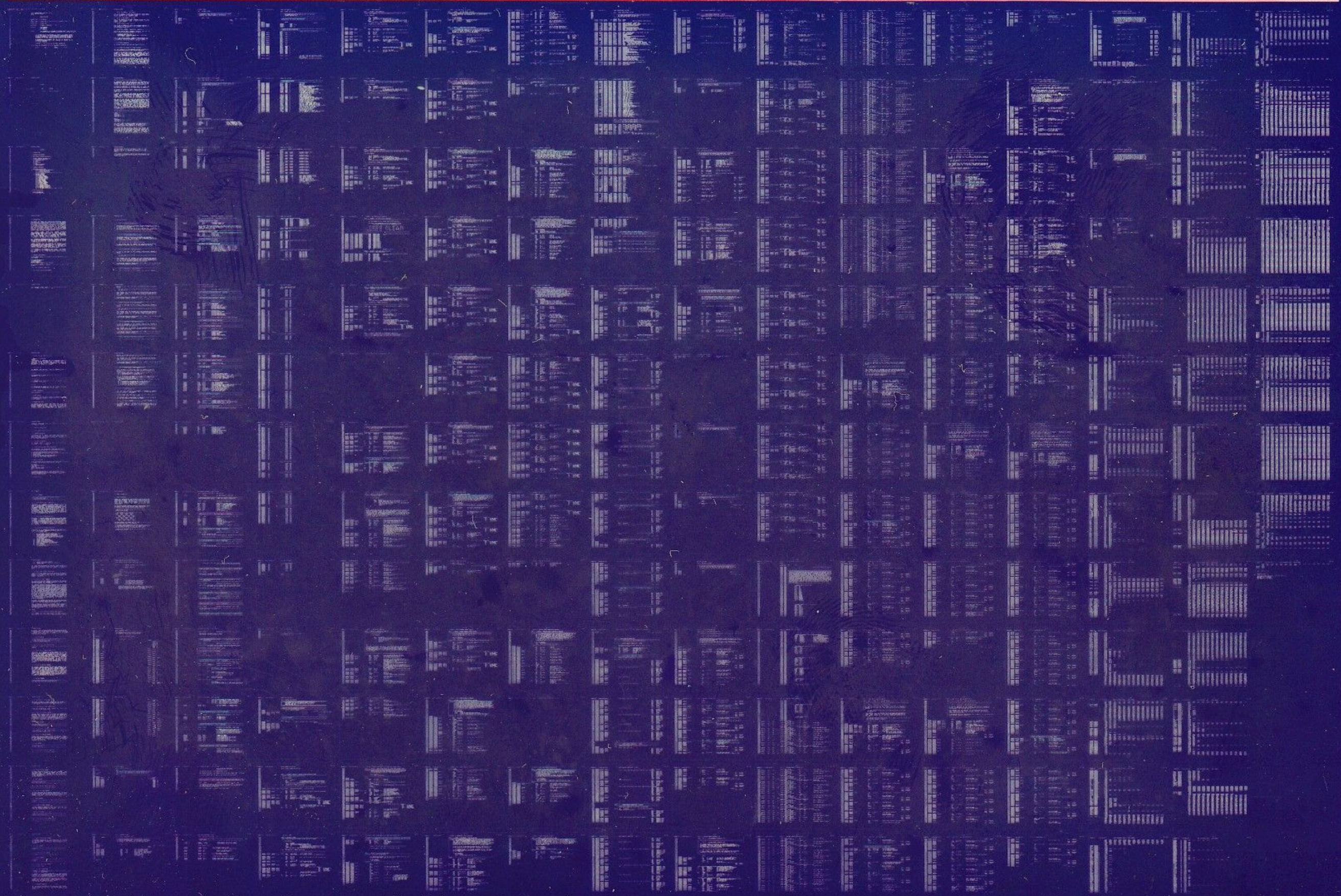


DMV11
M8053 M8064

DMV11 LINE UNIT DIAG 3
CVDMEA0

AH-S394A MC
FICHE 1 OF 1

MAY 1981
COPYRIGHT © 1981
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

.TITLE CVDMEA0 DMV11 LINE UNIT DIAG3
.SBTTL PROGRAM DOCUMENT
.REM @

IDENTIFICATION

PRODUCT CODE: AC-S392A-MC
PRODUCT NAME: CVDMEA0 DMV-11 LINE UNIT STATIC DIAGNOSTIC PART #3
PRODUCT DATE: JANUARY 1981
MAINTAINER: DIAGNOSTICS MERRIMACK CC:38P
AUTHORS: CHRIS BRIEMEN
DAVE HOFFMAN
RAY MARSHALL
PURPOSE: THIS DIAGNOSTIC IS DESIGNED TO PERFORM STATIC LOGIC TESTS FOR
THE M8053 OR M8064 (HEREAFTER REFERRED TO AS THE DMV OR DMV-11)

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) 1981 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP UNIBUS MASSBUS
DEC DECUS DECTAPE

CVDMEA.P11

10-DEC-80 09 16

PROGRAM DOCUMENT

41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

HISTORY

REV

DATE

REASON

0

14-JAN-81

INITIAL RELEASE

CVDMEA.P11

10-DEC-80 09:16

PROGRAM DOCUMENT

CONTENTS

56	
57	
58	
59	
60	1.0 INTRODUCTION
61	
62	2.0 HARDWARE REQUIREMENTS
63	
64	3.0 PRELIMINARY PROGRAM REQUIREMENTS
65	
66	4.0 GENERAL PROGRAM CONSIDERATIONS
67	4.1 DIAGNOSTIC SUPERVISOR
68	4.2 EXECUTION TIME
69	4.3 XXDP+
70	4.4 ACT/SLIDE
71	4.5 APT
72	4.6 MEMORY MANAGEMENT
73	4.7 ERROR LOGGING
74	
75	5.0 PROGRAM LOAD MEDIA
76	
77	6.0 OPERATING INSTRUCTIONS
78	6.1 LOADING AND STARTING PROCEDURES
79	6.1.1 LOADING PROCEDURES
80	6.1.2 STARTING PROCEDURES
81	6.1.3 ** STEPS FOR QUICK AND SIMPLE EXECUTION **
82	6.2 INITIAL DIALOGUE
83	6.3 PROGRAM OPTIONS
84	6.3.1 START COMMAND
85	6.3.2 RESTART COMMAND
86	6.3.3 CONTINUE COMMAND
87	6.3.4 PROCEED COMMAND
88	6.3.5 ADD COMMAND
89	6.3.6 DROP COMMAND
90	6.3.7 PRINT COMMAND
91	6.3.8 DISPLAY COMMAND
92	6.3.9 FLAGS COMMAND
93	6.3.10 ZFLAGS COMMAND
94	6.3.11 CONTROL CHARACTERS
95	6.3.12 HARDWARE PARAMETERS
96	6.3.13 SOFTWARE PARAMETERS
97	6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE
98	
99	7.0 TEST DESCRIPTIONS
100	
101	8.0 ERROR INFORMATION
102	8.1 ERROR REPORTING

CVDMEA.P11

10-DEC-80 09:16

PROGRAM DOCUMENT

103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158

1.0 INTRODUCTION

THE M8053 AND M8064 ARE SINGLE-LINE SYNCHRONOUS, MICRO-PROCESSOR BASED COMMUNICATIONS INTERFACES WHICH CAN SUPPORT BOTH CHARACTER-ORIENTED (DDCMP, BSC, ETC.) AND BIT-ORIENTED (SDLC, HDLC, ETC.) PROTOCOLS. THE PURPOSE OF THIS PROGRAM IS TO PERFORM STATIC DIAGNOSTIC TESTING OF THE VIA, FIFO, USYRT (BCP/BOP MODES), AND LINE DRIVERS ON THESE BOARDS. NOTE THAT ALL EXTERNAL LOOPBACK (XLB) TESTS ARE CONTAINED HERE. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: MODEM LOOPBACK AND ASSORTED EXTERNAL LOOPBACK TESTS (INCLUDING BCP:CRC-16/ODD VRC/EVEN VRC; BOP:CRC-CCITT-1'S/0'S).

THE STATIC LOGIC TESTS WILL PROVIDE EXTENSIVE TROUBLESHOOTING CAPABILITIES, SUCH AS TIGHT SCOPE LOOPS, SWITCH OPTIONS, AND ABILITY TO 'LOCK' ONTO INTERMITTENT ERRORS. IN ADDITION TESTS ARE DESIGNED AND STRUCTURED TO ACHIEVE MAXIMUM FAULT RESOLUTION AND FACILITATE REPLACEMENT OF THE SMALLEST FIELD REPLACEABLE UNIT.

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

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM ALLOWS MODIFICATION OF DEVICE PARAMETERS, SUCH AS LSI-BUS ADDRESS, VECTOR ADDRESSES AND DEVICE PRIORITY. IN ADDITION, THE OPERATOR CAN SPECIFY PARTICULAR TESTS TO BE RUN AND A VARIETY OF LOOPING, RUNNING, AND REPORTING MODES.

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

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8053/8064 STATIC LOGIC TESTS:

PDP-11/03 OR PDP-11/23
16K WORDS OF MEMORY
CONSOLE TERMINAL
M8053 OR M8064 COMMUNICATIONS INTERFACE

THE FOLLOWING HARDWARE IS REQUIRED TO FULLY TEST THE DMV-11 LINE DRIVERS:

H3254, H3255 LOOPBACK CONNECTORS

3.0 PRELIMINARY PROGRAM REQUIREMENTS

CVDMEA.P11 10-DEC-80 09:16

PROGRAM DOCUMENT

159
160
161

THIS PROGRAM (CVDME) SHOULD BE THE LAST OF THE FIVE DMV-11
STATIC DIAGNOSTICS TO BE RUN.

CVDMEA.P11 10-DEC-80 09:16

PROGRAM DOCUMENT

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

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

4.2 EXECUTION TIME

THE MAXIMUM TIME REQUIRED TO RUN THIS PROGRAM IS ABOUT 10 SECONDS PER PASS FOR EACH UNIT (5 SEC FOR PDP-11/23).

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.

4.7 ERROR LOGGING

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

5.0 PROGRAM LOAD MEDIA

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

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217

CVDME.A.P11

10-DEC-80 09:16

PROGRAM DOCUMENT

DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

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

6.1.2 STARTING PROCEDURES

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

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

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

6.2 INITIAL DIALOGUE

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

```
DRS LOADED
DIAG. RUN-TIME SERVICES
CVDME-A-0
DMV-11 LINE UNIT TESTS - PART 3 OF 3
UNIT IS M8053 OR M8064
DR>
```

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

6.3 PROGRAM OPTIONS

218
219
220
221
222
223
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
281
282
283
284
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

6.3.1 START COMMAND

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

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

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

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

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

6 3.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

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

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

CVDMEA.P11 10-DEC-80 09:16

PROGRAM DOCUMENT

ISR INHIBIT STATISTICAL REPORTS
 IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
 LOT LOOP ON TEST

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

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

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

6.3.1.5 EFFECT OF START COMMAND

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

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

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

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

EXAMPLE:

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

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST

330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385

CVDMEA.P11

10-DEC-80 09:16

PROGRAM DOCUMENT

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 CL. AND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

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

386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441

442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497

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:< AG-LIST>

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

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

6.3.4.2 EFFECT OF PROCEED COMMAND

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

6.3.5 ADD COMMAND

ADD/UNITS:<UNIT-LIST>

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

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

498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553

6.3.5.2 EFFECT OF ADD COMMAND

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

6.3.6 DROP COMMAND

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

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

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

6.3.6.2 EFFECT OF DROP COMMAND

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

6.3.7 PRINT COMMAND

PRI(NT)

6.3.7.1 EFFECT OF PRINT COMMAND

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

554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609

6.3.8.2 EFFECT OF DISPLAY COMMAND

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

6.3.9 FLAGS COMMAND

FLA(GS)

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

ZFL(AGS)

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

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

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

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC 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 3 QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.



CVDMEA.P11

10-DEC-80 09:16

PROGRAM DOCUMENT

610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665

1. DEVICE CSR ADDRESS : (0) 160020?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SEL0) RESIDE ON THE LSI-BUS. THE ALLOWABLE RANGE IS 160020-177760 (OCTAL), AND THE DEFAULT VALUE IS 160020.

2. DEVICE VECTOR ADDRESS : (0) 300 ?

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

3. DEVICE PRIORITY LEVEL : (0) 4 ?

THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF THIS DEVICE WILL BE EXECUTED. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS 4.

4. BOARD TYPE (0=M8064, 1=M8053-V35, 2=M8053-EIA) : (0) J ?

THIS IS THE TYPE OF DMV-11 CURRENTLY INSTALLED. NOTE THAT THE M8053 IS SWITCH SELECTABLE BETWEEN V.35 AND EIA.

5. TURNAROUND CONNECTOR TYPE -
(0=M3254/M3255, 1=INTEGRAL MODEM CABLE, 2=EIA CABLE,
3=V.35 CABLE, 4=NONE) : (0) 0 ?

THIS IS THE TYPE OF EXTERNAL LOOPBACK CONNECTOR BEING USED. IF NO LOOPBACK CONNECTOR IS PRESENT (4), THE EXTERNAL LOOPBACK TESTS WILL ALL BE RUN USING TTL-INTERNAL LOOPBACK.

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY THIS TEST.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

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

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

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

CVDMEA.P11 10-DEC-80 09:16

PROGRAM DOCUMENT

666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721

USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

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

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

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

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

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

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

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

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

THE SECOND TIME THRU THE SERIES, TABLES 7 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE 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

CVDMEA.P11 10-DEC-80 09:16

PROGRAM DOCUMENT

722
723
724
725
726
727
728
729
730
731

GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND
GETS AN 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN
TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7
THRU 15.

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

732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787

7.0 TEST DESCRIPTIONS

```

*****
* TEST 1 <RX DATA FLUSHING TEST>
*
* IN BCP MODE/HALF DUPLEX IT IS DESIRABLE TO HAVE THE ABILITY TO FLUSH
* THE USYRT OF ITS CRC CHARACTERS. THIS FLUSHING IS ACCOMPLISHED BY WRITING
* TO THE VIA SHIFT REGISTER.
* THIS TEST VERIFIES THAT WHEN THE VIA SR IS WRITTEN INTO, 8 PULSES WILL
* BE GENERATED AT THE CB1 PIN (WHICH DIRECTLY FEEDS THE CHARACTER FIFO).
*
*****

*****
* TEST 2 <INTEGRAL MODEM INTERFACE TEST>
*
* THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM AND A MESSAGE IS
* TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR ON
* THE BOARD OR AT THE END OF A CABLE. THE FOLLOWING MESSAGE WILL BE
* SENT IN BCP MODE WITH CRC-16 SPECIFIED:
*
* SYNC SYNC 000 125 252 377 000 CRC1 CRC2 SYNC
*
* IF THE P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL
* TURNAROUND IS PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
*
*****

*****
* TEST 3 <DATA TEST -- BCP XLB CRC-16>
*
* IF XLB IS SPECIFIED IN THE P-TABLE, THIS TEST WILL TRANSMIT &
* RECEIVE IN BCP MODE WITH CRC-16 ERROR DETECTION THE FOLLOWING
* MESSAGE:
*
* 125 252 000 377 001 002 004 010 020 040 100 200 376 375 373 367
* 357 337 277 177
*
* THIS MESSAGE WILL BE PRECEDED BY 3 SYNC CHARACTERS AND REPEATED
* THREE TIMES WITH CRC'S FOLLOWING EACH ONE. THE LAST TRANSMISSION OF
* THE CRC WILL BE FOLLOWED BY SEVERAL SYNC CHARACTERS BEFORE DROPPING
* TXE & RXE. 8-BIT CHARACTER LENGTHS ARE ALSO UTILIZED.
*
* IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
* USING INTERNAL LOOPBACK (TTLOOP=1).
*
*****

*****
* TEST 4 <DATA TEST -- BCP XLB ODD VRC>
*
* IF XLB IS SPECIFIED IN THE P-TABLE, THIS TEST WILL TRANSMIT &

```

PROGRAM DOCUMENT

788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843

* RECEIVE IN BCP MODE WITH ODD VRC ERROR DETECTION THE FOLLOWING
* MESSAGE:

125 252 000 377 001 002 004 010 020 040 100 200 376 375 373 367
357 337 277 177

* THIS MESSAGE WILL BE PRECEDED BY 3 SYNC CHARACTERS AND REPEATED
* THREE TIMES. AFTER THE LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
* SENT BEFORE DROPPING TXE & RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
* UTILIZED.

* IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
* USING INTERNAL LOOPBACK (TTLOOP=1).

* TEST 5 <DATA TEST -- BCP XLB EVEN VRC>

* IF XLB IS SPECIFIED IN THE P-TABLE, THIS TEST WILL TRANSMIT &
* RECEIVE IN BCP MODE WITH EVEN VRC ERROR DETECTION THE FOLLOWING
* MESSAGE:

125 252 000 377 001 002 004 010 020 040 100 200 376 375 373 367
357 337 277 177

* THIS MESSAGE WILL BE PRECEDED BY 3 SYNC CHARACTERS AND REPEATED
* THREE TIMES. AFTER THE LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
* SENT BEFORE DROPPING TXE & RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
* UTILIZED.

* IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
* USING INTERNAL LOOPBACK (TTLOOP=1).

* TEST 6 <DATA TEST -- BOP XLB CRC-CCITT-1>

* IF XLB IS SPECIFIED IN THE P-TABLE, THIS TEST WILL TRANSMIT &
* RECEIVE IN BOP MODE WITH CRC-CCITT-1 ERROR DETECTION THE FOLLOWING
* SHORT MESSAGE: 125 252 000 377 001

* THIS MESSAGE WILL BE PRECEDED BY FLAG CHARACTERS AND REPEATED
* THREE TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
* LENGTHS ARE ALSO UTILIZED.

* IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
* USING INTERNAL LOOPBACK (TTLOOP=1).

* TEST 7 <DATA TEST -- BOP XLB CRC-CCITT-0>

* IF XLB IS SPECIFIED IN THE P-TABLE, THIS TEST WILL TRANSMIT &

CVDMEA.P11

10-DEC-80 09:16

PROGRAM DOCUMENT

844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899

: * RECEIVE IN BOP MODE WITH CRC-CCITT-0 ERROR DETECTION THE FOLLOWING
: * SHORT MESSAGE: 125 252 000 377 001
: *
: * THIS MESSAGE WILL BE PRECEDED BY FLAG CHARACTERS AND REPEATED
: * THREE TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
: * LENGTHS ARE ALSO UTILIZED.
: *
: * IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
: * USING INTERNAL LOOPBACK (TTLOOP=1).
: *****

: *****
: * TEST 8 <MODEM CONTROL SIGNAL LOOPBACK TEST>
: *
: * FIRST, THE DMV-11 IS INITIALIZED. THEN, TTL LOOPBACK IS SELECTED,
: * AND THE FOLLOWING CHECKS ARE PERFORMED INVOLVING THE MODEM STATUS
: * REGISTER :
: * - RING, CARRIER, MODEM READY, TEST MODE, CTS ARE CHECKED FOR 1 STATE.
: * - RTS IS DE-ASSERTED AND CTS IS CHECKED FOR 0.
: * - RTS IS ASSERTED AND CTS IS CHECKED FOR 1.
: *
: * NEXT, IF THE OPTION IS AN M8053 WITH AN H3254 TEST CONNECTOR INSTALLED,
: * THE DMV-11 IN INITIALIZED AGAIN, (TTL LOOPBACK IS CLEARED), AND
: * THE FOLLOWING CHECKS ARE PERFORMED :
: * - RING, CARRIER, MODEM READY, CTS ARE CHECKED FOR . TEST MODE IS CHECKED
: * FOR 0.
: * - RTS IS DE-ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 0.
: * - RTS IS ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 1.
: * - DTR IS DE-ASSERTED, AND MODEM READY IS CHECKED FOR 0.
: * - DTR IS ASSERTED, AND MODEM READY IS CHECKED FOR 1.
: *****

: *****
: * TEST 9 <DDCMP MESSAGE TEST>
: *
: * THIS TEST WILL USE XLB IF IT IS ENABLED -- OTHERWISE TTL LOOPBACK
: * WILL BE UTILIZED. THIS ASSURES THAT IT CAN ALWAYS BE RUN AS A
: * GENERAL 'RINGOUT' OF THE M8053.
: *
: * INITIALIZATION: BCP MODE, CRC-16, IDLE = 0, SYNC (S/AR) = 226 OCT.
: * (96 HEX.), RXCL & TXCL = 0 (CHAR. LENGTH = 8).
: *
: * THE FOLLOWING SAMPLE DDCMP MESSAGE IS TRANSMITTED & RECEIVED AND ALL
: * DATA AND CRC CHARACTERS ARE CHECKED FOR ERRORS:
: *
: * ----- HEADER ----- DATA (PATTERN K) -----
: * SYNC SYNC 201 000 075 003 002 001 CRC CRC 000 377 ... 252 000 CRC CRC
: *
: * THE ATTEMPT HERE IS TO PROVIDE A TEST JUST BELOW THE LEVEL OF THE
: * FUNCTIONAL DIAGNOSTIC. THE USYRT WILL BE RESPONSIBLE FOR ALL CRC
: * GENERATION AND VERIFICATION BUT THE CRC'S WILL ALSO BE VERIFIED BY
: * SOFTWARE.
: *****

CVDMEA.P11

10-DEC-80 09:16

PROGRAM DOCUMENT

900

CVDMEA.P11

10-DEC-80 09:16

PROGRAM DOCUMENT

901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942

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.

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH DESCRIBES A 'MASTER CLEAR FAILURE' ERROR, AND PROVIDES THE PC OF THE ERROR CALL AND THE DEVICE REGISTER CONTENTS :

```
CVDMB DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
MASTER CLEAR FAILURE
```

THE CONTENTS OF ALL BYTE SELECT REG'S ARE:

```
BSEL0  BSEL1  BSEL2  BSEL3
000    000    000    000
      BSEL4  BSEL5  BSEL6  BSEL7
      000    000    121    000
BSEL10 BSEL11 BSEL12 BSEL13
000    000    000    000
      BSEL14 BSEL15 BSEL16 BSEL17
      000    000    000    000
```

FOR OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE, AND REQUIRE ADDITIONAL DATA TO BE REPORTED.

IF EXTENDED ERROR INFORMATION HAD BEEN INHIBITED USING THE IXE FLAG PRIOR TO RUNNING THE TEST, THE ABOVE ERROR WOULD HAVE BEEN REPORTED IN THE FOLLOWING SHORTENED FORM :

```
CVDMB DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
MASTER CLEAR FAILURE
```

CVDMEA.P11 10-DEC-80 09:16

GENERAL EQUATES AND DS INVOCATION & SETUP

.SBTTL GENERAL EQUATES AND DS INVOCATION & SETUP

```

943
944
945
946      000000
947
948
949
950
951      002000      .=2000
952
953
954      002000      .MCALL SVC
955                      SVC      ; INITIALIZE SUPERVISOR MACROS
956
957      002000      BGNMOD LU1MOD
958
959
960      000001      $LSTIN= 1
961      000001      $LSTTAG= 1
962      000001      SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
963      000001      SVCTST= 1      ; LIST TEST TAGS, SHIFTED RIGHT
964      000001      SVCSUB= 1      ; LIST SUBTEST TAGS, SHIFTED RIGHT
965      000001      SVCGBL= 1      ; LIST GLOBAL TAGS, SHIFTED RIGHT
966      000001      SVCTAG= 1      ; LIST OTHER TAGS, SHIFTED RIGHT
967
968      ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
969      ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
970      ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
971      ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.

```

CVDMEA.P11 10-DEC-80 09:16

PROGRAM HEADER

```

972
973
974
975
976
977
978 002000
979
980
981 002000
982 002000
983 002000 103
984 002001 126
985 002002 104
986 002003 115
987 002004 105
988 002005 000
989 002006 000
990 002007 000
991 002010
992 002010 101
993 002011
994 002011 060
995 002012
996 002012 000000
997 002014
998 002014 000017
999 002016
1000 002016 033644
1001 002020
1002 002020 000000
1003 002022
1004 002022 002150
1005 002024
1006 002024 000000
1007 002026
1008 002026 034506
1009 002030
1010 002030 000000
1011 002032
1012 002032 000000
1013 002034
1014 002034 000000
1015 002036
1016 002036 000000
1017 002040
1018 002040 002124
1019 002042
1020 002042 000000
1021 002044
1022 002044 000000
1023 002046
1024 002046 000000
1025 002050
1026 002050 003
1027 002051 003
    
```

```

.SBTTL PROGRAM HEADER
:++
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:--
        POINTER BGNAU,BGNDU,ERRTBL

        HEADER .CVDME,A,0,15.,0
    
```

```

LSNAME::
        .ASCII /C/
        .ASCII /V/
        .ASCII /D/
        .ASCII /M/
        .ASCII /E/
        .BYTE 0
        .BYTE 0
        .BYTE 0
LSREV::
        .ASCII /A/
LSDEPO::
        .ASCII /O/
LSUNIT::
        .WORD 0
LSTIML::
        .WORD 15.
LSHPOP::
        .WORD LSHARD
LSSPCP::
        .WORD 0
LSHPTP::
        .WORD LSHW
LSSPTP::
        .WORD 0
LSLADP::
        .WORD LSLAST
LSSSTA::
        .WORD 0
LSCO::
        .WORD 0
LSDTYP::
        .WORD 0
LSAPT::
        .WORD 0
LSDTP::
        .WORD LSDISPATCH
LSPRID::
        .WORD 0
LSEVI::
        .WORD 0
LSEXP1::
        .WORD 0
LSMREV::
        .BYTE CSREVISION
        .BYTE CREDIT
    
```

CVDMEA.P11 10-DEC-80 09:16

PROGRAM HEADER

1028	002052	
1029	002052	000000
1030	002054	000000
1031	002056	
1032	002056	000000
1033	002060	
1034	002060	003264
1035	002062	
1036	002062	000000
1037	002064	
1038	002064	000000
1039	002066	
1040	002066	000000
1041	002070	
1042	002070	023222
1043	002072	
1044	002072	023216
1045	002074	
1046	002074	000000
1047	002076	
1048	002076	003304
1049	002100	
1050	002100	104035
1051	002102	
1052	002102	002172
1053	002104	
1054	002104	022554
1055	002106	
1056	002106	023214
1057	002110	
1058	002110	023070
1059	002112	
1060	002112	022546
1061	002114	
1062	002114	000000
1063	002116	
1064	002116	000000
1065	002120	
1066	002120	000000
1067		
1068		
1069		

.EVEN

LSEF::	.WORD	0
	.WORD	0
LSSPC::	.WORD	0
LSDVVP::	.WORD	LSDVTYP
LSREPP::	.WORD	0
LSEXP4::	.WORD	0
LSEXP5::	.WORD	0
LSAUT::	.WORD	LSAU
LSDUT::	.WORD	LSDU
LSLUN::	.WORD	0
LSDESP::	.WORD	LSDESC
LSLOAD::	EMT	ESLOAD
LSETP::	.WORD	LSERRTBL
LSICP::	.WORD	LSINIT
LSCCP::	.WORD	LSCLEAN
LSACP::	.WORD	LSAUTO
LSPT::	.WORD	LSPROT
LSTEST::	.WORD	0
LSPLY::	.WORD	0
LSHIME::	.WORD	0

CVDMEA.P11 10-DEC-80 09:16

DISPATCH TABLE

.SBTTL DISPATCH TABLE

SLASH

::////////////////////////////////////

::/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.

::/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.

SLASH

::////////////////////////////////////

DISPATCH 9.

.WORD	9
LSDISPATCH::	
.WORD	T1
.WORD	T2
.WORD	T3
.WORD	T4
.WORD	T5
.WORD	T6
.WORD	T7
.WORD	T8
.WORD	T9

1070		
1071		
1072	002122	
1073		
1074		
1075		
1076	002122	
1077		
1078		
1079	002122	
1080	002122	000011
1081	002124	
1082	002124	023224
1083	002126	023524
1084	002130	026230
1085	002132	027106
1086	002134	027650
1087	002136	030412
1088	002140	031074
1089	002142	031556
1090	002144	032522
1091		

CVDMEA.P11 10-DEC-80 09:16

DEFAULT HARDWARE P-TABLE

.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.
:////////////////////

```

```

1092
1093
1094
1095
1096
1097
1098
1099
1100 002146
1101 002146 000010
1102 002150
1103 002150
1104
1105 002150 160020
1106 002152 000300
1107 002154 004000
1108 002156 000000
1109 002160 000000
1110 002162 000000
1111 002164 000000
1112 002166 000001
1113
1114
1115
1116 002170
1117 002170

```

BGNHW DFPTBL

.WORD L10000-LSHW/2

LSHW::
DFPTBL::

```

.WORD 160020
.WORD 300
.WORD 4000
.WORD 000
.WORD 000
.WORD 0
.WORD 0
.WORD 1

```

```

:DMV11 CSR UNIBUS ADDRESS
:DMV11 INTERRUPT VECTOR
:DMV11 INTERRUPT PRIORITY LEVEL = 4
:SWITCH REG. #1 (BOOT ADDRESS)
:SWITCH REG. #2 (DDCMP ADDRESS)
:MODULE IS M8064
:H3254&H3255 USED
:BAUD RATE = 56 K
: 0 = 19.2 K
: 1 = 56 K

```

ENDHW

L10000:

LDMPA.P11 10-DEC-80 09:16

SOFTWARE P-TABLE

.SBTTL SOFTWARE P-TABLE

```

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

```

```

1118
1119
1120
1121
1122
1123
1124
1125 002170
1126 002170 000000
1127 002172
1128 002172
1129
1130 002172
1131 002172

```

BGNSW SFPTBL

```

.LWORD L10001-LSSW/2
LSSW::
SFPTBL::

```

ENDSW

L10001:

CVDMEA.P11 10-DEC-80 09:16

GLOBAL EQUATES SECTION -- BASIC EQUATES

.SBTTL GLOBAL EQUATES SECTION -- BASIC EQUATES

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

EQUALS

:
: BIT DIFINITIONS

1132		
1133		
1134		
1135		
1136		
1137		
1138		
1139		
1140	002172	
1141		
1142		
1143		
1144	100000	BIT15== 100000
1145	040000	BIT14== 40000
1146	020000	BIT13== 20000
1147	010000	BIT12== 10000
1148	004000	BIT11== 4000
1149	002000	BIT10== 2000
1150	001000	BIT09== 1000
1151	000400	BIT08== 400
1152	000200	BIT07== 200
1153	000100	BIT06== 100
1154	000040	BIT05== 40
1155	000020	BIT04== 20
1156	000010	BIT03== 10
1157	000004	BIT02== 4
1158	000002	BIT01== 2
1159	000001	BIT00== 1

1160		
1161	001000	BIT9== BIT09
1162	000400	BIT8== BIT08
1163	000200	BIT7== BIT07
1164	000100	BIT6== BIT06
1165	000040	BIT5== BIT05
1166	000020	BIT4== BIT04
1167	000010	BIT3== BIT03
1168	000004	BIT2== BIT02
1169	000002	BIT1== BIT01
1170	000001	BIT0== BIT00

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

1171			
1172			
1173			
1174			
1175	000040	EF.START== 32.	: START COMMAND WAS ISSUED
1176	000037	EF.RESTART== 31.	: RESTART COMMAND WAS ISSUED
1177	000036	EF.CONTINUE== 30.	: CONTINUE COMMAND WAS ISSUED
1178	000035	EF.NEW== 29.	: A NEW PASS HAS BEEN STARTED
1179	000034	EF.PWR== 28.	: A POWER-FAIL/POWER-UP OCCURRED

:
: PRIORITY LEVEL DEFINITIONS

1180		
1181		
1182		
1183		
1184	000340	PRI07== 340
1185	000300	PRI06== 300
1186	000240	PRI05== 240
1187	000200	PRI04== 200

CVDMEA.P11

10-DEC-80 09:16

GLOBAL EQUATES SECTION -- BASIC EQUATES

1188 000140
1189 000100
1190 000040
1191 000000
1192
1193
1194
1195 000004
1196 000010
1197 000020
1198 000040
1199 000100
1200 000200
1201 000400
1202 001000
1203 002000
1204 004000
1205 010000
1206 020000
1207 040000
1208 100000

PRI03== 140
PRI02== 100
PRI01== 40
PRI00== 0
:
:OPERATOR FLAG BITS
:
EVL== 4
LOT== 10
ADR== 20
IDU== 40
ISR== 100
UAM== 200
BOE== 400
PNT== 1000
PRI== 2000
IXE== 4000
IBE== 10000
TER== 20000
LE== 40000
HOE== 100000

CVDMEA.P11 10-DEC-80 09:16

REGISTER DEFINITIONS -- MAINTENANCE REGISTERS -- SELN & BSELN

.SBTTL REGISTER DEFINITIONS -- MAINTENANCE REGISTERS -- SELN & BSELN

1209
 1210
 1211
 1212
 1213
 1214 000020
 1215 000001
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225 000200
 1226 000100
 1227 000001
 1228 000301
 1229
 1230
 1231
 1232
 1233 000200
 1234
 1235
 1236
 1237
 1238 000001
 1239 000002
 1240 000003
 1241 000004
 1242 000005
 1243 000007
 1244
 1245

```

:*****
:* MAINTENANCE REGISTER # 0 - BSEL0
:*****
IEO      = BIT4      ;'INTERRUPT ENABLE OUT'
IEI      = BIT0      ;'INTERRUPT ENABLE IN'

: BIT 7 IS ALSO USED BY THE MICROCODE. ITS LABEL IS 'RQI' WHICH STANDS FOR
: 'REQUIST IN'. IT'S PART OF THE HANDSHAKING FOR USING THE SEL & BSEL REG'S.
: HOWEVER, THE MAINT. LOOP DOES NOT MAKE USE OF THIS BIT AND IT IS THEREFORE
: UNNECESSARY TO DEFINE IT HERE.

