCIZ  / DATA COMPARE ERROR ON KMV11 REGISTER (SEL2 TO SEL16)/
2229
2230 015246      040      116      :17  EM0004:  .ASCIZ  / NO ANSWER FROM KMV11 /
2231
2232 015275      124      111      115  EM0006: .ASCIZ  /TIMEOUT DURING KMV11 MICRO TEST /
2233
2234 015336      111      116      124  EM0007: .ASCIZ  /INTERUPT OCCURED ON KMV11 AT INCORRECT VECTOR /
2235
2236 015416      113      115      126  EM0011: .ASCIZ  /KMV11 REAL TIME CLOCK FAILED TO INTERUPT /
2237
2238 015470      107      105      116  EM0012: .ASCIZ  /GENERATOR COUNT CAN'T BE READ OR WRITE CORRECTLY /
2239
2240 015552      107      105      116  EM0013: .ASCIZ  /GENERATOR OUTPUT ISN T IN A GOOD STATE(NO ACTION ON OUTPUT)/
2241
2242 015646      116      117      040  EM0033: .ASCIZ  /NO CHANGE IN BAUD RATE GENERATOR COUNT /
2243
2244 015716      116      117      040  EM0014: .ASCIZ  /NO ACTION ON BAUD RATE GENERATOR OUTPUT /
2245
2246 015767      105      122      122  EM0015: .ASCIZ  /ERROR WHEN TRANSMITTING IN INTERNAL LOOP WITHOUT INTERUPTS /
2247
2248 016063      105      122      122  EM0016: .ASCIZ  /ERROR WHEN TRANSMITTING FRAMES IN INTERNAL LOOPBACK MODE /
2249
2250 016156      105      122      122  EM0017: .ASCIZ  /ERROR WHEN TRANSMITTING FRAMES IN EXTERNAL LOOPBACK /
2251
2252 016243      105      122      122  EM0022: .ASCIZ  /ERROR DURING TRANSMISSION AND RECEPTION OF FRAMES /
2253
2254 016326      122      105      101  EM0023: .ASCIZ  /REAL TIME CLOCK INTERUPT OCCURED TOO EARLY /
2255
2256 016402      111      116      103  EM0024: .ASCIZ  /INCORRECT KMV11 REPLY /
2257
2258 016431      116      117      040  EM0027: .ASCIZ  /NO LOOP BACK CONNECTOR,TEST NOT FEXECUTED /
2259
2260 016503      105      122      122  EM0031: .ASCIZ  /ERROR WHEN TRANSMITTING IN INTERNAL LOOP WITHOUT INTERUPTS /
2261
2262 016577      115      117      104  EM0032: .ASCIZ  /MODEM SIGNAL ERROR ON CHANNEL IN EXTERNAL LOOPBACK
2263
2264 016663      040      120      122  EM0035: .ASCIZ  / PROM REVISION IS NOT COMPATIBLE WITH DIAGNUSTIC REVISION
2265
2266 016755      040      111      116  EM0036: .ASCIZ  / INTERUPT OCCURED ON DC'11 WHEN REAL TIME CLOCK IS DISABLE
2267

```

2269	017050	045	116	045	MSEL0: .ASCIZ /#N#A SEL0 = #06#A SHOULD BE = #06#N/
2270					
2271	017116	045	116	045	MREG0: .ASCIZ /#N#A SEL0 = #06#A SHOULD BE = #06/
2272	017162	045	116	045	MREG2: .ASCIZ /#N#A SEL2 = #06#A SHOULD BE = #06/
2273	017226	045	116	045	MREG4: .ASCIZ /#N#A SEL4 = #06#A SHOULD BE = #06/
2274	017272	045	116	045	MREG6: .ASCIZ /#N#A SEL6 = #06#A SHOULD BE = #06/
2275	017336	045	116	045	MREG10: .ASCIZ /#N#A SEL10 = #06#A SHOULD BE = #06/
2276	017402	045	116	045	MREG12: .ASCIZ /#N#A SEL12 = #06#A SHOULD BE = #06/
2277	017446	045	116	045	MREG14: .ASCIZ /#N#A SEL14 = #06#A SHOULD BE = #06/
2278	017512	045	116	045	MREG16: .ASCIZ /#N#A SEL16 = #06#A SHOULD BE = #06/
2279					
2280					
2281	017556	045	116	045	MINT: .ASCIZ /#N#A GOOD = #06#A BAD = #06/
2282					
2283	017612	045	116	045	MSEL0: .ASCIZ /#N#A BSEL0 = #06#A SHOULD BE = #06/
2284					
2285					
2286	017654	045	116	045	MVECT: .ASCIZ /#N#A RECEIVE BAD VECT = #06#A SHOULD BE = #06/
2287					
2288					
2289					
2290					
2291					
2292	017730	045	116	045	MT11V: .ASCIZ /#N#A RECEIVE VECTOR = #06#A SHOULD BE = #06/
2293	020006	045	116	045	MFRAM1: .ASCIZ /#N#A RECEIVE FRAME IS = #06#A SHOULD BE = #06/
2294	020064	045	116	045	MFRAM2: .ASCIZ /#N#A TRANSMIT SPEED IS = #06#A FRAME LENGTH = #06/
2295					
2296					
2297	020146	045	116	045	MSTER1: .ASCIZ /#N#A ERROR STATUS = #06/
2298	020200	045	116	045	MSTER2: .ASCIZ /#N#A WORD COUNT DISCREPANCY = #06/
2299					
2300	020243	045	116	045	MODEM1: .ASCIZ /#N#A TESTED MODEM SIGNAL IS = #06/
2301	020307	045	116	045	MODEM2: .ASCIZ /#N#A RESULT OF TEST IS = #06/
2302	020346	045	116	045	MODEM3: .ASCIZ /#N#A MODEM SIGNAL STATE IS = #06/
2303	020411	045	116	045	MODEM4: .ASCIZ /#N#A SEE TEST HEADER FOR SIGNAL DESCRIPTION /
2304					
2305	020466	045	116	045	MRAMEF: .ASCIZ /#N#A TXDATA = #06#A , RXDATA = #06/
2306					
2307	020537	045	116	045	MLOOP: .ASCIZ /#N#A NO LOOP BACK CONNECTOR, TEST NOT EXECUTED /
2308					.EVEN

```
2310 020616          BGNMSG  PRSELO          ;REPORT SELO
2311 020616          PRINTB  @MSELO,SEL 0,GOOD
2312 020646 004737 012732 JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
2313 020652          BREAK
2314 020654          ENDMSG
2315
2316
2317
2318 020656          BGNMSG  PRINT          ;
2319 020656          PRINTB  @MINT,GOOD,BAD
2320 020706 004737 012732 JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
2321 020712          BREAK
2322 020714          ENDMSG
2323
2324
2325 020716          BGNMSG  PRALL          ;REPORT CONTENT OF ALL CSR'S
2326 020716          PRINTB  @MREG0,SEL0,GOOD0
2327 020746          PRINTB  @MREG2,SEL2,GOOD2
2328 020776          PRINTB  @MREG4,SEL4,GOOD4
2329 021026          PRINTB  @MREG6,SEL6,GOOD6
2330 021056          PRINTB  @MREG10,SEL10,GOOD10
2331 021106          PRINTB  @MREG12,SEL12,GOOD12
2332 021136          PRINTB  @MREG14,SEL14,GOOD14
2333 021166          PRINTB  @MREG16,SEL16,GOOD16
2334 021216 004737 012732 JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
2335 021222          BREAK
2336 021224          ENDMSG
2337
2338
2339
2340
2341
2342
2343 021226          BGNMSG  PRREG          ;REPORT ALL CSR'S BUT SELO
2344 021226          PRINTB  @MREG2,SEL2,GOOD2
2345 021256          PRINTB  @MREG4,SEL4,GOOD4
2346 021306          PRINTB  @MREG6,SEL6,GOOD6
2347 021336          PRINTB  @MREG10,SEL10,GOOD10
2348 021366          PRINTB  @MREG12,SEL12,GOOD12
2349 021416          PRINTB  @MREG14,SEL14,GOOD14
2350 021446          PRINTB  @MREG16,SEL16,GOOD16
2351 021476 004737 012732 JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
2352 021502          BREAK
2353 021504          ENDMSG
2354
2355
2356
2357
2358 021506          BGNMSG  PADFLT          ;ADDRESS TEST
2359 021506          PRINTB  @TFM36,ADDR,UNIT
2360 021536 004737 012732 JSR      PC,CHKMAX
2361 021542          ENDMSG
2362
2363
2364
2365
2366
```

```

2367
2368
2369
2370 021544      BGNMSG  PBSELO      ;REPORT BSELO
2371 021544      PRINTB  #MBSELO,BSELO,GOOD
2372 021574 004737 012732 JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
2373 021600      BREAK
2374 021602      ENDMSG
2375
2376
2377
2378
2379
2380
2381
2382
2383 021604      BGNMSG  PVECT      ;REPORT VECTOR
2384 021604      PRINTB  #MVECT,VECT,GOOD
2385 021634 004737 012732 JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
2386 021640      BREAK
2387 021642      ENDMSG
2388
2389
2390
2391
2392 021644      BGNMSG  PRT11V     ;CHECK IF TOO MANY ERROR
2393 021644      PRINTB  #MT11V,VECT,GOOD
2394 021674 004737 012732 JSR      PC,CHKMAX
2395 021700      BREAK
2396 021702      ENDMSG
2397
2398
2399
2400
2401 021704      BGNMSG  PFRAME     ;REPORT FRAME ERROR
2402 021704      PRINTB  #MFRAM1,RXDATA,IXDATA
2403 021734      PRINTB  #MFRAM2,TSPEED,LENGTH
2404 021764 004737 012732 JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
2405 021770      BREAK
2406 021772      ENDMSG
2407
2408
2409
2410
2411
2412 021774      BGNMSG  PMODEM     ;REPORT MODEM SIGNAL ERROR
2413 021774      PRINTB  #MODEM1,GOOD
2414 022020      PRINTB  #MODEM2,BAD
2415 022044      PRINTB  #MODEM3,DATA
2416 022070      PRINTB  #MODEM4
2417 022110 004737 012732 JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
2418 022114      BREAK
2419 022116      ENDMSG
2420
2421
2422
2423

```

2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447

022120
022120
022150
022152

022154
022154
022200
022224
022230
022232

004737 012732

BGNMSG PRAMEF
PRINTB @MRAMEF, TXDATA, RXDATA
BREAK
ENDMSG

BGNMSG PRSTER
PRINTB @MSTER1, STAERR
PRINTB @MSTER2, WRDCNT
JSR PC, CHKMAX
BREAK
ENDMSG

;SHORT REPORT FOR FRAME ERROR

;REPORT ERROR STATUS ,WORD CNT

;CHECK IF TOO MANY ERROR

2449
2450
2451
2452
2453
2454
2455
2456
2457
2458 022234
2459
2465
2466 022234
2467
2474
2475 022240
2476
2477

.SBTTL REPORT CODING SECTION

; THE REPORT CODING SECTION CONTAINS THE
; 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
;

BGNRPT

EXIT RPT

ENDRPT