:*****
:* MAINTENANCE REGISTER # 1 - BSEL1
:*****
RUN      = BIT7      ;'RUN' & ALSO CONTROLS 6502 MICROPROCESSOR'S RDY STATE
MCLR     = BIT6      ;M-STER CLEAR
MREQ     = BIT0      ;M-LOOP ACCESS
STRML0P= RUN!MCLR!MREQ ;INITIATE M-LOOP

:*****
:* MAINTENANCE REGISTER # 2 - BSEL2
:*****
MRDY     = BIT7      ;M-LOOP READY

:*****
:* MAINTENANCE LOOP COMMAND DEFINITIONS
:*****
REDLOC   = 1         ;READ LOC. W/IN DMV-11 --- (SEL4) ==> BSEL6
WRIL0C   = 2         ;WRITE LOC. W/IN DMV-11 --- BSEL6 ==> (SEL4)
REDPAG   = 3         ;READ BLOCK W/IN DMV-11 --- (SEL6) ==> (SEL4)
WRIPAG   = 4         ;WRITE BLOCK W/IN DMV-11 --- (SEL4) ==> (SEL6)
EXECUT   = 5         ;SET 6502'S PC AND EXECUTE --- SEL6 ==> PC
DOTBMT   = 7         ;SET MAINTENANCE INTERRUPT DISABLE IN PROCESSOR
;STATUS --- [KB7] ==> BSEL3

```

CVDMEA.P11 10-DEC-80 09:16

REGISTER DEFINITIONS -- USYRT

.SBTTL REGISTER DEFINITIONS -- USYRT

1246
1247
1248
1249 120400
1250

USYRT = 120400 ;USYRT BASE ADDRESS = A100 (HEX)

1251
1252
1253

::*****
:* USYRT 'RECEIVER DATA BUFFER' REGISTER -- READ ONLY
:*****

1254
1255 120400
1256

RDSRL = 120400 ;ADDRESS OF THIS REG

1257
1258
1259

::*****
:* USYRT 'RECEIVER STATUS' REGISTER -- READ ONLY
:*****

1260
1261 120401
1262

RDSRH = 120401 ;ADDRESS OF THIS REG

1263

;BIT DEFINITIONS ON BYTE BASIS :

1264 000200
1265 000160
1266 000010
1267 000004
1268 000002
1269 000001

RERR = BIT7 ;ERROR CHECK
ABC = BIT6!BIT5!BIT4 ;ASSEMBLED BIT COUNT
ROR = BIT3 ;RECEIVER OVER RUN
RABGA = BIT2 ;RECEIVED ABORT/GA CHARACTER
REOM = BIT1 ;RECEIVED END-OF-MESSAGE
RSOM = BIT0 ;RECEIVED START-OF-MESSAGE

1270

;BIT DEFINITIONS ON WORD BASIS :

1271
1272 100000
1273 004000
1274 002000
1275 001000
1276 000400

RXERR = BIT15 ;RECEIVED CRC/VRC ERROR
RXOR = BIT11 ;RECEIVER OVER RUN
RXABGA = BIT10 ;RECEIVED ABORT/GO AHEAD CHARACTER
RXEOM = BIT9 ;RECEIVED END-OF-MESSAGE
RXSOM = BIT8 ;RECEIVED START-OF-MESSAGE

1277
1278 000001

RERCHK = BIT0 ;FLAG TO INVOKE RERR CHK IN SUBROUTINE RXCHAR

1279

::*****
:* USYRT 'TRANSMITTER DATA BUFFER' REGISTER
:*****

1280
1281
1282

1283
1284 120402

TDSRL = 120402 ;ADDRESS OF THIS REG

1285
1286

::*****
:* USYRT 'TX STATUS AND CONTROL' REGISTER
:*****

1287
1288

1289
1290 120403

TDSRH = 120403 ;ADDRESS OF THIS REG

1291
1292

;BIT DEFINITIONS ON BYTE BASIS :

1293 000200
1294 000010
1295 000004
1296 000002
1297 000001

TERR = BIT7 ;TRANSMITTER UNDERRUN ERROR
TGA = BIT3 ;TRANSMIT GO AHEAD
TAB = BIT2 ;TRANSMIT ABORT
TEOM = BIT1 ;TRANSMIT END-OF-MESSAGE
TSOM = BIT0 ;TRANSMIT START-OF-MESSAGE

1298
1299

;BIT DEFINITIONS ON WORD BASIS :

1300 100000
1301 004000

TXERR = BIT15 ;TRANSMITTER UNDERRUN ERROR
TXGA = BIT11 ;TRANSMIT GO AHEAD

CVDMEA.P11 10-DEC-80 09:16

REGISTER DEFINITIONS -- USYRT

```

1302      002000      TXAB   = BIT10      ;TRANSMIT ABORT
1303      001000      TXEOM  = BIT9       ;TRANSMIT END-OF-MESSAGE
1304      000400      TXSOM  = BIT8       ;TRANSMIT START-OF-MESSAGE
1305
1306      ;:*****
1307      ;* USYRT 'SYNC/SECONDARY ADDRESS' REGISTER
1308      ;:*****
1309
1310      120404      PCSARL  = 120404      ;ADDRESS OF THIS REG
1311      000226      SYNCH   = 226        ;STANDARD SYNCH CHARACTER
1312
1313      ;:*****
1314      ;* USYRT 'MODE CONTROL'
1315      ;:*****
1316
1317      120405      PCSARH  = 120405      ;ADDRESS OF THIS REG
1318
1319      ;:BIT DEFINITIONS ON BYTE BASIS:
1320
1321      000200      APA     = BIT7        ;'ALL PARTIES ADDRESS' ENABLE
1322      000100      PROTO  = BIT6        ;SPECIFIES BOP/CCP PROTOCOL -- 0 = BOP
1323      000040      STRIP  = BIT5        ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1324      000020      SECAD  = BIT4        ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1325      000010      IDLE   = BIT3        ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1326      000007      XYZ    = BIT2!BIT1!BIT0 ;CRC/PARITY SELECTION CONTROL
1327
1328      ;:BIT DEFINITIONS ON WORD BASIS:
1329
1330      100000      APAD    = BIT15       ;'ALL PARTIES ADDRESS' ENABLE
1331      040000      DDCMP  = BIT14       ;CODE FOR DDCMP MODE
1332      020000      STRIPS  = BIT13       ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1333      010000      SECADR  = BIT12       ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1334      004000      IDLES  = BIT11       ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1335      000400      CRCOS  = BIT8        ;CODE FOR CRC-CCITT-0 SELECTION
1336      001400      CRC16  = BIT9!BIT8    ;CODE FOR CRC-16 SELECTION
1337      003400      NOCHK  = BIT10!BIT9!BIT8 ;CODE FOR NO ERROR CHECKING
1338      002400      EVRC   = BIT10!BIT8    ;CODE FOR VRC EVEN CHECK
1339      002000      OVRC   = BIT10        ;CODE FOR VRC ODD CHECK
1340
1341      ;:*****
1342      ;* USYRT 'DATA LENGTH SELECT' REGISTER
1343      ;:*****
1344
1345      120407      PCR     = 120407      ;ADDRESS OF THIS REG
1346
1347      ;:BIT DEFINITIONS:
1348
1349      000340      TXDL    = BIT7!BIT6!BIT5 ;TRANSMIT DATA LENGTH SELECTION
1350      000020      EXADD  = BIT4        ;EXTENDED ADDRESS FIELD -- NOT USED OR TESTED
1351      000010      EXCON  = BIT3        ;EXTENDED CONTROL FIELD -- NOT USED OR TESTED
1352      000007      RXDL   = BIT2!BIT1!BIT0 ;RECEIVER DATA LENGTH SELECTION
1353
1354      ;:*****
1355      ;* USYRT STATUS REGISTER (ADDR. A400)
1356      ;:*****
1357      122000      USTATR = 122000      ;USYRT STATUS REGISTER ADDRESS = A400 (HEX)

```

CVDMEA.P11 10-DEC-80 09:16

REGISTER DEFINITIONS -- USYRT

1358
 1359
 1360
 1361 000200
 1362 000100
 1363 000040
 1364 000020
 1365 000010
 1366 000004
 1367 000002
 1368 000001

:BIT DEFINITIONS:

RDA = BIT7 ;RECEIVER DATA AVAILABLE
 TBMT = BIT6 ;TRANSMITTER BUFFER EMPTY
 RXACT = BIT5 ;RECEIVER ACTIVE
 RSA = BIT4 ;RECEIVER STATUS AVAILABLE
 TSO = BIT3 ;TRANSMITTER SERIAL OUTPUT
 TXACT = BIT2 ;TRANSMITTER ACTIVE
 TXU = BIT1 ;TRANSMITTER UNDERRUN
 SFR = BIT0 ;SYNC/FLAG RECEIVED

CVDMEA.P11 10-DEC-80 09:16

REGISTER DEFINITIONS -- 6522 VIA CHIP

.SBTTL REGISTER DEFINITIONS -- 6522 VIA CHIP

1369
1370
1371 120000
1372
1373
1374
1375
1376
1377 120000
1378
1379 000200
1380 000100
1381 000040
1382 000020
1383 000010
1384 000004
1385 000002
1386 000001
1387 000000
1388
1389
1390
1391
1392
1393 120001
1394
1395 000200
1396 000100
1397 000040
1398 000020
1399 000010
1400 000004
1401 000002
1402 000001
1403
1404
1405
1406
1407
1408
1409 120002
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421 120003
1422
1423
1424

VIA = 120000 ;VIA BASE ADDRESS = A000 (HEX)
:*****
:* MODEM & MAINTENANCE CONTROL -- 'DRB' 8 BIT PORT B -- WRITE ONLY
:*****
VIAORB = 120000 ;ADDRESS OF THIS REGISTER -- HEX = A0X0
MULCLK = BIT7 ;'NULL CLK L' -- NULL CLOCK
RXEN = BIT6 ;'RXENL' -- USYRT RECEIVER ENABLE
TXEN = BIT5 ;'TXENL' -- USYRT TRANSMITTER ENABLE
DTR = BIT4 ;'DTR' -- DATA TERMINAL READY
RTSND = BIT3 ;'RTSND' -- REQUEST TO SEND
HDX = BIT2 ;'HDX' -- HALF DUPLEX
TTLOOP = BIT1 ;'SELECT TTL LEVEL LOOPBACK'
PRESET = BIT0 ;'PRESET H' --
DTRL = 0 ;DTR IS ASSERTED LOW
:*****
:* MODEM STATUS REGISTER -- 'DRA' 8 BIT PORT A -- READ ONLY
:*****
VIAMS = 120001 ;ADDRESS OF THIS REGISTER -- HEX = A0X1
RING = BIT7 ;'RING H' --
CARRIER = BIT6 ;'CARRIER H' --
MDMRDY = BIT5 ;'MODEM RDY H' --
SPEED = BIT4 ;'BAUD RATE SWITCH -- (19.2K/56K)
CTS = BIT3 ;'CTS H -- CLEAR TO SEND
TM = BIT2 ;'TEST MODE H' --
RCVDAT = BIT1 ;'RCV DATA H' --
UMAINI = BIT0 ; SELECT USYRT INT LOOPBACK **SELECT BIT**
:*****
:* DATA DIRECTION FOR PORT B -- 'DDRB' -- READ/WRITE
:*****
VIADPB = 120002 ;ADDRESS OF THIS REGISTER -- HEX = A0X2
: ALL BITS ARE DEFINED THE SAME:
: THE BIT SETTING DEFINED THE DIRECTION OF ITS RELATED BIT IN BIT PORT B
:
: INITIALIZED TO 377 (HEX = FF) -- PORT B IS READ/WRITE
:*****
:* DATA DIRECTION FOR PORT A -- 'DDRA' -- READ/WRITE
:*****
VIADPA = 120003 ;ADDRESS OF THIS REGISTER -- HEX = A0X3
: ALL BITS ARE DEFINED THE SAME:
: THE BIT SETTING DEFINED THE DIRECTION OF ITS RELATED BIT IN BIT PORT A

CVDMEA.P11

10-DEC-80 09:16

REGISTER DEFINITIONS -- 6522 VIA CHIP

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
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480

120004

: INITIALIZED TO 001 (HEX = 01) -- PORT A IS READ ONLY (EXCEPT FOR
: BIT0 WHICH ENABLES USYRT INTERNAL LOOPBACK).

::*****
:* TIMER 1 LOW ORDER (LATCH & COUNTER) -- 'T1L-L' & 'T1C-L' -- WRITE & READ
:*****

VIAT1A = 120004 ;ADDRESS OF THIS REGISTER -- HEX = A0X4

: WHEN WRITING, LOW ORDER LATCH IS LOADED.
: WHEN READING, LOW ORDER COUNTER IS READ.

::*****
:* TIMER 1 HIGH ORDER COUNTER & TRIGGER -- 'T1L-H AND TRIGGER' & 'T1C-H'
:* -- WRITE & READ
:*****

120005

VIAT1B = 120005 ;ADDRESS OF THIS REGISTER -- HEX = A0X5

: WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
: ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.

: WHEN READING, THE HIGH ORDER COUNTER IS READ.

::*****
:* TIMER 1 LOW ORDER LATCH -- 'T1L-L' -- READ/WRITE
:*****

120006

VIAT1C = 120006 ;ADDRESS OF THIS REGISTER -- HEX = A0X6

: THE LOW ORDER LATCH IS READ OR LOADED. THIS LATCH IS USED TO LOAD THE
: COUNTER WHEN TIMODE (IN VIAACR) = 3

::*****
:* TIMER 1 HIGH ORDER LATCH -- 'T1L-H' -- READ/WRITE
:*****

120007

VIAT1D = 120007 ;ADDRESS OF THIS REGISTER -- HEX = A0X7

: THE HIGH ORDER LATCH IS READ OR LOADED. THIS LATCH IS USED TO LOAD THE
: COUNTER WHEN TIMODE (IN VIAACR) = 3

::*****
:* TIMER 2 LOW ORDER (LATCH & COUNTER) -- 'T2L-L' & 'T2C-L' -- WRITE & READ
:*****

CVDMEA.P11

10-DEC-80 09:16

REGISTER DEFINITIONS -- 6522 VIA CHIP

1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536

120010

VIAT2A = 120010 ;ADDRESS OF THIS REGISTER -- HEX = A0X8

: WHEN WRITING, LOW ORDER LATCH IS LOADED.
: WHEN READING, LOW ORDER COUNTER IS READ.

::*****
:* TIMER 2 HIGH ORDER COUNTER & TRIGGER -- 'T2L-H AND TRIGGER' & 'T2C-H'
:* -- WRITE & READ
:*****

120011

VIAT2B = 120011 ;ADDRESS OF THIS REGISTER -- HEX = A0X9

: WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
: ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.

: WHEN READING, THE HIGH ORDER COUNTER IS READ.

::*****
:* SHIFT REGISTER -- 'SR' -- READ/WRITE
:*****

120012

VIASR = 120012 ;ADDRESS OF THIS REGISTER -- HEX = A0XA

: SHIFTING IS CONTROLLED BY THE SETTING OF VIASRC (ACR2 ---> ACR4) IN VIAACR

::*****
:* AUXILIARY CONTROL REGISTER -- 'ACR' -- READ/WRITE
:*****

120013

VIAACR = 120013 ;ADDRESS OF THIS REGISTER -- HEX = A0XB

000300

T1MODE = BIT7!BIT6 ;CONTROL THE MODE OF TIMER # 1

:BIT 7:
: 0 PB7 DISABLED -- ONLY T1TO IN VIAIFR REFLECTS TIMEOUT
: 1 PB7 & T1TO REFLECT TIMEOUT

:BIT 6:
: 0 TIMER 1 IN ONE-SHOT MODE
: 1 TIMER 1 IN CONTINUOUS SQUARE WAVE MODE

000040

T2MODE = BIT5 ;CONTROLS THE MODE OF TIMER # 1

: 0 PULSE COUNTING MODE
: 1 INTERVAL TIMER MODE

000034

SRMODE = BIT4!BIT3!BIT2 ;CONTROLS THE MODE OF THE SHIFT REGISTER

: 0 SR DISABLED
: 1 SHIFT IN UNDER CONTROL OF T2, SHFT PULSES GEN'D ON CB1
: 2 SHIFT IN AT SYS. CLOCK RATE, SHFT PULSES GEN'D ON CB1

CVDMEA.P11 10-DEC-80 09:16

REGISTER DEFINITIONS -- 6522 VIA CHIP

```

1537                                     : 3  SHIFT IN UNDER CONTROL OF EXTERNAL INPUT PULSES
1538                                     : 4  SHIFT OUT -- FREE RUNNING -- RATE CONTROLLED BY T2
1539                                     : 5  SHIFT OUT -- RATE CONTROLLED BY T2 -- PULSES ON CB1
1540                                     : 6  SHIFT OUT -- SYS. CLOCK RATE -- PULSES ON CB1
1541                                     : 7  SHIFT OUT -- UNDER CONTROL OF PULSES APPLIED TO CB1
1542
1543      000002      PBLNGB = BIT1          ;PB LATCH CONTROL -- 1 ENABLES LATCH
1544      000001      PALENB = BIT0        ;PA LATCH CONTROL -- 1 ENABLES LATCH
1545
1546
1547
1548
1549      ;:*****
1550      ;* PERIPHERAL CONTROL REGISTER -- 'PCR' -- READ/WRITE
1551      ;:*****
1552
1553      120014      VIAPCR = 120014        ;ADDRESS OF THIS REGISTER -- HEX = A0XC
1554
1555      000340      CB2CTL = BIT7!BIT6!BIT5 ;CB2 MODE SELECT
1556      000020      CB1CTL = BIT4        ;CB1 MODE SELECT
1557      000016      CA2CTL = BIT3!BIT2!BIT1 ;CA2 MODE SELECT
1558      000001      CA1CTL = BIT0        ;CA1 MODE SELECT
1559
1560
1561
1562      ;:*****
1563      ;* INTERRUPT FLAG REGISTER -- 'IFR' -- READ ONLY
1564      ;:*****
1565
1566      120015      VIAIFR = 120015       ;ADDRESS OF THIS REGISTER -- HEX = A0XD
1567
1568      000200      FLGIRQ = BIT7         ;SET WHEN A FLAG IN THIS REG. GOES HIGH AND
1569      ;:ITS CORRESPONDING BIT IN VIAIER IS SET.
1570      ;:(I.E. VIAIER IS THE ENABLE REGISTER FOR THE
1571      ;:FOR THE SETTING OF IRQ AND THE ISSUANCE OF
1572      ;:AN INTERRUPT TO THE 6502 WHEN IRQ IS SET.)
1573
1574      000100      FLGT1 = BIT6          ;:TIMEOUT OF TIMER 1
1575      000040      FLGT2 = BIT5          ;:TIMEOUT OF TIMER 2
1576      000020      FLGCB1 = BIT4        ;:ACTIVE TRANSITION OF PIN 18 (CB1)
1577      000010      FLGCB2 = BIT3        ;:ACTIVE TRANSITION OF PIN 19 (CB2)
1578      000004      FLGSR = BIT2         ;:COMPLETION OF 8 SHIFTS
1579      000002      FLGCA1 = BIT1        ;:ACTIVE TRANSITION OF PIN 40 (CA1)
1580      000001      FLGCA2 = BIT0        ;:ACTIVE TRANSITION OF PIN 39 (CA2)
1581
1582
1583
1584      ;:*****
1585      ;* INTERRUPT ENABLE REGISTER -- 'IER' -- READ/WRITE
1586      ;:*****
1587
1588      120016      VIAIER = 120016       ;ADDRESS OF THIS REGISTER -- HEX = A0XE
1589
1590      000200      INTSC = BIT7         ;:CONTROLS THE SETTING OR CLEARING OF BITS IN
1591      ;:THE REST OF IER. IF = 0 THE OTHER BITS IN
1592      ;:THIS REG., IF SET, WILL CLEAR THEIR RESPECTIVE

```

CVL-2A.P11 10-DEC-80 09:16

REGISTER DEFINITIONS -- 6522 VIA CHIP

1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617

;BITS IN THE INT. ENAB. REG.. IF = 1, THE
;RESPECTIVE BITS WILL BE SET.

; WHEN WRITING THIS REG., THE COMMENT ABOVE HOLDS.
; WHEN READING THIS REG., THE CURRENT STATE OF THE INT. ENABLE REG. IS RETURNED.
; THE BIT ASSIGNMENTS ARE THE SAME AS FOR VIAIFR AS DEFINED ABOVE.

::*****
;* OUTPUT REGISTER A -- 'ORA' -- READ ONLY (OR READ/WRITE UNDER CONTROL OF 'DDPA')
;*****

120017

VIAORA = 120017 ;ADDRESS OF THIS REGISTER -- HEX = A0XF
; THIS ADDRESS ACCESSES THE SAME DATA AS 'VIAMS' EXCEPT THAT NO 'HANDSHAKING'
; WILL TAKE PLACE (I.E. THERE IS NO CHANGE IN IRQ OR CA2 AS A RESULT OF
; READING ORA THROUGH THIS ADDRESS)
;THE BIT ASSIGNMENTS ARE THE SAME AS FOR 'VIAMS' ABOVE.

CVDMEA.P11 10-DEC-80 09:16

REGISTER DEFINITIONS -- MISC

.SBTTL REGISTER DEFINITIONS -- MISC

```

:*****
:* SWITCH PACKS
:*****
    
```

1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643

121000
121400

100000
001000

000002
000001

040000
001000

000200

100000
040000
020000

```

SWPBOT = 121000          ;'BOOT ADDRESS'' SWITCH PACK [A200]
SWPDDCMP = 121400       ;'DDCMP ADDRESS'' SWITCH PACK [A300]

;MISCELLANEOUS EQUATES

TCCHEK = BIT15          ;FLAG TO REQUEST H3254,5 CHECK
RAMADR = 001000         ;STARTING ADRS OF RAM PAGE 2 (ADRS 0200 HEX)

EIAV35 = BIT1           ;SELECT V.35 OR EIA 423/232C
INTGRL = BIT0           ;SELECT INTEGRAL MODEM

NORXEN = BIT14          ;KILL RXEN DURING ''INITRN''
NOLOOP = BIT9           ;KILL TTLOOP DURING ''INITRN''

NCTBMT = BIT7           ;DISABLE INITIAL TBMT=0 CHECK IN TXCHAR

NOCRDA = BIT15          ;DISABLE INITIAL RDA=0 CHECK IN RXCHAR
NFCRDA = BIT14          ;DISABLE FINAL RDA=1 CHECK IN RXCHAR
NCRACT = BIT13          ;DISABLE RXACT=1 CHECK AFTER CLOCKING (RXCHAR)
    
```

CVDMEA.P11 10-DEC-80 09:16

GLOBAL DATA SECTION

1644
 1645
 1646
 1647
 1648
 1649
 1650
 1651
 1652
 1653
 1654
 1655 002172
 1656 002172
 1657 002172 000000
 1658 002174 000000
 1659 002176 000000
 1660 002200 000000
 1661
 1662
 1663
 1664
 1665 002202
 1666 002202 000000
 1667 002204
 1668 002204 000000
 1669 002206
 1670 002206 000000
 1671 002210
 1672 002210 000000
 1673 002212
 1674 002212 000000
 1675 002214
 1676 002214 000000
 1677 002216
 1678 002216 000000
 1679 002220
 1680 002220 000000
 1681 002222 000000
 1682 002224 000000
 1683 002226 000000
 1684 002230 000000
 1685 002232 000000
 1686 002234 000000
 1687 002236 000000
 1688 002240 000000
 1689
 1690 002242 000010
 1691
 1692
 1693 002262 000020

.SBTTL GLOBAL DATA SECTION

```

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

```

```

:*****
: CONTROL BLOCK FOR STACKED ERROR MESSAGES
:-----*****

```

ERRTBL

LSERRTBL::

```

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

```

```

:*****
:* STORAGE FOR DEVICE REGISTERS
:*****

```

```

WSR0: ;STORAGE FOR DEVICE CSR REGISTERS
BSR0: .WORD 0
WSR2:
BSR1: .WORD 0
WSR4:
BSR2: .WORD 0
WSR6:
BSR3: .WORD 0
WSR10:
BSR4: .WORD 0
WSR12:
BSR5: .WORD 0
WSR14:
BSR6: .WORD 0
WSR16:
BSR7: .WORD 0
BSR10: .WORD 0
BSR11: .WORD 0
BSR12: .WORD 0
BSR13: .WORD 0
BSR14: .WORD 0
BSR15: .WORD 0
BSR16: .WORD 0
BSR17: .WORD 0

```

UREGS: .BLKW 8.

```

:THE FIRST 7 ARE FOR THE USYRT'S ACTUAL
:REGISTERS. THE LAST ONE IS FOR THE STATUS
:REG. (USTATR).
:STORAGE FOR VIA REGISTERS FOR PRINTOUT

```

VREGS: .BLKW 16.

CVDMEA.P11 10-DEC-80 09:16

GLOBAL DATA SECTION

```

1694
1695
1696
1697 002322 000000
1698 002324 000000
1699 002326 000000
1700 002330 000000
1701 002332 000000
1702 002334 000000
1703 002336 000000
1704 002340 000000
1705 002342 000000
1706 002344 000000
1707 002346 000000
1708
1709 002350 000000
1710 002352 000000
1711 002354 000000
1712 002356 000000
1713 002360 000000
1714 002362 000000
1715 002364 000000
1716 002366 000000
1717 002370 000000
1718 002372 000000
1719 002374 000000
1720 002376 000000
1721 002400 000000
1722 002402 000000
1723 002404 000000
1724 002406 000000
1725 002410 000000
1726 002412 000000
1727 002414 000000
1728

:*****
:* MISCELLANEOUS STORAGE
:*****
TDATA: .WORD 0 ;TEST DATA
GDATA: .WORD 0 ;GOOD DATA
BDATA: .WORD 0 ;BAD DATA
XDATA: .WORD 0 ;EXCLUSIVE-OR BETWEEN GOOD AND BAD DATA
SCRACH: .WORD 0 ;GEN'L PURPOSE SCRATCH WORD
LOGDEV: .WORD 0 ;LOGICAL DEVICE NUMBER
REGNUM: .WORD 0 ;CONTAINS A DEVICE REGISTER NUMBER
PSTACK: .WORD 0 ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
PRIOR: .WORD 0 ;CPU PRIORITY FOR PRINTOUT
SUBRPC: .WORD 0 ;PC OF SUBR CALL FOR ERROR REPORTS
INTFLG: .WORD 0 ;INTERRUPT RECEIVED FLAGS
; BIT 0 FOR TX, BIT 1 FOR RCV
ERRFLG: .WORD 0 ;SUBROUTINE ERROR FLAG
TIMFLG: .WORD 0 ;EVENT TIME-OUT FLAG
RETADR: .WORD 0 ;SUBR ERROR RETURN ADDRESS
REDBYT: .WORD 0 ;LO BYTE CONTAINS BYTE READ FROM LU REG
WRIBYT: .WORD 0 ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
LOADAT: .WORD 0 ;CONTAINS TEST DATA LOADED INTO REG
GOODAT: .WORD 0 ;STORAGE FOR EXPECTED DATA
BADDAT: .WORD 0 ;STORAGE FOR ACTUAL DATA
FRSTIM: .WORD 0 ;FLAG=0 IF PROGRAM JUST LOADED
SAVE4: .WORD 0 ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
SAVE6: .WORD 0 ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
ERROR1: .WORD 0 ;SUBR ERR. BIT FLAGS (DEF'D IN GLOBAL EQUATES)
CHPTYP: .WORD 0 ;USYRT CHIP TYPE, =0 FOR SMC, ELSE =1
SAVLEN: .WORD 0 ;SAVED TX AND RCV CHAR LENGTHS
DEVMAP: .WORD 0 ;BIT MAP OF ACTIVE DEVICES
DEVPTR: .WORD 0 ;DEVICE MAP BIT POINTER
UNIT: .WORD 0 ;CONTAINS UNIT NO. (1 TO N)
STARES: .WORD 0 ;FLAG TO SHOW NO. OF PASSES SINCE STA OR RES
TSTNUM: .WORD 0 ;NO. OF CURRENT TEST (FOR SOME TESTS)

```


CVDMEA.P11 10-DEC-80 09:16

GLOBAL DATA SECTION

```

1729                                     :***** CURRENT DEVICE PARAMETERS *****
1730 002416 BSEL0:
1731 002416 SEL0:
1732 002416 160020 MPCSR: .WORD 160020 ;POINTER TO DMV11 CSR'S
1733 002420 160021 BSEL1: .WORD 160021 ;POINTER TO BSEL1
1734 002422 BSEL2:
1735 002422 160022 SEL2: .WORD 160022 ;POINTER TO SEL2
1736 002424 160023 BSEL3: .WORD 160023 ;POINTER TO BSEL3
1737 002426 BSEL4:
1738 002426 160024 SEL4: .WORD 160024 ;POINTER TO SEL4
1739 002430 160025 BSEL5: .WORD 160025 ;POINTER TO BSEL5
1740 002432 BSEL6:
1741 002432 160026 SEL6: .WORD 160026 ;POINTER TO SEL6
1742 002434 160027 BSEL7: .WORD 160027 ;POINTER TO BSEL7
1743 002436 BSEL10:
1744 002436 160030 SEL10: .WORD 160030 ;POINTER TO SEL10
1745 002440 160031 BSEL11: .WORD 160031 ;POINTER TO BSEL11
1746 002442 BSEL12:
1747 002442 160032 SEL12: .WORD 160032 ;POINTER TO SEL12
1748 002444 160033 BSEL13: .WORD 160033 ;POINTER TO BSEL13
1749 002446 BSEL14:
1750 002446 160034 SEL14: .WORD 160034 ;POINTER TO SEL14
1751 002450 160035 BSEL15: .WORD 160035 ;POINTER TO BSEL15
1752 002452 BSEL16:
1753 002452 160036 SEL16: .WORD 160036 ;POINTER TO SEL16
1754 002454 160037 BSEL17: .WORD 160037 ;POINTER TO BSEL17
1755
1756 002456 000300 MPIVEC: .WORD 300 ;DMV11 INPUT INTERRUPT VECTOR
1757 002460 000304 MPOVEC: .WORD 304 ;DMV11 OUTPUT INTERRUPT VECTOR
1758 002462 000240 MPRIOR: .WORD 240 ;DMV11 DEVICE PRIORITY
1759 002464 000000 LUSWI1: .WORD 0 ;LINE UNIT SWITCH PACK #1
1760 002466 000000 LUSWI2: .WORD 0 ;LINE UNIT SWITCH PACK #2
1761 002470 000000 BRDTYP: .WORD 0 ;0=M8064, 1=M8053/V.35, 2=M8053/EIA
1762 002472 000000 TSTCON: .WORD 0 ;TEST CONNECTOR INDICATOR
1763 002474 000001 BDRATE: .WORD 1 ;BAUD RATE = 56 K
1764 ; ; 0 = 19.2 K
1765 ; ; 1 = 56 K

```

CVDMEA.P11 10-DEC-80 09:16

GLOBAL DATA SECTION

```

1766
1767 002476 120400
1768 002500 120401
1769 002502 120402
1770 002504 120403
1771 002506 120404
1772 002510 120405
1773 002512 120407
1774 002514 122000
1775
1776
1777 002516 000010
1778
1779
1780 002526 000000
1781 002527 000000
1782 002532 000000
1783 002534 000000
1784 002536 000000
1785 002540 000000
1786 002542 000000
1787 002544 000000
1788
1789
1790 002546 000000
1791 002550 000000
1792 002552 000000
1793 002554 000000
1794 002556 000000
1795 002560 000000
1796 002562 000000
1797 002564 000000
1798
1799
1800 002566
1801 002566 377
1802 002567 000
1803 002570 000
1804 002571 360
1805 002572 000
1806 002573 000
1807 002574 347
1808
1809 002575 200

```

```

;TABLE OF USYRT REGISTER ADDRESSES
USYREG: .WORD 120400 ;ADDRESS OF RDSRL
        .WORD 120401 ;ADDRESS OF RDSRH
        .WORD 120402 ;ADDRESS OF TDSRL
        .WORD 120403 ;ADDRESS OF TDSRH
        .WORD 120404 ;ADDRESS OF PCSARL
        .WORD 120405 ;ADDRESS OF PCSARH
        .WORD 120407 ;ADDRESS OF PCR
        .WORD 122000 ;ADDRESS OF USYRT STATUS REG

;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
REDDAT: .BLKB 8.

;***** GEN'L PURPOSE SCRATCH STORAGE *****
REG0: .WORD 0
REG1: .WORD 0
REG2: .WORD 0
REG3: .WORD 0
REG4: .WORD 0
REG5: .WORD 0
REG6: .WORD 0
REG7: .WORD 0

;***** SCRATCH STORAGE FOR MESSAGE REPORTING *****
TMP0: .WORD 0
TMP1: .WORD 0
TMP2: .WORD 0
TMP3: .WORD 0
TMP4: .WORD 0
TMP5: .WORD 0
TMP6: .WORD 0
TMP7: .WORD 0

;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
UPBITS: .BYTE 377 ;MASK FOR RDBR
        .BYTE 000 ;MASK FOR RDSR
        .BYTE 000 ;MASK FOR TDBR
        .BYTE 360 ;MASK FOR TDSR
        .BYTE 000 ;MASK FOR SSAR
        .BYTE 000 ;MASK FOR PCSAR
        .BYTE 347 ;MASK FOR PCR

TDSRNRW: .BYTE 200 ;TDSR NON-R/W BITS

```

CVDMEA.P11 10-DEC-80 09:16

DATA TEST PATTERNS

1810		
1811		
1812	002576	
1813	002576	377
1814	002577	377
1815	002600	377
1816	002601	377
1817	002602	377
1818	002603	377
1819	002604	377
1820	002605	366
1821		
1822		
1823	002606	
1824	002606	000
1825	002607	000
1826	002610	000
1827	002611	000
1828	002612	000
1829	002613	000
1830	002614	000
1831	002615	110
1832		
1833		
1834	002616	
1835	002616	000
1836	002617	001
1837	002620	003
1838	002621	004
1839	002622	005
1840	002623	007
1841	002624	100
1842	002625	101
1843	002626	103
1844	002627	104
1845	002630	105
1846	002631	107
1847	002632	000
1848	002633	017
1849	002634	027
1850	002635	041
1851	002636	200
1852	002637	277
1853	002640	103
1854	002641	144
1855	002642	115
1856	002643	157
1857	002644	000
1858		
1859		
1860	002645	
1861	002645	125
1862	002646	252
1863	002647	000
1864	002650	377
1865	002651	001

```
.SBTTL DATA TEST PATTERNS
:***** DATA PATTERN E *****
PATE:
```

```
.BYTE 377
.BYTE 377
.BYTE 377
.BYTE 377
.BYTE 377
.BYTE 377
.BYTE 377
.BYTE 377
.BYTE 366
```

```
:***** DATA PATTERN F *****
PATF:
```

```
.BYTE 000
.BYTE 000
.BYTE 000
.BYTE 000
.BYTE 000
.BYTE 000
.BYTE 000
.BYTE 000
.BYTE 110
```

```
:***** DATA PATTERN G *****
PATG:
```

```
.BYTE 000
.BYTE 001
.BYTE 003
.BYTE 004
.BYTE 005
.BYTE 007
.BYTE 100
.BYTE 101
.BYTE 103
.BYTE 104
.BYTE 105
.BYTE 107
.BYTE 000
.BYTE 017
.BYTE 027
.BYTE 041
.BYTE 200
.BYTE 277
.BYTE 103
.BYTE 144
.BYTE 115
.BYTE 157
.BYTE 000
```

```
:***** DATA PATTERN X *****
PATX:
```

```
.BYTE 125
.BYTE 252
.BYTE 000
.BYTE 377
.BYTE 001
```

CVDMEA.P11 10-DEC-80 09:16

DATA TEST PATTERNS

1866	002652	002	.BYTE	002
1867	002653	004	.BYTE	004
1868	002654	010	.BYTE	010
1869	002655	020	.BYTE	020
1870	002656	040	.BYTE	040
1871	002657	100	.BYTE	100
1872	002660	200	.BYTE	200
1873	002661	376	.BYTE	376
1874	002662	375	.BYTE	375
1875	002663	373	.BYTE	373
1876	002664	367	.BYTE	367
1877	002665	357	.BYTE	357
1878	002666	337	.BYTE	337
1879	002667	277	.BYTE	277
1880	002670	177	.BYTE	177
1881				
1882	002671	125	.BYTE	125
1883	002672	252	.BYTE	252
1884	002673	000	.BYTE	000
1885	002674	377	.BYTE	377
1886	002675	001	.BYTE	001
1887	002676	002	.BYTE	002
1888	002677	004	.BYTE	004
1889	002700	010	.BYTE	010
1890	002701	020	.BYTE	020
1891	002702	040	.BYTE	040
1892	002703	100	.BYTE	100
1893	002704	200	.BYTE	200
1894	002705	376	.BYTE	376
1895	002706	375	.BYTE	375
1896	002707	373	.BYTE	373
1897	002710	367	.BYTE	367
1898	002711	357	.BYTE	357
1899	002712	337	.BYTE	337
1900	002713	277	.BYTE	277
1901	002714	177	.BYTE	177
1902				
1903	002715	125	.BYTE	125
1904	002716	252	.BYTE	252
1905	002717	000	.BYTE	000
1906	002720	377	.BYTE	377
1907	002721	001	.BYTE	001
1908	002722	002	.BYTE	002
1909	002723	004	.BYTE	004
1910	002724	010	.BYTE	010
1911	002725	020	.BYTE	020
1912	002726	040	.BYTE	040
1913	002727	100	.BYTE	100
1914	002730	200	.BYTE	200
1915	002731	376	.BYTE	376
1916	002732	375	.BYTE	375
1917	002733	373	.BYTE	373
1918	002734	367	.BYTE	367
1919	002735	357	.BYTE	357
1920	002736	337	.BYTE	337
1921	002737	277	.BYTE	277

CVDM:EA.P11 10-DEC-80 09:16

DATA TEST PATTERNS

1922	002740	177
1923		
1924		
1925	002741	
1926	002741	000
1927	002742	041
1928	002743	102
1929	002744	143
1930	002745	204
1931	002746	245
1932	002747	306
1933	002750	347
1934	002751	000
1935	002752	001
1936	002753	002
1937	002754	004
1938	002755	040
1939	002756	100
1940	002757	200
1941	002760	000
1942	002761	346
1943	002762	345
1944	002763	343
1945	002764	307
1946	002765	247
1947	002766	147
1948	002767	347
1949	002770	242
1950	002771	105
1951	002772	347
1952	002773	010
1953	002774	020
1954	002775	367
1955	002776	357
1956	002777	030
1957	003000	027
1958	003001	377
1959		
1960		
1961	003002	
1962	003002	000
1963	003003	000
1964	003004	001
1965	003005	002
1966	003006	004
1967	003007	020
1968	003010	040
1969	003011	010
1970		
1971		
1972	003012	
1973	003012	000
1974	003013	377
1975	003014	376
1976	003015	375
1977	003016	373

EPATX: .BYTE 177

;***** DATA PATTERN I *****

PATI:

.BYTE	000
.BYTE	041
.BYTE	102
.BYTE	143
.BYTE	204
.BYTE	245
.BYTE	306
.BYTE	347
.BYTE	000
.BYTE	001
.BYTE	002
.BYTE	004
.BYTE	040
.BYTE	100
.BYTE	200
.BYTE	000
.BYTE	346
.BYTE	345
.BYTE	343
.BYTE	307
.BYTE	247
.BYTE	147
.BYTE	347
.BYTE	242
.BYTE	105
.BYTE	347
.BYTE	010
.BYTE	020
.BYTE	367
.BYTE	357
.BYTE	030
.BYTE	027
.BYTE	377

;***** DATA PATTERN J *****

PATJ:

.BYTE	000
.BYTE	000
.BYTE	001
.BYTE	002
.BYTE	004
.BYTE	020
.BYTE	040
.BYTE	010

;***** DATA PATTERN K *****

PATK:

.BYTE	000
.BYTE	377
.BYTE	376
.BYTE	375
.BYTE	373

CVDMEA.P11 10-DEC-80 09:16

DATA TEST PATTERNS

1978	003017	376	.BYTE	376
1979	003020	177	.BYTE	177
1980	003021	377	.BYTE	377
1981	003022	000	.BYTE	000
1982	003023	001	.BYTE	001
1983	003024	002	.BYTE	002
1984	003025	004	.BYTE	004
1985	003026	010	.BYTE	010
1986	003027	200	.BYTE	200
1987	003030	125	.BYTE	125
1988	003031	252	.BYTE	252
1989	003032	000	.BYTE	000

***** DATA PATTERN L *****
 PATL:

1991				
1992	003033			
1993	003033	000	.BYTE	000
1994	003034	017	.BYTE	017
1995	003035	016	.BYTE	016
1996	003036	015	.BYTE	015
1997	003037	013	.BYTE	013
1998	003040	016	.BYTE	016
1999	003041	017	.BYTE	017
2000	003042	017	.BYTE	017
2001	003043	000	.BYTE	000
2002	003044	001	.BYTE	001
2003	003045	002	.BYTE	002
2004	003046	004	.BYTE	004
2005	003047	010	.BYTE	010
2006	003050	000	.BYTE	000
2007	003051	005	.BYTE	005
2008	003052	012	.BYTE	012
2009	003053	000	.BYTE	000

CVDMEA.P11 10-DEC-80 09:16

DATA TEST PATTERNS

2010
2011
2012 003054 000
2013 003055 002
2014 003056 014
2015 003057 060
2016 003060 001
2017 003061 007
2018 003062 037
2019 003063 177
2020
2021
2022 003064
2023

:***** DATA PATTERN Q *****
PATQ: .BYTE 000
.BYTE 002
.BYTE 014
.BYTE 060
.BYTE 001
.BYTE 007
.BYTE 037
.BYTE 177

ENDPAT:
.EVEN

CVDMEA.P11 10-DEC-80 09:16

DATA TEST PATTERNS

2024
2025
2026
2027
2028 003064 000100
2029
2030
2031
2032

:*** RECEIVED DATA BUFFER (64. WORDS) ***
RCVBUF: .BLKW 64.



CVDMEA.P11 10-DEC-80 09:16

GLOBAL TEXT SECTION

2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068

003264				
003264				
003264	034115	032460	020063	
003272	051117	046440	030070	
003300	032066	000		
	003304			
003304	000012			
003304				
003304	046504	026526	030461	
003312	046040	047111	020105	
003320	047125	052111	052040	
003326	051505	051524	026440	
003334	050040	051101	020124	
003342	020063	043117	031440	
003350	000			
	003352			
	000010			

```

.SBTTL GLOBAL TEXT SECTION
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:% THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
:% MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
:% MORE THAN ONE TEST.
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
:*****
:* NAMES OF DEVICES SUPPORTED BY PROGRAM
:*****
DEV TYP <M8053 OR M8064>
LSDVTYP::
.ASCIZ /M8053 OR M8064/
.EVEN
:*****
:* TITLE OF PROGRAM
:*****
.RADIX 10.
DESCRIPT <DMV-11 LINE UNIT TESTS - PART 3 OF 3>
LSDDESC::
.ASCIZ /DMV-11 LINE UNI
.EVEN
.RADIX 8.

```

CVDMEA.P11 10-DEC-80 09:16

GLOBAL SUBROUTINE SECTION

.SBTTL GLOBAL SUBROUTINE SECTION

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
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114

```

.SBTTL ....M-LOOP -- MSTCLR -- MASTER CLEAR AND ENTER M-LOOP
*****
MSTCLR -- MASTER CLEAR & ENTER M-LOOP

CALLING SEQUENCE:

      JSR      PC,MSTCLR
      BCC     NS          ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
      ERROR   ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
      <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>

NS:   <RESUMPTION OF NORMAL PROCESSING>
*****

```

```

2090 003352 112777 000301 177040 MSTCLR: MOVB  #RUN!MCLR!MREQ,@BSEL1 ;INITIATE M-LOOP
2091
2092 003360 010346          MOV    R3,-(SP)
2093 003362 012703 000030          MOV    #24,R3          ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
2094 003366 077301          SOB   R3,1$
2095 003370 012603          MOV    (SP)+,R3
2096
2097 003372 132777 000200 177022          BITB  #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
2098 003400 001023          BNE   5$             ;YES, GOOD. RETURN
2099 003402 004737 004166          JSR   PC,GETWSR      ;GET BYTE SELECT REGISTERS
2100 003406 012737 000301 002324          MOV   #RUN!MCLR!MREQ,GDATA ;IDENTIFY REQUESTED FUNCTION
2101 003414          GTDF  EM3,ERR4    ;'MRDY' TIMEOUT
2102          ;          QUEUE 'DEVICE FATAL' ERROR # 1
2103 003414 012737 000001 002172          MOV   #T.EDF,ERRTYP
2104 003422 012737 000001 002174          MOV   #1,ERRNBR
2105 003430 012737 014263 002176          MOV   #EM3,ERRMSG
2106 003436 012737 020120 002200          MOV   #ERR4,ERRBLK
2107 003444 000261          SEC
2108 003446 000401          BR   9$             ;SET CARRY TO INDICATE ERROR
2109 003450 000241          5$: CLC
2110 003452 000207          9$: RTS    PC      ;EXIT WITH THE 'ERROR' FLAG (CARRY BIT) SET
                          ;CLEAR C BIT FOR NO ERRORS
                          ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....M-LOOP -- READ

```

.SBTTL ....M-LOOP -- READ
:*****
: READ - READ THE SPECIFIED ADDRESS WITHIN THE DMV-11 (M8053)
:
: CALLING SEQUENCE:
:
:     JSR     R5,READ
:     .WORD  <ADDRESS OF REGISTER WITHIN DMV-11>
:     .WORD  <DESTINATION ADDRESS WITHIN LSI-11>
:     BCC   NS      ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
:     ERROR ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
:     <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
:
: NS:  <RESUMPTION OF NORMAL PROCESSING>
:-----*****

```

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

```

003454 012577 176746
003460 112777 000001 176734
003466 010346
003470 012703 000050
003474 077301
003476 012603
003500 132777 000200 176714
003506 001023
003510 004737 004166
003514 012737 000001 002324
003522
003522 012737 000001 002172
003530 012737 000002 002174
003536 012737 014307 002176
003544 012737 020120 002200
003552 000261
003554 000401
003556 000241
003560 117735 176646
003564 000205

```

```

READ:  MOV     (R5)+,@SEL4      ;SETUP SOURCE POINTER
        MOVB   #REDLOC,@SEL2   ;TELL M-LOOP TO GIVE US THE REQUESTED DATA
:
:
1$:    MOV     R3,-(SP)
        MOV     #40,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
        SOB    R3,1$
        MOV     (SP)+,R3
:
:
        BITB   #MRDY,@SEL2     ;DID THE M-LOOP FINISH
        BNE    $$              ;YES, GOOD. RETURN
:
:
        JSR    PC,GETWSR       ;GET BYTE SELECT REGISTERS
        MOV     #REDLOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
        GTDF   EM4,ERR4        ;'MRDY' TIMEOUT
:                               ; QUEUE 'DEVICE FATAL' ERROR # 2
:                               MOV     #T.EDF,ERRTYP
:                               MOV     #2,ERRNBR
:                               MOV     #EM4,ERRMSG
:                               MOV     #ERR4,ERRBLK
:
:
        SEC
        BR     6$              ;INDICATE AN ERROR HAS BEEN STACKED
:                               ;RETURN WITH THAT INDICATION
:
5$:    CLC
6$:    MOVB   @SEL6,@(R5)+     ;INDICATE 'NO ERROR'
        RTS    R5              ;PUT DATA WHERE CALLER WANTS IT
:                               ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....M-LOOP -- READ IMMEDIATE

.SBTTLM-LOOP -- READ IMMEDIATE

: READI - READ IMMEDIATE THE SPECIFIED ADDRESS WITHIN THE DMV-11 (MB053)

: CALLING SEQUENCE:

: JSR R5,READI
: .WORD <ADDRESS OF REGISTER WITHIN DMV-11>
: .WORD <DESTINATION -- CONTENTS OF REG. IS PUT HERE>
: BCC NS ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
: ERROR ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
: <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>

: NS: <RESUMPTION OF NORMAL PROCESSING>

2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178 003566
2179 003566 012577 176634
2180 003572 112777 000001 176622
2181
2182 003600 010346
2183 003602 012703 000050
2184 003606 077301
2185 003610 012603
2186
2187 003612 132777 000200 176602
2188 003620 001023
2189
2190 003622 004737 004166
2191 003626 012737 000001 002324
2192 003634
2193
2194 003634 012737 000001 002172
2195 003642 012737 000000 002174
2196 003650 012737 014307 002176
2197 003656 012737 020120 002200
2198 003664 000261
2199 003666 000401
2200
2201 003670 000241
2202 003672 017725 176534
2203 003676 000205
2204
2205
2206
2207

READI:

MOV (R5)+,@SEL4 ;SETUP SOURCE POINTER
MOV #REDLOC,@BSEL2 ;TELL M-LOOP TO GIVE US THE REQUESTED DATA
MOV R3,-(SP)
MOV #40,R3 ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
1\$: SOB R3,1\$
MOV (SP)+,R3
BITB #MRDY,@BSEL2 ;DID THE M-LOOP FINISH
BNE 5\$;YES, GOOD. RETURN
JSR PC,GETWSR ;GET BYTE SELECT REGISTERS
MOV #REDLOC,GDATA ;IDENTIFY REQUESTED FUNCTION
GTFD EM4,ERR4 ;'MRDY' TIMEOUT
; QUEUE 'DEVICE FATAL' ERROR # 3
MOV #T.EDF,ERRTYP
MOV #3,ERRNBR
MOV #EM4,ERRMSG
MOV #ERR4,ERRBLK
SEC
BR 6\$;INDICATE AN ERROR HAS BEEN STACKED
;RETURN WITH THAT INDICATION
5\$: CLC ;INDICATE 'NO ERROR'
6\$: MOV @SEL6,(R5)+ ;PUT DATA WHERE CALLER WANTS IT
RTS R5 ;RETURN

CVDMEA.P11 10-DEC-80 09:16

2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231

003700 012577 176522
003704 113577 176522
003710 000404

```

....M-LOOP -- WRITE
.SBTTL ....M-LOOP -- WRITE
:*****
:WRITE - WRITE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
:
:CALLING SEQUENCE:
:
:   JSR    R5,WRITE
:   .WORD  <ADDRESS OF REGISTER WITHIN DMV-11>
:   .WORD  <ADDRESS OF DATA BYTE>
:   BCC   NS           ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
:   ERROR          ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
:   <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
:
:   NS:  <RESUMPTION OF NORMAL PROCESSING>
:
:-----*****
WRITE:  MOV    (R5)+,@SEL4    ;SETUP SOURCE POINTER
        MOVB  @(R5)+,@SEL6  ;MAKE DATA AVAILABLE TO M-LOOP
        BR    MLWRI        ;THE REST OF THIS ROUTINE IS THE SAME AS 'WRITEI'

```

CVDMEA.P11 10-DEC-80 09:16

....M-LOOP -- WRITE IMMEDIATE

2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277

.SBTTLM-LOOP -- WRITE IMMEDIATE
:*****
:WRITEI - WRITE IMMEDIATE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
:CALLING SEQUENCE:
: JSR R5,WRITEI
: .WORD <ADDRESS OF REGISTER WITHIN DMV-11>
: .WORD <DATA FIELD -- DATA TO BE WRITTEN IN DMV-11>
: BCC NS ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
: ERROR ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
: <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
: NS: <RESUMPTION OF NORMAL PROCESSING>
:-----*****

WRITEI:
MOV (R5)+,@SEL4 ;SETUP SOURCE POINTER
MOV (R5)+,@SEL6 ;MAKE DATA AVAILABLE TO M-LOOP
MLWRI: MOVB #WRILOC,@BSSEL2 ;TELL M-LOOP TO WRITE THE DATA

1\$: MOV R3,-(SP)
MOV #40,R3 ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
SOB R3,1\$
MOV (SP)+,R3

BITB #MRDY,@BSSEL2 ;DID THE M-LOOP FINISH
BNE 5\$;YES, GOOD. RETURN
JSR PC,GETWSR ;GET BYTE SELECT REGISTERS
MOV #WRILOC,GDATA ;IDENTIFY REQUESTED FUNCTION
GTDF EM4,ERR4 ;'MRDY' TIMEOUT
: QUEUE 'DEVICE FATAL' ERROR # 4
MOV #T.EDF,ERRTYP
MOV #4,ERRNBR
MOV #EM4,ERRMSG
MOV #ERR4,ERRBLK

SEC ;INDICATE AN ERROR HAS BEEN STACKED
BR 6\$;RETURN WITH THAT INDICATION

5\$: CLC ;INDICATE 'NO ERROR'
6\$: RTS R5 ;RETURN

CVDMEA.P11 10-DEC-80 09:16

....GETBSR -- GET BYTE SELECT REGISTERS

2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294 004024 117737 176366 002202
2295 004032 117737 176362 002204
2296 004040 117737 176356 002206
2297 004046 117737 176352 002210
2298 004054 117737 176346 002212
2299 004062 117737 176342 002214
2300 004070 117737 176336 002216
2301 004076 117737 176332 002220
2302 004104 117737 176326 002222
2303 004112 117737 176322 002224
2304 004120 117737 176316 002226
2305 004126 117737 176312 002230
2306 004134 117737 176306 002232
2307 004142 117737 176302 002234
2308 004150 117737 176276 002236
2309 004156 117737 176272 002240
2310 004164 000207
2311
2312
2313
2314
2315 004166 017737 176224 002202
2316 004174 017737 176222 002204
2317 004202 017737 176220 002206
2318 004210 017737 176216 002210
2319 004216 017737 176214 002212
2320 004224 017737 176212 002214
2321 004232 017737 176210 002216
2322 004240 017737 176206 002220
2323 004246 000207

```
.SBTTL ....GETBSR -- GET BYTE SELECT REGISTERS
:*****
GET THE CONTENTS OF ALL CONTROL AND STATUS REGISTERS
FUNCTION - THIS SUBROUTINE COLLECTS THE CONTENTS OF THE
          BYTE SELECT REGISTERS FOR THE PURPOSE OF DISPLAY.
ENTRY CONDITIONS - NONE      ##  #   ###  #   ##  #
EXIT CONDITIONS - NONE      #  #  #   ##  #  #  #  ##
REGISTERS DESTROYED - NONE  ##  ###  ###  #  #  #  #
```

```
GETBSR:  MOV  @BSEL0,BSR0      ;PUT THE CURRENT CSR VALUES INTO THE PRINT-OUT
         MOV  @BSEL1,BSR1      ;TABLE
         MOV  @BSEL2,BSR2
         MOV  @BSEL3,BSR3
         MOV  @BSEL4,BSR4
         MOV  @BSEL5,BSR5
         MOV  @BSEL6,BSR6
         MOV  @BSEL7,BSR7
         MOV  @BSEL10,BSR10
         MOV  @BSEL11,BSR11
         MOV  @BSEL12,BSR12
         MOV  @BSEL13,BSR13
         MOV  @BSEL14,BSR14
         MOV  @BSEL15,BSR15
         MOV  @BSEL16,BSR16
         MOV  @BSEL17,BSR17
         RTS  PC              ;RETURN TO CALLER
```

```
.SBTTL ....GETWSR -- GET WORD SELECT REGISTERS
; "WORD" VERSION OF ABOVE SUBROUTINE
```

```
GETWSR:  MOV  @SEL0,WSR0      ;MOVE THE 4 WORD REGISTERS TO THE OTHERWISE
         MOV  @SEL2,WSR2      ;BYTE TABLE
         MOV  @SEL4,WSR4
         MOV  @SEL6,WSR6
         MOV  @SEL10,WSR10
         MOV  @SEL12,WSR12
         MOV  @SEL14,WSR14
         MOV  @SEL16,WSR16
         RTS  PC              ;RETURN TO CALLER
```

CVDMEA.P11 10-DEC-80 09:16

....STUREG -- STATIC TEST OF SPECIFIED USYRT REGISTER

2324
 2325
 2326
 2327
 2328
 2329
 2330
 2331
 2332
 2333
 2334
 2335
 2336
 2337
 2338
 2339
 2340
 2341
 2342
 2343
 2344 004250 010037 004264
 2345 004254 010037 004302
 2346
 2347 004260 004537 003700
 2348 004264 000000
 2349 004266 002322
 2350 004270 103431
 2351
 2352 004272 005037 002326
 2353 004276 004537 003454
 2354 004302 000000
 2355 004304 002326
 2356 004306 103422
 2357
 2358 004310 123737 002324 002326
 2359 004316 000241
 2360
 2361 004320 001415
 2362 004322
 2363
 2364 004322 012737 000001 002172
 2365 004330 012737 000005 002174
 2366 004336 012737 014360 002176
 2367 004344 012737 020244 002200
 2368 004352 000261
 2369 004354 000207
 2370
 2371
 2372
 2373
 2374
 2375
 2376
 2377 004356 000207
 2378
 2379

```
.SBTTL ....STUREG -- STATIC TEST OF SPECIFIED USYRT REGISTER
:*****
:STUREG -- PERFORM A STATIC TEST OF THE SPECIFIED USYRT REGISTER
:
:CALLING SEQUENCE:
:
:  <R0 CONTAINS THE ADDRESS OF THE REGISTER TO BE TESTED>
:  <'TDATA' CONTAINS THE TEST BYTE>
:  <'GDATA' CONTAINS THE EXPECTED DATA>
:  <'REGNUM' CONTAINS REG INDEX FOR POSSIBLE ERRORS>
:
:  JSR    PC,STUREG
:  BCC   NS          ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
:  ERROR ;AN ERROR MESSAGE HAS BEEN STACKED: PRINT IT
:  <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE (I.E. CKLOOP)>
:
:  NS:  <RESUMPTION OF NORMAL PROCESSING>
:-----
```

```
STUREG: MOV    R0,2$          ;PUT SPECIFIED REGISTER'S ADDRESS IN I/O CALLS
        MOV    R0,4$
:
:  2$:  JSR    R5,WRITE      ;WRITE IT
:      .WORD  0            ;*** MODIFIED FROM ABOVE ***
:      .WORD  TDATA
:      BCS   10$          ;ON ERROR, EXIT
:
:      CLR   BDATA        ;CLEAR BOTH BYTES -- JUST IN CASE....
:      JSR   R5,READ      ;READ IT BACK AGAIN
:      .WORD  0            ;*** MODIFIED FROM ABOVE ***
:      .WORD  BDATA
:      BCS   10$          ;ON ERROR, EXIT
:
:  4$:  CMPB  GDATA,BDATA   ;DID WE READ WHAT WE WROTE?
:      CLC
:      ; (THIS ISN'T NEEDED FOR THE ERROR TEST BUT
:      ; MUST BE CLEARED ON EXIT IF NO ERROR OCCURED)
:      BEQ   10$          ;YES, EXIT FROM SUBTEST
:      GTDF  EM25,ERR7A   ;REPORT READ/WRITE ERROR
:      ;          QUEUE 'DEVICE FATAL' ERROR # 5
:
:      MOV   #T.EDF,ERRTYP
:      MOV   #5,ERRNBR
:      MOV   #EM25,ERRMSG
:      MOV   #ERR7A,ERRBLK
:
:  10$: SEC
:      RTS   PC          ;INDICATE THAT AN ERROR WAS DETECTED
```

```
.SBTTL ....STALL -- DELAY FOR 10.5 MICRO-SEC'S (ON LSI-11)
:*****
:STALL -- THIS SUBROUTINE STALLS FOR ABOUT 10.5 MICRO-SECONDS
:-----
STALL: RTS   PC
```


CVDMEA.P11 10-DEC-80 09:16

2380

....STALL -- DELAY FOR 10.5 MICRO-SEC'S (ON LSI-11)

C. JME.A.P11 10-DEC-80 09:16

```

2381
2382
2383
2384
2385
2386
2387
2388
2389
2390 004360 012737 002242 004422
2391 004366 012737 120400 004420
2392
2393 004374 005037 002260
2394 004400 004537 003454
2395 004404 122000
2396 004406 002260
2397
2398 004410 005077 000006
2399 004414 004537 003454
2400 004420 000000
2401 004422 000000
2402
2403 004424 005237 004420
2404 004430 023727 004420 120406
2405 004436 001772
2406
2407 004440 062737 000002 004422
2408 004446 023727 004420 120410
2409 004454 001355
2410
2411 004456 000207
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421 004460 012737 002262 004506
2422 004466 012737 120000 004504
2423 004474 005077 000006
2424 004500 004537 003454
2425 004504 000000
2426 004506 000000
2427 004510 005237 004504
2428 004514 062737 000002 004506
2429 004522 023727 004504 120020
2430 004530 001361
2431 004532 000207

.SBTTL
:*****
:* GETURS - LOAD INTO THE 8 WORD STORAGE AREA (UREGS) THE CONTENTS OF THE
:* VARIOUS USYRT REGISTERS
:*
:* CALLING SEQUENCE:
:*****
GETURS: MOV #UREGS,5$ ;INIT POINTER TO REG STORAGE TABLE
MOV #USYRT,4$ ;INIT POINTER TO REGISTER ADDRESSES
3$: CLR UREGS+14. ;CLEAR STORAGE WORD
JSR R5,READ ;READ THE USYRT STATUS REGISTER
;STATUS REGISTER'S ADDRESS WITHIN DMV-11
;ADDRESS ALLOCATED TO THAT REG. W/IN 'UREGS'
4$: .WORD 0
5$: .WORD 0
6$: INC 4$ ;INCREMENT REG NO.
CMP 4$,#USYRT+6 ;THIS IS NOT A VALID REGISTER ADDRESS
BEQ 6$ ;SO IT MUST BE BYPASSED
ADD #2,5$ ;ADVANCE ADDRESS OF STORAGE AREA POINTER
CMP 4$,#USYRT+10 ;SEE IF ALL REGS READ YET
BNE 3$ ;BR IF NOT
RTS PC ;RETURN

:*****
:* GETVRS: - LOAD INTO THE 16 WORD STORAGE AREA (VREGS) THE CONTENTS OF THE
:* VARIOUS VIA REGISTERS.
:*
:* CALLING SEQUENCE :
:*****
GETVRS: MOV #VREGS,5$ ;INIT POINTER TO REG STORAGE TABLE
MOV #VIA,4$ ;INIT POINTER TO REGISTER ADDRESSES
3$: CLR 2$ ;CLEAR STORAGE WORD
JSR R5,READ ;READ A VIA REG
4$: .WORD 0 ;REGISTER ADDRESS GOES HERE
5$: .WORD 0 ;STORAGE ADRS IN TABLE GOES HERE
6$: INC 4$ ;INCREMENT REG NO.
ADD #2,5$ ;INCREMENT STORAGE ADRS
CMP 4$,#VIA+16. ;SEE IF ALL VIA REGS READ YET
BNE 3$ ;BR IF NOT
RTS PC ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....INITT1 -- INITIALIZE TIMER #1

2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459 004534 010146
2460 004536 012537 004660
2461 004542 012537 004706
2462 004546 111501
2463 004550 143701 000077
2464 004554 010137 004650
2465 004560 112501
2466
2467 004562 106301
2468 004564 106301
2469
2470
2471
2472
2473 004566 143701 000177
2474 004572 153701 000100
2475 004576 010137 004610
2476
2477 004602 004537 003712
2478 004606 120016
2479 004610 000000
2480
2481 004612 004537 003566
2482 004616 120013
2483 004620 000000
2484
2485 004622 013701 004620
2486 004626 143701 000300
2487 004632 053701 004650

```
.SBTTL ....INITT1 -- INITIALIZE TIMER #1
*****
* INITT1 - INITIALIZE TIMER # 1
*
* CALLING SEQUENCE:
*
* JSR R5,INITT1
* .WORD <VALUE LOADED INTO THE T1 LATCH @ VIAT1C & VIAT1D>
* .WORD <VALUE LOADED INTO 'T1L-L' & 'T1C-H'>
* .BYTE <BITS 6 & 7 WILL BE LOADED INTO 'ACR', BIT 5 WILL BE
* USED TO SET OR CLEAR BIT 6 ('T1') OF THE INTERRUPT
* ENABLE REGISTER ('IER')>
* .BYTE <UNUSED>
*
* NOTE:
*
* BEFORE LOADING AND STARTING THE COUNTER, THE LATCH REGISTER (ACCESSED THRU
* 'VIAT1C') IS LOADED. THEN, T1L-L IS LOADED AND NEXT, T1C-H. THIS LAST
* LOAD WILL RESET THE TIMEOUT BIT AND COUNTER LOGIC. IT IS EXPECTED AT THIS
* TIME (5/25/79) THAT THE INTERRUPT FACILITY OF THE VIA CHIP WILL NOT BE USED
* -- HOWEVER, ACCESS TO THE INTERRUPT ENABLE BIT IS GIVEN THROUGH THE THIRD
* PARAMETER IN THE CALLING SEQUENCE (BIT 5 = 0 WILL CAUSE THIS ROUTINE TO
* CLEAR THE ENABLE BIT ('T1') IN 'IER'.)
*****
```

```
INITT1: MOV R1, -(SP) ;SAVE THE REGISTER WE WILL BE USING
MOV (R5)+, 7$ ;SETUP VALUE TO BE WRITTEN IN LATCH
MOV (R5)+, 10$ ;SETUP VALUE TO BE WRITTEN IN COUNTER
MOVB (R5), R1 ;GET & PROCESS BITS FOR ACR 6 & 7
BICB 077, R1
MOV R1, 4$ ;SETUP CALL SET ACR'S BITS 6 & 7
MOVB (R5)+, R1 ;NOW, GET THE BIT TO BE USED IN SETTING OR
;CLEARING BIT 6 OF 'IER'
ASLB R1 ;THE PASSED BIT IS IN THE WRONG POSITION
ASLB R1 ;BUT, THE PASSED BIT SHOULD CONTROL THE OPERATION.
;WE KNOW WE ARE SETTING OR CLEARING BIT 6 --
;THUS, THE PASSED BIT WILL BECOME THE CONTROLLING
;BIT 7 AND WE WILL 'OR' IN THE BIT WE WISH TO
;BE CONTROLLED (BIT 6).
BICB 177, R1 ;FIRST, MAKE SURE ALL UNWANTED BITS ARE CLEARED
BISB 100, R1 ;THEN SET BIT 6
MOV R1, 2$ ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE

2$: JSR R5, WRITEI ;WRITE TO
VIAIER ;THE VIA'S IER
.WORD 0 ;INTERRUPT ENABLE/DISABLE INFORMATION

3$: JSR R5, READI ;READ THE CURRENT SETTING OF
VIAACR ;THE VIA'S ACR
.WORD 0 ;INTO '3$'

MOV 3$, R1 ;GET THAT VALUE
BICB 300, R1 ;CLEAR THE CURRENT SETTING OF BITS 6 & 7
BIS 4$, R1 ;SET THEM ACCORDING TO THE PASSED VALUES
```