```

2479          .SBTTL  INITIALIZE SECTION
2480
2481          ;////////////////////////////////////
2482          ;// THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
2483          ;// AT THE BEGINNING OF EACH PASS.
2484          ;////////////////////////////////////
2485
2486 022242          BGNINIT
2487
2488
2523          .EVEN
2524
2525          .EVEN
2526
2527          .EVEN
2528
2529 022242          SETVEC  #140,#170000,#340          ;ODT ROM ADDRESS          ;JB REV A 0
2530
2531          ;INITIALIZE SUBROUTINE STACK
2532 022270 012705 012672          MOV      #SSTACK,R5
2533          ;STORE BASE LEVEL PROGRAM STACK POINTER
2534 022274 010637 002246          MOV      SP,PSTACK
2535 022300 005737 002250          TST      FTIME
2536 022304 001011          BNE      1$
2537 022306 013737 000004 002252          MOV      @#4,SAVE4
2538 022314 013737 000006 002254          MOV      @#6,SAVE6
2539 022322 012737 000001 002250          MOV      #1,FTIME
2540 022330 013737 002252 000004 1$: MOV      SAVE4,@#4
2541 022336 013737 002254 000006          MOV      SAVE6,@#6
2542
2543 022344          READEF  #EF.START          ;START COMMAND?
2544 022352          BCOMPLETE          SETUP          ;IF YES BRANCH
2545
2546 022354          READEF  #EF.CONTINUE          ;CONTINUE COMMAND?
2547 022362          BCOMPLETE          END
2548
2549 022364          READEF  #EF.NEW          ;NEW PASS?
2550 022372          BNCOMPLETE          NEXT          ;IF NOT EXIT SETUP
2551
2552 022374 012737 177777 012432 SETUP: MOV      # 1,UUT          ;INITIALISE UNIT NUMBER
2553
2554 022402 005237 012432          NEXT:  INC      UUT          ;POINT NEXT UNIT
2555 022406 023737 012432 002240          CMP      UUT,L$UIT          ;ALL DONE?
2556 022414 001521          BEQ      ABORT          ;IF YES END OF PASS
2557
2558 022416 013701 012432          MOV      UUT,R1
2559 022422          PRINTF  #RUNNING,R1          ;PRINT RUNNING MESSAGE
2560          .EVEN
2561
2562
2563 022444          GPHARD  UUT,R1          ;GET P TABLE
2564 022454          BNCOMPLETE          NEXT          ;IF NOT AVAILABLE GET NEXT
2565
2566
2567 022456          GETPRM:
2568
2569 022456 011137 012450          MOV      (R1),KMVCSR          ;GET ADDRESS OF KMV11

```

```

2570
2571 022462 011137 012452          MOV    (R1),KMVP02          ;GET POINTER TO KMV11 SEL02 REG
2572 022466 062737 000002 012452  ADD    #2,KMVP02
2573
2574 022474 011137 012454          MOV    (R1),KMVP04          ;GET POINTER TO KMV11 PORT REG  SEL 4
2575 022500 062737 000004 012454  ADD    #4,KMVP04
2576
2577 022506 011137 012456          MOV    (R1),KMVP06          ;GET POINTER TO KMV11 PORT REG - SEL 6
2578 022512 062737 000006 012456  ADD    #6,KMVP06
2579
2580 022520 011137 012460          MOV    (R1),KMVP10         ;GET POINTER TO KMV11 REG 10
2581 022524 062737 000010 012460  ADD    #10,KMVP10
2582
2583 022532 011137 012462          MOV    (R1),KMVP12         ;GET POINTER TO KMV11 REG 12
2584 022536 062737 000012 012462  ADD    #12,KMVP12
2585
2586 022544 011137 012464          MOV    (R1),KMVP14         ;GET POINTER TO KMV11 REG 14
2587 022550 062737 000014 012464  ADD    #14,KMVP14
2588
2589 022556 012137 012466          MOV    (R1)+,KMVP16        ;GET POINTER TO KMV11 REG 16
2590 022562 062737 000016 012466  ADD    #16,KMVP16
2591
2592 022570 011137 012434          MOV    (R1),KMVV00         ;GET POINTER TO VECTOR 0
2593
2594 022574 011137 012442          MOV    (R1),KMVV02         ;GET POINTER TO VECTOR 2
2595 022600 062737 000002 012442  ADD    #2,KMVV02
2596
2597 022606 011137 012440          MOV    (R1),KMVV04         ;GET POINTER TO VECTOR 4
2598 022612 062737 000004 012440  ADD    #4,KMVV04
2599
2600 022620 012137 012444          MOV    (R1)+,KMVV06        ;GET POINTER TO VECTOR 6
2601 022624 062737 000006 012444  ADD    #6,KMVV06
2602
2603 022632 012137 012436          MOV    (R1)+,KMVLVL        ;GET POINTER TO TX PRIORITY LEVEL
2604 022636 062737 000006 012446  ADD    #6,KMTLVL
2605
2606 022644 011137 012470          MOV    (R1),LOOP          ;GET LOOPBACK PARAMETERS:
2607
2608 022650 005037 002234          CLR    ERRCNT             ;CLEAR ERROR COUNT
2609 022654          EXIT    INIT
2610
2611
2612
2613 022660          ABORT: DOCLN              ;CLEAN UP AND ABORT PASS
2614 022662          EXIT INIT                ;EXIT
2615
2616
2617
2618
2619
2620 022666          045      116      045  RUNNING: .NLIST BEX          .ASCIZ /NNA RUNNING ON UNIT #D2#A
2621          .LIST BEX
2622          .EVEN
2623
2624
2625
2626

```

KMV11 A LINE CNT DIAGNOSTIC
INITIALIZE SECTION

MACRO M1200 09-APR 84 17:38 PAGE 51 2

B6

580 55

022'04
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

END: ENDINIT

2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675

022726

.SBTTL AUTODROP SECTION

;;
; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
; THE 'ADR' FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
; SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
; DROPPED FROM TESTING.

.EVEN
BGNAUTO

;DEVICE DOES NOT HAVE A "READY"
MOV KMVCSR,R1 ;R1 CONTAINS BASE KMV11 ADDRESS
MOV #7,R5 ;7 REGISTERS TO BE TESTED
MOV #2,R4 ;SET OUT TIMEOUT TRAP
; ;
MOV #340,6 ;LEVEL 7
MOV #300,6 ;LEVEL 6
1\$: TST (R1) ;REFERENCE DEVICE REGISTER
NOP
ADD #2,R1 ;NEXT REGISTER
DEC R5 ;DEC REGISTER COUNT
BNE 1\$;BR IF NOT LAST REGISTER
BR 3\$

;JB REV A-0
;JB REV A 0

2\$: ADD #4,SP
DODU LOGDEV

3\$: MOV SAVE4,4
MOV SAVF6,6
ENDAUTO

2677
2678
2679
2680
2681
2682
2683
2684 023020
2685
2686
2706
2707
2708
2709 023020
2710
2711 023022

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

f r)

- 2713
- 2714
- 2715
- 2716
- 2717
- 2718
- 2719
- 2720
- 2721
- 2722
- 2723
- 2724
- 2733
- 2734
- 2746
- 2747
- 2748
- 2749
- 2750
- 2751
- 2752
- 2753
- 2754
- 2755
- 2756
- 2757
- 2758
- 2759
- 2760
- 2761
- 2762
- 2763
- 2764
- 2765
- 2766

.SBTTL DROP UNIT SECTION

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

023024

BGNDU

023024

.EVEN
PRINTF @DROPD,RO ;UNIT DROPPED

023046

EXIT DU

023052

045

116

045

DROPD:

.NLIST BEX
.ASCIZ /N#A UNIT #D2#A DROPPED/
.LIST BEX
.EVEN

023102

ENDDU

2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2786
2787
2788
2789
2790
2791
2792
2793
2794

.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.
:/

023104
023104

BGNAU
ENDAU

2796
2797
2798
2799
2800
2801
2802
2803
2804
2811
2817
2818
2819
2825
2826
2827
2839
2840
2841
2842
2848

023106

```
.SBTTL  HARDWARE TESTS

;START OF CODE BLOCK WHICH IS USED AS DATA
ROMMAP:;+
; TEST TO ...
;

;      BGNTST

;      EXIT   TST

;      .EVEN
;      ENDTST
```


2850 023106

2851
2852
2853 023106

2854
2855 023106

2856 023106 013701 012450
2857 023112 012705 000007
2858 023116 012737 023154 000004

2859
2860 023124 012737 000300 000006

2861 023132 005711
2862 023134 000240

2863 023136
2864 023142 062701 000002

2865 023146 005305
2866 023150 001370

2867 023152 000413
2868

2869 023154 062706 000004 2\$:

2870 023160 010137 002354
2871 023164 013737 012432 012422

2872 023172
2873

2874 023202 013737 002252 000004 3\$:

2875 023210 013737 002254 000006
2876 023216

2877
2878 023222

2879
2880

```

BADHEAD
;***** TEST1 *****
; *VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
; *DOES NOT CAUSE A TIME OUT TRAP
BADHEAD
;***** TEST1 *****

BGNTST
MOV    KMVCSR,R1    ;R1 CONTAINS KMV11 ADDRESSES
MOV    #7,R5        ;7 REGISTERS TO BE TESTED
MOV    #2$,4        ;SET OUT TIMEOUT TRAP
;
MOV    #340,6       ;LEVEL 7
MOV    #300,6       ;LEVEL 6
1$:    TST    (R1)   ;REFERENCE DEVICE REGISTERS
      NOP
      ESCAPE TST
      ADD    #2,R1   ;NEXT REGISTER
      DEC    R5      ;DEC REGISTER COUNT
      BNE   1$      ;BR IF NOT LAST REGISTER
      BR    3$

2$:    ADD    #4,SP
      MOV    R1,ADDR
      MOV    UUT,UNIT
      ERRHRD 0,TIM,PADFLT ;TIME OUT ERROR

3$:    MOV    SAVE4,4
      MOV    SAVE6,6
      ESCAPE TST

ENDTST
.EVEN

```

;JB REV A-0
;JB REV A-0

2882 023224

BADHEAD
:***** TEST2 *****
:CHECK PROM REVISION TO SEE IF COMPATIBLE WITH DIAGNOSTIC
BADHEAD
:***** TEST2 *****

2883
2884 023224

2885
2886
2887
2888
2889 023224

STARS 1
:READ LOCATION 2 OF THE PROM (ADDRESS 160002) WHICH CONTAINS PROM VERSION
: NUMBER
:CHECK IF DIAGNOSTIC AND PROM ARE COMPATIBLE AND GIVE AN ERROR IF NOT
STARS 1

2890
2891
2892
2893 023224
2894
2895
2896
2897
2898

2899 023224

BGNTST
JSR PC,CLRKMV ;CLEAR ALL REGISTERS
JSR PC,MAINM1 ;SET MAINT MODE

2900 023224 004737 014372
2901 023230 004737 014474

2902
2903

2904 023234 004537 014632
2905 023240 160002

REVPRO: JSR R5,READ ;READ LOCATION 160002
.WORD 160002

2906
2907

2908 023242 023737 012424 012366
2909 023250 001410

CMP GDREV,BAD ;LOOK IF COMPATIBLE
BEQ 1\$;YES

2910
2911 023252

2912 023262 004737 012732
2913 023266

ERRHRD 7,EM0035 ;REPORT THE ERROR
JSR PC,CHKMAX ;CHECK IF TOO MANY ERROR
ESCAPE TST

2914 023272
2915 023272

1\$:
ENDTST

2917
2918
2919 023274

2920
2921 023274

2922
2923
2924
2925
2926
2927
2928 023274
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971 023274