CVDMEA.P11 10-DEC-80 09:16

....INIT1 -- INITIALIZE TIMER #1

```

2488 004636 010137 004650      MOV      R1,4$      ;PASS THE NEW REG. SETTING TO APPROPRIATE CALL
2489
2490 004642 004537 003712      JSR      R5,WRITEI  ;WRITE TO
2491 004646 120013                VIAACR                ;THE VIA'S ACR
2492 004650 000000      4$:      .WORD      0      ;THE NEW REGISTER SETTING
2493
2494 004652 004537 003712      JSR      R5,WRITEI  ;WRITE TO
2495 004656 120006                VIAT1C                ;LOW ORDER LATCH REGISTER (T1L-L)
2496 004660 000000      7$:      .WORD      0      ;THE VALUE PASSED
2497
2498 004662 113737 004661 004676      MOVB     7$+1,8$     ;SETUP FOR AND
2499 004670 004537 003712      JSR      R5,WRITEI  ;WRITE TO
2500 004674 120007                VIAT1D                ;HIGH ORDER LATCH REGISTER (T1L-H)
2501 004676 000000      8$:      .WORD      0      ;THE VALUE PASSED
2502
2503 004700 004537 003712      JSR      R5,WRITEI  ;WRITE TO
2504 004704 120004                VIAT1A                ;LOW ORDER LATCH & COUNTER (T1L-L & T1C-L)
2505 004706 000000      10$:     .WORD      0      ;THE VALUE PASSED
2506
2507 004710 113737 004707 004724      MOVB     10$+1,11$  ;SETUP FOR AND
2508 004716 004537 003712      JSR      R5,WRITEI  ;WRITE TO
2509 004722 120005                VIAT1B                ;HIGH ORDER COUNTER (T1C-H) <ALSO STARTS CTR>
2510 004724 000000      11$:     .WORD      0      ;THE VALUE PASSED
2511
2512      ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2513
2514 004726 012601      MOV      (SP)+,R1    ;BUT FIRST RESTORE R1
2515 004730 005205      INC      R5          ;AND PUT R5 BACK ON A WORD BOUNDARY (THE LAST
2516                                     ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2517
2518 004732 000205      RTS      R5          ;NOW, RETURN
2519
2520

```

CVDMEA.P11 10-DEC-80 09:16

....INITT2 -- INITIALIZE TIMER #2

2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576

004734 010146
004736 012537 005056
004742 111501
004744 143701 000337
004750 010137 005046
004754 112501
004756 106301
004760 106301
004762 106301
004764 143701 000177
004770 153701 000040
004774 010137 005006
005000 004537 003712
005004 120016
005006 000000
005010 004537 003566
005014 120013
005016 000000
005020 013701 005016
005024 143701 000040
005030 053701 005046
005034 010137 005046

```
.SBTTL ....INITT2 -- INITIALIZE TIMER #2
*****
* INITT2 - INITIALIZE TIMER # 2
*
*      CALLING SEQUENCE:
*
*      JSR      R5,INITT2
*      .WORD    <VALUE LOADED INTO 'T2L-L' & 'T2C-H'>
*      .BYTE    <BIT 5 WILL BE LOADED INTO 'ACR', BIT 4 WILL BE USED
*              TO SET OR CLEAR BIT 5 ('T2') OF THE INTERRUPT ENABLE
*              REGISTER ('IER')>
*      .BYTE    <UNUSED>
*
* NOTE:
* FIRST T2L-L IS LOADED, THEN T2C-H. THIS SECOND LOAD WILL RESET THE TIMEOUT
* BIT AND COUNTER LOGIC. IT IS EXPECTED AT THIS TIME (5/25/79) THAT THE
* INTERRUPT FACILITY OF THE VIA CHIP WILL NOT BE USED -- HOWEVER, ACCESS TO
* THE INTERRUPT ENABLE BIT IS GIVEN THROUGH THE SECOND PARAMETER IN THE
* CALLING SEQUENCE (BIT 4 = 0 WILL CAUSE THIS ROUTINE TO CLEAR THE ENABLE BIT
* ('T2') IN 'IER'.)
*****
```

```
INITT2: MOV      R1, -(SP)      ;SAVE THE REGISTER WE WILL BE USING
        MOV      (R5)+, 10$  ;SETUP VALUE TO BE WRITTEN IN COUNTER
        MOV      (R5), R1    ;GET & PROCESS BIT FOR ACR 5
        BICB     337, R1
        MOV      R1, 4$      ;SETUP CALL TO SET OR CLEAR ACR'S BIT 5
        MOV      (R5)+, R1   ;NOW, GET THE BIT TO BE USED IN SETTING OR
        ;CLEARING BIT 5 OF 'IER'
        ASLB     R1          ;THE PASSED BIT IS IN THE WRONG POSITION
        ASLB     R1          ;BUT, THE PASSED BIT SHOULD CONTROL THE
        ASLB     R1          ;OPERATION.
        ;WE KNOW WE ARE SETTING OR CLEARING BIT 5 --
        ;THUS, THE PASSED BIT WILL BECOME THE CONTROLLING
        ;BIT & WE WILL 'OR' IN THE BIT WE WISH TO
        ;BE CONTROLLED (BIT 5).
        BICB     177, R1     ;FIRST, MAKE SURE ALL UNWANTED BITS ARE CLEARED
        BISB     040, R1    ;THEN SET BIT 5
        MOV      R1, 2$     ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE
        JSR      R5, WRITEI  ;WRITE TO
        VIAIER   ;THE VIA'S IER
2$: .WORD    0              ;INTERRUPT ENABLE/DISABLE INFORMATION
        JSR      R5, READI   ;READ THE CURRENT SETTING OF
        VIAACR   ;THE VIA'S ACR
3$: .WORD    0              ;INTO '3$'
        MOV      3$, R1      ;GET THAT VALUE
        BICB     040, R1    ;CLEAR THE CURRENT SETTING OF BIT 5
        BIS      4$, R1     ;SET IT ACCORDING TO THE PASSED VALUE
        MOV      R1, 4$     ;PASS NEW REG. SETTING TO APPROPRIATE CALL
```

CVDMEA.P11 10-DEC-80 09:16

....INITT2 -- INITIALIZE TIMER #2

```

2577 005040 004537 003712      JSR    R5,WRITEI      ;WRITE TO
2578 005044 120013              VIAACR                ;THE VIA'S ACR
2579 005046 000000      4$:  .WORD    0        ;THE NEW REGISTER SETTING
2580
2581 005050 004537 003712      JSR    R5,WRITEI      ;WRITE TO
2582 005054 120010              VIAT2A                ;LOW ORDER LATCH & COUNTER (T2L-L & T2C-L)
2583 005056 000000      10$: .WORD    0        ;THE VALUE PASSED
2584
2585 005060 113737 005057 005074  MOVB   10$+1,11$      ;SETUP FOR AND
2586 005066 004537 003712      JSR    R5,WRITEI      ;WRITE TO
2587 005072 120011              VIAT2B                ;HIGH ORDER COUNTER (T2C-H) <ALSO STARTS CTR>
2588 005074 000000      11$: .WORD    0        ;THE VALUE PASSED
2589
2590 ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2591
2592 005076 012601      MOV    (SP)+,R1       ;BUT FIRST RESTORE R1
2593 005100 005200      INC   R5              ;AND PUT R5 BACK ON A WORD BOUNDARY (THE LAST
2594                                     ;PASSED PARAM WAS A BYTE, NOT A WORD!)
2595
2596 005102 000205      RTS   R5              ;THEN RETURN
2597

```

CVDMEA.P11 10-DEC-80 09:16

....RSTCHK -- RESET USYRT/VERIFY ALL USYRT REGS @ RESET STATE

2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651

005104
005104 010146
005106 010246

005110 004537 003712
005114 120000
005116 000031
005120 004537 003712
005124 120000
005126 000030

005130 005001
005132 012702 002606
005136 016137 002476 005150 6\$:
005144 004537 003566
005150 000000 7\$:
005152 000000 8\$:
005154 123722 005152
005160 001432

005162 010137 002336
005166 006237 002336
005172 005037 002324
005176 116237 177777 002324
005204 013737 005152 002326

005212

005212 012737 000001 002172
005220 012737 000006 002174
005226 012737 014214 002176
005234 012737 020364 002200
005242 000261
005244 000406

005246 062701 000002 9\$:
005252 020127 000020
005256 002727
005260 000241
005262 012602 10\$:
005264 012601
005266 000205

```
.SBTTL ....RSTCHK -- RESET USYRT/VERIFY ALL USYRT REGS @ RESET STATE
*****
: RSTCHK - MANUALLY RESET THE USYRT AND VERIFY THAT ALL USYRT REGISTERS
: ARE IN THEIR RESET STATE. AN ERROR MESSAGE IDENTIFYING THE
: FAILING REGISTER IS STACKED IF ONE IS ENCOUNTERED.
:
: CALLING SEQUENCE:
: JSR R5,RSTCHK
*****

RSTCHK:
MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2

JSR R5,WRITEI ;SET PROGRAM RESET BIT IN VIA ORB REG
VIAORB
DTR!RTSND!PRESET
JSR R5,WRITEI ;CLEAR PROGRAM RESET BIT IN VIA ORB REG
VIAORB
DTR!RTSND

CLR R1 ;INIT USYRT REG ADRS PTR
MOV #PATF,R2 ;INIT DATA PATTERN POINTER
MOV USYREG(R1),7$ ;SET USYRT READ ADDRESS
JSR R5,READI ;READ A USYRT REG
;WORD 0 ;USYRT REG ADRS GOES HERE
;WORD 0 ;DATA READ IS RETURNED HERE
CMPB 8$,(R2)+ ;SEE IF REG CONTAINS EXPECTED DATA
BEQ 9$ ;BR IF MATCH

MOV R1,REGNUM ;SET USYRT REG NO. FOR PRINTOUT
ASR REGNUM ;GET WORD OFFSET
CLR GDATA ;GET EXPECTED DATA
MOVB -1(R2),GDATA
MOV 8$,BDATA ;GET ACTUAL DATA
;STACK 'USYRT NOT CLEARED BY PROGRAM RESET' MSG
GTDF EM2,ERR10

; QUEUE 'DEVICE FATAL' ERROR # 6
MOV #T.EDF,ERRTYP
MOV #6,ERRNBR
MOV #EM2,ERRMSG
MOV #ERR10,ERRBLK

SEC ;SET C BIT TO FLAG ERROR
BR 10$ ;TAKE ERROR EXIT

9$: ADD #2,R1 ;INCR USYRT REG ADRS PTR
CMP R1,#16. ;SEE IF ALL REGS READ YET
BLT 6$ ;BR IF NOT
CLC ;** CLEAR C BIT FOR NO ERRORS
10$: MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
RTS R5 ;** RETURN
```

CVDMEA.P11 10-DEC-80 09:16

```

2652
2653
2654
2655 005270 010146
2656 005272 012701 000005
2657 005276 077101
2658 005300 012601
2659 005302 000207
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678 005304
2679 005304 004537 003712
2680 005310 120002
2681 005312 000377
2682 005314 004537 003712
2683 005320 120003
2684 005322 000001
2685 005324 004537 003712
2686 005330 120017
2687 005332 000000
2688 005334 004537 003712
2689 005340 120000
2690 005342 000030
2691 005344 004537 003712
2692 005350 120013
2693 005352 000350
2694 005354 004537 003712
2695 005360 120014
2696 005362 000022
2697 005364 004537 003712
2698 005370 120016
2699 005372 000177
2700 005374 000207
2701
2702
    
```

....RSTCHK -- RESET USYRT/VERIFY ALL USYRT REGS @ RESET STATE

```

:*****
:* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
:*****
WAIT50: MOV R1,-(SP) ;SAVE R1
        MOV #5,R1 ;INIT COUNTER
3$: SOB R1,3$ ;DELAY HERE FOR 23.8 MICRO-SEC'S
        MOV (SP)+,R1 ;RESTORE R1
        RTS PC ;RETURN

; OVERHEAD (JSR, MOV, MOV, MOV, & RTS) ADD UP TO 25.25 MICRO-SEC'S
; THEREFORE, ACTUAL TOTAL DELAY IS 49.35 MICRO-SECONDS
    
```

.SBTTLSETVIA -- SET UP VIA REGISTERS

```

:*****
:* SETVIA - SET UP THE VIA REGISTERS
:*
:* THIS SUBROUTINE PROGRAMS THE VIA REGISTERS FOR NORMAL OPERATION, BY
:* LOADING THE DDRB, DDRA, ORB, ACR, PCR, IER.
:*
:* CALLING SEQUENCE :
:* JSR PC,SETVIA
:*****
SETVIA: JSR R5,WRITEI ;SET PORT B FOR OUTPUT MODE
        VIADPB
        377
        JSR R5,WRITEI ;SET PORT A FOR INPUT MODE
        VIADPA ;(BIT0 IS ONLY OUTPUT BIT)
        001
        JSR R5,WRITEI ;DISABLE USYRT INTERNAL LOOPBACK
        VIAORA
        000
        JSR R5,WRITEI ;INIT PORT B
        VIAORB
        DTR!RTSND
        JSR R5,WRITEI ;SET ACR FOR : T1 SQUARE WAVE OUTPUT MODE,
        VIAACR ; T2 ONE-SHOT OUTPUT MODE,
        350 ; SR AT SYS CLOCK RATE ON CB1
        JSR R5,WRITEI ;SET PCR FOR : CB1 NEG TRANS INPUT MODE,
        VIAPCR ; CA2 NEG TRANS INPUT MODE,
        022 ; CA1 NEG TRANS INPUT MODE
        JSR R5,WRITEI ;DISABLE ALL MICRO-INTRPTS
        VIAIER
        177
        RTS PC ;RETURN
    
```


CVDMEA.P11 10-DEC-80 09:16

....INIDMV -- INIT DMV (MCLR, VIA SETUP)

2703
2704
2705
2706
2707
2708
2709
2710
2711
2712 005376 004737 003352
2713 005402 004737 005304
2714 005406 000207
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726 005410
2727 005410 004537 003566
2728 005414 122000
2729 005416 000000
2730 005420 122537 005416
2731 005424 000241
2732 005426 001430
2733 005430 012737 000007 002336
2734 005436 016537 177777 002324
2735 005444 005037 002326
2736 005450 113737 005416 002326
2737
2738 005456
2739
2740 005456 012737 000001 002172
2741 005464 012737 000007 002174
2742 005472 012737 014650 002176
2743 005500 012737 020364 002200
2744 005506 000261
2745 005510 005205
2746 005512 000205
2747
2748
2749
2750

```

.SBTTL ....INIDMV -- INIT DMV (MCLR, VIA SETUP)
*****
* INIDMV - THIS SUBROUTINE INITIALIZES THE DMV-11, BY DOING A MASTER CLEAR,
* ENTERING THE M-LOOP. AND PROGRAMMING THE VIA REGS FOR DEFAULT
* OPERATION.
*
* CALLING SEQUENCE :
*   JSR PC,INIDMV
*****
INIDMV: JSR   PC,MSTCLR      ;MASTER CLR, M-LOOP
        JSR   PC,SETVIA   ;PROGRAM VIA
        RTS   PC          ;RETURN

.SBTTL ....CKUSTS -- CHECK USYRT STATUS REGISTERS
*****
* CKUSTS - THIS SUBROUTINE CHECKS THE USYRT STATUS BY READING THE USYRT
* STATUS REGISTER AND COMPARING IT TO THE LOW BYTE OF THE WORD FOLLOWING
* THE CALL. IF THERE IS A MISMATCH, THE SUBROUTINE STACKS THE ERROR
* INFORMATION, AND SETS THE 'C' BIT AND RETURNS.
*****
CKUSTS: JSR   R5,READI      ;READ USYRT STATUS REGISTER
        USTATR
1$:      .WORD 0
        CMPB (R5)+,1$     ;SEE IF STATUS MATCHES EXPECTED
        CLC              ;CLEAR C BIT
        BEQ  2$          ;BR IF STATUS OK
        MOV  #7,REGNUM    ;SET USYRT REG NO. FOR PRINTOUT
        MOV  -1(R5),GDATA ;GET EXPECTED DATA
        CLR  BDATA        ;GET ACTUAL DATA
        MOVB 1$,BDATA
;STACK 'USYRT STATUS INCORRECT' ERROR
        GTDF EM68,ERR10
;          QUEUE 'DEVICE FATAL' ERROR # 7
        MOV  #T.EDF,ERRTYP
        MOV  #7,ERRNBR
        MOV  #EM68,ERRMSG
        MOV  #ERR10,ERRBLK
2$:      SEC              ;SET C BIT FOR ERROR
        INC  R5           ;INCREMENT R5 PAST ARGUMENT
        RTS  R5          ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....CKTACT -- CHECK TRANSMITTER ACTIVE (TXACT)

2751
 2752
 2753
 2754
 2755
 2756
 2757
 2758
 2759
 2760
 2761 005514
 2762 005514 012737 000007 002336
 2763 005522 004537 003566
 2764 005526 122000
 2765 005530 000000
 2766 005532 032725 000001
 2767 005536 001422
 2768 005540 132737 000004 005530
 2769 005546 001040
 2770
 2771 005550
 2772
 2773 005550 012737 000001 002172
 2774 005556 012737 000010 002174
 2775 005564 012737 014677 002176
 2776 005572 012737 020714 002200
 2777 005600 000261
 2778 005602 000423
 2779 005604 132737 000004 005530
 2780 005612 001416
 2781
 2782 005614
 2783
 2784 005614 012737 000001 002172
 2785 005622 012737 000011 002174
 2786 005630 012737 014715 002176
 2787 005636 012737 020714 002200
 2788 005644 000261
 2789 005646 000401
 2790 005650 000241
 2791 005652 000205
 2792
 2793
 2794
 2795

```

.SBTTL ....CKTACT -- CHECK TRANSMITTER ACTIVE (TXACT)
*****
* CKTACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TXACT IN THE USYRT
* STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
* STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
*
* CALLING SEQUENCE :
* JSR R5,CKTACT
* .WORD <BIT 0 IS EXPECTED VALUE OF TXACT>
*****
CKTACT:
MOV #7,REGNUM ;SET REG NO. FOR POSSIBLE ERROR REPORT
JSR R5,READI ;READ USYRT STATUS
USTATR
1$: .WORD 0
BIT #BIT0,(R5)+ ;GET EXPECTED STATE OF TXACT
BEQ 2$ ;BR IF EXPECTED TXACT = 0
BITB #TXACT,1$ ;SEE IF TXACT = 1
BNE 3$ ;BR IF TXACT = 1
;STACK 'TXACT NOT SET' MSG
GTDF EM69,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 8
MOV #T.EDF,ERRTYP
MOV #8,ERRNBR
MOV #EM69,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
2$: BITB #TXACT,1$ ;SEE IF TXACT = 0
BEQ 3$ ;BR IF TXACT = 0
;STACK 'TXACT NOT CLEARED' MSG
GTDF EM70,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 9
MOV #T.EDF,ERRTYP
MOV #9,ERRNBR
MOV #EM70,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
3$: CLC ;CLEAR C BIT FOR NO ERRORS
4$: RTS R5 ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....CKRACT -- CHECK RECEIVER ACTIVE (RXACT)

2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806 005654
2807 005654 012737 000007 002336
2808 005662 004537 003566
2809 005666 122000
2810 005670 000000
2811 005672 032725 000001
2812 005676 001422
2813 005700 132737 000040 005670
2814 005706 001040
2815
2816 005710
2817
2818 005710 012737 000001 002172
2819 005716 012737 000012 002174
2820 005724 012737 014737 002176
2821 005732 012737 020714 002200
2822 005740 000261
2823 005742 000423
2824 005744 132737 000040 005670
2825 005752 001416
2826
2827 005754
2828
2829 005754 012737 000001 002172
2830 005762 012737 000013 002174
2831 005770 012737 014755 002176
2832 005776 012737 020714 002200
2833 006004 000261
2834 006006 000401
2835 006010 000241
2836 006012 000205
2837
2838
2839
2840

```

.SBTTL ....CKRACT -- CHECK RECEIVER ACTIVE (RXACT)
*****
;* CKRACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RXACT IN THE USYRT
;* STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
;* STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
;*
;* CALLING SEQUENCE :
;* JSR R5,CKRACT
;* .WORD <BIT 0 IS EXPECTED VALUE OF RXACT>
*****
CKRACT:
MOV #7,REGNUM ;SET REG NO. FOR POSSIBLE ERROR REPORT
JSR R5,READI ;READ USYRT STATUS
USTATR
1$: .WORD 0
BIT #BIT0,(R5)+ ;GET EXPECTED STATE OF RXACT
BEQ 2$ ;BR IF EXPECTED RXACT = 0
BITB #RXACT,1$ ;SEE IF RXACT = 1
BNE 3$ ;BR IF RXACT = 1
;STACK 'RXACT NOT SET' MSG
GTDF EM71,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 10
MOV #T.EDF,ERRTYP
MOV #10,ERRNBR
MOV #EM71,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
2$: BITB #RXACT,1$ ;SEE IF RXACT = 0
BEQ 3$ ;BR IF RXACT = 0
;STACK 'RXACT NOT CLEARED' MSG
GTDF EM72,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 11
MOV #T.EDF,ERRTYP
MOV #11,ERRNBR
MOV #EM72,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
3$: CLC ;CLEAR C BIT FOR NO ERRORS
4$: RTS R5 ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....CKTBMT -- CHECK TRANSMIT BUFFER EMPTY

2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851 006014
2852 006014 012737 000007 002336
2853 006022 004537 003566
2854 006026 122000
2855 006030 000000
2856 006032 032725 000001
2857 006036 001422
2858 006040 132737 000100 006030
2859 006046 001040
2860
2861 006050
2862
2863 006050 012737 000001 002172
2864 006056 012737 000014 002174
2865 006064 012737 014777 002176
2866 006072 012737 020714 002200
2867 006100 000261
2868 006102 000423
2869 006104 132737 000100 006030
2870 006112 001416
2871
2872 006114
2873
2874 006114 012737 000001 002172
2875 006122 012737 000015 002174
2876 006130 012737 015014 002176
2877 006136 012737 020714 002200
2878 006144 000261
2879 006146 000401
2880 006150 000241
2881 006152 000205
2882
2883
2884
2885

```

.SBTTL ....CKTBMT -- CHECK TRANSMIT BUFFER EMPTY
:*****
:* CKTBMT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TBMT IN THE USYRT
:* STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
:* STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
:*
:* CALLING SEQUENCE :
:* JSR R5,CKTBMT
:* .WORD <BIT 0 IS EXPECTED VALUE OF TBMT>
:*****
CKTBMT:
MOV #7,REGNUM ;SET REG NO. FOR POSSIBLE ERROR REPORT
JSR R5,READI ;READ USYRT STATUS
USTATR
1$: .WORD 0
BIT #BIT0,(R5)+ ;GET EXPECTED STATE OF TBMT
BEQ 2$ ;BR IF EXPECTED TBMT = 0
BITB #TBMT,1$ ;SEE IF TBMT = 1
BNE 3$ ;BR IF TBMT = 1
;STACK 'TBMT NOT SET' MSG
GTDF EM73,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 12
MOV #T.EDF,ERRTYP
MOV #12,ERRNBR
MOV #EM73,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
2$: BITB #TBMT,1$ ;SEE IF TBMT = 0
BEQ 3$ ;BR IF TBMT = 0
;STACK 'TBMT NOT CLEARED' MSG
GTDF EM74,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 13
MOV #T.EDF,ERRTYP
MOV #13,ERRNBR
MOV #EM74,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
3$: CLC ;CLEAR C BIT FOR NO ERRORS
4$: RTS R5 ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....CKRDA -- CHECK RECEIVE DATA AVAILABLE

2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896 006154
2897 006154 012737 000007 002336
2898 006162 004537 003566
2899 006166 122000
2900 006170 000000
2901 006172 032725 000001
2902 006176 001422
2903 006200 132737 000200 006170
2904 006206 001040
2905
2906 006210
2907
2908 006210 012737 000001 002172
2909 006216 012737 000016 002174
2910 006224 012737 015035 002176
2911 006232 012737 020714 002200
2912 006240 000261
2913 006242 000423
2914 006244 132737 000200 006170
2915 006252 001416
2916
2917 006254
2918
2919 006254 012737 000001 002172
2920 006262 012737 000017 002174
2921 006270 012737 015051 002176
2922 006276 012737 020714 002200
2923 006304 000261
2924 006306 000401
2925 006310 000241
2926 006312 000205
2927
2928
2929
2930

```

.SBTTL ....CKRDA -- CHECK RECEIVE DATA AVAILABLE
*****
* CKRDA - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RDA IN THE USYRT
* STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
* STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
*
* CALLING SEQUENCE :
* JSR R5,CKRDA
* .WORD <BIT 0 IS EXPECTED VALUE OF RDA>
*****
CKRDA:
MOV #7,REGNUM ;SET REG NO. FOR POSSIBLE ERROR REPORT
JSR R5,READI ;READ USYRT STATUS
USTATR
1$: .WORD 0
BIT #BIT0,(R5)+ ;GET EXPECTED STATE OF RDA
BEQ 2$ ;BR IF EXPECTED RDA = 0
BITB #RDA,1$ ;SEE IF RDA = 1
BNE 3$ ;BR IF RDA = 1
;STACK 'RDA NOT SET' MSG
GTFD EM75,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 14
MOV #T.EDF,ERRTYP
MOV #14,ERRNBR
MOV #EM75,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
2$: BITB #RDA,1$ ;SEE IF RDA = 0
BEQ 3$ ;BR IF RDA = 0
;STACK 'RDA NOT CLEARED' MSG
GTFD EM76,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 15
MOV #T.EDF,ERRTYP
MOV #15,ERRNBR
MOV #EM76,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
3$: CLC ;CLEAR C BIT FOR NO ERRORS
4$: RTS R5 ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....CKRSA -- CHECK RECEIVER STATUS AVAILABLE

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
2972
2973

006314
006314 012737 000007 002336
006322 004537 003566
006326 122000
006330 000000
006332 032725 000001
006336 001422
006340 132737 000020 006330
006346 001040
006350
006350 012737 000001 002172
006356 012737 000020 002174
006364 012737 015071 002176
006372 012737 020714 002200
006400 000261
006402 000423
006404 132737 000020 006330
006412 001416
006414
006414 012737 000001 002172
006422 012737 000021 002174
006430 012737 015105 002176
006436 012737 020714 002200
006444 000261
006446 000401
006450 000241
006452 000205

```
.SBTTL ....CKRSA -- CHECK RECEIVER STATUS AVAILABLE
*****
* CKRSA - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RSA IN THE USYRT
* STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
* STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
*
* CALLING SEQUENCE :
* JSR R5,CKRSA
* .WORD <BIT 0 IS EXPECTED VALUE OF RSA>
*****
CKRSA:
MOV #7,REGNUM ;SET REG NO. FOR POSSIBLE ERROR REPORT
JSR R5,READI ;READ USYRT STATUS
USTATR
1$: .WORD 0
BIT #BIT0,(R5)+ ;GET EXPECTED STATE OF RSA
BEQ 2$ ;BR IF EXPECTED RSA = 0
BITB #RSA,1$ ;SEE IF RSA = 1
BNE 3$ ;BR IF RSA = 1
;STACK 'RSA NOT SET' MSG
GDF EM77,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 16
MOV #T.EDF,ERRTYP
MOV #16,ERRNBR
MOV #EM77,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
2$: BITB #RSA,1$ ;SEE IF RSA = 0
BEQ 3$ ;BR IF RSA = 0
;STACK 'RSA NOT CLEARED' MSG
GDF EM78,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 17
MOV #T.EDF,ERRTYP
MOV #17,ERRNBR
MOV #EM78,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
3$: CLC ;CLEAR C BIT FOR NO ERRORS
4$: RTS R5 ;RETURN
```

CVDMEA.P11 10-DEC-80 09:16

....CKROR -- CHECK RECEIVER OVERRUN

2974
 2975
 2976
 2977
 2978
 2979
 2980
 2981
 2982
 2983
 2984 006454
 2985 006454 012737 000001 002336
 2986 006462 004537 003566
 2987 006466 120401
 2988 006470 000000
 2989 006472 032725 000001
 2990 006476 001422
 2991 006500 132737 000010 006470
 2992 006506 001040
 2993
 2994 006510
 2995
 2996 006510 012737 000001 002172
 2997 006516 012737 000022 002174
 2998 006524 012737 015450 002176
 2999 006532 012737 020714 002200
 3000 006540 000261
 3001 006542 000423
 3002 006544 132737 000010 006470
 3003 006552 001416
 3004
 3005 006554
 3006
 3007 006554 012737 000001 002172
 3008 006562 012737 000023 002174
 3009 006570 012737 015501 002176
 3010 006576 012737 020714 002200
 3011 006604 000261
 3012 006606 000401
 3013 006610 000241
 3014 006612 000205
 3015
 3016
 3017

```

.SBTTL ....CKROR -- CHECK RECEIVER OVERRUN
*****
* CKROR - THIS SUBROUTINE CHECKS FOR THE OCCURANCE OF RECEIVER OVERRUN IN THE
* USYRT RECEIVER STATUS REGISTER (RDSRH), AND REPORTS AN ERROR IF IT IS
* NOT PROPERLY SET TO THE STATE OF P' 0 IN THE WORD FOLLOWING THE CALL.
*
* CALLING SEQUENCE :
* JSR R5,CKROR
* .WORD <BIT 0 IS EXPECTED VALUE OF ROR>
*****
CKROR:
MOV #1,REGNUM ;SET REG NO. FOR POSSIBLE ERROR REPORT
JSR R5,READI ;READ RECEIVER STATUS
RDSRH
1$: .WORD 0
BIT #BIT0,(R5)+ ;GET EXPECTED STATE OF ROR
BEQ 2$ ;BR IF EXPECTED ROR = 0
BITB #ROR,1$ ;SEE IF ROR = 1
BNE 3$ ;BR IF ROR = 1
;STACK 'RECEIVER OVRN NOT SET' MSG
GTFD EM90,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 18
MOV #T.EDF,ERRTYP
MOV #18,ERRNBR
MOV #EM90,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
2$: BITB #ROR,1$ ;SEE IF ROR = 0
BEQ 3$ ;BR IF ROR = 0
;STACK 'ROR NOT CLEARED' MSG
GTFD EM91,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 19
MOV #T.EDF,ERRTYP
MOV #19,ERRNBR
MOV #EM91,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
3$: CLC ;CLEAR C BIT FOR NO ERRORS
4$: RTS R5 ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....CKSEOM -- CHECK RSOM, REOM

```

3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031 006614
3032 006614 012737 000007 002336
3033 006622 004537 003566
3034 006626 120401
3035 006630 000000
3036 006632 032725 000001
3037 006636 001422
3038 006640 132737 000001 006630
3039 006646 001040
3040
3041 006650
3042
3043 006650 012737 000001 002172
3044 006656 012737 000024 002174
3045 006664 012737 014427 002176
3046 006672 012737 020714 002200
3047 006700 000261
3048 006702 000473
3049 006704 132737 000001 006630
3050 006712 001416
3051
3052 006714
3053
3054 006714 012737 000001 002172
3055 006722 012737 000025 002174
3056 006730 012737 014406 002176
3057 006736 012737 020714 002200
3058 006744 000261
3059 006746 000451
3060 006750 032765 000002 177776
3061 006756 001422
3062 006760 132737 000002 006630
3063 006766 001040
3064
3065 006770
3066
3067 006770 012737 000001 002172
3068 006776 012737 000026 002174
3069 007004 012737 014465 002176
3070 007012 012737 020714 002200
3071 007020 000261
3072 007022 000423
3073 007024 132737 000002 006630

```

.SBTTLCKSEOM -- CHECK RSOM, REOM

```

*****
* CKSEOM - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF RSOM, REOM IN THE
* USYRT RECEIVER STATUS REG (RDSRH) AND REPORTS AN ERROR IF THEY ARE NOT
* PROPERLY SET TO THE STATES OF BITS 0,1 IN THE WORD FOLLOWING THE CALL.
* IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
* IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
* DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.