```
BADHEAD
;***** TEST3 *****
;REAL TIME CLOCK TEST
BADHEAD
;***** TEST3 *****

STARS 1
;THIS TEST CHECK KMV11 REAL TIME CLOCK.
;THE DCT11 FULLY EXECUTE THIS MICRO TEST AND GIVE A RESULT VIA CSR'S
;TO THE HOST. (TIMING IN CHECKED BY DCT11)
;
;
;TEST DESCRIPTION:
;
;DCT11 ENABLE KMV11 CLOCK,AND THEN SET UP A 80 MS PERIODE CLOCK.
;
;DCT11 WAIT FOR AT LEAST 80 MS AND CHECK IF AN INTERUPT OCCUR
;ON DCT11 CHIP AT VECTOR 130
;
;
;DCT11 TURN OF CLOCK, WAIT AGAIN FOR MORE THAN 80 MS AND CHECK THAT
;NO INTERUPT OCCUR
;
;ERROR REPORTING:      BSELO=200      IF TIMEOUT DURING TEST
;                        BSELO=100      IF ERROR DURING TEST
;                        BSELO=TEST NUB  IF NO KMV11 ANSWER
;                        BSELO=0        IF TEST IS OK
;
;IF ERROR              SEL6=1          IF NO INTERUPT OCCUR
;                        SEL6=2          IF BAD VECTOR
;                        SEL6=4          IF INTERUPT OCCUR WHEN CLOCK
;                                      IS NOT ENABLE
;                        SEL6=10         INTERUPT OCCUR TOO EARLY
;
;
;                        SEL2=EXPECTED VECTOR
;
;
; MICRO TEST NB: 27
;
;
;CAUTION:              KMV11 CRISTAL FREQUENCY CAN'T BE CHECKED WITH THIS TEST;
;                        FOR THAT THE OPERATOR MUST SCOPE THE CRISTAL SIGNAL
;                        DIRECTLY ON THE MODULE ON IC 12 (13824 KHZ)
STARS 1
```

2973								
2974	023274				BGNTST			
2975	023274	004737	014372		JSR	PC,CLRKMV		;CLR REG
2976	023300	004737	014474		JSR	PC,MAINM1		;SET MAINT MODE
2977	023304	004537	014556		RTCLK: JSR	R5,TSTNU8		
2978	023310	000027			.WORD	27		
2979								
2980	023312				WAITB	0,2		;WAIT FOR TEST EXECUTION
2981								
2982								
2983	023332	004737	013074		JSR	PC,TSTERR		;CHECK BSELO
2984	023336	000522			BR	1\$;TEST OK
2985	023340	000423			BR	2\$;TIMEOUT ERROR
2986	023342	000432			BR	3\$;NO KMV ANSWER
2987								
2988								
2989	023344	022777	000001	167104	CMP	#1,@KMVP06		;ERROR DURING TEST ,SEE WHICH ONE
2990	023352	001436			BEQ	4\$;NO INTERRUPT OCCUR
2991								
2992	023354	022777	000002	167074	CMP	#2,@KMVP06		
2993	023362	001442			BEQ	5\$;INT ON BAD VECTOR
2994								
2995	023364	022737	000004	012456	CMP	#4,KMVP06		
2996	023372	001454			BEQ	6\$;INT OCCUR WHEN CLOCK IS DESABLE
2997								
2998								
2999								
3000	023374	022737	000010	012456	CMP	#10,KMVP06		;INTERUPT OCCUR TOO EARLY
3001	023402	001460			BEQ	7\$		
3002								
3003	023404	000137	023564		IMP	10\$;WRO'G KMV11 ANSWER
3004								
3005								
3006								
3007								
3008	023410				2\$: ERRHRD	8,EM0006		;TIMEOUT ERROR
3009	023420	004737	012732		JSR	PC,CHKMAX		;CHECK IF TOO MANY ERROR
3010	023424				ESCAPE	TST		
3011								
3012								
3013								
3014	023430				3\$: ERRHRD	9,EM0004		;NO KMV11 ANSWER
3015	023440	004737	012732		JSR	PC,CHKMAX		;CHECK IF TOO MANY ERROR
3016	023444				ESCAPE	TST		
3017								
3018								
3019	023450				4\$: ERRHRD	10,EM0011		;NO INTERRUPT OCCUR
3020	023460	004737	012732		JSR	PC,CHKMAX		;CHECK IF TOO MANY ERROR
3021	023464				ESCAPE	TST		
3022								
3023								
3024								
3025	023470	017737	166760	012374	5\$: MOV	@KMVP04,VECT		;READ BAD VECT
3026	023476	012737	000130	002264	MOV	#130,GOOD		
3027	023504				ERRHRD	11,EM0007		;INTERUPT OCCUR AT A BAD VECTOR
3028	023514	004737	012732		JSR	PC,CHKMAX		;CHECK IF TOO MANY ERROR
3029	023520				ESCAPE	TST		

10

```
3030
3031
3032 023524          6$:  ERRHRD  12,EM0036      ;INT OCCUR WHEN CHIP IS DESABLE
3033 023534 004737 012732  JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
3034 023540          ESCAPE TST
3035
3036
3037
3038
3039 023544          7$:  ERRHRD  13,EM0023      ;INTERUPT OCCUR TOO EARLY ON KMV11
3040 023554 004737 012732  JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
3041 023560          ESCAPE TST
3042
3043
3044
3045
3046
3047
3048 023564          10$: ERRHRD  14,EM0024      ;INCORRECT KMV11 RESULT
3049 023574 004737 012732  JSR      PC,CHKMAX      ;CHECK IF TOO MANY ERROR
3050 023600          ESCAPE TST
3051
3052
3053
3054
3055 023604 000240          1$:  NOP
3056 023606          ENDTST
3057
3058
3059
```

3061 023610
3062
3063 023610
3064
3065
3066
3067
3068
3069
3070
3071
3072 023610
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115

```
BADHEAD
:***** TEST4 *****
:BAUD RATE GENERATOR TEST
BADHEAD
:***** TEST4 *****

STARS 1
:THIS TEST READ THE STATUS AND THE OUTPUT OF THE BAUD RATE GENERATOR
:DURING EACH PHASE OF THE CLOCK PULSE.
:NOTE:THIS TEST AND ALL THE VERIFICATIONS ARE MADE BY THE DCT11 WHICH
:ONLY GIVE TEST RESULT VIA CSR'S TO THE HOST(TIMING IS CHECKED BY DCT11)
:
:
:TEST DESCRIPTION:
:   DCT11 LOAD GENERATOR COUNT WITH MAX COUNT (-+4.74 MSEC)
:
:   READ BACK GENERATOR COUN ,STATUS AND VALIDATE REPOSE.
:
:
:   STEP 1:READ COUNT AFTER STARTING CLOCK
:           CLOCK COUNT MUST BE NEGATIVE
:           OUTPUT MUST BE = 1
:
:
:           ERROR REPORTING:
:           IF COUNT=POSITIVE           BSELO=100=ERROR
:                                       SEL6 =1  *GENE COUNT CAN T BE READ OR
:                                       WRITEN CORRECTLY
:
:           IF OUTPUT=0                 BSELO=100=ERROR
:                                       SEL6 =2  *GENE OUTPUT IS NOT CORRECT
:
:   STEP 2: WAIT 2,5MSEC AND READ BACK AGAIN GENERATOR COUNT AND STATUS
:           OUTPUT MUST BE = 0
:
:
:           ERROR REPORTING:
:           IF OUTPUT =1                 BSELO=100= ERROR
:                                       SEL6=10  *OUTPUT ISN T IN A GOOD STATE
:
:   STEP3:WAIT 2.5 MSEC MORE AND READ BACK AGAIN GENERATOR COUNT AND STATUS
:           OUTPUT MUST BE = 1
```

3116
3117
3118
3119
3120
3121
3122
3123
3124
3125
3126 023610

```

:      ERROR REPORTING:
:
:      IF OUTPUT=0
:
:      -ELSE EXIT
:
:
:
: TEST 30=      TEST GENERATOR A
STARS 1
```

BSELO=100=ERROR
SEL6=40 =NO ACTION ON GENERATOR OUTPUT

```

3128 023610          BGNTST
3129 023610 004737 014372      JSR    PC,CLRKMV      ;CLR REG
3130 023614 004737 014474      JSR    PC,MAINM1     ;SET MAINT MODE
3131 023620 004537 014556      JSR    R5,TSTNU8
3132 023624 000030          .WORD  30
3133
3134
3135 023626          BDRGEN: WAITB  0.1      ;WAIT FOR TEST EXECUTION
3136
3137 023646 004737 013074      JSR    PC,TSTERR     ;CHECK BSELO TO SEE IF ERROR
3138 023652 000137 024052      JMP    BDROK0        ;TEST OK BR AT END
3139 023656 000402          BR     2$            ;TIME OUT ERROR
3140 023660 000401          BR     2$            ;NO KMV11 ANSWER
3141 023662 000410          BR     3$            ;ERROR DURING TEST
3142
3143
3144
3145 023664          2$:  ERRHRD  15,EM0004      ;NO KMV11 ANSWER
3146 023674 004737 012732      JSR    PC,CHKMAX     ;CHECK IF TOO MANY ERROR
3147 023700          ESCAPE TST
3148
3149
3150
3151 023704          3$:
3152 023704 017737 166546 002320  MOV    @KMVPO6,SEL6  ;LOOK WHICH ERROR
3153 023712 022737 000001 002320  CMP    #1,SEL6       ;READ SEL6
3154 023720 001010          BNE    4$            ;LOOK IF ERROR 1
3155
3156 023722          ;GENE COUNT CAN'T BE READ OR WRITE CORRECTLY
3157 023732 004737 012732      ERRHRD  16,EM0012   ;CHECK IF TOO MANY ERROR
3158 023736          JSR    PC,CHKMAX
3159          ESCAPE TST
3160
3161 023742 022737 000002 002320  4$:  CMP    #2,SEL6       ;LOOK IF ERROR 2
3162 023750 001010          BNE    5$            ;NO
3163
3164
3165
3166 023752          ;GENE OUTPUT ISN'T IN A GOOD STATE
3167 023762 004737 012732      ERRHRD  17,EM0013   ;CHECK IF TOO MANY ERROR
3168 023766          JSR    PC,CHKMAX
3169          ESCAPE TST
3170
3171
3172 023772 022737 000010 002320  5$:  CMP    #10,SEL6      ;ERROR 10?
3173 024000 001414          BEQ    GENOUT
3174 024002 022737 000040 002320  CMP    #40,SEL6
3175 024010 001410          BEQ    GENOUT
3176
3177 024012          ;WRONG KMV11 ANSWER
3178 024022 004737 012732      ERRHRD  18,EM0024   ;CHECK IF TOO MANY ERROR
3179 024026          JSR    PC,CHKMAX
3180          ESCAPE TST
3181
3182
3183
3184

```


C/

KM11 A LINE CNT DIAGNOSTIC
HARDWARE TESTS

MACRO M1200 09 APR 84 17:38 PAGE 62 1

SEQ 50

3185
3186 024032
3187 024042 004737 01.7'52
3188 024046
3189
3190
3191
3192 024052
3193 024052

GENOUT: ERRMND 19,EM0014
JSR PC,CH*MAX
ESCAPE TST

;NO ACTION ON GENERATOR OUTPUT
;CHECK IF TOO MANY ERROR

BDROKO:
ENDTST

3195
3196 024054

3197
3198
3199 024054

3200
3201
3202
3203
3204
3205
3206
3207 024054

3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249

BADHEAD
:***** TESTS *****
:TRANSMIT DIFFERENT FRAMES (OF 500 WORDS) AT 1.2 KBAUDS SPEED IN
:INTERNAL MODE WITHOUT ANY INTERRUPT ON CHANNEL A .
BADHEAD
:***** TESTS *****

STARS 1
:QBUS WRITE DIFFERENT TX TABLE OF 500 WORDS, LOAD IN KMV11 CSR'S
:THE TX AND RX TABLE ADDRESS ,THE TABLE LENGTH AND TRANSFER SPEED
:
:
:OCT11 EXECUTE THE TRANSFER IN INTERNAL MODE ON CHA AND WRITE BACK
:IN RX TABLE (TRANSFER FROM QBUS TO KMV11 -DMA)
:QBUS CHECK BSEL0 TO SEE THE STATUS OF THE TEST AND IF TEST DONE CHECK IF
:RX TABLE =TX TABLE

:
:PARAMETERS SELECTION:
: SEL2= TX TABLE ADDRESS
: SEL4= TX TABLE LENGTH
: BSEL6= EXTENDED ADDRESS OF TX TABLE
: BSEL7= " " RX
: SEL12= RX TABLE ADDRESS
: SEL14= SPEED SELECTION
: BSEL16= ERROR STATUS
: SEL10= RECEIVED BYTE COUNT DIFFERENCE BETWEEN RX AND TX TABLE
: >0 IF TX>RX
: <0 IF TX<RX
: BSEL0= TEST STATUS

:
:TEST STATUS DESCRIPTION:
: BSEL0= 0 *TEST DONE CHECK RX TABLE
: BSEL0= 200 *TIMEOUT ERROR
: BSEL0= TSTNB *NO KMV11 ANSWER
: BSEL0= 100 *ERROR DURING TEST ,IN THAT CASE SEE WHICH KIND OF
: ERROR BY TESTING BSEL16.

:
:ERROR STATUS DESCRIPTION:
:
: WHEN BSEL0=100,GIVE STATUS AND WORD COUNT DISCREPANCY
:
: BSEL16= BIT14=1 *FCS ERROR
: BSEL16= BIT13=1 *OVERFLOV ERROR

3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264 024054

```

; BSEL16= BIT8 =1 =ILLEGAL INTERRUPT ERROR
; BSEL16= BIT7 =1 =RX ABORT ERROR
; BSEL16= BIT6 =1 =UNDERRUN ERROR
; BSEL16= BIT5 =1 =WORD COUNT DISCREPANCY
; BSEL16= BIT4 =1 =DMA IN TIMEOUT ERROR
; BSEL16= BIT3 =1 =DMA OUT TIMEOUT ERROR
; BSEL16= BIT2 =1 =CLOCK PROBLEM (NO BUFFER EMPTY)
; BSEL16= BIT1 =1 =DATA COMPARE ERROR BETWEEN TX AND RX TABLE (USED
; ONLY DURING SELF TEST)
;
; MICRO DIAG TEST DESCRIPTION:
; TEST 36 =TRANSMIT FRAMES AT 1.2KB SPEED ON CHANNEL A WITHOUT INTERRUPT
;
; STARS 1
```

```

3266 024054          BGNTST
3267 024054 004737 014372      JSR    PC,CLRKMV      ;CLR REG
3268 024060 005037 012400      CLR    CHANEL
3269 024064 005037 002256      CLR    FLAG
3270 024070 004737 014474      JSR    PC,MAINM1     ;SET MAINT MODE
3271 024074 012737 000500 012410  MOV    #500,LENGTH  ;SELECT LENGTH
3272
3273 024102 012737 013224 012406  MOV    #KB1.2,TSPEED ;SELECT SPEED
3274
3275 024110 012703 000001      INTTX: MOV    #1,R3   ;SELECT A PATTERN
3276
3277
3278 024114 005203      TXSTAR: INC    R3     ;NEW ONE
3279 024116          BREAK
3280 024120 013704 012410      MOV    LENGTH,R4    ;LOAD LENGTH
3281 024124 012702 002362      MOV    #TABLE,R2   ;TX TABLE ADDRESS
3282 024130 004737 013152      10$:  JSR    PC,GENER  ;WRITE TX TABLE
3283 024134 013722 012372      MOV    DATA,(R2)+
3284 024140 005304          DEC    R4           ;ALL DONE?
3285 024142 001372      BNE    10$
3286
3287
3288
3289 024144 013704 012410      MOV    LENGTH,R4
3290 024150 012702 006362      MOV    #TABLE,R2   ;CLEAR RX TABLE
3291 024154 005022      11$:  CLR    (R2)+
3292 024156 005304          DEC    R4
3293 024160 001375      BNE    11$
3294
3295
3296
3297
3298 024162 013777 012406 166274  MOV    TSPEED,@KMVP14 ;SEND TX SPEED
3299 024170 012777 002362 166254  MOV    #TABLE,@KMVP02 ;SEND TX TABLE ADDRESS
3300 024176 013777 012410 166250  MOV    LENGTH,@KMVP04 ;LOAD TX TABLE ADDRESS
3301 024204 012777 006362 166250  MOV    #TABLE,@KMVP12 ;LOAD RX TABLE ADDRESS
3302 024212 005077 166240      CLR    @KMVP06
3303
3304
3305
3306
3307
3308 024216 004537 014556      JSR    #5,TSTNUB
3309 024222 000036          .WORD 36           ;DO TEST 36- CHA TEST
3310
3311
3312
3313 024224          WAITB 0.20      ;WAIT FOR TEST EXECUTION
3314
3315
3316 024244 004737 013074      JSR    PC,TSTERR   ;CHECK BSELO
3317
3318 024250 000427          BR    6$           ;TEST OK CHECK RX TABLE
3319 024252 000402          BR    3$           ;TIMEOUT ERROR
3320 024254 000401          BR    3$           ;NO KMV11 ANSWER
3321 024256 000410          BR    4$           ;CHECK SEL16 TO SEE WHICH ONE
3322

```

```

3323
3324 024260          3$:  ERRHRD  25,EM0004          ;NO KVM11 ANSWER
3325 024270 004737 012732      JSR    PC,CHKMAX          ;CHECK IF TOO MANY ERROR
3326 024274          ESCAPE  TST
3327
3328
3329
3330 024300          4$:                          ;ERROR DURING TEST READ ERROR STATUS
3331                          ;TO CHECK WHICH ONE
3332
3333 024300 017737 166162 012416      MOV    @KMVP16,STAERR          ;READ ERROR STATUS
3334
3335 024306 017737 166146 012420      MOV    @KMVP10,WRDCNT          ;READ WORD COUNT DISCREPANCY
3336
3337 024314          ERRHRD  26,EM0031,PRSTER          ;ERROR WHILE TX,RX FRAMES,GIVE ERROR
3338                          ;GIVE ERROR STATUS,WORD CNT DISCREPANCY
3339 024324          ESCAPE  TST
3340
3341
3342
3343
3344
3345
3346
3347
3348 024330 012702 002362          6$:  MOV    @TTABLE,R2          ;LOAD TXTABLE ADDRESS
3349 024334 012705 006362          MOV    @RTABLE,R5          ;      RXTABLE ADDRESS
3350 024340 013704 012410          MOV    LENGTH,R4          ;TABLE LENGTH
3351
3352 024344 022225          RXCK:  CMP    (R2)+,(R5)+          ;CHECK RX AND TX TABLE
3353 024346 001007          BNE    RXERR
3354 024350 005374          DEC    R4          ;ALL CHECK?
3355 024352 001374          BNE    RXCK          ;NO BRANCH
3356
3357
3358
3359 024354 022703 000005          CMP    @5,R3          ;ALL KIND OF PATTERN DONE?
3360 024360 001255          BNE    TXSTAR          ;NO TRY WITH NEW ONE
3361
3362 024362 000137 024504          JMP    RXEND
3363
3364 024366 162705 000002          RXERR: SUB    @2,R5
3365 024372 162702 000002          SUB    @2,R2
3366
3367 024376 011237 012402          MOV    (R2),TXDATA
3368 024402 011537 012404          MOV    (R5),RXDATA
3369
3370 024406 005737 002256          TST    FLAG
3371 024412 001014          BNE    7$          ;LOOK IF 1ST ERROR
3372
3373 024414          ERRHRD  27,EM0015,PFRAME          ;DATA CMP ERROR
3374 024424 005237 002256          INC    FLAG
3375 024430 062702 000002          ADD    @2,R2          ;POINT NEXT ADDRESS
3376 024434 062705 000002          ADD    @2,R5
3377 024440 000137 024344          JMP    RXCK
3378
3379 024444          7$:  ERRHRD  27,0,PRAMEF          ;SHORT REPORT

```

```

3380 024454 005237 002256      INC      FLAG
3381 024460 062702 000002      ADD      #2,R2
3382 024464 062705 000002      ADD      #2,R5
3383 024470 022737 000010 002256  CMP      #10,FLAG
3384 024476 001322                BNE      RXCK
3385
3386 024500                ESCAPE  TST
3387
3388
3389 024504                RXEND:
3390
3391
3392
3393 024504                ENDTST

```

;POINT NEXT ADDRESS
;LOOK IF 10 REPORT

3395
3396
3397
3398 024506

3399
3400
3401 024506

3402
3403
3404
3405
3406
3407
3408
3409 024506
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449

```

BADHEAD
:***** TEST6 *****
:TRANSMIT DIFFERENT FRAME OF VARIOUS LENGTH (FROM 2BYTES TO 2K BYTES)
:AT 72 KBAUDS IN INTERNAL MODE ON CHANNEL A (TRANSMISSION WITH INTERRUPT)
BADHEAD
:***** TEST6 *****

STARS 1
:QBUS WRITE DIFFERENT TX TABLE OF VARIOUS LENGTH, LOAD IN KMV11 CSR'S
:THE TX AND RX TABLE ADDRESS ,THE TABLE LENGTH AND TRANSFER SPEED
:
:
:
:DCT11 EXECUTE THE TRANSFER IN INTERNAL MODE ON CHA AND WRITTE BACK
:IN RX TABLE
:QBUS CHECK BSEL0 TO SEE THE STATUS OF THE TEST AND IF TEST DONE CHECK IF
:RX TABLE =TX TABLE
:SPEED=72 KBAUDS
:
:
:PARAMETERS SELECTION:
:   SEL2= TX TABLE ADDRESS
:   SEL4= TX TABLE LENGTH
:   BSEL6= EXTENDED ADDRESS OF TX TABLE
:   BSEL7= " " RX
:   SEL12= RX TABLE ADDRESS
:   SEL14= SPEED SELECTION ( = 141 IF 72KBAUDS)
:   BSEL16= ERROR STATUS
:   BSEL0= TEST STATUS
:   SEL10= BYTE COUNT DESCREPANCY >0 IF TX>RX
:                                     <0 IF TX<RX
:
:
:TEST STATUS DESCRIPTION:
:   BSEL0= 0 =TEST DONE CHECK RX TABLE
:   BSEL0= 200 =TIMEOUT ERROR
:   BSEL0= 151NB =NO KMV11 ANSWER
:   BSEL0= 100 =ERROR DURING TEST ,LOOK WHICH ONE BY TESTING BSEL16
:
:
:ERROR STATUS DESCRIPTION:
:
:   WHEN BSEL0=100,GIVE CONTAINT OF ERROR STATUS AND WORD COUNT DISCREPANCY
:
:
:   BSEL16= BIT14=1 =PCS ERROR
:   BSEL16= BIT13=1 =OVERFLOW ERRJR

```