```

```

* CALLING SEQUENCE :
* JSR R5,CKSEOM
* <BIT 0 IS EXPECTED VALUE OF RSOM, BIT 1 IS VALUE OF REOM>
*****

```

```

CKSEOM:
MOV #7,REGNUM ;SET REG NO. FOR POSSIBLE ERROR REPORT
JSR R5,READI ;READ USYRT RECEIVER STATUS
RDSRH
1$: .WORD 0
BIT #BIT0,(R5)+ ;GET EXPECTED STATE OF RSOM
BEQ 2$ ;BR IF EXPECTED RSOM = 0
BITB #RSOM,1$ ;SEE IF RSOM = 1
BNE 3$ ;BR IF RSOM = 1
;STACK 'RSOM NOT SET' MSG
GTFD EM29,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 20
MOV #T.EDF,ERRTYP
MOV #20,ERRNBR
MOV #EM29,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 6$ ;TAKE ERROR EXIT
2$: BITB #RSOM,1$ ;SEE IF RSOM = 0
BEQ 3$ ;BR IF RSOM = 0
;STACK 'RSOM NOT CLEARED' MSG
GTFD EM28,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 21
MOV #T.EDF,ERRTYP
MOV #21,ERRNBR
MOV #EM28,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 6$ ;TAKE ERROR EXIT
3$: BIT #BIT1,-2(R5) ;GET EXPECTED STATE OF REOM
BEQ 4$ ;BR IF EXPECTED REOM = 0
BITB #REOM,1$ ;SEE IF REOM = 1
BNE 5$ ;BR IF REOM = 1
;STACK 'REOM NOT SET' MSG
GTFD EM31,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 22
MOV #T.EDF,ERRTYP
MOV #22,ERRNBR
MOV #EM31,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 6$ ;TAKE ERROR EXIT
4$: BITB #REOM,1$ ;SEE IF REOM = 0

```


CVDMEA.P11 10-DEC-80 09:16

....CKSEOM -- CHECK RSOM, REOM

3074 007032 001416
 3075
 3076 007034
 3077
 3078 007034 012737 000001 002172
 3079 007042 012737 000027 002174
 3080 007050 012737 014444 002176
 3081 007056 012737 020714 002200
 3082 007064 000261
 3083 007066 000401
 3084 007070 000241
 3085 007072 000205
 3086
 3087

BEQ 5\$
 ;STACK 'REOM NOT CLEARED' MSG
 GTDF EM30,ERR12

;BR IF REOM = 0

; QUEUE 'DEVICE FATAL' ERROR # 23

MOV #T.EDF,ERRTYP
 MOV #23,ERRNBR
 MOV #EM30,ERRMSG
 MOV #ERR12,ERRBLK

SEC
 BR 6\$
 5\$: CLC
 6\$: RTS R5

;SET C BIT TO FLAG ERROR
 ;TAKE ERROR EXIT
 ;CLEAR C BIT FOR NO ERRORS
 ;RETURN

CVDMEA.P11 10-DEC-80 09:16

....CHKTSO -- CHECK TRANSMIT SERIAL OUT BIT

```

3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098 007074
3099 007074 012737 000007 002336
3100 007102 004537 003566
3101 007106 122000
3102 007110 000000
3103 007112 032725 000001
3104 007116 001422
3105 007120 132737 000010 007110
3106 007126 001040
3107
3108 007130
3109
3110 007130 012737 000001 002172
3111 007136 012737 000030 002174
3112 007144 012737 015536 002176
3113 007152 012737 020714 002200
3114 007160 000261
3115 007162 000423
3116
3117 007164 132737 000010 007110 2$:
3118 007172 001416
3119
3120 007174
3121
3122 007174 012737 000001 002172
3123 007202 012737 000031 002174
3124 007210 012737 015556 002176
3125 007216 012737 020714 002200
3126 007224 000261
3127 007226 000401
3128 007230 000241 3$:
3129 007232 000205 4$:
3130

```

```

.SBTTL ....CHKTSO -- CHECK TRANSMIT SERIAL OUT BIT
*****
* CHKTSO - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TSO IN THE USYRT
* STATUS REGISTER, AND SETS THE 'C' BIT IF IT IS NOT SET TO THE STATE
* OF BIT 0 IN THE WORD FOLLOWING THE CALL.
*
* CALLING SEQUENCE :
* JSR R5,CHKTSO
* .WORD <BIT 0 IS EXPECTED VALUE OF TSO>
*****
CHKTSO:
MOV #7,REGNUM ;SET REG NO. FOR POSSIBLE ERROR REPORT
JSR R5,READI ;READ USYRT STATUS
USTATR
1$: .WORD 0
BIT #BIT0,(R5)+ ;GET EXPECTED STATE OF TSO
BEQ 2$ ;BR IF EXPECTED TSO = 0
BITB #TSO,1$ ;SEE IF TSO = 1
BNE 3$ ;BR IF TSO = 1
;*** STACK 'TSO NOT SET' ERROR ***
GTDF EM100,ERR12 ; QUEUE 'DEVICE FATAL' ERROR # 24
;
MOV #T.EDF,ERRTYP
MOV #24,ERRNBR
MOV #EM100,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
2$: BITB #TSO,1$ ;SEE IF TSO = 0
BEQ 3$ ;BR IF TSO = 0
;*** STACK 'TSO NOT CLEARED' ERROR ***
GTDF EM101,ERR12 ; QUEUE 'DEVICE FATAL' ERROR # 25
;
MOV #T.EDF,ERRTYP
MOV #25,ERRNBR
MOV #EM101,ERRMSG
MOV #ERR12,ERRBLK
SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
3$: CLC ;CLEAR C BIT FOR NO ERRORS
4$: RTS R5 ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

....INITRN -- INIT TRANSMISSION OF A MESSAGE

3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148 007234
3149 007234 010146
3150 007236 004537 003712
3151 007242 120000
3152 007244 000031
3153 007246 004537 003712
3154 007252 120000
3155 007254 000030
3156 007256 112537 007270
3157 007262 004537 003712
3158 007266 120404
3159 007270 000000 1\$:
3160 007272 112537 007304
3161 007276 004537 003712
3162 007302 120405
3163 007304 000000 2\$:
3164 007306 112537 007332
3165 007312 005037 002402
3166 007316 113737 007332 002402
3167 007324 004537 003712
3168 007330 120407
3169 007332 000000 3\$:
3170 007334 004537 003712
3171 007340 120013
3172 007342 000200
3173 007344 004537 003712
3174 007350 120006
3175 007352 000300
3176 007354 004537 003712
3177 007360 120607
3178 007362 000000
3179 007364 004537 005410
3180 007370 000110
3181 007372 103454
3182
3183 007374 013737 007530 007414
3184 007402 142537 007414
3185
3186 007406 004537 003712

```

.SBTTL ....INITRN -- INIT TRANSMISSION OF A MESSAGE
*****
* INITRN - THIS SUBROUTINE INITIATES TRANSMISSION OF A MESSAGE, BY LOADING
* THE USYRT PCSARL,H AND THE PCR WITH THE DATA PASSED IN THE 2 WORDS
* FOLLOWING THE CALL ; LOADING AND CLOCKING 1 SOM UNTIL THE FIRST
* SYNCH OR FLAG HAS BEEN SERIALIZED IN THE USYRT. THE PROGRAM MONITORS
* ALL THE FLAGS IN THE USYRT STATUS REGISTER THROUGHOUT THE PROCESS.
* IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION IS STACKED
* AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE DISCRETION
* OF THE CALLING ROUTINE OR SUBROUTINE.
*
* CALLING SEQUENCE :
* JSR R5,INITRN
* .WORD <VALUE TO LOAD INTO USYRT PCSARL,H>
* .WORD <VALUE TO LOAD INTO USYRT PCR (PASSED IN LO BYTE)>
* <SPECIAL VIAORB MASKING VALUE (PASSED IN HI BYTE)>
*****
INITRN:
MOV R1,-(SP) ;SAVE R1
JSR R5,WRITEI ;RESET THE USYRT
VIAORB
RTSND!DTR!PRESET
JSR R5,WRITEI ;CLEAR USYRT RESET BIT
VIAORB
RTSND!DTR
MOVB (R5)+,1$ ;GET VALUE TO LOAD INTO USYRT PCSARL
JSR R5,WRITEI ;LOAD USYRT PCSARL
PCSARL
1$:
.WORD 0
MOVB (R5)+,2$ ;GET VALUE TO LOAD INTO PCSARH
JSR R5,WRITEI ;LOAD USYRT PCSARH
PCSARH
2$:
.WORD 0
MOVB (R5)+,3$ ;GET VALUE TO LOAD INTO PCR
CLR SAVLEN
MOVB 3$,SAVLEN ;SAVE CHAR LENGTH BITS
JSR R5,WRITEI ;LOAD USYRT PCR
PCR
3$:
.WORD 0
JSR R5,WRITEI ;SET ACR FOR T1 ONE-SHOT MODE
VIAACR
200
JSR R5,WRITEI ;LOAD VIA T1L-L
VIAT1C
300
JSR R5,WRITEI ;LOAD VIA T1L-H
VIAT1D
000
JSR R5,CKUSTS ;CHK USYRT STATUS FOR INIT'D STATE
110 ; TBMT = 1, TSO = 1
BCS 7$ ;IF ERROR, EXIT SUBROUTINE

MOV 20$,13$ ;* SET UP DEFAULT VIAORB PARAMETERS
BICB (R5)+,13$ ;* CLEAR ANY SPECIFIED VIAORB BITS.

JSR R5,WRITEI ;SET UP USYRT

```

CVDMEA.P11 10-DEC-80 09:16

....INITRN -- INIT TRANSMISSION OF A MESSAGE

```

3187 007412 120000
3188 007414 000142
3189
3190 007416 004537 003712
3191 007422 120403
3192 007424 000001
3193 007426 004537 003712
3194 007432 120402
3195 007434 000226
3196 007436 004537 006014
3197 007442 000000
3198 007444 103427
3199 007446 005001
3200 007450 004537 012072
3201 007454 000001
3202 007456 004537 003566
3203 007462 122000
3204 007464 000000
3205 007466 132737 000100 007464
3206 007474 001010
3207 007476 005201
3208 007500 020127 000003
3209 007504 002761
3210 007506 004537 006014
3211 007512 000001
3212 007514 103403
3213 007516 004537 005514
3214 007522 000001
3215 007524 012601
3216 007526 000205
3217
3218 007530 000142
3219
3220

```

```

VIAORB
13$: TXEN!RXEN!TTLOOP ;* THIS VALUE MIGHT BE MODIFIED ABOVE
      JSR R5,WRITEI ;SET TSOM IN USYRT
      TDSRH
      TSOM
      JSR R5,WRITEI ;LOAD SYNCH CHAR INTO TX BUF
      TDSRL
      SYNCH
      JSR R5,CKTBMT ;CHK FOR TBMT = 0
      0
      BCS 7$ ;IF ERROR, EXIT SUBROUTINE
      CLR R1 ;INIT CYCLE COUNTER
4$: JSR R5,STEPLU ;CLOCK LU FOR 1 CYCLE
      1
      JSR R5,READI ;READ USYRT STATUS REG
      USTATR
5$: .WORD 0
      BITB #TBMT,5$ ;SEE IF TBMT IS SET YET
      BNE 6$ ;BR IF YES
      INC R1 ;INCR CYCLE COUNTER
      CMP R1,#3 ;SEE IF 3 CYCLES DONE YET
      BLT 4$ ;BR IF LESS THAN 3 CYCLES
      JSR R5,CKTBMT ;GO STACK 'TBMT NOT SET' MSG
      1
      BCS 7$ ;IF ERROR, EXIT SUBROUTINE
6$: JSR R5,CKTACT ;CHK FOR TFACT = 1
      1
7$: MOV (SP)+,R1 ;RESTORE R1
      RTS R5 ;RETURN (IF C = 1, WE HAD AN ERROR)
20$: TXEN!RXEN!TTLOOP ;DEFAULT VALUE FOR VIAORB: ENABLE
      ;TX AND RX ON USYRT, ASSERT RTS, DTR

```

CVDMEA.P11 10-DEC-80 09:16

....CKLPBK -- DETERMINE IF TEST CAN BE RUN

3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242 007532
3243 007532 032725 100000
3244 007536 001407
3245 007540 005737 002472
3246 007544 001002
3247 007546 000137 010072
3248 007552 000137 010076
3249
3250 007556 023727 002472 000004
3251 007564 001002
3252 007566 000137 010076
3253
3254
3255
3256 007572 026527 177776 000001
3257 007600 001406
3258 007602 026527 177776 000002
3259 007610 001422
3260 007612 000137 007742
3261
3262
3263 007616 005737 002470
3264 007622 001402
3265 007624 000137 010040
3266 007630 005737 002472
3267 007634 001406
3268 007636 023727 002472 000001
3269 007644 001402
3270 007646 000137 010006
3271 007652 000137 010072
3272
3273
3274 007656 005737 002470
3275 007662 001002
3276 007664 000137 010040

```

.SBTTL ....CKLPBK -- DETERMINE IF TEST CAN BE RUN
*****
* CKLPBK - THIS SUBROUTINE DETERMINES IF THE TEST CALLING IT CAN BE RUN. THE
* TEST PASSES THE DESIRED MODEM INTERFACE TYPE IN THE WORD FOLLOWING THE
* CALL, AND IF A PROPER EXTERNAL LOOPBACK HAS BEEN PROVIDED BY THE
* OPERATOR FOR THAT INTERFACE, AND IF THE BAUD RATE IS CORRECT, A RETURN
* IS MADE WITH THE C BIT CLEARED, TO RUN THE TEST. IF NOT, A RETURN IS
* MADE WITH THE C BIT SET TO 1, SO THAT THE TEST CAN BE SKIPPED.
*
* IF BIT 15 IS SET IN THE WORD FOLLOWING THE CALL, THE TEST WILL NOT
* BE RUN UNLESS THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED.
*
* IF THE PROGRAM PASSES '0' IN THE WORD FOLLOWING THE CALL, THE SUBRTN
* WILL ATTEMPT TO RUN WHICHEVER MODEM INTERFACE IS SELECTED BY CABLE
* OR TEST CONNECTOR.
*
* CALLING SEQUENCE :
*   JSR R5,CKLPBK
*   .WORD <DESIRED MODEM INTERFACE INFO>
*****
CKLPBK:
      BIT      #ITCCHK,(R5)+ ;SEE IF H3254,5 CHECK IS DESIRED
      BEQ      2$           ;BR IF NOT
      TST      TSTCON      ;SEE IF H3254,5 INSTALLED
      BNE      1$           ;BR IF NOT
      JMP      46$          ;BR TO RUN TEST
1$:    JMP      48$          ;GO TO SKIP TEST
      ;IF NO EXTERNAL LPBK, SKIP TEST
2$:    CMP      TSTCON,#4   ;SEE IF NO LPBK
      BNE      3$           ;BR IF LOOPBACK
      JMP      48$          ;GO TO SKIP TEST

      ;*** SEE IF AN INTERFACE IS REQUESTED ***
3$:    CMP      -2(R5),#INTGRL ;SEE IF INTEGRAL MODEM REQUESTED
      BEQ      8$           ;BR IF INTGRL MODEM REQUESTED
      CMP      -2(R5),#EIAV35 ;SEE IF V.35 OR EIA REQUESTED
      BEQ      16$          ;BR IF V.35 REQUESTED
      JMP      32$          ;NONE REQUESTED, FIND AN INTERFACE TO TEST

      ;SEE IF INTEGRAL MODEM CAN BE RUN
8$:    TST      BRDTYP      ;SEE IF M8064
      BEQ      10$          ;BR IF M8064
      JMP      42$          ;WRONG OPTION, GO TO SKIP TEST
10$:   TST      TSTCON      ;SEE IF H3254, H3255 USED
      BEQ      12$          ;BR IF YES
15$:   CMP      TSTCON,#1   ;SEE IF OPERATOR SPEC'D INTEGRAL MODEM
      BEQ      12$          ;BR IF YES, TO RUN TEST
      JMP      40$          ;WRONG INTERFACE, GO SKIP TEST
12$:   JMP      46$          ;GO TO RUN TEST

      ;SEE IF V.35 OR EIA CAN BE RUN
16$:   TST      BRDTYP      ;SEE IF M8053 BOARD
      BNE      18$          ;BR IF M8053
      JMP      42$          ;WRONG OPTION, GO TO SKIP TEST

```

CVDMEA.P11 10-DEC-80 09:16

....CKLPBK -- DETERMINE IF TEST CAN BE RUN

```

3277 007670 005737 002472 18$: TST TSTCON ;SEE IF H3254, H3255 USED
3278 007674 001002 BNE 23$ ;BR IF NOT
3279 007676 000137 010072 20$: JMP 46$ ;GO RUN THE TEST
3280 007702 023727 002472 000003 23$: CMP TSTCON,#3 ;SEE IF OPERATOR SPEC'D V.35
3281 007710 001006 BNE 28$ ;BR IF NO
3282 007712 023727 002470 000001 CMP BRDTYP,#1 ;TSTCOM MATCH BRDTYP?
3283 007720 001766 BEQ 20$ ;YES: RUN TEST
3284 007722 000137 010006 JMP 40$ ;WRONG INTERFACE, GO SKIP TEST
3285
3286 007726 023727 002472 000002 28$: CMP TSTCON,#2 ;SEE IF OPERATOR SPEC'D EIA
3287 007734 001760 BEQ 20$ ;BR IF YES, TO RUN EIA
3288 007736 000137 010006 JMP 40$ ;WRONG INTERFACE, GO SKIP TEST
3289
3290 ;*** NO INTERFACE REQUESTED - FIND ONE TO TEST ***
3291
3292 007742 005737 002470 37$: TST BRDTYP ;SEE IF INTEGRAL MODEM SELECTED
3293 007746 001343 BNE 16$ ;BR IF NOT (TEST FOR V35/EIA)
3294 007750 000137 007652 JMP 12$ ;SEE IF INTEGRAL MODEM CAN BE RUN
3295
3296 ;PRINT 'FOR BAUD RATE SPECIFIED,'
3297 007754 38$:
3298 007754 023727 002412 000001 CMP STARES,#1 ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3299 007762 001063 BNE 50$ ;BR IF NOT, TO SKIP PRINTING
3300 007764 PRINTF #FMT30
3301 007764 012746 013762 MOV #FMT30,-(SP)
3302 007770 012746 000001 MOV #1,-(SP)
3303 007774 010600 MOV SP,R0
3304 007776 104417 TRAP C$PNTF
3305 010000 062706 000004 ADD #4,SP
3306 010004 000434 BR 48$ ;GO TO PRINT 'TEST NOT RUN'
3307 ;PRINT 'IMPROPER CONNECTOR TYPE SPECIFIED'
3308 010006 40$:
3309 010006 023727 002412 000001 CMP STARES,#1 ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3310 010014 001046 BNE 50$ ;BR IF NOT, TO SKIP PRINTING
3311 010016 PRINTF #FMT31 ;*****
3312 010016 012746 014017 MOV #FMT31,-(SP)
3313 010022 012746 000001 MOV #1,-(SP)
3314 010026 010600 MOV SP,R0
3315 010030 104417 TRAP C$PNTF
3316 010032 062706 000004 ADD #4,SP
3317 010036 000417 BR 48$ ;GO TO PRINT 'TEST NOT RUN'
3318
3319 ;PRINT 'FOR OPTION SPECIFIED,'
3320 010040 023727 002412 000001 42$: CMP STARES,#1 ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3321 010046 001031 BNE 50$ ;BR IF NOT TO SKIP PRINTING
3322 010050 PRINTF #FMT32
3323 010050 012746 014065 MOV #FMT32,-(SP)
3324 010054 012746 000001 MOV #1,-(SP)
3325 010060 010600 MOV SP,R0
3326 010062 104417 TRAP C$PNTF
3327 010064 062706 000004 ADD #4,SP
3328 010070 000402 BR 48$ ;GO TO PRINT 'TEST NOT RUN'
3329
3330 ;*** BRANCH HERE TO RUN TEST ***
3331 010072 000241 46$: CLC ;CLEAR C BIT TO RUN TEST
3332 010074 000417 BR 52$ ;EXIT

```

CVDMEA.P11 10-DEC-80 09:16

....CKLPBK -- DETERMINE IF TEST CAN BE RUN

3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350

*** BRANCH HERE TO SKIP TEST ***
;PRINT 'TEST XX NOT RUN'

48\$:
CMP STARES,#1 ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
BNE 50\$;BR IF NOT, TO SKIP PRINTING
PRINTF #FMT19,TSTNUM

MOV TSTNUM,-(SP)
MOV #FMT19,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTF
ADD #6,SP

50\$: SEC ;SET C BIT TO SKIP TEST
52\$: RTS R5 ;RETURN

CVDMEA.P11 10-DEC-80 09:16

....TXCHAR -- TRANSMIT A CHARACTER

3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368 010136
3369 010136 010146
3370 010140 010246
3371 010142 012537 010154
3372 010146 004537 003712
3373 010152 120402
3374 010154 000000
3375 010156 005001
3376 010160 005002
3377 010162 112502
3378 010164 001425
3379 010166 004537 005514
3380 010172 000001
3381 010174 103421
3382 010176 020102
3383 010200 001414
3384
3385 010202 131527 000200
3386 010206 001004
3387
3388 010210 004537 006014
3389 010214 000000
3390 010216 103410
3391 010220 004537 012072
3392 010224 000001
3393 010226 005201
3394 010230 000756
3395 010232 004537 006014
3396 010236 000001
3397 010240 012602
3398 010242 012601
3399 010244 005205
3400 010246 000205
3401
3402
3403
3404

```

.SBTTL ....TXCHAR -- TRANSMIT A CHARACTER
*****
* TXCHAR - THIS SUBROUTINE INITIATES TRANSMISSION OF A CHAR BY LOADING
* THE USYRT TDSRL WITH THE DATA PASSED IN THE LO BYTE OF THE WORD
* FOLLOWING THE CALL, AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES
* PASSED IN THE SECOND WORD FOLLOWING THE CALL. THE PROGRAM CONTINUALLY
* MONITORS TBMT AND TXACT THROUGHOUT THE PROCESS.
* IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
* IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
* DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
*
* CALLING SEQUENCE :
* JSR R5,TXCHAR
* .WORD <DATA FOR TDSRL IN LO BYTE>
* .WORD <NUMBER OF CYCLES TO CLOCK (IN LO BYTE)>
* <SWITCH TO DISABLE INITIAL TBMT=0 CHECK (MSB IN HI BYTE)>
*****
TXCHAR:
MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2
MOV (R5)+,1$ ;GET DATA FOR TDSRL
JSR R5,WRITEI ;LOAD DATA INTO TDSRL
TDSRL
1$: .WORD 0
CLR R1 ;INIT CYCLE COUNT AND CLEAR C BIT
CLR R2 ;CLEAR REQ'D CYCLE COUNT
MOVB (R5)+,R2 ;GET DESIRED NO. OF CYCLES
BEQ 6$ ;BR IF NO CLOCKING DONE
3$: JSR R5,CKTACT ;CHECK TXACT = 1
1
BCS 6$ ;BR TO EXIT IF ERROR
CMP R1,R2 ;SEE IF REQUIRED CYCLES DONE YET
BEQ 5$ ;BR IF YES
3385 BITB (R5),#NCTBMT ;* CHECK FOR 'TBMT=0 CHECK' DISABLE
3386 BNE 7$ ;* BR IF MSB IS NOT SET
JSR R5,CKTBMT ;CHECK FOR TBMT = 0
0
BCS 6$ ;BR TO EXIT IF ERROR
7$: JSR R5,STEPLU ;CLOCK LU FOR 1 CYCLE
1
INC R1 ;INCR CYCLE COUNT
BR 3$ ;KEEP CLOCKING
5$: JSR R5,CKTBMT ;CHK TBMT = 1
1
6$: MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
INC R5 ;ADJUST R5 FOR SAME RETURN
RTS R5 ;RETURN (WITH C BIT = 1 IF ERROR)

```


CVDMEA.P11 10-DEC-80 09:16

....TXCTRL -- CONTROL MESSAGE TRANSMISSION (TDSRH)

3405
3406
3407
3408
3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421 010250
3422 010250 010146
3423 010252 010246
3424 010254 012537 010266
3425 010260 004537 003712
3426 010264 120403
3427 010266 000000
3428 010270 005001
3429 010272 012502
3430 010274 001422
3431 010276 004537 005514
3432 010302 000001
3433 010304 103416
3434 010306 020102
3435 010310 001411
3436 010312 004537 006014
3437 010316 000000
3438 010320 103410
3439 010322 004537 012072
3440 010326 000001
3441 010330 005201
3442 010332 000761
3443 010334 004537 006014
3444 010340 000001
3445 010342 012602
3446 010344 012601
3447 010346 000205
3448
3449

```
.SBTTL ....TXCTRL -- CONTROL MESSAGE TRANSMISSION (TDSRH)
*****
* TXCTRL - THIS SUBROUTINE ALLOWS CONTROL OF MESSAGE TRANSMISSION BY LOADING
* THE USYRT TDSRH WITH THE DATA PASSED IN THE LO BYTE OF THE WORD
* FOLLOWING THE CALL, AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES
* PASSED IN THE SECOND WORD FOLLOWING THE CALL. THE PROGRAM CONTINUALLY
* MONITORS TBMT AND TXACT THROUGHOUT THE PROCESS.
* IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
* IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
* DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
*
* CALLING SEQUENCE :
* JSR R5,TXCTRL
* .WORD <DATA FOR TDSRH IN LO BYTE>
* .WORD <NUMBER OF CYCLES TO CLOCK>
*****
TXCTRL:
MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2
MOV (R5)+,2$ ;GET DATA FOR TDSRH
JSR R5,WRITEI ;LOAD DATA INTO TDSRH
TDSRH
2$: .WORD 0
CLR R1 ;INIT CYCLE COUNT AND CLEAR C BIT
MOV (R5)+,R2 ;GET DESIRED NO. OF CYCLES
BEQ 6$ ;BR IF NO CLOCKING DONE
3$: JSR R5,CKTACT ;CHECK TXACT = 1
1
BCS 6$ ;BR TO EXIT IF ERROR
CMP R1,R2 ;SEE IF REQUIRED CYCLES DONE YET
BEQ 5$ ;BR IF YES
JSR R5,CKTBMT ;CHECK FOR TBMT = 0
0
BCS 6$ ;BR TO EXIT IF ERROR
JSR R5,STEPLU ;CLOCK LU FOR 1 CYCLE
1
INC R1 ;INCR CYCLE COUNT
BR 3$ ;KEEP CLOCKING
5$: JSR R5,CKTBMT ;CHK TBMT = 1
1
6$: MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
RTS R5 ;RETURN (WITH C BIT = 1 IF ERROR)
```

CVDMEA.P11 10-DEC-80 09:16

....RXCHAR -- RECEIVE A CHARACTER

3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469 010350
3470 010350 010146
3471 010352 010246
3472 010354 004537 003566
3473 010360 120401
3474 010362 000000
3475 010364 004537 003566
3476 010370 120400
3477 010372 000000
3478 010374 111501
3479 010376 042701 177400
3480 010402 023727 002402 000347
3481 010410 001005
3482 010412 142737 000200 010372
3483 010420 ;.2701 000200
3484 010424 123701 010372
3485 010430 001462
3486 010432 004537 003566
3487 010436 122000
3488 010440 000000
3489 010442 132737 000002 010440
3490 010450 001421
3491 010452 012737 000007 002336
3492
3493 010460
3494
3495 010460 012737 000001 002172
3496 010466 012737 000032 002174
3497 010474 012737 014626 002176
3498 010502 012737 020714 002200
3499 010510 000137 011610
3500 010514 005037 002336
3501 010520 005037 002324
3502 010524 110137 002324
3503 010530 005037 002326
3504 010534 113737 010372 002326
3505

.SBTTLRXCHAR -- RECEIVE A CHARACTER

 * RXCHAR - THIS SUBROUTINE READS THE USYRT RDSR AND CHECKS THE CONTENTS
 * AGAINST THE DATA PASSED IN THE WORD FOLLOWING THE CALL.
 * IF BIT0 = 0 IN THE SECOND WORD FOLLOWING THE CALL, THE RERR BIT IS
 * NOT CHECKED AGAINST THE EXPECTED VALUE. THEN, IT CLOCKS
 * THE LINE UNIT FOR THE NO. OF CYCLES PASSED IN THE THIRD WORD
 * FOLLOWING THE CALL. THE PROGRAM CONTINUALLY MONITORS RDA AND RXACT.
 * IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
 * IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
 * DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
 *
 * CALLING SEQUENCE :
 * JSR R5,RXCHAR
 * .WORD <EXPECTED RDSRL IN LO BYTE, RDSRH IN HI BYTE>
 * .WORD <=0 FOR NO RERR CHK, =1 FOR RERR CHK>
 * .WORD <NUMBER OF CYCLES TO CLOCK (IN LO BYTE)>
 * <SPECIAL DISABLE SWITCHES: NOCRDA,NFCRDA,NCRACK(IN HI BYTE)>

RXCHAR:
 MOV R1,-(SP) ;SAVE R1
 MOV R2,-(SP) ;SAVE R2
 JSR R5,READI ;READ RDSRH
 RDSRH
 2\$: .WORD 0
 JSR R5,READI ;READ RDSRL
 RDSRL
 1\$: .WORD 0
 MOVB (R5),R1 ;GET EXPECTED RDSRL
 BIC #177400,R1 ;MASK OFF UNUSED BITS
 CMP SAVLEN,#TXDL!RXDL ;SEE IF 7-BIT CHARS BEING USED
 BNE 3\$;BR IF NOT 7-BIT CHARS
 BICB #BIT7,1\$;CLEAR 8TH BIT FOR COMPARE
 BICB #BIT7,R1
 3\$: CMPB 1\$,R1 ;COMPARE RCV'D CHAR TO EXPECTED
 BEQ 6\$;BR IF MATCH
 JSR R5,READI ;READ USYRT STATUS REG
 USTATR
 4\$: .WORD 0
 BITB #TXU,4\$;SEE IF TX UNDERRUN OCCURRED
 BEQ 5\$;BR IF NOT
 MOV #7,REGNUM ;SET USYRT REG NO. FOR STATUS REG
 ;STACK 'TX UNDERRUN' ERROR
 GTDF EM54,ERR12
 ; QUEUE 'DEVICE FATAL' ERROR # 26
 MOV #T.EDF,ERRTYP
 MOV #26,ERRNBR
 MOV #EM54,ERRMSG
 MOV #ERR12,ERRBLK
 5\$: JMP 20\$;TAKE ERROR EXIT
 CLR REGNUM ;SET USYRT REG NO. FOR RDSRL
 CLR GDATA ;SET EXPECTED DATA
 MOVB R1,GDATA
 CLR BDATA ;SET ACTUAL DATA
 MOVB 1\$,BDATA
 ;STACK 'RCV'D DATA MISCOMPARE' ERROR

CVDMEA.P11 10-DEC-80 09:16

....RXCHAR -- RECEIVE A CHARACTER

```

3506 010542          GTDF      EM34,ERR10
3507                                     ;      QUEUE 'DEVICE FATAL' ERROR # 27
3508 010542 012737 000001 002172                                     MOV      #T.EDF,ERRTYP
3509 010550 012737 000033 002174                                     MOV      #27,ERRNBR
3510 010556 012737 014502 002176                                     MOV      #EM34,ERRMSG
3511 010564 012737 020364 002200                                     MOV      #ERR10,ERRBLK
3512 010572 000137 011610
3513 010576 116501 000001      6$:  JMP      20$      ;TAKE ERROR EXIT
3514 010602 042701 177400      MOVB     1(R5),R1  ;GET RDSRH
3515 010606 123701 010362      BIC     #177400,R1 ;MASK OFF UNUSED BITS
3516 010612 001016      CMPB     2$,R1    ;COMPARE RCV'D STATUS TO EXPECTED
3517 010614 000137 011474      BNE     7$      ;BR IF MISMATCH
3518 010620 012737 000001 002336      JMP     17$     ;CONTINUE
3519 010626 005037 002324      MOV     #1,REGNUM ;SET USYRT REG NO. FOR RDSRH
3520 010632 110137 002324      CLR     GDATA   ;SET EXPECTED DATA
3521 010636 005037 002326      MOVB   R1,GDATA
3522 010642 113737 010362 002326      CLR     BDATA   ;SET ACTUAL DATA
3523 010650 012737 000001 002336      MOVB   2$,BDATA
3524 010656 032765 000001 000002      7$:  MOV     #1,REGNUM ;SET REG NO. FOR PRINTOUT
3525 010664 001447      BIT     #RERRCHK,2(R5) ;SEE IF RCV ERROR BIT SHOULD BE IGNORED
3526                                     BEQ     9$      ;BR IF YES
3527 010666 132701 000200      ;CHECK RERR BIT
3528 010672 001022      BITB   #RERR,R1  ;SEE IF EXPECTED BIT = 1
3529 010674 132737 000200 010362      BNE     8$      ;BR IF YES
3530 010702 001440      BITB   #RERR,2$  ;SEE IF ACTUAL BIT = 0
3531                                     BEQ     9$      ;BR IF YES
3532 010704      ;STACK 'RERR NOT CLEARED' MSG
3533      GTDF      EM35,ERR12
3534                                     ;      QUEUE 'DEVICE FATAL' ERROR # 28
3535 010704 012737 000001 002172                                     MOV      #T.EDF,ERRTYP
3536 010712 012737 000034 002174                                     MOV      #28,ERRNBR
3537 010720 012737 014530 002176                                     MOV      #EM35,ERRMSG
3538 010726 012737 020714 002200                                     MOV      #ERR12,ERRBLK
3539 010734 000137 011610
3540 010740 132737 000200 010362      8$:  JMP     20$     ;TAKE ERROR EXIT
3541 010746 001016      BITB   #RERR,2$  ;SEE IF ACTUAL BIT = 1
3542                                     BNE     9$      ;BR IF YES
3543      ;STACK 'RERR NOT SET' MSG
3544      GTDF      EM36,ERR12
3545                                     ;      QUEUE 'DEVICE FATAL' ERROR # 29
3546 010750 012737 000001 002172                                     MOV      #T.EDF,ERRTYP
3547 010756 012737 000035 002174                                     MOV      #29,ERRNBR
3548 010764 012737 014551 002176                                     MOV      #EM36,ERRMSG
3549 010772 012737 020714 002200                                     MOV      #ERR12,ERRBLK
3550 011000 000137 011610
3551      ;CHECK ROR BIT
3552 011004 132701 000010      9$:  JMP     20$     ;TAKE ERROR EXIT
3553 011010 001022      BITB   #ROR,R1  ;SEE IF EXPECTED BIT = 1
3554 011012 132737 000010 010362      BNE     10$     ;BR IF YES
3555 011020 001440      BITB   #ROR,2$  ;SEE IF ACTUAL BIT = 0
3556                                     BEQ     11$     ;BR IF YES
3557      ;STACK 'ROR NOT CLEARED' MSG
3558      GTDF      EM16,ERR12
3559                                     ;      QUEUE 'DEVICE FATAL' ERROR # 30
3560 011022 012737 000001 002172                                     MOV      #T.EDF,ERRTYP
3561 011030 012737 000036 002174                                     MOV      #30,ERRNBR
3562 011036 012737 014340 002176                                     MOV      #EM16,ERRMSG
3563 011044 012737 020714 002200                                     MOV      #ERR12,ERRBLK
3564 011052 000137 011610      JMP     20$     ;TAKE ERROR EXIT

```

CVDMPA.P11 10-DEC-80 09:16

....RXCHAR -- RECEIVE A CHARACTER

```

3562 011056 132737 000010 010362 10$: BITB #ROR,2$ ;SEE IF ACTUAL BIT = 1
3563 011064 001016 ;BNE 11$ ;BR IF YES
3564 ;STACK 'ROR NOT SET' MSG
3565 011066 GTDF EM14,ERR12
3566 ; QUEUE 'DEVICE FATAL' ERROR # 31
3567 011066 012737 000001 002172 MOV #T.EDF,ERRTYP
3568 011074 012737 000037 002174 MOV #31,ERRNBR
3569 011102 012737 014324 002176 MOV #EM14,ERRMSG
3570 011110 012737 020714 002200 MOV #ERR12,ERRBLK
3571 011116 000137 011610 ;TAKE ERROR EXIT
3572 ;CHECK RABGA BIT
3573 011122 132701 000004 11$: BITB #RABGA,R1 ;SEE IF EXPECTED BIT = 1
3574 011126 001022 ;BNE 12$ ;BR IF YES
3575 011130 132737 000004 010362 BITB #RABGA,2$ ;SEE IF ACTUAL BIT = 0
3576 011136 001440 ;BEQ 13$ ;BR IF YES
3577 *;STACK 'RABGA NOT CLEARED' MSG
3578 011140 GTDF EM39,ERR12
3579 ; QUEUE 'DEVICE FATAL' ERROR # 32
3580 011140 012737 000001 002172 MOV #T.EDF,ERRTYP
3581 011146 012737 000040 002174 MOV #32,ERRNBR
3582 011154 012737 014566 002176 MOV #EM39,ERRMSG
3583 011162 012737 020714 002200 MOV #ERR12,ERRBLK
3584 011170 000137 011610 ;TAKE ERROR EXIT
3585 011174 132737 000004 010362 12$: BITB #RABGA,2$ ;SEE IF ACTUAL BIT = 1
3586 011202 001016 ;BNE 13$ ;BR IF YES
3587 ;STACK 'RABGA NOT SET' MSG
3588 011204 GTDF EM40,ERR12
3589 ; QUEUE 'DEVICE FATAL' ERROR # 33
3590 011204 012737 000001 002172 MOV #T.EDF,ERRTYP
3591 011212 012737 000041 002174 MOV #33,ERRNBR
3592 011220 012737 014610 002176 MOV #EM40,ERRMSG
3593 011226 012737 020714 002200 MOV #ERR12,ERRBLK
3594 011234 000137 011610 ;TAKE ERROR EXIT
3595 ;CHECK REOM BIT
3596 011240 132701 000002 13$: BITB #REOM,R1 ;SEE IF EXPECTED BIT = 1
3597 011244 001022 ;BNE 14$ ;BR IF YES
3598 011246 132737 000002 010362 BITB #REOM,2$ ;SEE IF ACTUAL BIT = 0
3599 011254 001440 ;BEQ 15$ ;BR IF YES
3600 ;STACK 'REOM NOT CLEARED' MSG
3601 011256 GTDF EM30,ERR12
3602 ; QUEUE 'DEVICE FATAL' ERROR # 34
3603 011256 012737 000001 002172 MOV #T.EDF,ERRTYP
3604 011264 012737 000042 002174 MOV #34,ERRNBR
3605 011272 012737 014444 002176 MOV #EM30,ERRMSG
3606 011300 012737 020714 002200 MOV #ERR12,ERRBLK
3607 011306 000137 011610 ;TAKE ERROR EXIT
3608 011312 132737 000002 010362 14$: BITB #REOM,2$ ;SEE IF ACTUAL BIT = 1
3609 011320 001016 ;BNE 15$ ;BR IF YES
3610 ;STACK 'REOM NOT SET' MSG
3611 011322 GTDF EM31,ERR12
3612 ; QUEUE 'DEVICE FATAL' ERROR # 35
3613 011322 012737 000001 002172 MOV #T.EDF,ERRTYP
3614 011330 012737 000043 002174 MOV #35,ERRNBR
3615 011336 012737 014465 002176 MOV #EM31,ERRMSG
3616 011344 012737 020714 002200 MOV #ERR12,ERRBLK
3617 011352 000137 011610 JMP 20$ ;TAKE ERROR EXIT

```

CVDMEA.P11 10-DEC-80 09:16

....RXCHAR -- RECEIVE A CHARACTER

```

3618
3619 011356 132701 000001
3620 011362 001022
3621 011364 132737 000001 010362
3622 011372 001440
3623
3624 011374
3625
3626 011374 012737 000001 002172
3627 011402 012737 000044 002174
3628 011410 012737 014406 002176
3629 011416 012737 020714 002200
3630 011424 000137 011610
3631 011430 132737 000001 010362
3632 011436 001016
3633
3634 011440
3635
3636 011440 012737 000001 002172
3637 011446 012737 000045 002174
3638 011454 012737 014427 002176
3639 011462 012737 020714 002200
3640 011470 000137 011610
3641
3642 011474 116502 000004
3643 011500 005001
3644
3645 011502 136527 000005 000040
3646 011510 001004
3647 011512 004537 005654
3648 011516 000001
3649 011520 103433
3650
3651 011522 020102
3652 011524 001415
3653
3654 011526 136527 000005 000200
3655 011534 001004
3656 011536 004537 006154
3657 011542 000000
3658 011544 103421
3659
3660 011546 004537 012072
3661 011552 000001
3662 011554 005201
3663 011556 000751
3664
3665 011560 136527 000005 000100
3666 011566 001004
3667 011570 004537 006154
3668 011574 000001
3669 011576 103404
3670
3671 011600 062705 000006
3672 011604 000241
3673 011606 000403

;CHECK RSOM BIT
15$: BITB #RSOM,R1 ;SEE IF EXPECTED BIT = 1
      BNE 16$ ;BR IF YES
      BITB #RSOM,2$ ;SEE IF ACTUAL BIT = 0
      BEQ 17$ ;BR IF YES
;STACK 'RSOM NOT CLEARED' MSG
      GTDF EM28,ERR12
;
; QUEUE 'DEVICE FATAL' ERROR # 36
      MOV #T.EDF,ERRTYP
      MOV #36,ERRNBR
      MOV #EM28,ERRMSG
      MOV #ERR12,ERRBLK
;TAKE ERROR EXIT
16$: JMP 20$ ;SEE IF ACTUAL BIT = 1
      BITB #RSOM,2$ ;BR IF YES
      BNE 17$
;STACK 'RSOM NOT SET' MSG
      GTDF EM29,ERR12
;
; QUEUE 'DEVICE FATAL' ERROR # 37
      MOV #T.EDF,ERRTYP
      MOV #37,ERRNBR
      MOV #EM29,ERRMSG
      MOV #ERR12,ERRBLK
;TAKE ERROR EXIT
      JMP 20$
17$: MOVB 4(R5),R2 ;GET DESIRED NO. OF CYCLES
      CLR R1 ;INIT CYCLE COUNT
18$: BITB 5(R5),#BIT5 ;* IS RXACT CHECK TO BE DISABLED ?
      BNE 31$ ;* BR IF YES
      JSR R5,CKRACT ;CHK FOR RACT = 1
      T 1
      BCS 20$ ;BR TO EXIT IF ERROR
31$: CMP R1,R2 ;SEE IF REQUIRED CYCLES DONE YET
      BEQ 19$ ;BR IF YES
      BITB 5(R5),#BIT7 ;* SEE IF INITIAL RDA CHECK DESIRED
      BNE 22$ ;* BR IF NO
      JSR R5,CKRDA ;CHK FOR RDA = 0
      O 0
      BCS 20$ ;BR TO EXIT IF ERROR
22$: JSR R5,STEPLU ;CLOCK LU FOR 1 CYCLE
      T 1
      INC R1 ;INCR CYCLE COUNT
      BR 18$ ;CONTINUE CLOCKING
19$: BITB 5(R5),#BIT6 ;* IS FINAL RDA CHECK TO BE SKIPPED ?
      BNE 30$ ;* BR IF YES
      JSR R5,CKRDA ;CHK RDA = 1
      T 1
      BCS 20$ ;BR IF ERROR
30$: ADD #6,R5 ;FIX UP RETURN ADRS
      CLC ;SET C = 0 FOR NO ERROR
      BR 21$ ;TAKE ERROR-FREE EXIT

```

CVDMEA.P11 10-DEC-80 09:16

....RXCHAR -- RECEIVE A CHARACTER

3674	011610	062705	000006
3675	011614	000261	
3676	011616	012602	
3677	011620	012601	
3678	011622	000205	
3679			

20\$:	ADD	#6,R5	:FIX UP RETURN ADDRESS
	SEC		:SET C BIT FOR ERROR
21\$:	MOV	(SP)+,R2	:RESTORE R2
	MOV	(SP)+,R1	:RESTORE R1
	RTS	R5	:RETURN

CVDMEA.P11 10-DEC-80 09:16

....RCV1ST -- RECEIVE FIRST CHARACTER OF MESSAGE

3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695 011624
3696 011624 010146
3697 011626 010246
3698 011630 005001
3699 011632 012502
3700 011634 062702 000003
3701 011640 004537 005654
3702 011644 000000
3703 011646 103446
3704 011650 004537 006154
3705 011654 000000
3706 011656 103442
3707 011660 004537 006614
3708 011664 000000
3709 011666 103436
3710 011670 004537 012072
3711 011674 000001
3712 011676 005201
3713 011700 004537 003566
3714 011704 122000
3715 011706 000000
3716 011710 132737 000200 011706
3717 011716 001006
3718 011720 020102
3719 011722 002762
3720 011724 004537 006154
3721 011730 000001
3722 011732 103414
3723 011734 020165 177776
3724 011740 002004
3725 011742 004537 006154
3726 011746 000000
3727 011750 103405
3728 011752 004537 005654
3729 011756 000001
3730 011760 103401
3731 011762 000241
3732 011764 012602
3733 011766 012601
3734 011770 000205
3735

.SBTTLRCV1ST -- RECEIVE FIRST CHARACTER OF MESSAGE

* RCV1ST - THIS SUBROUTINE RECEIVES THE FIRST CHAR OF A MESSAGE AND MONITORS
* THE STATUS OF THE RECEIVER. FIRST, A CHECK IS MADE FOR RXACT = 0,
* RDA = 0, RSA = 0, RSOM = 0. THEN, THE LINE UNIT IS CLOCKED UNTIL
* RDA = 1. THE PROGRAM CHECKS FOR THIS TO OCCUR WITHIN 3 CYCLES AFTER
* THE NO. OF CYCLES PASSED IN THE SECOND WORD FOLLOWING THE CALL.
* IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
* IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
* DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.

CALLING SEQUENCE :
JSR R5,RCV1ST
.WORD <EXPECTED RECEIVER CYCLE COUNT>

RCV1ST:
MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2
CLR R1 ;INIT CYCLE COUNT
MOV (R5)+,R2 ;GET CYCLE COUNT LIMIT
ADD #3,R2
JSR R5,CKRACT ;CHK FOR RXACT = 0
0
BCS 6\$;BR TO EXIT IF ERROR
JSR R5,CKRDA ;CHK FOR RDA = 0
0
BCS 6\$;BR TO EXIT IF ERROR
JSR R5,CKSEOM ;CHK FOR RSOM = 0, REOM = 0
0
BCS 6\$;BR TO EXIT IF ERROR
1\$: JSR R5,STEPLU ;CLOCK LU FOR 1 CYCLE
1
INC R1 ;INCREMENT CYCLE COUNT
JSR R5,READJ ;READ USYRT STATUS REG
USTATR
2\$: .WORD 0
BITB #RDA,2\$;SEE IF RDA SET YET
BNE 3\$;BR IF YES
CMP R1,R2 ;SEE IF LIMIT EXCEEDED
BLT 1\$;BR IF NOT YET
JSR R5,CKRDA ;GO STACK 'RDA NOT SET' MSG
1
BCS 6\$;BR TO EXIT IF ERROR
3\$: CMP R1,-2(R5) ;SEE IF LESS THAN REQUIRED CYCLES
RGE 4\$;BR IF NOT
JSR R5,CKRDA ;GO STACK 'RDA NOT CLEARED' MSG
0
BCS 6\$;BR TO EXIT IF ERROR
4\$: JSR R5,CKRACT ;CHK FOR RXACT = 1
1
BCS 6\$;BR TO EXIT IF ERROR
5\$: CLC ;CLEAR C BIT FOR NO ERRORS
6\$: MOV (SP)+,R2 ;RESTORE R2
MOV (SP)+,R1 ;RESTORE R1
RTS R5 ;RETURN (WITH C BIT = 1 IF ERROR)

L 7

CVDMEA.P11

10-DEC-80 09:16

....RCV1ST -- RECEIVE FIRST CHARACTER OF MESSAGE

SEQ 89

3736

CVDMEA.P11 10-DEC-80 09:16

....ENTRAN -- SHUT DOWN TRANSMITTER/RECEIVER

```

3737
3738
3739
3740
3741
37 2
3743
3744
3745
3746
3747
3748
3749
3750
3751 011772
3752 011772 012737 000002 012042
3753 012000 112537 012050
3754 012004 105725
3755 012006 100002
3756 012010 005037 012042
3757 012014 004537 005514
3758 012020 000001
3759 012022 103422
3760 012024 004537 005654
3761 012030 000001
3762 012032 103416
3763 012034 004537 003712
3764 012040 120000
3765 012042 000002
3766 012044 004537 012072
3767 012050 000000
3768 012052 004537 005514
3769 012056 000000
3770 012060 103403
3771 012062 004537 005654
3772 012066 000000
3773 012070 000205
3774
3775

```

```

.SBTTL ....ENTRAN -- SHUT DOWN TRANSMITTER/RECEIVER
*****
* ENTRAN - THIS SUBROUTINE TERMINATES A MESSAGE BY CLEARING TXEN AND RXEN,
* CLOCKING THE LINE UNIT FOR THE NUMBER OF CYCLES PASSED IN THE WORD
* FOLLOWING THE CALL, AND CHECKING FOR THE USYRT TRANSMITTER AND
* RECEIVER TO BE SHUT DOWN.
* IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
* IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
* DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
*
* CALLING SEQUENCE :
* JSR R5,ENTRAN
* MSB SET=NO TTLOOP ! LOWER BYTE = <NO. OF CYCLES TO CLOCK>
*****
ENTRAN:
MOV #TTLOOP,1$ ;INIT DEFAULT VIAORB (TTLOOP=1)
MOVB (R5)+,2$ ;GET DESIRED # OF TICKS (LOWER BYTE)
TSTB (R5)+ ;SEE IF MSB SET (TTLOOP DISABLE BIT)
BPL 4$ ;IS IT?
CLR 1$ ;IF YES: CLEAR VIAORB VALUE
4$: JSR R5,CKTACT ;CHK FOR TXACT = 1
1 BCS 6$ ;BR IF ERROR
JSR R5,CKRACT ;CHK FOR RXACT = 1
1 BCS 6$
JSR R5,WRITEI ;CLEAR TXEN AND RXEN IN USYRT
VIAORB
1$: TTLOOP ;** HOLE FOR ACTUAL VIAORB WORD **
JSR R5,STEPLU ;CLOCK LU FOR DESIRED NO. OF CYCLES
2$: .WORD 0
JSR R5,CKTACT ;CHK FOR TXACT = 0
0
BCS 6$ ;BR IF ERROR
JSR R5,CKRACT ;CHK FOR RXACT = 0
0
6$: RTS R5

```

CVDMEA.P11 10-DEC-80 09:16

....STEPLU -- CLOCK THE USYRT N TIMES

3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789 012072
3790 012072 010146
3791 012074 012501
3792 012076 004537 003712
3793 012102 120005
3794 012104 000000
3795 012106 005301
3796 012110 001372
3797 012112 012601
3798 012114 000205
3799
3800
3801
3802
3803
3804

```

.SBTTL ....STEPLU -- CLOCK THE USYRT N TIMES
*****
* STEPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NUMBER OF CYCLES
* PASSED IN THE WORD FOLLOWING THE CALL. THE VIA ACR MUST BE PREVIOUSLY
* SET UP FOR T1 ONE-SHOT MODE, AND THE T1 LATCHES MUST BE PREVIOUSLY SET
* TO CONTROL THE WIDTH OF THE CLOCK PULSE. ALL THAT THIS SUBROUTINE
* DOES IS TO LOAD 000 INTO THE HI BYTE OF THE T1 COUNTER, FOR THE
* DESIRED NUMBER OF TIMES.
*
* CALLING SEQUENCE :
* JSR R5,STEPLU
* .WORD <NUMBER OF CYCLES TO CLOCK>
*****

```

```

STEPLU:
MOV R1,-(SP)          ;SAVE R1
MOV (R5)+,R1         ;INIT CYCLE COUNTER
JSR R5,WRITEI        ;... T1C-H, START COUNTER, CLOCK 1 CYCLE
VIAT1B
000
DEC R1                ;DECR CYCLE COUNTER
BNE 1$                ;BR IF ALL CYCLES NOT DONE YET
MOV (SP)+,R1         ;RESTORE R1
RTS R5                ;RETURN

```

CVDMEA.P11 10-DEC-80 09:16

GLOBAL ERROR REPORT SECTION

3805
3806
3807
3808
3809
3810
3811
3812

.SBTTL GLOBAL ERROR REPORT SECTION

:/ THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
:/ THAT ARE USED IN MORE THAN ONE TEST.

```

.NLIST BEX
ENDEMB: .ASCIZ /%N%N/
NEWLIN: .ASCIZ /%N/ ;USED TO TERMINATE ERROR MESSAGES

012116 047045 047045 000
012123 045 000116

012126 047045 040445 040506 FMT2: .ASCIZ /%N%AFAILING REG = %T%ASEL%01/
012163 045 022516 020101 FMT3: .ASCIZ /%N%A EXPECTED: %03%A ACTUAL: %03%A XOR: %03/
012247 045 022516 052101 FMT4: .ASCIZ /%N%ATHE CONTENTS OF ALL%T%N%T/
012305 045 022516 030523 FMT4A: .ASCIZ /%N%S1%03%S5%03%S5%03%S5%03/
012340 047045 052045 000 FMT4B: .ASCIZ /%N%T/
012345 045 022516 032523 FMT4C: .ASCIZ /%N%S5%03%S5%03%S5%03%S5%03/
012400 047045 040445 020040 FMT5: .ASCIZ /%N%A WHEN %03%A LOADED INTO BSEL1/
012443 045 022516 020101 FMT5A: .ASCIZ /%N%A ATTEMPTING 'M-LOOP' FUNCTION CODE %02%A (%T%A)/
012530 047045 040445 042115 FMT7: .ASCIZ /%N%AMDJAG #%03%A FAILED/

012560 047045 040445 020040 FMT10: .ASCIZ /%N%A EXPECTED:%08%A ACTUAL:%08%A XOR:%08/
012634 040445 020040 051514 FMT10A: .ASCIZ /%A LSI ADDR:%08/
012655 045 022516 034117 FMT11: .ASCIZ /%N%08%08%08%08/
012674 047045 047045 052045 FMT12: .ASCIZ /%N%N%T/
012703 045 022516 022524 FMT13: .ASCIZ /%N%T%03%S2%03%S2%03%S2%03%S2%03%S2%03/
012751 045 031123 047445 FMT14: .ASCIZ /%S2%03%S2%03/
012766 040445 020040 042504 FMT15: .ASCIZ /%A DETECTED IN %T%T%A --/
013020 040445 020040 042504 FMT15A: .ASCIZ /%A DETECTED @ TEST PATTERN ELEMENT # %D2/
013072 047045 052045 047445 FMT16: .ASCIZ /%N%T%03%S4%03%S%03/
013115 045 022516 022524 FMT16A: .ASCIZ /%N%T%03%S%03%S%03%S4%03%S%03%S%03/
013157 045 020101 020040 FMT17: .ASCIZ /%A VALUE SENT TO NPR CONTROL REGISTER: %03/
013236 047045 040445 020040 FMT17A: .ASCIZ /%N%A VALUE READ FROM CONTROL REGISTER: %03/
013317 045 022516 020101 FMT17B: .ASCIZ /%N%A LSI-11 MEMORY ADDRESS ACCESSED:%08/
013372 047045 040445 020040 FMT17C: .ASCIZ /%N%A INFORMATION ON THE FIRST OF %D5%A ERRORS:/

013452 047045 040445 042524 FMT19: .ASCIZ /%N%ATEST %D2%A NOT RUN%N/
013503 045 022524 033117 FMT21: .ASCIZ /%T%06%N/
013513 045 022516 043101 FMT22: .ASCIZ /%N%AFAILING REG: /
013535 045 042501 050130 FMT23: .ASCIZ /%AEXPECTED: %03%S5%A ACTUAL: %03%S5%A XOR: %03%N/
013614 047045 052045 047045 FMT24: .ASCIZ /%N%T%N%T%N/
013627 045 031517 051445 FMT25: .ASCIZ /%03%S5%03%S5%03%S5%03%N/
013657 045 032123 047445 FMT26: .ASCIZ /%S4%03%S5%03%S5%03%S5%03%N/
013712 052045 052045 047045 FMT27: .ASCIZ /%T%T%N/
013721 045 042501 052130 FMT28: .ASCIZ /%AEXTENDED REG AX%01%A-%T%N/
013755 045 022524 000116 FMT29: .ASCIZ /%T%N/
013762 047045 040445 047506 FMT30: .ASCIZ /%N%AFOR BAUD RATE SPECIFIED,/
014017 045 022516 044501 FMT31: .ASCIZ /%N%AIMPROPER CONNECTOR TYPE SPECIFIED/
014065 045 022516 043101 FMT32: .ASCIZ /%N%AFOR OPTION SPECIFIED,/
014117 045 022516 052101 FMT39: .ASCIZ /%N%ATEST %D2%A NOT RUN%N/
014150 047045 040445 040506 FMT40: .ASCIZ /%N%AFAILING RAM ADRS: %06%A (OCT)%N/

014214 051525 051131 020124 EM2: .ASCIZ /USYRT NOT INITIALIZED BY PROGRAM RESET/
014263 115 041511 047522 EM3: .ASCIZ /MICRO-DIAG. FAILURE/
014307 115 042122 020131 EM4: .ASCIZ /MRDY TIMEOUT/
014324 047522 020122 047516 EM14: .ASCIZ /ROR NOT SET/

```

CVDMEA.P11 10-DEC-80 09:16

GLOBAL ERROR REPORT SECTION

014340	047522	020122	047516	EM16:	.ASCIZ	/ROR NOT CLEARED/
014360	042522	042101	053457	EM25:	.ASCIZ	'READ/WRITE DATA ERROR'
014406	051522	046517	047040	EM28:	.ASCIZ	/RSOM NOT CLEARED/
014427	122	047523	020115	EM29:	.ASCIZ	/RSOM NOT SET/
014444	042522	046517	047040	EM30:	.ASCIZ	/REOM NOT CLEARED/
014465	122	047505	020115	EM31:	.ASCIZ	/REOM NOT SET/
014502	041522	023526	020104	EM34:	.ASCIZ	/RCV'D DATA MISCOMPARE/
014530	042522	051122	047040	EM35:	.ASCIZ	/RERR NOT CLEARED/
014551	122	051105	020122	EM36:	.ASCIZ	/RERP NOT SET/
014566	040522	043502	020101	EM39:	.ASCIZ	/RABGA NOT CLEARED/
014610	040522	043502	020101	EM40:	.ASCIZ	/RABGA NOT SET/
014626	054124	052440	042116	EM54:	.ASCIZ	/TX UNDERRUN ERROR/
014650	051525	051131	020124	EM68:	.ASCIZ	/USYRT STATUS INCORRECT/
014677	124	040530	052103	EM69:	.ASCIZ	/TXACT NOT SET/
014715	124	040530	052103	EM70:	.ASCIZ	/TXACT NOT CLEARED/
014737	122	040530	052103	EM71:	.ASCIZ	/RXACT NOT SET/
014755	122	040530	052103	EM72:	.ASCIZ	/RXACT NOT CLEARED/
014777	124	046502	020124	EM73:	.ASCIZ	/TBMT NOT SET/
015014	041124	052115	047040	EM74:	.ASCIZ	/TBMT NOT CLEARED/
015035	122	040504	047040	EM75:	.ASCIZ	/RDA NOT SET/
015051	122	040504	047040	EM76:	.ASCIZ	/RDA NOT CLEARED/
015071	122	040523	047040	EM77:	.ASCIZ	/RSA NOT SET/
015105	122	040523	047040	EM78:	.ASCIZ	/RSA NOT CLEARED/
015125	122	046501	042440	EM79:	.ASCIZ	/RAM ERROR LOADING MICROCODE/
015161	103	051101	044522	EM80:	.ASCIZ	/CARRIER NOT SET/
015201	103	051101	044522	EM81:	.ASCIZ	/CARRIER NOT CLEARED/
015225	111	053116	046101	EM82:	.ASCIZ	/INVALID ERROR CODE FROM 6502/
015262	047515	042504	020115	EM83:	.ASCIZ	/MODEM STATUS INCORRECT/
015311	103	051524	047040	EM84:	.ASCIZ	/CTS NOT CLRD/
015326	052103	020123	047516	EM85:	.ASCIZ	/CTS NOT SET/
015342	040503	051122	042511	EM86:	.ASCIZ	/CARRIER NOT CLRD/
015363	103	051101	044522	EM87:	.ASCIZ	/CARRIER NOT SET/
015403	115	042117	046505	EM88:	.ASCIZ	/MODEM RDY NOT CLRD/
015426	047515	042504	020115	EM89:	.ASCIZ	/MODEM RDY NOT SET/
015450	042522	042503	053111	EM90:	.ASCIZ	/RECEIVER OVERRUN NOT SET/
015501	122	041505	044505	EM91:	.ASCIZ	/RECEIVER OVERRUN NOT CLEARED/
015536	051524	020117	044502	EM100:	.ASCIZ	/TSO BIT NOT SET/
015556	051524	020117	044502	EM101:	.ASCIZ	/TSO BIT NOT CLEARED/

..SBTTLTEXT STRINGS FOR ERROR HANDLERS -- 'TXT___'

----- TEXT USED BY ERROR HANDLERS -----

015602	051502	046105	020060	TXT1:	.ASCIZ	/BSEL0 BSEL1 BSEL2 BSEL3/
015640	020040	020040	051502	TXT2:	.ASCIZ	/ BSEL4 BSEL5 BSEL6 BSEL7/
015702	051502	046105	030061	TXT2A:	.ASCIZ	/BSEL10 BSEL11 BSEL12 BSEL13/
015741	040	020040	041040	TXT2B:	.ASCIZ	/ BSEL14 BSEL15 BSEL16 BSEL17/
016004	041040	052131	020105	TXT3:	.ASCIZ	/ BYTE SELECT REG'S ARE:/
016034	020040	051440	046105	TXT4:	.ASCIZ	/ SEL0 SEL2 SEL4 SEL6/
016074	020040	051440	046105	TXT4A:	.ASCIZ	/ SEL10 SEL12 SEL14 SEL16/
016135	102	000		TXT5:	.ASCIZ	/B/
016137	040	042523	042514	TXT6:	.ASCIZ	/ SELECT REG'S ARE:/
016162	051040	043505	051511	TXT7:	.ASCIZ	/ REGISTERS ORB ORA DDRB DDRA T1CL T1CH T1LL T1LH /
016252	020040	020040	020040	TXT7A:	.ASCIZ	/ T2CL T2CH SR ACR PCR IFR IER ORA /

CVDMEA.P11 10-DEC-80 09:16

....TEXT STRINGS FOR ERROR HANDLERS -- 'TXT_--'

```

016342 042440 050130 041505 TXT8: .ASCIZ / EXPECTED: /
016362 040440 052103 040525 TXT9: .ASCIZ / ACTUAL: /
016402 054040 051117 020072 TXT10: .ASCIZ / XOR: /
016422 020040 020116 050040 TXT11: .ASCIZ / N P R R E G I S T E R S:/
016474 020040 020040 020040 TXT11A: .ASCIZ / CONTROL DATA/
016532 020040 020040 020040 TXT11B: .ASCIZ / OUT ADDR. IN ADDR./
016602 042504 044526 042503 TXT12: .ASCIZ /DEVICE CSR ADDRESS : /
016630 051525 051131 020124 TXT13: .ASCIZ /USYRT REGS :/
016645 122 051504 046122 TXT14: .ASCIZ /RDSRL RDSRH TDSRL TDSRH/
016703 040 020040 050040 TXT15: .ASCIZ / PCSARL PCSARH PCR USTAT/
016745 126 040511 051040 TXT16: .ASCIZ /VIA REGS :/
016760 051117 020102 020040 TXT17: .ASCIZ /ORB JRA DDRB DDRA/
017015 040 020040 052040 TXT18: .ASCIZ / T1CL T1CH T1LL T1LH/
017056 031124 046103 020040 TXT19: .ASCIZ /T2CL T2CH SR ACR/
017112 020040 020040 041520 TXT20: .ASCIZ / PCR IFR IER ORA/

017152 021 000 TXTNUL: .BYTE 21,0 ;CTL-Q -- THIS (WE HOPE) IS HARMLESS

017154 047516 000120 TXTML0: .ASCIZ /NOP/
017160 042522 042101 030440 TXTML1: .ASCIZ /READ 1 BYTE/
017174 051127 052111 020105 TXTML2: .ASCIZ /WRITE 1 BYTE/
017211 116 051120 047455 TXTML3: .ASCIZ /NPR-OUT 256 BYTES/
017233 116 051120 044455 TXTML4: .ASCIZ /NPR-IN 256 BYTES/
017254 042523 020124 044515 TXTML5: .ASCIZ /SET MICROPROCESSOR'S PC/
017304 047125 042504 044506 TXTML6: .ASCIZ /UNDEFINED/
017316 046101 047514 020127 TXTML7: .ASCIZ /ALLOW U-PROCESSOR INTERRUPTS/

017353 126 040511 051040 TXTVR: .ASCIZ /VIA REGISTER /
017371 117 041122 000 TXTVR0: .ASCIZ /ORB/
017375 117 040522 000 TXTVR1: .ASCIZ /ORA/
017401 104 051104 000102 TXTVR2: .ASCIZ /DDRBL/
017406 042104 040522 000 TXTVR3: .ASCIZ /DDRA/
017413 124 041461 000114 TXTVR4: .ASCIZ /T1CL/
017420 030524 044103 000 TXTVR5: .ASCIZ /T1CH/
017425 124 046061 000114 TXTVR6: .ASCIZ /T1LL/
017432 030524 044114 000 TXTVR7: .ASCIZ /T1LH/
017437 124 041462 000114 TXTVR8: .ASCIZ /T2CL/
017444 031124 044103 000 TXTVR9: .ASCIZ /T2CH/
017451 123 000122 TXTVRA: .ASCIZ /SR/
017454 041501 000122 TXTVRB: .ASCIZ /ACR/
017460 041520 000122 TXTVRC: .ASCIZ /PCR/
017464 043111 000122 TXTVRD: .ASCIZ /IFR/
017470 042511 000122 TXTVRE: .ASCIZ /IER/
017474 051117 000101 TXTVRF: .ASCIZ /ORA/