```
3450      ; BSEL16= BIT8 =1 =ILLEGAL INTERRUPT ERROR
3451      ; BSEL16= BIT7 =1 =RX ABORT ERROR
3452      ; BSEL16= BIT6 =1 =UNDERRUN ERROR
3453      ; BSEL16= BIT5 =1 =BYTE COUNT DISCREPANCY
3454      ; BSEL16= BIT4 =1 =DMA IN TIMEOUT ERROR
3455      ; BSEL16= BIT3 =1 =DMA OUT TIMEOUT ERROR
3456      ; BSEL16= BIT2 =1 =CLOCK PROBLEM (NO BUFFER EMPTY)
3457      ; BSEL16= BIT1 =1 =DATA COMPARE ERROR BETWEEN TX AND RX TABLE (USE
3458      ;                                     ONLY DURING SELF TEST)
3459      ;
3460      ;
3461      ; MICRO DIAG TEST DESCRIPTION:
3462      ; TEST 40      = TRANSMIT VARIOUS LENGTH FRAME AT 72 KBAUDS ON CHANNEL A
3463      ;
3464      ;
3465 024506 STARS 1
```



```

3467 024506          BGNST
3468 024506 004737 014372      JSR      PC,CLRKMV          ;CLR REG
3469 024512 005037 012400      CLR      CHANEL
3470 024516 004737 014474      JSR      PC,MAINM1        ;SET MAINT MODE
3471 024522 005037 002256      CLR      FLAG
3472
3473
3474 024526 012703 000005      MOV      #5,R3            ;SELECT RANDOM PATTERN
3475          ; THE FOLLOWING RATE WAS CHANGED FROM 72KB TO 64KB, AS 72KB CAUSED
3476          ; INTERMITTENT FAILURES OF THIS TEST. THIS SHOULD BE CHANGED TO
3477          ; 72KB IN A FUTURE RELEASE. IT IS RECOMMENDED THAT THIS TEST BE RUN
3478          ; AT 72KB (BY USING ODT).
3479          ;
3480 024532 012737 000154 012406      MOV      #KB72,TSPEED    ;SELECT SPEED ;JB REV A 0
3481          MOV      #KB64,TSPEED    ;SELECT SPEED ;JB REV A 0
3482 024540 012737 000001 012410      TXLTAR: MOV      #1,LENGTH ;START WITH 2 CHARACTERS
3483
3484 024546 013704 012410      TXLBGN: MOV      LENGTH,R4
3485 024552 012702 002362      MOV      #RTABLE,R2
3486 024556 004737 013152      10$:   JSR      PC,GENER      ;WRITE TX TABLE
3487 024562 013722 012372      MOV      DATA,(R2)+
3488 024566 005304              DEC      R4
3489 024570 001372              BNE      10$
3490
3491 024572              BREAK
3492
3493 024574 013704 012410      MOV      LENGTH,R4          ;CLEAR RX TABLE
3494 024600 012702 006362      MOV      #RTABLE,R2
3495 024604 005022              20$:   CLR      (R2)+
3496 024606 005304              DEC      R4
3497 024610 001375              BNE      20$
3498
3499
3500
3501
3502
3503
3504 024612 013777 012406 165644      MOV      TSPEED,@KMVP14    ;SEND TX SPEED
3505 024620 012777 002362 165624      MOV      #RTABLE,@KMVP06  ; ' TX TABLE ADDRESS
3506 024626 013777 012410 165620      MOV      LENGTH,@KMVP04   ; ' ' ' LENGTH
3507 024634 012777 006362 165620      MOV      #RTABLE,@KMVP12  ;SEND RX TABLE ADDRESS
3508 024642 005077 165610      CLR      @KMVP06          ;CLR EXTENDED ADDRESS
3509
3510
3511
3512 024646 004537 014556      JSR      R5,ISTNUB
3513 024652 000040              .WORD   40                ;DO TEST 40= CHA TEST
3514
3515 024654              WAITB   0.2                ;WAIT FOR TEST EXECUTION
3516
3517
3518 024674 004737 013074      JSR      PC,ISTERR        ;CHECK BSELO
3519
3520 024700 000427              BR      6$                ;TEST OK CHECK RX TABLE
3521 024702 000402              BR      3$                ;TIMEOUT ERROR
3522 024704 000401              BR      3$                ;NO KMV11 ANSWER
3523 024706 000410              BR      4$                ;CHECK SEL 16 TO SEE WHICH ONE

```

```

3524
3525
3526 024710          3$:  ERRHRD  28,EM0004          ;NO KMV11 ANSWER
3527 024720 004737 012732  JSR      PC,CHKMAX          ;CHECK IF TOO MANY ERROR
3528 024724          ESCAPE  TST
3529
3530 024730          4$:
3531
3532
3533 024730 017737 165532 012416  MOV      @KMVP16,STAERR          ;READ ERROR STATUS
3534
3535 024736 017737 165516 012420  MOV      @KMVP10,WRDCNT          ;READ WORD COUNT DISCREPANCY
3536
3537 024744          ERRHRD  29,EM0022,PRSTER          ;ERROR WHILE TX,RX FRAMES,GIVE ERROR
3538
3539 024754          ESCAPE  TST          ;GIVE ERROR STATUS,WORD CNT DISCREPANCY
3540
3541
3542
3543
3544 024760 012702 002362          6$:  MOV      @TABLE,R2          ;LOAD TX TABLE ADDRESS
3545 024764 012705 006362          MOV      @RTABLE,R5          ; '   RX   '
3546 024770 013704 012410          MOV      LENGTH,R4          ; "   TX TABLE LENGTH
3547
3548
3549 024774 022522          RXLCK:  CMP      (R5), (R2)          ;CMP TX AND RX TABLE
3550 024776 001015          BNE      RXLERR          ;BR IF ERROR
3551 025000 005304          DEC      R4          ;ALL DONE
3552 025002 001374          BNE      RXLCK          ;NO
3553
3554 025004 062737 000400 012410  ADD      @400,LENGTH          ;CHANGE LENGTH
3555 025012 022737 002000 012410  CMP      @2000,LENGTH          ;IS IT MAX?
3556 025020 100252          BPL      TXLBGN          ;NO DO TEST AGAIN WITH NEW TABLE
3557
3558
3559 025022 005303          DEC      R3          ;SELECT OTHER PATERNS
3560 025024 001245          BNE      TXLTAR
3561
3562 025026 000137 025150          JMP      RXLEND
3563
3564
3565
3566 025032 162705 000002          RXLERR:  SUB      @2,R5
3567 025036 162702 000002          SUB      @2,R2
3568
3569 025042 011237 012402          MOV      (R2),TXDATA
3570 025046 011537 012404          MOV      (R5),RXDATA
3571
3572 025052 005737 002256          TST      FLAG          ;LOOK IF 1ST ERROR
3573 025056 001014          BNE      30$
3574
3575 025060          ERRHRD  30,EM0016,PFRAME          ;DATA CMP ERROR
3576 025070 005237 002256          INC      FLAG
3577 025074 062702 000002          ADD      @2,R2          ;POINT NEXT ADDRESS
3578 025100 062705 000002          ADD      @2,R5
3579 025104 000137 024344          JMP      RXCK
3580

```

```

3581 025110          30$:  ERRHRD  30,0,PRAMEF          ;SHORT REPORT
3582 025120 005237 002256      INC    FLAG
3583 025124 062702 000002      ADD    #2,R2
3584 025130 062705 000002      ADD    #2,R5          ;POINT NEXT ADDRESS
3585 025134 022737 000010 002256  CMP    #10,FLAG      ;LOOK IF 10 REPORT
3586 025142 001314
3587
3588 025144          ESCAPE  TST
3589
3590
3591
3592
3593 025150          RXLEND:
3594 025150          ENDTST

```

3596
3597 025152

```
BADHEAD
;***** TEST7 *****
;TRANSMIT DIFFERENT FRAMES OF VARIOUS LENGTH IN EXTERNAL LOOP BACK
;MODE ON CHANNEL A AT 72KB
BADHEAD
;***** TEST7 *****
```

3601
3602
3603
3604
3605
3606

3607 025152

```
STARS 1
;
;AT BEGINNING OF TEST ,CHECK IF LOOP BACK CONNECTORS ARE INSTALLED
;OR NOT:IF NOT INSTALLED = EXIT TEST AND GIVE ERROR MESSAGE
;*****
;
;
;
;QBUS WRITE DIFFERENT TX TABLE OF VARIOUS LENGTH, LOAD IN KMV11 CSR'S
;THE TX AND RX TABLE ADDRESS ,THE TABLE LENGTH AND TRANSFER SPEED (72KB)
;
;
;DCT11 EXECUTE THE TRANSFER IN EXTERNAL MODE ON CHA AND WRITE BACK
;IN RX TABLE
;QBUS CHECK BSELO TO SEE THE STATUS OF THE TEST AND IF TEST DONE CHECK IF
;RX TABLE =TX TABLE
;
;
;PARAMETERS SELECTION:
```

3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650

```
SEL2= TX TABLE ADDRESS
SEL4= TX TABLE LENGTH
BSEL6= EXTENDED ADDRESS OF TX TABLE
BSEL7= " " RX
SEL12= RX TABLE ADDRESS
SEL14= SPEED SELECTION (=141 IF 72KB)
BSEL16= ERROR STATUS
BSELO= TEST STATUS
SEL10= RECEIVE BYTE COUNT >0 IF TX>RX
<0 IF TX<RX