017500 050116 020122 000 TXTNP: .ASCIZ /NPR /
017505 103 047117 051124 TXTNP0: .ASCIZ /CONTROL/
017515 104 052101 020101 TXTNP1: .ASCIZ /DATA HI/
017525 104 052101 020101 TXTNP2: .ASCIZ /DATA LO/
017535 101 042104 027122 TXTNP3: .ASCIZ /ADDR. OUT EX/
017552 042101 051104 020056 TXTNP4: .ASCIZ /ADDR. OUT HI/
017567 101 042104 027122 TXTNP5: .ASCIZ /ADDR. OUT LO/
017604 042101 051104 020056 TXTNP6: .ASCIZ /ADDR. IN EX/
017620 042101 051104 020056 TXTNP7: .ASCIZ /ADDR. IN HI/
017634 042101 051104 020056 TXTNP8: .ASCIZ /ADDR. IN LO/

```

CVDMEA.P11 10-DEC-80 09:16

....TEXT STRINGS FOR ERROR HANDLERS -- 'TXT_--'

017650	051525	051131	020124	TXTUR:	.ASCIZ	/USYRT REG /
017663	122	051504	046122	TXTURO:	.ASCIZ	/RDSRL/
017671	122	051504	044122	TXTUR1:	.ASCIZ	/RDSRH/
017677	124	051504	046122	TXTUR2:	.ASCIZ	/TDSRL/
017705	124	051504	044122	TXTUR3:	.ASCIZ	/TDSRH/
017713	120	051503	051101	TXTUR4:	.ASCIZ	/PCSARL/
017722	041520	040523	044122	TXTUR5:	.ASCIZ	/PCSARH/
017731	120	051103	000	TXTUR6:	.ASCIZ	/PCR/
017735	125	052123	052101	TXTUR7:	.ASCIZ	/USTAT/

.LIST BEX
.EVEN

3813 017744

3814

3815

3816

3817

3818

3819

3820

.SBTTLTEXT ADDRESS TABLES FOR ERROR HANDLERS -- 'TXT_T'

----- TEXT ADDRESS TABLES USED BY ERROR HANDLERS -----

3821 017744 017154 017160 017174 TXTMLT: .WORD TXTML0,TXTML1,TXTML2,TXTML3,TXTML4,TXTML5,TXTML6,TXTML7

3822 017752 017211 017233 017254

3823 017760 017304 017316

3824

3825 017764 017353 .WORD TXTVR

3826 017766 017371 017375 017401 TXTVRT: .WORD TXTVR0,TXTVR1,TXTVR2,TXTVR3,TXTVR4,TXTVR5,TXTVR6,TXTVR7

3827 017774 017406 017413 017420

3828 020002 017425 017432 .WORD TXTVR8,TXTVR9,TXTVRA,TXTVRB,TXTVRC,TXTVRD,TXTVRE,TXTVRF

3829 020006 017437 017444 017451

3830 020014 017454 017460 017464

3831 020022 017470 017474

3832

3833 020026 017500 .WORD TXTNP

3834 020030 017505 017515 017525 TXTNPT: .WORD TXTNP0,TXTNP1,TXTNP2,TXTNP3,TXTNP4,TXTNP5,TXTNP6,TXTNP7,TXTNP8

3835 020036 017535 017552 017567

3836 020044 017604 017620 017634

3837 020052 017663 017671 017677 TXTURT: .WORD TXTUR0,TXTUR1,TXTUR2,TXTUR3,TXTUR4,TXTUR5,TXTUR6,TXTUR7

3838 020060 017705 017713 017722

3839 020066 017731 017735

3840

3841

3842

CVDMEA.P11 10-DEC-80 09:16

....TEXT ADDRESS TABLES FOR ERROR HANDLERS -- 'TXT_T'

```

3843
3844
3845
3846 020072
3847 020072
3848 020072 004737 021306
3849 020076
3850 020076 012746 012116
3851 020102 012746 000001
3852 020106 010600
3853 020110 104414
3854 020112 062706 000004
3855 020116
3856 020116
3857 020116 104423
3858
3859
3860
3861 020120
3862 020120
3863 020120 105037 002325
3864 020124 010146
3865 020126 013701 002324
3866 020132 022701 000017
3867 020136 002012
3868 020140
3869 020140 010146
3870 020142 012746 012400
3871 020146 012746 000002
3872 020152 010600
3873 020154 104415
3874 020156 062706 000006
3875 020162 000424
3876
3877 020164 001001
3878 020166 005001
3879 020170 022701 000007
3880 020174 002002
3881 020176 012701 000006
3882 020202 006301
3883 020204
3884 020204 016146 017744
3885 020210 013746 002324
3886 020214 012746 012443
3887 020220 012746 000003
3888 020224 010600
3889 020226 104415
3890 020230 062706 000010
3891
3892 020234 012601
3893 020236 004737 021634
3894 020242
3895 020242
3896 020242 104423
3897
3898
    
```

```

-----
:SBTTL ....ERROR HANDLER -- ERR3 -- DUMP THE BYTE SELECT REGISTERS
-----
    
```

BGNMSG ERR3

ERR3::

```

JSR PC,ERR4$
PRINTB #ENDEMB
    
```

```

MOV #ENDEMB,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
    
```

ENDMSG

L10002:

TRAP C\$MSG

```

-----
:SBTTL ....ERROR HANDLER -- ERR4 -- M-LOOP TIMEOUT ERROR HANDLING
-----
    
```

BGNMSG ERR4

ERR4::

```

CLRB GDATA+1
MOV R1,-(SP)
MOV GDATA,R1
CMP #17,R1
BGE 5$
PRINTX #FMT5,R1
    
```

```

:MAKE SURE BIT 8 DOESN'T PRINT!
:SAVE THE WORKING REGISTER
:SAVE THIS FOR LATER
:WAS THIS AN M-LOOP REQUEST?
:YES, THEN REPORT THE FUNCTION CODE
:NO, THEN IT MUST BE A BSEL1 SETTING
    
```

```

MOV R1,-(SP)
MOV #FMT5,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #6,SP
    
```

BR 20\$

5\$: BNE 6\$

```

:IF IT WAS A 17, THIS IS A 'NOP' AND
: THE TEXT POINTER MUST SO REFLECT.
:IS FUNCTION CODE > 7?
:NO, THEN WE CAN HANDLE IT
:YES, THEN IT'S UNDEFINED -- SAY SO
    
```

6\$: CMP #7,R1

```

BGE 7$
MOV #6,R1
    
```

7\$: ASL R1

```

:CONVERT TO A WORD OFFSET
:REPORT THE FAILING FUNCTION
    
```

PRINTX #FMT5A,GDATA,TXTMLT(R1)

```

MOV TXTMLT(R1),-(SP)
MOV GDATA,-(SP)
MOV #FMT5A,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #10,SP
    
```

20\$: MOV (SP)+,R1

```

:RESTORE THE WORKING REGISTER
:DUMP THE SELECT REGISTERS
    
```

```

JSR PC,ERR5$
ENDMSG
    
```

L10003:

TRAP C\$MSG

```

-----
:SBTTL ....ERROR HANDLER -- ERR7A -- USYRT REGISTER ERRORS
-----
    
```

CVDMEA.P11 10-DEC-80 09:16

....ERROR HANDLER -- ERR7A -- USYRT REGISTER ERRORS

```

3899
3900 020244
3901 020244
3902 020244 113701 002336
3903 020250 006301
3904 020252
3905 020252 016146 020052
3906 020256 012746 017650
3907 020262 012746 012766
3908 020266 012746 000003
3909 020272 010600
3910 020274 104414
3911 020276 062706 000010
3912 020302 004737 021262
3913 020306
3914 020306 013746 002330
3915 020312 013746 002326
3916 020316 013746 002324
3917 020322 012746 012163
3918 020326 012746 000004
3919 020332 010600
3920 020334 104414
3921 020336 062706 000012
3922 020342
3923 020342 012746 012116
3924 020346 012746 000001
3925 020352 010600
3926 020354 104414
3927 020356 062706 000004
3928 020362
3929 020362
3930 020362 104423
3931
3932
3933
3934
3935 020364
3936 020364
3937 020364
3938 020364 013746 002416
3939 020370 012746 016602
3940 020374 012746 013503
3941 020400 012746 000003
3942 020404 010600
3943 020406 104414
3944 020410 062706 000010
3945 020414
3946 020414 012746 013513
3947 020420 012746 000001
3948 020424 010600
3949 020426 104414
3950 020430 062706 000004
3951 020434 013701 002336
3952 020440 006301
3953 020442
3954 020442 016146 020052

```

```

-----
:-----
BGNMSG ERR7A
ERR7A::
MOV REGNUM,R1
ASL R1 ;AS PASSED, THIS WAS A BYTE OFFSET
PRINTB #FMT15,#TXTUR,TXTURT(R1)
MOV TXTURT(R1),-(SP)
MOV #TXTUR,-(SP)
MOV #FMT15,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
JSR PC,XORGB
PRINTB #FMT3,GDATA,BDATA,XDATA
MOV XDATA,-(SP)
MOV BDATA,-(SP)
MOV GDATA,-(SP)
MOV #FMT3,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #12,SP
PRINTB #ENDEMB
MOV #ENDEMB,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
ENDMSG
L10004:
TRAP C$MSG
:-----
:SBTTL ....ERROR HANDLER -- ERR10 -- USYRT REG ERROR (XOR, REG PRINTOUT)
:-----
BGNMSG ERR10
ERR10::
PRINTB #FMT21,#TXT12,MPCSR
MOV MPCSR,-(SP)
MOV #TXT12,-(SP)
MOV #FMT21,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
PRINTB #FMT22
MOV #FMT22,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
MOV REGNUM,R1
ASL R1 ;GET PTR TO USYRT REG ASCII
PRINTB #FMT27,#TXTUR,TXTURT(R1)
MOV TXTURT(R1),-(SP)

```


CVDMEA.P11 10-DEC-80 09:16

....ERROR HANDLER -- ERR10 -- USYRT REG ERROR (XOR, REG PRINTOUT)

```

3955 020446 012746 017650          MOV      #TXTUR,-(SP)
3956 020452 012746 013712          MOV      #FMT27,-(SP)
3957 020456 012746 000003          MOV      #3,-(SP)
3958 020462 010600                    MOV      SP,R0
3959 020464 104414                    TRAP    C$PNTB
3960 020466 062706 000010          ADD     #10,SP
3961 020472 004737 021262          JSR     PC,XORGB      ;COMPUTE XOR OF GOOD AND BAD DATA
3962 020476                    PRINTB  #FMT23,GDATA,BDATA,XDATA
3963 020476 013746 002330          MOV      XDATA,-(SP)
3964 020502 013746 002326          MOV      BDATA,-(SP)
3965 020506 013746 002324          MOV      GDATA,-(SP)
3966 020512 012746 013535          MOV      #FMT23,-(SP)
3967 020516 012746 000004          MOV      #4,-(SP)
3968 020522 010600                    MOV      SP,R0
3969 020524 104414                    TRAP    C$PNTB
3970 020526 062706 000012          ADD     #12,SP
3971 020532 004737 022364          JSR     PC,ERR12$    ;GET & PRINT USYRT REGISTERS
3972 020536                    ENDMSG
3973 020536
3974 020536 104423                    L10005: TRAP    C$MSG
3975
3976
3977 :-----:
3978 :SBTTL ....ERROR HANDLER -- ERR11 -- VIA REG ERROR (XOR, REG PRINTOUT)
3979 :-----:
3980 020540                    BGNMSG  ERR11
3981 020540
3982 020540                    PRINTB  #FMT21,#TXT12,MPCSR          ERR11::
3983 020540 013746 002416          MOV      MPCSR,-(SP)
3984 020544 012746 016602          MOV      #TXT12,-(SP)
3985 020550 012746 013503          MOV      #FMT21,-(SP)
3986 020554 012746 000003          MOV      #3,-(SP)
3987 020560 010600                    MOV      SP,R0
3988 020562 104414                    TRAP    C$PNTB
3989 020564 062706 000010          ADD     #10,SP
3990 020570                    PRINTB  #FMT22
3991 020570 012746 013513          MOV      #FMT22,-(SP)
3992 020574 012746 000001          MOV      #1,-(SP)
3993 020600 010600                    MOV      SP,R0
3994 020602 104414                    TRAP    C$PNTB
3995 020604 062706 000004          ADD     #4,SP
3996 020610 013701 002336          MOV     REGNUM,R1
3997 020614 006301                    ASL     R1
3998 020616                    PRINTB  #FMT27,#TXTVR,TXTVRT(R1) ;GET PTR TO VIA REG ASCII
3999 020616 016146 017766          MOV      TXTVRT(R1),-(SP)
4000 020622 012746 017353          MOV      #TXTVR,-(SP)
4001 020626 012746 013712          MOV      #FMT27,-(SP)
4002 020632 012746 000003          MOV      #3,-(SP)
4003 020636 010600                    MOV      SP,R0
4004 020640 104414                    TRAP    C$PNTB
4005 020642 062706 000010          ADD     #10,SP
4006 020646 004737 021262          JSR     PC,XORGB      ;COMPUTE XOR OF GOOD AND BAD DATA
4007 020652                    PRINTB  #FMT23,GDATA,BDATA,XDATA
4008 020652 013746 002330          MOV      XDATA,-(SP)
4009 020656 013746 002326          MOV      BDATA,-(SP)
4010 020662 013746 002324          MOV      GDATA,-(SP)

```

CVDMEA.P11 10-DEC-80 09:16

....ERROR HANDLER -- ERR11 -- VIA REG ERROR (XOR, REG PRINTOUT)

4011	020666	012746	013535			MOV	#FMT23,-(SP)
4012	020672	012746	000004			MOV	#4,-(SP)
4013	020676	010600				MOV	SP,R0
4014	020700	104414				TRAP	C\$PNTB
4015	020702	062706	000012			ADD	#12,SP
4016	020706	004737	022032	JSR	PC,ERR11\$		
4017	020712			ENDMSG			
4018	020712						
4019	020712	104423				L10006:	TRAP C\$MSG

 :SBTTLERROR HANDLER -- ERR12 -- USYRT REG ERROR (USYRT PRINTOUT)
 :-----

4025	020714			BGNMSG	ERR12		
4026	020714					ERR12::	
4027	020714			PRINTB	#FMT21,#TXT12,MPCSR		
4028	020714	013746	002416			MOV	MPCSR,-(SP)
4029	020720	012746	016602			MOV	#TXT12,-(SP)
4030	020724	012746	013503			MOV	#FMT21,-(SP)
4031	020730	012746	000003			MOV	#3,-(SP)
4032	020734	010600				MOV	SP,R0
4033	020736	104414				TRAP	C\$PNTB
4034	020740	062706	000010			ADD	#10,SP
4035	020744			PRINTB	#FMT22		
4036	020744	012746	013513			MOV	#FMT22,-(SP)
4037	020750	012746	000001			MOV	#1,-(SP)
4038	020754	010600				MOV	SP,R0
4039	020756	104414				TRAP	C\$PNTB
4040	020760	062706	000004			ADD	#4,SP
4041	020764	013701	002336	MOV	REGNUM,R1		
4042	020770	006301		ASL	R1		
4043	020772			PRINTB	#FMT27,#TXTUR,TXTURT(R1)		
4044	020772	016146	020052			MOV	TXTURT(R1),-(SP)
4045	020776	012746	017650			MOV	#TXTUR,-(SP)
4046	021002	012746	013712			MOV	#FMT27,-(SP)
4047	021006	012746	000003			MOV	#3,-(SP)
4048	021012	010600				MOV	SP,R0
4049	021014	104414				TRAP	C\$PNTB
4050	021016	062706	000010			ADD	#10,SP
4051	021022	004737	022364	JSR	PC,ERR12\$		
4052	021026			ENDMSG			
4053	021026						
4054	021026	104423				L10007:	TRAP C\$MSG

 :SBTTLERROR HANDLER -- ERR13 -- RAM ADDRESS ERRORS
 :-----

4060	021030			BGNMSG	ERR13		
4061	021030					ERR13::	
4062	021030			PRINTB	#FMT21,#TXT12,MPCSR		
4063	021030	013746	002416			MOV	MPCSR,-(SP)
4064	021034	012746	016602			MOV	#TXT12,-(SP)
4065	021040	012746	013503			MOV	#FMT21,-(SP)
4066	021044	012746	000003			MOV	#3,-(SP)

CVDMEA.P11 10-DEC-80 09:16

....ERROR HANDLER -- ERR13 -- RAM ADDRESS ERRORS

```

4067 021050 010600
4068 021052 104414
4069 021054 062706 000010
4070 021060
4071 021060 013746 002336
4072 021064 012746 014150
4073 021070 012746 000002
4074 021074 010600
4075 021076 104414
4076 021100 062706 000006
4077 021104 004737 021262
4078 021110
4079 021110 013746 002330
4080 021114 013746 002326
4081 021120 013746 002324
4082 021124 012746 013535
4083 021130 012746 000004
4084 021134 010600
4085 021136 104414
4086 021140 062706 000012
4087 021144
4088 021144
4089 021144 104423
4090
4091
4092
4093
4094
4095 021146
4096 021146
4097 021146
4098 021146 013746 002416
4099 021152 012746 016602
4100 021156 012746 013503
4101 021162 012746 000003
4102 021166 010600
4103 021170 104414
4104 021172 062706 000010
4105 021176
4106 021176 012746 013513
4107 021202 012746 000001
4108 021206 010600
4109 021210 104414
4110 021212 062706 000004
4111 021216 013701 002336
4112 021222 006301
4113 021224
4114 021224 016146 017766
4115 021230 012746 017353
4116 021234 012746 013712
4117 021240 012746 000003
4118 021244 010600
4119 021246 104414
4120 021250 062706 000010
4121 021254 004737 022032
4122 021260

MOV SP,RO
TRAP C$PNTB
ADD #10,SP

PRINTB #FMT40,REGNUM

MOV REGNUM,-(SP)
MOV #FMT40,-(SP)
MOV #2,-(SP)
MOV SP,RO
TRAP C$PNTB
ADD #6,SP

JSR PC,XORGB ;COMPUTE XOR OF GOOD AND BAD DATA
PRINTB #FMT23,GDATA,BDATA,XDATA

MOV XDATA,-(SP)
MOV BDATA,-(SP)
MOV GDATA,-(SP)
MOV #FMT23,-(SP)
MOV #4,-(SP)
MOV SP,RO
TRAP C$PNTB
ADD #12,SP

L10010: TRAP C$MSG

-----
:SBTTL ....ERROR HANDLER -- ERR14 -- VIA REG ERRORS (VIA PRINTOUT)
-----
BGNMSG ERR14
PRINTB #FMT21,#TXT12,MPCSR

ERR14::
MOV MPCSR,-(SP)
MOV #TXT12,-(SP)
MOV #FMT21,-(SP)
MOV #3,-(SP)
MOV SP,RO
TRAP C$PNTB
ADD #10,SP

PRINTB #FMT22

MOV #FMT22,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C$PNTB
ADD #4,SP

MOV REGNUM,R1
ASL R1 ;GET PTR TO VIA REG ASCII
PRINTB #FMT27,#TXTVR,TXTVRT(R1)

MOV TXTVRT(R1),-(SP)
MOV #TXTVR,-(SP)
MOV #FMT27,-(SP)
MOV #3,-(SP)
MOV SP,RO
TRAP C$PNTB
ADD #10,SP

JSR PC,ERR11$ ;GET & PRINT VIA REGISTERS
ENDMSG

```

CVDMEA.P11 10-DEC-80 09:16

....ERROR HANDLER -- ERR14 -- VIA REG ERRORS (VIA PRINTOUT)

L10011: TRAP C\$MSG

4123 021260
4124 021260 104423
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140 021262 010146
4141 021264 013701 002324
4142 021270 013737 002326 002330
4143 021276 074137 002330
4144 021302 012601
4145 021304 000207
4146
4147
4148
4149
4150
4151
4152 021306
4153 021306 012746 015602
4154 021312 012746 016004
4155 021316 012746 012247
4156 021322 012746 000003
4157 021326 010600
4158 021330 104415
4159 021332 062706 000010
4160 021336
4161 021336 013746 002210
4162 021342 013746 002206
4163 021346 013746 002204
4164 021352 013746 002202
4165 021356 012746 012305
4166 021362 012746 000005
4167 021366 010600
4168 021370 104415
4169 021372 062706 000014
4170 021376
4171 021376 012746 015640
4172 021402 012746 012340
4173 021406 012746 000002
4174 021412 010600
4175 021414 104415
4176 021416 062706 000006
4177 021422
4178 021422 013746 002220

.SBTTLERROR HANDLER SUBROUTINES

SUBROUTINES USED ONLY BY ERROR HANDLERS

.SBTTLERROR HANDLER SUBROUTINE -- XORGB

: PERFORM EXCLUSIVE OR BETWEEN 'GDATA' & 'BDATA' PUTTING THE RESULT IN 'XDATA'

```
XORGB:  MOV    R1,-(SP)      ;PRESERVE WORKING REGISTER
        MOV    GDATA,R1   ;GET 'GOOD' DATA
        MOV    BDATA,XDATA ;AND 'BAD' DATA
        XOR    R1,XDATA   ;PERFORM EXCLUSIVE OR
        MOV    (SP)+,R1   ;RESTORE R1
        RTS    PC        ;RETURN
```

.SBTTLERROR HANDLER SUBROUTINE -- ERR4\$

: IDENTIFY & DUMP THE BYTE SELECT REGISTERS

ERR4\$: PRINTX #FMT4,#TXT3,#TXT1

```
MOV    #TXT1,-(SP)
MOV    #TXT3,-(SP)
MOV    #FMT4,-(SP)
MOV    #3,-(SP)
MOV    SP,R0
TRAP  C$PNTX
ADD    #10,SP
```

PRINTX #FMT4A,BSR0,BSR1,BSR2,BSR3

```
MOV    BSR3,-(SP)
MOV    BSR2,-(SP)
MOV    BSR1,-(SP)
MOV    BSR0,-(SP)
MOV    #FMT4A,-(SP)
MOV    #5,-(SP)
MOV    SP,R0
TRAP  C$PNTX
ADD    #14,SP
```

PRINTX #FMT4B,#TXT2

```
MOV    #TXT2,-(SP)
MOV    #FMT4B,-(SP)
MOV    #2,-(SP)
MOV    SP,R0
TRAP  C$PNTX
ADD    #6,SP
```

PRINTX #FMT4C,BSR4,BSR5,BSR6,BSR7

MOV BSR7,-(SP)

CVDMEA.P11 10-DEC-80 09:16

.....ERROR HANDLER SUBROUTINE -- ERR4\$

4179 021426 013746 002216
4180 021432 013746 002214
4181 021436 013746 002212
4182 021442 012746 012345
4183 021446 012746 000005
4184 021452 010600
4185 021454 104415
4186 021456 062706 000014
4187 021462
4188 021462 012746 015702
4189 021466 012746 012340
4190 021472 012746 000002
4191 021476 010600
4192 021500 104415
4193 021502 062706 000006
4194 021506
4195 021506 013746 002230
4196 021512 013746 002226
4197 021516 013746 002224
4198 021522 013746 002222
4199 021526 012746 012305
4200 021532 012746 000005
4201 021536 010600
4202 021540 104415
4203 021542 062706 000014
4204 021546
4205 021546 012746 015741
4206 021552 012746 012340
4207 021556 012746 000002
4208 021562 010600
4209 021564 104415
4210 021566 062706 000006
4211 021572
4212 021572 013746 002240
4213 021576 013746 002236
4214 021602 013746 002234
4215 021606 013746 002232
4216 021612 012746 012345
4217 021616 012746 000005
4218 021622 010600
4219 021624 104415
4220 021626 062706 000014
4221 021632 000207
4222
4223
4224
4225
4226
4227 021634
4228 021634
4229 021634 012746 016034
4230 021640 012746 016137
4231 021644 012746 012247
4232 021650 012746 000003
4233 021654 010600
4234 021656 104415

PRINTX #FMT4B,#TXT2A

PRINTX #FMT4A,BSR10,BSR11,BSR12,BSR13

PRINTX #FMT4B,#TXT2B

PRINTX #FMT4C,BSR14,BSR15,BSR16,BSR17

RTS PC

MOV BSR6,-(SP)
MOV BSR5,-(SP)
MOV BSR4,-(SP)
MOV #FMT4C,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP
MOV #TXT2A,-(SP)
MOV #FMT4B,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #6,SP
MOV BSR13,-(SP)
MOV BSR12,-(SP)
MOV BSR11,-(SP)
MOV BSR10,-(SP)
MOV #FMT4A,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP
MOV #TXT2B,-(SP)
MOV #FMT4B,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #6,SP
MOV BSR17,-(SP)
MOV BSR16,-(SP)
MOV BSR15,-(SP)
MOV BSR14,-(SP)
MOV #FMT4C,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP

:SBTTLERROR HANDLER SUBROUTINE -- ERR5\$
:-----

:COMMON ERROR SUBROUTINE TO PRINT SELECT REGISTERS

:ERR5\$:

PRINTX #FMT4,#TXT6,#TXT4

MOV #TXT4,-(SP)
MOV #TXT6,-(SP)
MOV #FMT4,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTX

CVDMEA.P11 10-DEC-80 09:16

.....ERROR HANDLER SUBROUTINE -- ERR5\$

4235	021660	062706	000010		ADD	#10,SP
4236	021664			PRINTX	#FMT11,WSR0,WSR2,WSR4,WSR6 ;DUMP THE SELECT REGISTERS	
4237	021664	013746	002210		MOV	WSR6,-(SP)
4238	021670	013746	002206		MOV	WSR4,-(SP)
4239	021674	013746	002204		MOV	WSR2,-(SP)
4240	021700	013746	002202		MOV	WSR0,-(SP)
4241	021704	012746	012655		MOV	#FMT11,-(SP)
4242	021710	012746	000005		MOV	#5,-(SP)
4243	021714	010600			MOV	SP,R0
4244	021716	104415			TRAP	C\$PNTX
4245	021720	062706	000014		ADD	#14,SP
4246	021724			PRINTX	#FMT4B,#TXT4A	
4247	021724	012746	016074		MOV	#TXT4A,-(SP)
4248	021730	012746	012340		MOV	#FMT4B,-(SP)
4249	021734	012746	000002		MOV	#2,-(SP)
4250	021740	010600			MOV	SP,R0
4251	021742	104415			TRAP	C\$PNTX
4252	021744	062706	000006		ADD	#6,SP
4253	021750			PRINTX	#FMT11,WSR10,WSR12,WSR14,WSR16 ;DUMP THE SELECT REGISTERS	
4254	021750	013746	002220		MOV	WSR16,-(SP)
4255	021754	013746	002216		MOV	WSR14,-(SP)
4256	021760	013746	002214		MOV	WSR12,-(SP)
4257	021764	013746	002212		MOV	WSR10,-(SP)
4258	021770	012746	012655		MOV	#FMT11,-(SP)
4259	021774	012746	000005		MOV	#5,-(SP)
4260	022000	010600			MOV	SP,R0
4261	022002	104415			TRAP	C\$PNTX
4262	022004	062706	000014		ADD	#14,SP
4263	022010			PRINTB	#ENDEMB	
4264	022010	012746	012116		MOV	#ENDEMB,-(SP)
4265	022014	012746	000001		MOV	#1,-(SP)
4266	022020	010600			MOV	SP,R0
4267	022022	104414			TRAP	C\$PNTB
4268	022024	062706	000004		ADD	#4,SP
4269	022030	000207		RTS	PC	

 :SBTTLERROR HANDLER SUBROUTINE -- ERR11\$

: COMMON ERROR SUBROUTINE TO GET/PRINT VIA REGISTERS

4276	022032	004737	004460	ERR11\$:	JSR	PC,GETVRS ;GET VIA REGS FOR PRINTOUT
4277	022036				PRINTX	#FMT24,#TXT16,#TXT17
4278	022036	012746	016760		MOV	#TXT17,-(SP)
4279	022042	012746	016745		MOV	#TXT16,-(SP)
4280	022046	012746	013614		MOV	#FMT24,-(SP)
4281	022052	012746	000003		MOV	#3,-(SP)
4282	022056	010600			MOV	SP,R0
4283	022060	104415			TRAP	C\$PNTX
4284	022062	062706	000010		ADD	#10,SP
4285	022066			PRINTX	#FMT25,VREGS+0,VREGS+2,VREGS+4,VREGS+6	
4286	022066	013746	002270		MOV	VREGS+6,-(SP)
4287	022072	013746	002266		MOV	VREGS+4,-(SP)
4288	022076	013746	002264		MOV	VREGS+2,-(SP)
4289	022102	013746	002262		MOV	VREGS+0,-(SP)
4290	022106	012746	013627		MOV	#FMT25,-(SP)

CVDMEA.P11 10-DEC-80 09:16

.....ERROR HANDLER SUBROUTINE -- ERR11\$

4291	022112	012746	000005		MOV	#5,-(SP)
4292	022116	010600			MOV	SP,R0
4293	022120	104415			TRAP	C\$PNTX
4294	022122	062706	000014		ADD	#14,SP
4295	022126			PRINTX	#FMT29	#TXT18
4296	022126	012746	017015		MOV	#TXT18,-(SP)
4297	022132	012746	013755		MOV	#FMT29,-(SP)
4298	022136	012746	000002		MOV	#2,-(SP)
4299	022142	010600			MOV	SP,R0
4300	022144	104415			TRAP	C\$PNTX
4301	022146	062706	000006		ADD	#6,SP
4302	022152			PRINTX	#FMT26,VREGS+8.,	VREGS+10.,VREGS+12.,VREGS+14.
4303	022152	013746	002300		MOV	VREGS+14.,-(SP)
4304	022156	013746	002276		MOV	VREGS+12.,-(SP)
4305	022162	013746	002274		MOV	VREGS+10.,-(SP)
4306	022166	013746	002272		MOV	VREGS+8.,-(SP)
4307	022172	012746	013657		MOV	#FMT26,-(SP)
4308	022176	012746	000005		MOV	#5,-(SP)
4309	022202	010600			MOV	SP,R0
4310	022204	104415			TRAP	C\$PNTX
4311	022206	062706	000014		ADD	#14,SP
4312	022212			PRINTX	#FMT29,#TXT19	
4313	022212	012746	017056		MOV	#TXT19,-(SP)
4314	022216	012746	013755		MOV	#FMT29,-(SP)
4315	022222	012746	000002		MOV	#2,-(SP)
4316	022226	010600			MOV	SP,R0
4317	022230	104415			TRAP	C\$PNTX
4318	022232	062706	000006		ADD	#6,SP
4319	022236			PRINTX	#FMT25,VREGS+16.,	VREGS+18.,VREGS+20.,VREGS+22.
4320	022236	013746	002310		MOV	VREGS+22.,-(SP)
4321	022242	013746	002306		MOV	VREGS+20.,-(SP)
4322	022246	013746	002304		MOV	VREGS+18.,-(SP)
4323	022252	013746	002302		MOV	VREGS+16.,-(SP)
4324	022256	012746	013627		MOV	#FMT25,-(SP)
4325	022262	012746	000005		MOV	#5,-(SP)
4326	022266	010600			MOV	SP,R0
4327	022270	104415			TRAP	C\$PNTX
4328	022272	062706	000014		ADD	#14,SP
4329	022276			PRINTX	#FMT29,#TXT20	
4330	022276	012746	017112		MOV	#TXT20,-(SP)
4331	022302	012746	013755		MOV	#FMT29,-(SP)
4332	022306	012746	000002		MOV	#2,-(SP)
4333	022312	010600			MOV	SP,R0
4334	022314	104415			TRAP	C\$PNTX
4335	022316	062706	000006		ADD	#6,SP
4336	022322			PRINTX	#FMT27,VREGS+24.,	VREGS+26.,VREGS+28.,VREGS+30.
4337	022322	013746	002320		MOV	VREGS+30.,-(SP)
4338	022326	013746	002316		MOV	VREGS+28.,-(SP)
4339	022332	013746	002314		MOV	VREGS+26.,-(SP)
4340	022336	013746	002312		MOV	VREGS+24.,-(SP)
4341	022342	012746	013657		MOV	#FMT26,-(SP)
4342	022346	012746	000005		MOV	#5,-(SP)
4343	022352	010600			MOV	SP,R0
4344	022354	104415			TRAP	C\$PNTX
4345	022356	062706	000014		ADD	#14,SP
4346	022362	000207		RTS	PC	

CVDMEA.P11 10-DEC-80 09:16

.....ERROR HANDLER SUBROUTINE -- ERR11\$

4347
 4348
 4349
 4350
 4351
 4352
 4353 022364 004737 004360
 4354 022370
 4355 022370 012746 016645
 4356 022374 012746 016630
 4357 022400 012746 013614
 4358 022404 012746 000003
 4359 022410 010600
 4360 022412 104415
 4361 022414 062706 000010
 4362 022420
 4363 022420 013746 002250
 4364 022424 013746 002246
 4365 022430 013746 002244
 4366 022434 013746 002242
 4367 022440 012746 013627
 4368 022444 012746 000005
 4369 022450 010600
 4370 022452 104415
 4371 022454 062706 000014
 4372 022460
 4373 022460 012746 016703
 4374 022464 012746 013755
 4375 022470 012746 000002
 4376 022474 010600
 4377 022476 104415
 4378 022500 062706 000006
 4379 022504
 4380 022504 013746 002260
 4381 022510 013746 002256
 4382 022514 013746 002254
 4383 022520 013746 002252
 4384 022524 012746 013657
 4385 022530 012746 000005
 4386 022534 010600
 4387 022536 104415
 4388 022540 062706 000014
 4389 022544 000207
 4390
 4391

```

-----
:SBTTL .....ERROR HANDLER SUBROUTINE -- ERR12$
:
:      COMMON ERROR ROUTINE TO GET AND PRINTOUT USYRT REGISTERS
:
    
```

```

ERR12$: JSR    PC,GETURS          ;GET USYRT REGS FOR PRINTOUT
         PRINTX #FMT24,#TXT13,#TXT14
    
```

```

         MOV    #TXT14,-(SP)
         MOV    #TXT13,-(SP)
         MOV    #FMT24,-(SP)
         MOV    #3,-(SP)
         MOV    SP,R0
         TRAP   C$PNTX
         ADD    #10,SP
    