;
;
;TEST STATUS DESCRIPTION:
BSELO= 0 =TEST DONE CHECK RX TABLE
BSELO= 200 =TIMEOUT ERROR
BSELO= TSTNB =NO KMV11 ANSWER
BSELO= 100 =ERROR DURING TEST ,LOOK WHICH ONE BY TESTING BSEL16
;
;
;ERROR STATUS DESCRIPTION:
```

3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707

;
; WHEN BSEL0=100,GIVE CONTAINT OF ERROR STATUS AND WORD COUNT DISCREPANCY
;
;
; BSEL16= BIT14=1 =FCS ERROR
; BSEL16= BIT13=1 =OVERRUN ERROR
; BSEL16= BIT8 =1 =ILLEGAL INTERRUPT ERROR
; BSEL16= BIT7 =1 =RX ABORT ERROR
; BSEL16= BIT6 =1 =UNDERRUN ERROR
; BSEL16= BIT5 =1 =BYTE COUNT DISCREPANCY
; BSEL16= BIT4 =1 =DMA IN TIMEOUT ERROR
; BSEL16= BIT3 =1 =DMA OUT TIMEOUT ERROR
; BSEL16= BIT2 =1 =CLOCK PROBLEM
; BSEL16= BIT1 =1 =DATA COMPARE ERROR BETWEEN TX AND RX TABLE (USE
; ONLY DURING SELF TEST)
;
; MICRO DIAG TEST DESCRIPTION:
; TEST 42 =TRANSMIT VARIOUS LENGTH FRAME AT 72 KBAUDS SPEED ON CHANNEL A
; IN EXTERNAL LOOP BACK MODE
;
;
; CAUTION:
;
; RUN ONLY WITH EXTERNAL LOOP BACK CONNECTOR:
;
;
; NOTE:
;
; TO FULLY TEST KMV11 DIAGNOSTIC MUST BE RUN WITH RS422 AND RS423
; EXTERNAL LOOP BACK CONECTOR
;
; EXTERNAL LOOP BACK CONNECTOR:
;
; KMV11 A CAN OPERATE EITHER IN RS422 OR RS 423 LEVEL CONVERTERS
;
;
; RS422 LOOP BACK:
; TO TEST COMPLETELY A KMV11 B IN RS422 MODE ,RUN THIS DIAGNOSTIC
; WITH LOOP BACK CONNECTOR PLUG ;
; -USE H3255 TO LOOP DIRECTLY AT THE OUTPUT OF THE MODULE
; USE H3251 PLUG AT THE END OF BC55U MODEM CABLE CONNECTOR ASSY.
;
;
; RS423 LOOP BACK:
; TO TEST COMPLETELY A KMV11 A IN RS423 MODE ,RUN THIS DIAGNOSTIC
; WITH LOOP BACK CONNECTOR PLUG ;
; -USE H3255 TO LOOP AT THE OUTPUT OF THE MODULE
; USE H3251 PLUG AT THE END OF BC55H MODEM CABLE CONNECTOR ASSY.
;
;
;
; RS422 LOOP BACK:
; SAME AS FOR RS423.
;

CX

S708
S709
S710
S711
S712
S713
S714
S715
S716 025152

!CAUTION:
!USE OF M325 LOOP BACK CONNECTOR WILL CAUSE MESSAGES ERROR IN TEST A.
!
!
!
!
!
!STARS 1

```

3718 025152          BGNTST
3719 025152 004737 014372      JSR    PC,CLRKMV      ;CLEAR REGISTERS
3720 025156 005737 012470      TST    LOOP          ;IS LOOP BIT=1?
3721 025162 001012              BNE    BGNTXA        ;YES GO ON TEST
3722 025164              PRINTF  @MLOOP       ;NO LOOP BACK CONNECTOR
3723                          ;TEST NOT EXECUTED
3724 025204          EXIT    TST
3725
3726
3727 025210 004737 014474      BGNTXA: JSR    PC,MAINM1 ;SET MAINT MODE
3728 025214 005037 002256      CLR    FLAG
3729
3730 025220 012703 000005      MOV    @5,R3         ;SELECT RANDOM PATTERN
3731 025224 012737 000141 012406  MOV    @KB72,TSPEED ;SELECT SPEED
3732
3733 025232 012737 000001 012410  TXATAR: MOV    @1,LENGTH ;1ST TABLE LENGTH(1 WORD)
3734
3735 025240 013704 012410      TXABGN: MOV    LENGTH,R4
3736 025244          BREAK
3737
3738 025246 012702 002362      MOV    @TABLE,R2
3739 025252 004737 013152      10$:  JSR    PC,GENER    ;WRITE TABLE
3740 025256 013722 012372      MOV    DATA,(R2).
3741 025262 005304              DEC    R4
3742 025264 001372              BNE    10$
3743
3744
3745
3746 025266 013704 012410      MOV    LENGTH,R4     ;CLEAR RX TABLE
3747 025272 012702 006362      MOV    @RTABLE,R2
3748 025276 005022              CLR    (R2).
3749 025300 005304              DEC    R4
3750 025302 001375              BNE    20$
3751
3752
3753
3754
3755
3756
3757
3758 025304 013777 012406 165152  MOV    TSPEED,@KMVP14 ;SEND TX SPEED
3759 025312 012777 002362 165132  MOV    @TABLE,@KMVP02 ; " TX TABLE ADDRESS
3760 025320 013777 012410 165126  MOV    LENGTH,@KMVP04 ; " " LENGTH
3761 025326 012777 006362 165126  MOV    @RTABLE,@KMVP12 ;SEND RX TABLE ADDRESS
3762 025334 005077 165116      CLR    @KMVP06       ;CLR EXTENDED ADDRESS
3763
3764
3765
3766
3767
3768 025340 004537 014556      1$:  JSR    R5,TSTNUB
3769 025344 000042              .WORD  42           ;DO TEST 42- CHB TEST
3770
3771
3772
3773 025346          2$:  WAITB  0.3       ;WAIT FOR TEST EXECUTION
3774

```

3775									
3776	025366	004737	013074		JSR	PC, TSTERR			;CHECK BSELO
3777									
3778	025372	000427			BR	6:			;TEST OK CHECK RX TABLE
3779	025374	000402			BR	3:			;TIMEOUT ERROR
3780	025376	000401			BR	3:			;NO KMV11 ANSWER
3781	025400	000410			BR	4:			;CHECK SEL16 TO SEE WHICH ONE
3782									
3783									
3784	025402			3:	ERRHRD	32,EM0004			;NO KMV11 ANSWER
3785	025412	004737	012732		JSR	PC,CHKMAX			;CHECK IF TOO MANY ERROR
3786	025416				ESCAPE	TST			
3787									
3788									
3789	025422			4:					;ERROR DURING TEST READ ERROR STATUS
3790									;TO CHECK WHICH ONE
3791									
3792	025422	017737	165040	012416	MOV	8KMVP16,STAERR			;READ ERROR STATUS
3793									
3794	025430	017737	165024	012420	MOV	8KMVP10,WRDCNT			;READ WORD COUNT DISCREPANCY
3795									
3796	025436				ERRHRD	33,EM0022,PRSTER			;ERROR WHILE TX,RX FRAMES,GIVE ERROR
3797									;GIVE ERROR STATUS,WORD CNT DISCREPANCY
3798	025446				ESCAPE	TST			
3799									
3800									
3801									
3802									
3803									
3804	025452	012702	002362	6:	MOV	0TABLE,R2			;LOAD TABLE PARAMETERS
3805	025456	012705	006362		MOV	0RTABLE,R5			
3806	025462	013704	012410		MOV	LENGTH,R4			
3807									
3808	025466	022225			RXACK:	CMP	(R2), (R5)		;CHECK TX AND RX TABLE
3809	025470	001015				BNE	RXAERR		
3810	025472	005304				DEC	R4		
3811	025474	001374				BNE	RXACK		
3812									
3813	025476	062737	000400	012410	ADD	0400,LENGTH			;CHANGE LENGTH
3814	025504	022737	002000	012410	CMP	02000,LENGTH			
3815	025512	100252			BPL	TXABGN			
3816	025514	005303			DEC	R3			;SELECT NEW PATTERN
3817	025516	001245			BNE	TXATAR			;ALL DONE
3818	025520	000137	025642		JMP	RXAEND			
3819									
3820									
3821									
3822	025524	162705	000002		RXAERR:	SUB	02,R5		
3823	025530	162702	000002			SUB	02,R2		
3824									
3825	025534	011237	012402		MOV	(R2),TXDATA			
3826	025540	011537	012404		MOV	(R5),RXDATA			
3827									
3828	025544	005737	002256		TST	FLAG			;LOOK IF 1ST ERROR
3829	025550	001014			BNE	30:			
3830									
3831	025552				ERRHRD	34,EM0015,PR/ME			;DATA CMP ERROR


```

3832 025562 005237 002256      INC    FLAG
3833 025566 062702 000002      ADD    02,R2      ;POINT NEXT ADDRESS
3834 025572 062705 000002      ADD    02,R5
3835 025576 000137 024774      JMP    RXLCK
3836
3837 025602          304:  ERRHRD 34,0,PRAMEF      ;SHORT REPORT
3838 025612 005237 002256      INC    FLAG
3839 025616 062702 000002      ADD    02,R2
3840 025622 062705 000002      ADD    02,R5      ;POINT NEXT ADDRESS
3841 025626 022737 000010 002256  CMP    010,FLAG    ;LOOK IF 10 REPORT
3842 025634 001314
3843
3844 025636          ESCAPE TST
3845
3846
3847
3848
3849
3850 025642          RXAEND:
3851 025642          ENDTST

```

3853
3854 025644

BADHEAD
:***** TEST8 *****
:TEST MODEM SIGNALS IN EXTERNAL LOOP BACK
BADHEAD
:***** TEST8 *****

3855
3856 025644

3857
3858
3859
3860
3861
3862

3863 025644

STARS 1
:HOST SFI TEST NUMBER 45
:DCT11 TEST MODEM SIGNAL 105,106,109,111,112,107,108,125,140,141
:BY SETTING AND CLEARING BIT 105,108,111,141,TIS AND TESTING
:BIT 106,109,125,107,112,142.

3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879

:
:IF TEST =OK,DCT11 CLEAR BSELO
:IF ERROR SET 100 IN BSELO AND REPORT ERROR
:
:ERROR REPORT DESCRIPTION:
:SEL2 INDICATE WHICH MODEM SIGNAL IS TESTED
:SEL4 INDICATE THE RESULT OF THE TEST
:SEL10 INDICATE IF IT WAS DURING A CLEAR OR A SET OPERATION
:
:SEL 10 BIT 1=0 INDICATE A CLEAR OPERATION ON TESTED MODEM SIGNAL
: =1 " SET

3880
3881

:
:

3882
3883

:SEL2 FORMAT (TESTED SIGNAL):

3884
3885

: / / / / 141 / TIS / 111 / 108 / 105 /
: BIT 7 BIT 6 BIT5 BIT4 BIT3 BIT2 BIT1 BIT0

3886
3887

:
:RESULT OF TEST (SEL4):

3888
3889

: / / / 106 / 125 / 109 / 142 / 112 / 107 /
: BIT 7 BIT 6 BIT5 BIT4 BIT3 BIT2 BIT1 BIT0

3890
3891

3892
3893

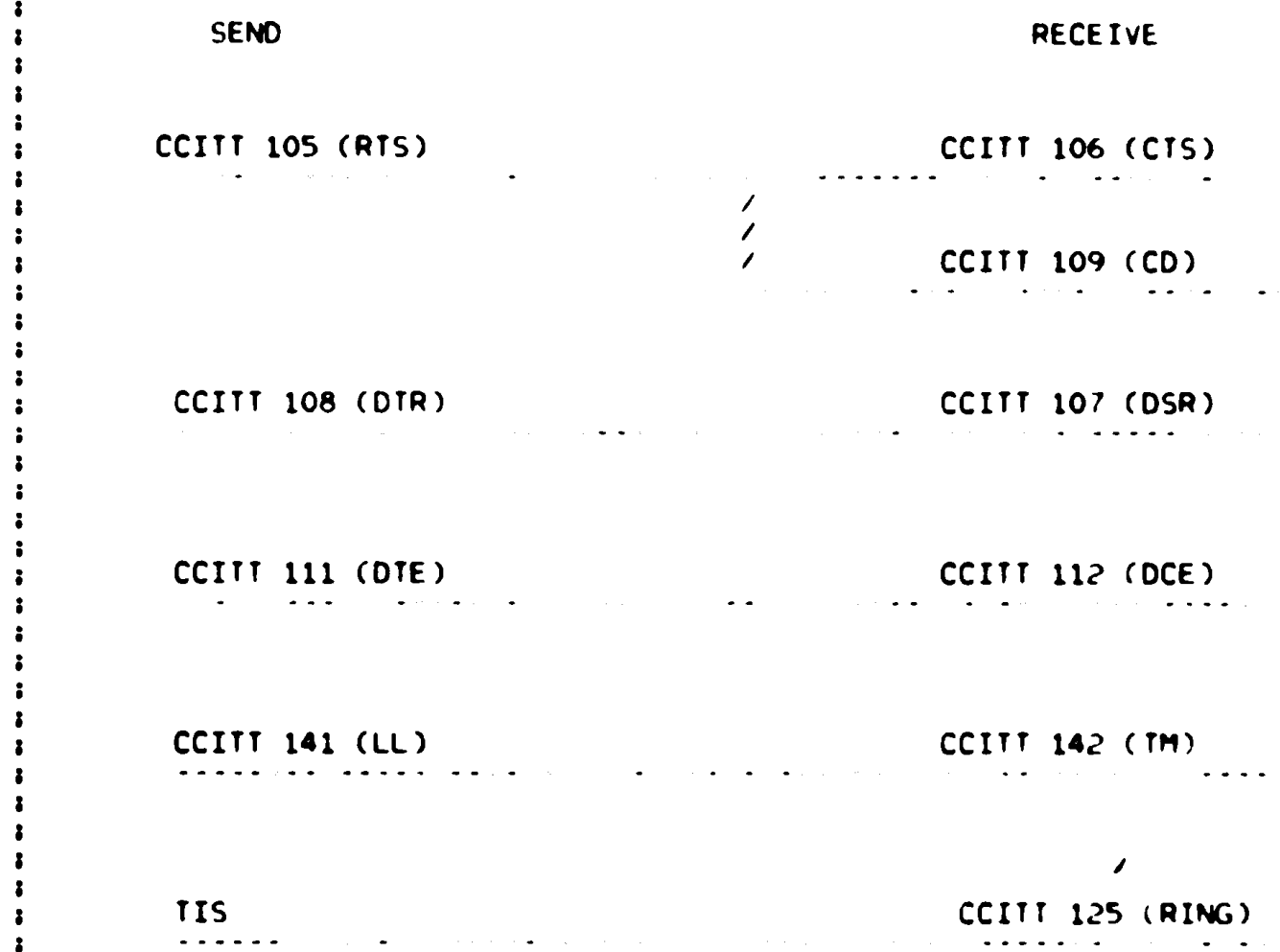
3894
3895

3896
3897

3898
3899
3900

3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958

;MODEM SIGNAL LINK:



;CAUTION:

;RUN ONLY WITH EXTERNAL LOOP BACK CONNECTOR:

;TO BE FULLY TESTED ,KMV11 DIAGNOSTIC MUST BE RUN WITH RS422 AND RS423
;EXTERNAL LOOP BACK CONECTOR

;EXTERNAL LOOP BACK CONNECTOR:

;KMV11 A CAN OPERATE EITHER IN RS422 OR RS 423 LEVEL CONVERTERS

;RS422 LOOP BACK:

3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982 025644
3983
3984

; TO TEST COMPLETELY A KMV11 B IN RS422 MODE ,RUN THIS DIAGNOSTIC
; WITH LOOP BACK CONNECTOR PLUG :
; -USE H3255 TO LOOP DIRECTLY AT THE OUTPUT OF THE MODULE
; -USE H3251 PLUG AT THE END OF BC55U MODEM CABLE CONNECTOR ASSY.
;
;
; RS423 LOOP BACK:
; TO TEST COMPLETELY A KMV11 A IN RS423 MODE ,RUN THIS DIAGNOSTIC
; WITH LOOP BACK CONNECTOR PLUG :
; USE H3255 TO LOOP AT THE OUTPUT OF THE MODULE
; USE H3251 PLUG AT THE END OF BC55H MODEM CABLE CONNECTOR ASSY.
;
;
;
; RS232 LOOP BACK:
; SAME AS FOR RS423.
;
; CAUTION:
; USE OF H325 LOOP BACK CONNECTOR WILL CAUSE MESSAGES ERROR IN TEST A.
;
;
;
;
STARS 1

```

3986 025644          RGNTST
3987 025644 004737 014372      JSR      PC,CLRKMV          ;CLEAR ALL REGISTERS
3988
3989 025650 005737 012470      TST      LOOP
3990 025654 001012              BNE      MODSIG          ;LOOP BACK PRESENT GO ON
3991
3992 025656          PRINTF  @MLOOP          ;NO LOOP BACK CONNECTOR
3993
3994
3995 025676          EXIT      TST          ;GO TO FOLLOWING TEST
3996
3997
3998
3999 025702 004737 014474      MODSIG: JSR      PC,MAINM1      ;SET MAINTENANCE MODE
4000 025706 004537 014556      JSR      R5,TSTNUB
4001 025712 000045              .WORD    45          ;SEND TEST 45
4002
4003 025714          WAITB    0,4
4004
4005 025734 004737 013074      JSR      PC,TSTERR        ;CHECK TEST RESULT
4006 025740 000430              BR       3$              ;TEST OK GO ON
4007 025742 000402              BR       4$              ;TIMEOUT
4008 025744 000401              BR       4$              ;NO TEST ANSWER
4009 025746 000406              BR       5$              ;ERROR DURING TEST ,LOOK WHICH ONE
4010
4011
4012
4013 025750          4$:      ERRHRD  36,EM0004      ;NO ANSWER
4014 025760          ESCAPE  TST
4015
4016 025764 017737 164462 002264 5$:      MOV      @KMVP02,GOOD      ;READ WHICH SIGNAL WAS TESTED
4017 025772 017737 164456 012366      MOV      @KMVP04,BAD      ;
4018 026000 017737 164454 012372      MOV      @KMVP10,DATA     ;READ SIGAL VALUE
4019
4020 026006          ERRHRD  37,EM0032,PMODEM      ;REPORT ERROR
4021 026016          ESCAPE  TST
4022
4023 026022          3$:
4024 026022          MODEND:
4025
4026
4027 026022          ENDTST

```

4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048 026024
4049
4050 026026
4051 026036
4052 026046
4053 026060
4054 026072
4055
4062
4063
4064 026072
026075
026100
026103
026106
026111
026114
026117
026122
4065 026124
026127
026132
026135
026140
026143
026146
026151
026154
026157
4066 026160
026163
026166
026171
026174
026177
026202
026205
026210

.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,2,0,0,674,YES
GPRMD PRIRTY,4,0,7000,4,7,YES
GPRMD LOOPBK,6,0,1,0,1,YES
ENDHRD

ADDRESS: .ASCIZ /MICRO-CPU CSR ADDRESS : /

VECTOR: .ASCIZ /MICRO CPU VECTOR ADDRESS : /

PRIRTY: .ASCIZ /MICRO CPU PRIORITY LEVEL : /

4067	026213	000			
	026214	111	123	040	LOOPBK: .ASCIZ /IS LOOP BACK CONNECTOR PLUGGED? 0=NO,1=YES: /
	026217	114	117	117	
	026222	120	040	102	
	026225	101	103	113	
	026230	040	103	117	
	026233	116	116	105	
	026236	103	124	117	
	026241	122	040	120	
	026244	114	125	107	
	026247	107	105	104	
	026252	077	040	060	
	026255	075	116	117	
	026260	054	061	075	
	026263	131	105	123	
	026266	072	040	000	
4068					.EVEN
4069					
4070					
4071					
4072					
4073					
4074					

4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088 026272
4089
4098
4099
4100 026274
4101
4102
4109
4110

.SBTTL SOFTWARE PARAMETER CODING SECTION

```

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

```

BGNSFT

ENDSFT


```
4112
4113 026274          $PATCH::
4114 026274          .BLKW  50
4115
4122
4123 026414          LASTAD
         026420      L$LAST::
4124 026420          ENDMOD
4125
4126
```

```
4128  
4129  
4142  
4143 026420          BGNSETUP          1  
4144 026420          BGNPTAB  
4145 026424 177000   .WORD 177000  
4146 026426 000300   .WORD 300  
4147 026430 004000   .WORD 4000  
4148 026432 000001   .WORD 1  
4149 026434          ENDPTAB  
4150 026434          ENDSETUP  
4151  
4152  
4153  
4154  
4155  
4156          000001          .END
```

ABORT	022660	C\$CVEC=	000036	EM0002	015115	GENER1	013274	I\$DUJ	=	000041
ADDR	002354	C\$DCLN=	000044	EM0003	015160	GENEX	013432	I\$HRD	=	000041
ADDRESS	026072	C\$DODU=	000051	EM0004	015246	GENINC	013424	I\$INIT	=	000041
ADR	=	C\$DRPT=	000024	EM0006	015275	GENISH	013302	I\$MOD	=	000041
ASSEMB	=	C\$DU	=	EM0007	015336	GENOUT	024032	I\$MLG	=	000041
BAD	012366	C\$EDIT=	000003	EM0011	015416	GENRAN	013304	I\$PROT	=	000040
BDDAT	002360	C\$ERDF=	000055	EM0012	015470	GENROT	013260	I\$PTAB	=	000041
BDRGEN	023626	C\$ERMR=	000056	EM0013	015552	GENRO	013246	I\$PWR	=	000041
BDROKO	024052	C\$ERRR=	000060	EM0014	015716	GENR1	013236	I\$RPT	=	000041
BGNXA	025210	C\$ERSF=	000054	EM0015	015767	GFNSL	013170	I\$SEG	=	000041
BIT0	=	C\$ERSO=	000057	EM0016	016063	GENO	013210	I\$SETU	=	000041
BIT00	=	C\$ESCA=	000010	EM0017	016156	GEN1	013214	I\$SFT	=	000041
BIT01	=	C\$ESEG=	000005	EM0022	016243	GEN25	013230	I\$SRV	=	000041
BIT02	=	C\$ESUB=	000003	EM0023	016326	GEN52	013222	I\$SUB	=	000041
BIT03	=	C\$ETST=	000001	EM0024	016407	GETPRM	022456	I\$TST	=	000041
BIT04	=	C\$EXIT=	000032	EM0027	016431	GOOD	002264	J\$JMP	=	000167
BIT05	=	C\$GETB=	000026	EM0031	016503	GOOD0	002266	KB1.2	=	013224 G
BIT06	=	C\$GETW=	000027	EM0032	016577	GOOD1	002270	KB56	=	000174 G
BIT07	=	C\$GMAN=	000043	EM0033	015646	GOOD10	002300	KB64	=	000154 G
BIT08	=	C\$GPHR=	000042	EM0035	016663	GOOD12	002302	KB68	=	000146 G
BIT09	=	C\$GPLO=	000030	EM0036	016755	GOOD14	002304	KB72	=	000141 G
BIT1	=	C\$GPRI=	000040	END	022724	GOOD16	002306	KIND	=	012376
BIT10	=	C\$INIT=	000011	ERRBLK	002226 G	GOOD2	002272	KMTLVL	=	012446
BIT11	=	C\$INLP=	000020	ERRCNT	002234	GOOD4	002274	KMVCSR	=	012450
BIT12	=	C\$MANI=	000050	ERRMSG	002224 G	GOOD6	002276	KMVLVL	=	012436
BIT13	=	C\$MEM	=	ERRNBR	002222 G	G\$CNT0=	000200	KMVP02	=	012452
BIT14	=	C\$MSG	=	ERRTYP	002220 G	G\$DELM=	000372	KMVP04	=	012454
BIT15	=	C\$OPEN=	000034	EVL	=	G\$DISP=	000003	KMVP06	=	012456
BIT2	=	C\$PNTB=	000014	EXADDR	012362	G\$EXCP=	000400	KMVP10	=	012460
BIT3	=	C\$PNTF=	000017	E\$END	=	G\$HILI=	000002	KMVP12	=	012462
BIT4	=	C\$PNTS=	000016	F\$LOAD=	000035	G\$LOLI=	000001	KMVP14	=	012464
BIT5	=	C\$PNTX=	000015	FLAG	002256	G\$NO	=	KMVP16	=	012466
BIT6	=	C\$QIO	=	FTIME	002250	G\$OFFS=	000400	KMVV00	=	012434
BIT7	=	C\$RDBU=	000007	F\$AU	=	G\$OFSI=	000376	KMVV02	=	012442
BIT8	=	C\$REFG=	000047	F\$AUTO=	000020	G\$PRMA=	000001	KMVV04	=	012440
BIT9	=	C\$RESE=	000033	F\$BGN	=	G\$PRMD=	000002	KMVV06	=	012444
BOE	=	C\$REVI=	000003	F\$CLEA=	000007	G\$PRML=	000000	KMV11A	=	002000 G
BSELO	012370	C\$RFLA=	000021	F\$DU	=	G\$RADA=	000140	LENGTH	=	012410
BSEL1	002332	C\$RPT	=	F\$END	=	G\$RADB=	000000	LOCK	=	002230
CBSELO	013644	C\$SEFG=	000046	F\$HARD=	000004	G\$RADD=	000040	LOE	=	040000 G
CHANEL	012400	C\$SPRI=	000041	F\$HW	=	G\$RADL=	000120	LOGDEV	=	002242
CHMAX	012732	C\$SVEC=	000037	F\$INIT=	000006	G\$RADO=	000020	LOKFLG	=	012430
CKALL	013702	C\$TPRI=	000013	F\$JMP	=	G\$XFER=	000004	LOOP	=	012470
CKREG	014150	DATA	012372	F\$MOD	=	G\$YES	=	LOOPBK	=	026214
CKSELO	013612	DATA1	=	F\$MSG	=	HELP	=	LOT	=	000010 G
CLRKMV	014372	DATA2	=	F\$PROT=	000021	MOE	=	L\$ACP	=	002110 G
COUNT	002350	DELCT1	002260	F\$PWR	=	IBE	=	L\$APT	=	002036 G
C\$AU	=	DELCT2	002262	F\$RPT	=	IDU	=	L\$AU	=	023104 G
C\$AUTO=	000061	DFPTBL	002154 G	F\$SEG	=	IER	=	L\$AUT	=	002070 G
C\$BRK	=	DIAGMC=	000000	F\$SOFT=	000005	INIFLG	012426	L\$AUTO	=	022726 G
C\$BSEG=	000004	DROPD	023052	F\$SRV	=	INTFLG	012364	L\$CCP	=	002106 G
C\$BSUB=	000002	EF.CON=	000036 G	F\$SUB	=	INTTX	024110	L\$CLEA	=	023020 G
C\$CEFG=	000045	EF.NEW=	000035 G	F\$SW	=	ISR	=	L\$CO	=	002032 G
C\$CLCP=	000062	EF.PWR=	000034 G	F\$TEST=	000001	IXE	=	L\$DEPO	=	002011 G
C\$CLEA=	000012	EF.RES=	000037 G	GDDAT	002356	I\$AU	=	L\$DESC	=	002164 G
C\$CLOS=	000035	EF.STA=	000040 G	GDREV	012424	I\$AUTO=	000041	L\$DESP	=	002076 G
C\$CLP1=	000006	EM0001	015021	GENER	013152	I\$CIN	000041	L\$DEVP	=	002070 G

00

L\$DISP 002132 G
L\$DLY 002116 G
L\$DTP 002040 G
L\$DTYP 002034 G
L\$DU 023024 G
L\$DUT 002072 G
L\$DVTY 012672 G
L\$EF 002052 G
L\$ENVI 002044 G
L\$ERRT 002220 G
L\$ETP 002102 G
L\$EXP1 002046 G
L\$EXP3 002064 G
L\$EXPS 002066 G
L\$HARD 026026 G
L\$HIME 002120 G
L\$MPCP 002016 G
L\$MPTP 002022 G
L\$HW 002154 G
L\$ICP 002104 G
L\$INIT 022242 G
L\$LADP 002026 G
L\$LAST 026420 G
L\$LOAD 002100 G
L\$LUN 002074 G
L\$MREV 002050 G
L\$NAME 002000 G
L\$PRIO 002042 G
L\$PROT 002122 G
L\$PRT 002112 G
L\$REPP 002062 G
L\$REV 002010 G
L\$RPT 022234 G
L\$SOFT 026274 G
L\$SPC 002056 G
L\$SPCP 002020 G
L\$SPTP 002024 G
L\$STA 002030 G
L\$SW 002236 G
L\$TEST 002114 G
L\$TIML 002014 G
L\$UIT 002240 G
L\$UNIT 002012 G
L10001 002164
L10002 020654
L10003 020714
L10004 021224
L10005 021504
L10006 021542
L10007 021602
L10010 021642
L10011 021702
L10012 021772
L10013 022116
L10014 022152
L10015 022232

L10016 022240
L10017 022724
L10020 023016
L10021 023022
L10022 023102
L10023 023104
L10024 023222
L10025 023272
L10026 023606
L10027 024052
L10030 024504
L10031 025150
L10032 025642
L10033 026022
L10034 026072
L10035 026274
L10036 026424
L10040 026434
MAINM1 014474
MAINT0 = 054000 G
MAINT1 = 044000 G
MAXERR 002232
MAXPRI = 000300 G
MSELO 017612
MCLR = 040000 G
MFRAM1 020006
MFRAM2 020064
MINT 017556
MLOOP 020537
MODEM1 020243
MODEM2 020307
MODEM3 020346
MODEM4 020411
MODEND 026022
MODSIG 025702
MRAMEF 020466
MREG0 017116
MREG10 017336
MREG12 017402
MREG14 017446
MREG16 017512
MREG2 017162
MREG4 017226
MREG6 017272
MSELO 017050
MSTER1 020146
MSTER2 020200
MT11V 017730
MVECT 017654
NERRS 013024
NEXT 022402
NUB 012412
NUMBER 002352
O\$APTS = 000000
O\$AU = 000000
O\$BGNR = 000000

O\$BGNS = 000000
O\$DU = 000001
O\$ERRT = 000000
O\$GNSW = 000001
O\$POIN = 000001
O\$SETU = 000001
PADFLT 021506 G
PBSELO 021544 G
PFRAME 021704 G
PMODEM 021774 G
PNT = 001000 G
PRALL 020716 G
PRAMEF 022120 G
PRI = 002000 G
PRINT 020656 G
PRIRTY 026160
PRI00 = 000000 G
PRI01 = 000040 G
PRI02 = 000100 G
PRI03 = 000140 G
PRI04 = 000200 G
PRI05 = 000240 G
PRI06 = 000300 G
PRI07 = 000340 G
PRREG 021226 G
PRSELO 020616 G
PRSTER 022154 G
PRT11V 021644 G
PSTACK 002246
PVECT 021604 G
QV.FLG 012431
RANCLC 013404
RANDN 002342
RANEX 013422
RANGEN 013324
RANMTA 002340
RANSEC 013410
RANSEL 002336
RANST 002334
RAN1 013336
RAN2 013354
RAN4 013412
READ 014632
REGADR 012472
REVPRO 023234
ROMMAP 023106
RSTREG 013532
RTABLE 006362
RTCLK 023304
RUNNIN 022666
RXACK 025466
RXAEND 025642
RXAERR 025524
RXCK 024344
RXCNT 012414
RXDATA 012404

RXEND 024504
RXERR 024366
RXLCK 024774
RXLEND 025150
RXLERR 025032
SAVE4 002252
SAVE6 002254
SAVPC 002244
SAVPC1 002344
SAVREG 013452
SAVSTA 002346
SELO 002310
SEL1 002312
SEL10 002322
SEL12 002324
SEL14 002326
SEL16 002330
SEL2 002314
SEL4 002316
SEL6 002320
SETUP 022374
SSTACK 012672
STAERR 012416
SVCGBL = 000000
SVCINS = 177777
SVCSUB = 177777
SVCTAG = 177777
SVCTS = 177777
SVCTST = 177777
S\$LSYM = 010000
TFM36 014733
TIM 014716
TSPEED 012406
TSTERR 013074
TSTNUB 014556
TTABLE 002362
TXABGN 025240
TXATIR 025232
TXDATA 012402
TXLBGN 024546
TXLTAR 024540
TXSTAR 024114
T\$ARGC = 000001
T\$CODE = 003032
T\$ERRN = 000045
T\$EXCP = 000000
T\$FLAG = 000040
T\$FREE = 026434
T\$GMAN = 000000
T\$HILI = 000001
T\$LAST = 000001
T\$LOLI = 000000
T\$LSYM = 010000
T\$LTNO = 000010
T\$NE ST = 177777

T\$NS0 = 000000
T\$NS1 = 000005
T\$PCNT = 000000
T\$PTAB = 010037
T\$PTHV = 000001
T\$PTNU = 000001
T\$SAVL = 177777
T\$SEGL = 177777
T\$SIZE = 000006
T\$SUBN = 000000
T\$TAGL = 177777
T\$TAGN = 010041
T\$TEMP = 000000
T\$TEST = 000010
T\$TSTM = 177777
T\$TSTS = 000001
T\$AU = 010023
T\$AUT = 010020
T\$CLE = 010021
T\$DAT = 010040
T\$DU = 010022
T\$HAR = 010034
T\$HW = 010001
T\$INI = 010017
T\$MSG = 010015
T\$PC = 000001
T\$PRO = 010000
T\$PTA = 010037
T\$RPT = 010016
T\$SOF = 010035
T\$TES = 010033
T1 023106 G
T2 023224 G
T3 023274 G
T4 023610 G
T5 024054 G
T6 024506 G
T7 025152 G
T8 025644 G
UAM = 000200 G
UNIT 012422
UUT 012432
VECT 012374
VECTOR 026124
WAIT1 012722
WAIT2 012702
WRDCNT 012420
WRITE 014604
X\$ALWA = 000000
X\$FALS = 000040
X\$OFFS = 000400
X\$TRUE = 000020
\$LSTIN = 000000
\$LSTTA = 000000
\$PATCH 026274 G

. ABS. 026434 000
 000000 001
ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28944 WORDS (114 PAGES)
DYNAMIC MEMORY: 20060 WORDS (77 PAGES)
ELAPSED TIME: 00:02:40
CNKMBA.BIC,CNKMBA.SEQ/ SP=SVC34.MLB/ML,CNKMBA.MAC