```

```

         PRINTX #FMT25,UREGS+0,UREGS+2,UREGS+4,UREGS+6
    
```

```

         MOV    UREGS+6,-(SP)
         MOV    UREGS+4,-(SP)
         MOV    UREGS+2,-(SP)
         MOV    UREGS+0,-(SP)
         MOV    #FMT25,-(SP)
         MOV    #5,-(SP)
         MOV    SP,R0
         TRAP   C$PNTX
         ADD    #14,SP
    
```

```

         PRINTX #FMT29,#TXT15
    
```

```

         MOV    #TXT15,-(SP)
         MOV    #FMT29,-(SP)
         MOV    #2,-(SP)
         MOV    SP,R0
         TRAP   C$PNTX
         ADD    #6,SP
    
```

```

         PRINTX #FMT26,UREGS+10,UREGS+12,UREGS+14,UREGS+16
    
```

```

         MOV    UREGS+16,-(SP)
         MOV    UREGS+14,-(SP)
         MOV    UREGS+12,-(SP)
         MOV    UREGS+10,-(SP)
         MOV    #FMT26,-(SP)
         MOV    #5,-(SP)
         MOV    SP,R0
         TRAP   C$PNTX
         ADD    #14,SP
    
```

```

         RTS    PC
    
```

.EVEN

CVDMEA.P11 10-DEC-80 09:16

LOAD DEVICE PROTECTION TABLE

.SBTTL LOAD DEVICE PROTECTION TABLE

```

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

```

```

4392
4393
4394
4395
4396
4397
4398
4399 022546
4400 022546
4401 022546 177777
4402 022550 177777
4403 022552 177777
4404 022554

```

BGNPROT

```

.WORD -1 ;DON'T CHK CSR ADRS
.WORD -1 ;DON'T CHK MASSBUS UNIT NO.
.WORD -1 ;DON'T CHK DRIVE NO.
ENDPROT

```

L\$PROT::

CVDMEA.P11 10-DEC-80 09:16

INITIALIZE SECTION

.SBTTL INITIALIZE SECTION

```

:////////////////////
:/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
:/ AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.
:////////////////////
    
```

```

4405
4406
4407
4408
4409
4410
4411
4412 022554          BGNINIT
4413 022554
4414
4415 022554 010637 002340          MOV     SP,PSTACK      ;SAVE BASE-LEVEL STACK POINTER
4416 022560 005037 002344          CLR     SUBRPC        ;CLEAR SUBR CALL PC
4417 022564 005037 002400          CLR     CHPTYP       ;CLEAR USYRT CHIP TYPE INDICATOR
4418 022570 005037 002376          CLR     ERROR1       ;CLEAR ERROR FLAG
4419 022574 005037 002402          CLR     SAVLEN       ;CLEAR CHAR LENGTH FROM SETUP
4420 022600 005737 002370          TST     FRSTIM        ;SEE IF FIRST TIME THROUGH AFTER LOAD
4421 022604 001007
4422 022606 013737 000004 002372    BNE     6$            ;BR IF NOT
4423 022614 013737 000006 002374    MOV     @#4,SAVE4     ;SAVE ERROR TRAP VECTOR
4424 022622 000406          MOV     @#6,SAVE6
4425
4426 022624 013737 002372 000004 6$:  MOV     SAVE4,@#4     ;RESTORE ERROR TRAP VECTOR
4427 022632 013737 002374 000006    MOV     SAVE6,@#6
4428
4429 022640 012737 000001 002370 9$:  MOV     #1,FRSTIM     ;MARK FLAG FOR NEXT TIME THROUGH
4430
4431          ;SEE IF PROGRAM JUST STARTED, BR IF YES
4432 022646          READEF #EF.START
4433 022646 012700 000040          MOV     TRAP #EF.START,RO
4434 022652 104447          TRAP   CSREFG
4435 022654          BCOMPLETE          STARST
4436 022654 103415          BCS     STARST
4437
4438          ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
4439 022656          READEF #EF.RESTART
4440 022656 012700 000037          MOV     TRAP #EF.RESTART,RO
4441 022662 104447          TRAP   CSREFG
4442 022664          BCOMPLETE          STARST
4443 022664 103411          BCS     STARST
4444
4445          ;SEE IF THIS IS A NEW PASS, BR IF YES
4446 022666          READEF #EF.NEW
4447 022666 012700 000035          MOV     TRAP #EF.NEW,RO
4448 022672 104447          TRAP   CSREFG
4449 022674          BCOMPLETE          NEWST
4450 022674 103411          BCS     NEWST
4451
4452          ;SEE IF PROGRAM WAS JUST CONTINUED
4453 022676          READEF #EF.CONTINUE
4454 022676 012700 000036          MOV     TRAP #EF.CONTINUE,RO
4455 022702 104447          TRAP   CSREFG
4456 022704          BCOMPLETE          ENDIT
4457 022704 103470          BCS     ENDIT
4458 022706 000414          BR      GETPRM
4459
4460 022710          STARST:
    
```

CVDMEA.P11 10-DEC-80 09:16

INITIALIZE SECTION

```

4461 022710 005037 002412          CLR      STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
4462
4463          ;CLEAR DEVICE MAP
4464 022714 005037 002404          CLR      DEVMAP
4465 022720
4466 022720 012737 177777 002334  NEWST:  MOV      #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
4467 022726 005237 002412          INC      STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
4468 022732 012737 000001 002406  MOV      #BIT0,DEVPTR     ;INIT DEVICE MAP BIT POINTER
4469
4470          ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
4471          ; CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
4472 022740
4473 022740 005237 002334  GETPRM:  INC      LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
4474 022744          GPHARD LOGDEV,R1      ;GET P-TABLE POINTER INTO R1
4475 022744 013700 002334          MOV      LOGDEV,R0
4476 022750 104442          TRAP
4477 022752 010001          MOV      R0,R1
4478 022754          BCOMPLETE      10$      ;BR IF DEVICE AVAILABLE
4479 022754 103403          BCS      10$
4480 022756 006337 002406          ASL      DEVPTR          ;SHIFT DEVICE POINTER
4481 022762 000766          BR      GETPRM          ;SKIP THIS DEVICE
4482 022764 053737 002406 002404 10$:  BIS      DEVPTR,DEVMAP     ;SET BIT FOR THIS DEVICE
4483 022772 006337 002406          ASL      DEVPTR          ;SHIFT BIT POINTER
4484
4485 022776 012102          MOV      (R1)+,R2        ;R2=CSR ADDR VALUE
4486 023000 012703 002416          MOV      #MPCSR,R3      ;R3=POINTER TO CSR ADDR STORAGE AREA
4487
4488 023004 010223 11$:  MOV      R2,(R3)+        ;PUT CSR ADDRESSES IN 'BSEL' AREA
4489 023006 005202          INC      R2              ;BUMP BSEL ADDR
4490 023010 022703 002456          CMP      #BSEL17+2,R3   ;ALL 16 ADDRESSES MOVED ?
4491 023014 001373          BNE
4492          ;NO: DO ANOTHER ADDRESS
4493          ;YES: CONTINUE
4494 023016 011137 002456          MOV      (R1),MPIVEC     ;GET DMV11 INPUT INTRPT VECTOR
4495 023022 012137 002460          MOV      (R1)+,MPOVEC
4496 023026 062737 000004 002460  ADD      #4,MPOVEC       ;GET DMV11 OUTPUT INTRPT VECTOR
4497 023034 012137 002462          MOV      (R1)+,MPRIOR   ;GET DMV11 DEVICE PRIORITY
4498 023040 012137 002464          MOV      (R1)+,LUSW1    ;GET LU SWITCH PACK #1
4499 023044 012137 002466          MOV      (R1)+,LUSW2    ;GET LU SWITCH PACK #2
4500 023050 012137 002470          MOV      (R1)+,BRDTYP   ;GET DMV-11 BOARD TYPE
4501 023054 012137 002472          MOV      (R1)+,TSTCON   ;GET TEST CONNECTOR INDICATOR
4502 023060 011137 002474          MOV      (R1),BDRATE    ;GET BAUD RATE FOR THIS DEVICE
4503          ;ISSUE LSI BUS RESET, TO INIT DMV11
4504 023064          BRESET
4505 023064 104433          TRAP      CS$RESET
4506 023066
4507 023066
4508 023066
4509 023066 104411          L10013:  TRAP      CS$INIT

```

CVDMEA.P11 10-DEC-80 09:16

AUTO DROP UNIT SECTION

4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526 023070
4527 023070
4528
4529 023070
4530 023070 012746 000000
4531 023074 012746 023206
4532 023100 012746 000004
4533 023104 012746 000003
4534 023110 104437
4535 023112 062706 000010
4536 023116 005037 002546
4537 023122 012702 000001
4538 023126 013703 002416
4539
4540 023132 105723
4541 023134 006302
4542 023136 103375
4543
4544 023140 013703 002416
4545 023144 012702 000001
4546 023150 005723
4547 023152 006302
4548 023154 006302
4549 023156 103374
4550
4551 023160
4552 023160 012700 000004
4553 023164 104436
4554 023166 005737 002546
4555 023172 001403
4556 023174
4557 023174 013700 002334
4558 023200 104451
4559
4560 023202 000240
4561
4562 023204
4563 023204
4564 023204 104461
4565

.SBTTL AUTO DROP UNIT SECTION

```

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

```

```

:*****
:
: THIS ALGORITHM IS THE SAME A CVDMA TEST # 1 EXCEPT THAT TEST
: WILL JUST REPORT THE FAILURE AND GO ON -- THIS ROUTINE WILL CAUSE THE
: DEVICE TO BE DROPPED IF A BUS-TIMEOUT OCCURS WHEN ANY OF THE CSR'S
: ARE ACCESSED WITH EITHER A 'TST' OR 'TSTB' INSTRUCTION.
:
:-----*****

```

```

      BGNAUTO
      LSAUTO::
      SETVEC #4,#AD.HIT,#0 ;SETUP INVALID-ADDRESS TRAP VECTOR
                                MOV #0,-(SP)
                                MOV #AD.HIT,-(SP)
                                MOV #4,-(SP)
                                MOV #3,-(SP)
                                TRAP C$SVEC
                                ADD #10,SP
      CLR TMP0 ;INITIALIZE TRAP FLAG REGISTER
      MOV #1,R2 ;FLAG BIT
      MOV BSEL0,R3 ;INIT ADDRESS POINTER
1$: TSTB (R3)+ ;ACCESS THE CSR'S BY BYTES.
    ASL R2
    BCC 1$
      MOV BSEL0,R3 ;RE-INIT ADDRESS POINTER
      MOV #1,R2 ;RE-INIT FLAG BIT
2$: TST (R3)+ ;ACCESS THE CSR'S BY WORDS.
    ASL R2
    ASL R2
    BCC 2$
      CLRVEC #4 ;RESTORE THE VECTOR TO DS
                                MOV #4,R0
                                TRAP C$CVFC
      TST TMP0 ;DID WE GET HIT WITH AN INVALID ADDRESS TRAP
      BEQ AD.OK ;NO, EXIT TEST
      DODU LOGDEV ;YES, DROP THIS LOGICAL DEV.
                                MOV LOGDEV,R0
                                TRAP C$DODU
      AD.OK: NOP ;(FOR PATCHING IN A HALT IF NECESSARY)
      ENDAUTO
                                L10014:
                                TRAP C$AUTO

```

CVDMEA.P11 10-DEC-80 09:16

AUTO DROP UNIT SECTION

4566 023206 050237 002546
4567 023212 000002
4568

AD.HIT: BIS R2, TMP0
RTI

;FLAG THE HIT IF WE GET IT!
;RETURN

CVDMEA.P11 10-DEC-80 09:16

CLEANUP CODING SECTION

4569
 4570
 4571
 4572
 4573
 4574
 4575
 4576 023214
 4577 023214
 4578
 4579
 4580 023214
 4581 023214
 4582 023214 104412

.SBTTL CLEANUP CODING SECTION

:///
 :// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
 :// AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
 :///

BGNCLN

L\$CLEAN::

ENDCLN

L10015:
 TRAP C\$CLEAN

CVDMEA.P11 10-DEC-80 09:16

DROP UNIT SECTION

4583
 4584
 4585
 4586
 4587
 4588
 4589
 4590 023216
 4591 023216
 4592
 4593 023216
 4594 023216 104433
 4595 023220
 4596 023220
 4597 023220 104453

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

      BGNDU
:ISSUE UNIBUS RESEi TO CLEAN UP
      BRESET

      ENDDU

      LSDU::
              TRAP    CSRESET
L10016:
              TRAP    CSDU
```

CVDMEA.P11 10-DEC-80 09:16

ADD UNIT SECTION

4598
4599
4600
4601
4602
4603
4604
4605
4606 023222
4607 023222
4608 023222
4609 023222
4610 023222 104452

.SBTTL ADD UNIT SECTION

```

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

```

BGNAU

ENDAU

LSAU::

L10017:

TRAP CSAU

CVDMEA.P11 10-DEC-80 09:16

TEST 1 -- RX DATA FLUSHING TEST

.SBTTL TEST 1 -- RX DATA FLUSHING TEST

```

*****
*
* TEST 1 -- RX DATA FLUSHING TEST
*
* IN BCP MODE/HALF DUPLEX IT IS DESIRABLE TO HAVE THE ABILITY TO FLUSH
* THE USYRT OF ITS CRC CHARACTERS. THIS FLUSHING IS ACCOMPLISHED BY READING
* TO THE VIA SHIFT REGISTER.
* THIS TEST VERIFIES THAT WHEN THE VIA SR IS READ, 8 PULSES WILL
* BE GENERATED AT THE CB1 PIN (WHICH DIRECTLY FEEDS THE CHARACTER FIFO).
*
*****

```

4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666

023224
023224 004737 005376
023230 004537 007234
023234 063626
023236 000000
023240 103003
023242
023242 104460
023244
023244 104410
023246 000254
023250 004537 003712
023254 120013
023256 000210
023260 103003
023262
023262 104460
023264
023264 104410
023266 000234
023270 004537 010250
023274 000001
023276 000007
023300 004537 010250
023304 000001
023306 000010
023310 004537 010250
023314 000000
023316 000000
023320 004537 010136
023324 000125
023326 000010
023330 103003

BGNTST

T1::

```

JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP

JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!NOCHK!SYNCH!STRIPS ;SET DDCMP,NO CHECK,SYNCH=226
0 ;USE 8 BIT CHARS
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR

ESCAPE TST ;SKIP TO END OF TEST TRAP C$ERROR

JSR R5,WRITEI ;SET SHIFT REGISTER TO
VIAACR ;"SYSTEM CLOCK RATE" MODE (CB1=CLK)
210 ;(BIT 7 PREVIOUSLY SET)
BCC .+8. ;BR IF NO ERROR
ERROR ; REPORT STACKED ERROR

ESCAPE TST ;SKIP TO END OF TEST TRAP C$ERROR

JSR R5,TXCTRL ;OUTPUT 1ST SYNC CHARACTER
TSOM ;AND KNOCK DOWN TBMT
7.

JSR R5,TXCTRL ;OUTPUT 2ND SYNC CHARACTER
TSOM ;AND KNOCK DOWN TBMT
8.

JSR R5,TXCTRL ;CLEAR TSOM (CCT READY TO SEND DATA)
000
0

JSR R5,TXCHAR ;LOAD 125, TX 3RD SYNCH
125
8.
BCC .+8. ;BR IF NO ERROR

```

CVDMEA.P11 10-DEC-80 09:16

TEST 1 -- RX DATA FLUSHING TEST

```

4667 023332
4668 023332 104460
4669 023334
4670 023334 104410
4671 023336 000164
4672
4673 023340 012702 000004
4674
4675 023344 004537 003712
4676 023350 120000
4677 023352 000142
4678 023354 103003
4679 023356
4680 023356 104460
4681 023360
4682 023360 104410
4683 023362 000140
4684
4685 023364 004537 012072
4686 023370 000001
4687
4688 023372 004537 003712
4689 023376 120000
4690 023400 000156
4691 023402 103003
4692 023404
4693 023404 104460
4694 023406
4695 023406 104410
4696 023410 000112
4697
4698 023412 004537 003566
4699 023416 120012
4700 023420 000000
4701 023422 103003
4702 023424
4703 023424 104460
4704 023426
4705 023426 104410
4706 023430 000072
4707
4708 023432 077234
4709
4710 023434 004537 010350
4711 023440 000377
4712 023442 000000
4713 023444 100000
4714 023446 103003
4715 023450
4716 023450 104460
4717 023452
4718 023452 104410
4719 023454 000046
4720
4721 023456 004537 010350
4722 023462 000003

```

```

ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST
;-----
MOV #4,R2 ;** TRANSFER 4 CHARACTERS **
1$: JSR R5,WRITEI ;SET RTS & FULL DUPLEX (SO STEPLU WORKS)
VIAORB
TXEN!RXEN!TTLOOP
BCC .+8. ;BR IF NO ERROR
ERROR ; REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST
;-----
JSR R5,STEPLU ;FLIP TSO BIT VALUE(WILL BE SHIFTED INTO
1 ; FIFO DURING FLUSHING).
VIAORB
JSR R5,WRITEI ;CLEAR RTS, SET HDX (SO THAT SR CLOCK WORKS)
VIAORB
TXEN!RXEN!RTSND!HDX!TTLOOP
BCC .+8. ;BR IF NO ERROR
ERROR ; REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST
;-----
JSR R5,READI ;READ VIA SHIF? REGISTER (SHOULD CAUSE
VIA SR ; 8 CLOCKS FROM CB1 LEAD => FIFO)
000
BCC .+8. ;BR IF NO ERROR
ERROR ; REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST
;-----
SOB R2,1$ ;** LOOP UNTIL ALL 4 SENT (VIA CB1) **
;-----
JSR R5,RXCHAR ;READ AND CHECK FOR 377
377 ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
0 ;* DIDN'T WORK.
NOCRDA
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST
;-----
JSR R5,RXCHAR ;READ AND CHECK FOR 000
003 ;* ERROR HERE INDICATES HI-SPEED SR CLOCK

```

CVDMEA.P11 10-DEC-80 09:16

TEST 1 -- RX DATA FLUSHING TEST

```

4723 023464 000000
4724 023466 100000
4725 023470 103003
4726 023472
4727 023472 104460
4728 023474
4729 023474 104410
4730 023476 000024
4731
4732 023500 004537 010350
4733 023504 000360
4734 023506 000000
4735 023510 100000
4736 023512 103003
4737 023514
4738 023514 104460
4739 023516
4740 023516 104410
4741 023520 000002
4742 023522
4743 023522
4744 023522 104401

```

```

0
NOCRDA
BCC .+8.
ERROR
ESCAPE TST
JSR R5,RXCHAR
360
0
NOCRDA
BCC .+8.
ERROR
ESCAPE TST
ENDTST

```

```

;* DIDN'T WORK.
;BR IF NO ERROR
;REPORT STACKED ERROR
;SKIP TO END OF TEST
TRAP C$ERROR
TRAP C$ESCAPE
.WORD L10020-.
;READ AND CHECK FOR 377
;* ERROR HERE INDICATES HI-SPEED SR CLOCK
;* DIDN'T WORK
;BR IF NO ERROR
;REPORT STACKED ERROR
;SKIP TO END OF TEST
TRAP C$ERROR
TRAP C$ESCAPE
.WORD L10020-.
L10020:
TRAP C$ETST

```

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

.SBTTL TEST 2 -- INTEGRAL MODEM INTERFACE TEST

4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800

023524
023524 012737 000002 002414
023532 004737 005376
023536 004537 007532
023542 000001
023544 103002
023546
023546 104432
023550 001630

023552 012701 025402
023556 012702 001000
023562 112137 023600
023566 010237 023576
023572 004537 003712
023576 000000
023600 000000
023602 005202
023604 020127 026227
023610 103764

023612 012701 025402
023616 012702 001000
023622 010237 023632
023626 004537 003566
023632 000000
023634 000000
023636 122137 023634
023642 001422
023644 010237 002336
023650 005037 002324
023654 116137 177777 002324
023662 005037 002326
023666 113737 023634 002326

* TEST 2 -- INTEGRAL MODEM INTERFACE TEST
* THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM AND A MESSAGE IS
* TRANSMITTED, RECEIVED, AND CHECKED USING A TURNAROUND CONNECTOR ON
* THE BOARD OR AT THE END OF A CABLE. THE FOLLOWING MESSAGE WILL BE
* SENT IN BCP MODE WITH CRC-16 SPECIFIED:
* SYNC SYNC 000 125 252 377 000 CRC1 CRC2 SYNC
* IF THE P-TABLE FOR THE CURRENT UNIT INDICATES THAT NO EXTERNAL
* TURNAROUND IS PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.

```

: BGNTEST
:
: T2:
: MOV #2, TSTNUM ;SET TEST NO. FOR POSSIBLE PRINTOUT
: JSR PC, INIDMV ;INIT DMV-11, ENTER MAINT LOOP
: JSR R5, CKLPBK ;SEE IF THIS INTERFACE CAN BE RUN
: INTGRL
: BCC 2$ ;BR IF YES
: EXIT TST ;WRONG INTERFACE - SKIP TEST
: TRAP C$EXIT
: .WORD L10021-

: LOAD 6502 MICROCODE FOR INTEGRAL MODEM TEST INTO RAM PAGE 2
2$: MOV #MPCODE, R1 ;GET STARTING ADRS OF DMV MICROCODE
: MOV #RAMADR, R2 ;GET STARTING ADRS OF RAM PAGE 2
3$: MOVB (R1)+, 6$ ;SET DATA BYTE TO BE WRITTEN
: MOV R2, 4$ ;SET RAM WRITE ADRS
: JSR R5, WRITEI ;WRITE A DATA BYTE INTO RAM
4$: .WORD 0
6$: .WORD 0
: INC R2 ;INCR RAM ADRS
: CMP R1, #ENDCOD ;SEE IF ALL CODE LOADED YET
: BLO 3$ ;BR IF NOT

: READ AND VERIFY 6502 MICROCODE IN RAM
: MOV #MPCODE, R1 ;GET STARTING ADRS OF DMV MICROCODE TO CHECK
: MOV #RAMADR, R2 ;GET STARTING ADRS OF RAM PAGE 2
8$: MOV R2, 10$ ;SET RAM READ ADRS
: JSR R5, READI ;READ A RAM BYTE
10$: .WORD 0
12$: .WORD 0
: CMPB (R1)+, 12$ ;SEE IF BYTE IS CORRECT
: BEQ 16$ ;BR IF CORRECT
: MOV R2, REGNUM ;SET RAM ADRS FOR ERROR REPORT
: CLR GDATA ;SET EXPECTED RAM DATA
: MOVB -1(R1), GDATA
: CLR BDATA ;SET ACTUAL RAM DATA
: MOVB 12$, BDATA
: REPORT RAM ERROR LOADING MICROCODE
```

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

4801 023674          GF = EM79,ERR13          'DEVICE FATAL' ERROR # 38
4802
4803 023674 104455          TRAP C$ERDF
4804 023676 000046          .WORD 38
4805 023700 015125          .WORD EM79
4806 023702 021030          .WORD ERR13
4807 023704          ESCAPE TST
4808 023704 104410          TRAP C$ESCAPE
4809 023706 001472          .WORD L10021-.
4810 023710 005202
4811 023712 020127 026227 16$: INC R2          ;INCR RAM ADRS
4812 023716 103741          CMP R1,#ENDCOD ;SEE IF ALL CODE CHECKED YET
4813          BLO 8$          ;BR IF NOT
4814
4815 023720 004537 003712 ;SET UP VIA AND USYRT FOR OPERATION
4816 023724 120000          JSR R5,WRITEI ;RESET THE USYRT
4817 023726 000031          VIAORB
4818 023730 004537 003712 RTSND!DTR!PRESET
4819 023734 120000          JSR R5,WRITEI ;CLEAR USYRT RESET BIT
4820 023736 000030          VIAORB
4821 023740 004537 003712 RTSND!DTR
4822 023744 120404          JSR R5,WRITEI ;SET SYNCH CHAR = 226
4823 023746 000226          PCSARL
4824 023750 012737 065400 023770 MOV #DDCMP!STRIPS!IDLES!CRC16,18$ ;SET DDCMP,STRIP,IDLE, CRC16
4825 023756 000337 023770 SWAB 18$          ;GET DATA INTO LO BYTE
4826 023762 004537 003712 JSR R5,WRITEI ;PROGRAM THE PCSARH
4827 023766 120405          PCSARH
4828 023770 000000          18$: .WORD 0
4829 023772 004537 005410 JSR R5,CKUSTS ;CHK USYRT STATUS FOR INITIALIZED STATE
4830 023776 000110          110
4831 024000 103003          BCC .+8. ;IF ERROR, PRINT REPORT
4832 024002          ERROR
4833 024002 104460          TRAP C$ERROR
4834 024004          ESCAPE TST
4835 024004 104410          TRAP C$ESCAPE
4836 024006 001372          .WORD L10021-.
4837 024010 004537 003712 JSR R5,WRITEI ;SET TSOM IN USYRT
4838 024014 120403          TDSRH
4839 024016 000001          TSOM
4840 024020 004537 003712 JSR R5,WRITEI ;LOAD 237 CHAR FOR INTGRL MODEM SYNCHRONIZATION
4841 024024 120402          TDSRL
4842 024026 000237          237
4843 024030 004537 006014 JSR R5,CKT$MT ;CHK FOR TBMT = 0
4844 024034 000000          0
4845 024036 103003          BCC .+8. ;IF ERROR, REPORT ERROR
4846 024040          ERROR
4847 024040 104460          TRAP C$ERROR
4848 024042          ESCAPE TST
4849 024042 104410          TRAP C$ESCAPE
4850 024044 001334          .WORD L10021-.
4851
4852 ;INITIATE 6502 TEST OF INTEGRAL MODEM
4853 024046 005077 156354 CLR @SEL4 ;CLEAR SEL4
4854 024052 012777 001000 156346 MOV #RAMADR,@SEL4 ;SET START ADRS OF RAM CODE IN SEL4
4855 024060 112777 000005 156334 MOVB #EXECUT,@SEL2 ;ISSUE M-LOOP CMND TO EXECUTE AT PC IN SEL4
4856 ;WAIT SEVERAL MILLI-SEC FOR COMPLETION OF TEST

```

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

4857 024066 012701 001750
4858 024072 005301
4859 024074 000240
4860 024076 001375
4861 024100 132777 000200 156314
4862 024106 001013
4863 024110 004737 004166
4864 024114 012737 000005 002324
4865
4866 024122
4867
4868 024122 104455
4869 024124 000047
4870 024126 014307
4871 024130 020120
4872 024132
4873 024132 104410
4874 024134 001244
4875
4876 024136 105777 156270
4877 024142 001002
4878 024144 000137 025400
4879 024150 012737 000007 002336
4880 024156 005037 002324
4881 024162 005037 002326
4882
4883 024166 127727 156240 000001
4884 024174 001006
4885
4886 024176
4887
4888 024176 104455
4889 024200 000050
4890 024202 014777
4891 024204 020714
4892 024206
4893 024206 104410
4894 024210 001170
4895
4896 024212 127727 156214 000002
4897 024220 001006
4898
4899 024222
4900
4901 024222 104455
4902 024224 000051
4903 024226 014777
4904 024230 020714
4905 024232
4906 024232 104410
4907 024234 001144
4908
4909 024236 127727 156170 000003
4910 024244 001006
4911
4912 024246

22$: MOV #1000.,R1 ;INIT WAIT LOOP COUNTER
DEC R1 ;DECREMENT COUNTER
NOP
BNE 22$ ;BR IF NOT DONE COUNTING YET
BITB #MRDY,@BSEL2 ;SEE IF M-LOOP FINISHED PROPERLY
BNE 24$ ;BR IF YES
JSR PC,GETWSR ;GET CSR'S FOR PRINTOUT
MOV #EXECUT,GDATA ;IDENTIFY REQUESTED FUNCTION
;REPORT MRDY TIMEOUT ERROR
GEDF EM4,ERR4
; 'DEVICE FATAL' ERROR # 39
TRAP C$ERDF
.WORD 39
.WORD EM4
.WORD ERR4
ESCAPE TST
TRAP C$ESCAPE
.WORD L10021-.

24$: TSTB @BSEL6 ;SEE IF ANY ERRORS OCCURRED IN TEST
BNE 26$ ;BR IF ERROR DETECTED
JMP 90$ ;SUCCESSFUL COMPLETION OF TEST
26$: MOV #7,REGNUM ;SET REG NO. FOR PRINTOUT
CLR GDATA ;CLEAR EXPECTED DATA AREA
CLR BDATA ;CLEAR ACTUAL DATA AREA
;CHK FOR ERROR 1
CMPB @BSEL6,#1 ;CHK FOR ERROR 1
BNE 28$ ;BR IF NOT
;REPORT TBMT NOT SET
GEDF EM73,ERR12
; 'DEVICE FATAL' ERROR # 40
TRAP C$ERDF
.WORD 40
.WORD EM73
.WORD ERR12
ESCAPE TST
TRAP C$ESCAPE
.WORD L10021-.

;CHK FOR ERROR 2
28$: CMPB @BSEL6,#2 ;CHK FOR ERROR 2
BNE 30$ ;BR IF NOT
;REPORT TBMT NOT SET
GEDF EM73,ERR12
; 'DEVICE FATAL' ERROR # 41
TRAP C$ERDF
.WORD 41
.WORD EM73
.WORD ERR12
ESCAPE TST
TRAP C$ESCAPE
.WORD L10021-.

;CHK FOR ERROR 3
30$: CMPB @BSEL6,#3 ;CHK FOR ERROR 3
BNE 31$ ;BR IF NOT
;REPORT TBMT NOT SET
GEDF EM73,ERR12

```

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

4913                                     ; 'DEVICE FATAL' ERROR # 42
4914 024246 104455                       TRAP  C$ERDF
4915 024250 000052                       .WORD 42
4916 024252 014777                       .WORD EM73
4917 024254 020714                       .WORD ERR12
4918 024256                               ESCAPE TST
4919 024256 104410                       TRAP  C$ESCAPE
4920 024260 001120                       .WORD L10021-.
4921                                     ;CHK FOR ERROR 4
4922 024262 127727 156144 000004 31$:  CMPB  @BSEL6,#4      ;CHK FOR ERROR 4
4923 024270 001011                       BNE  32$      ;BR IF NOT
4924                                     ;REPORT CARRIER NOT SET
4925 024272 012737 000001 002336  MOV  #1,REGNUM  ;SET REG NO. FOR PRINTOUT
4926 024300                       GEDF  EM80,ERR14
4927                                     ; 'DEVICE FATAL' ERROR # 43
4928 024300 104455                       TRAP  C$ERDF
4929 024302 000053                       .WORD 43
4930 024304 015161                       .WORD EM80
4931 024306 021146                       .WORD ERR14
4932 024310                               ESCAPE TST
4933 024310 104410                       TRAP  C$ESCAPE
4934 024312 001066                       .WORD L10021-.
4935                                     ;CHK FOR ERROR 5
4936 024314 127727 156112 000005 32$:  CMPB  @BSEL6,#5      ;CHK FOR ERROR 5
4937 024322 001006                       BNE  34$      ;BR IF NOT
4938                                     ;REPORT TBMT NOT SET
4939 024324                       GEDF  EM73,ERR12
4940                                     ; 'DEVICE FATAL' ERROR # 44
4941 024324 104455                       TRAP  C$ERDF
4942 024326 000054                       .WORD 44
4943 024330 014777                       .WORD EM73
4944 024332 020714                       .WORD ERR12
4945 024334                               ESCAPE TST
4946 024334 104410                       TRAP  C$ESCAPE
4947 024336 001042                       .WORD L10021-.
4948                                     ;CHK FOR ERROR 6
4949 024340 127727 156066 000006 34$:  CMPB  @BSEL6,#6      ;CHK FOR ERROR 6
4950 024346 001006                       BNE  36$      ;BR IF NOT
4951                                     ;REPORT TBMT NOT SET
4952 024350                       GEDF  EM73,ERR12
4953                                     ; 'DEVICE FATAL' ERROR # 45
4954 024350 104455                       TRAP  C$ERDF
4955 024352 000055                       .WORD 45
4956 024354 014777                       .WORD EM73
4957 024356 020714                       .WORD ERR12
4958 024360                               ESCAPE TST
4959 024360 104410                       TRAP  C$ESCAPE
4960 024362 001016                       .WORD L10021-.
4961                                     ;CHK FOR ERROR 7
4962 024364 127727 156042 000007 36$:  CMPB  @BSEL6,#7      ;CHK FOR ERROR 7
4963 024372 001006                       BNE  38$      ;BR IF NOT
4964                                     ;REPORT TBMT NOT SET
4965 024374                       GEDF  EM73,ERR12
4966                                     ; 'DEVICE FATAL' ERROR # 46
4967 024374 104455                       TRAP  C$ERDF
4968 024376 000056                       .WORD 46

```

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

4969 024400 014777 .WORD EM73
4970 024402 020714 .WORD ERR12
4971 024404 ESCAPE TST
4972 024404 104410 TRAP C$ESCAPE
4973 024406 000772 .WORD L10021-.
4974
4975 024410 127727 156016 000010 ;CHK FOR ERROR 8
38$: CMPB @BSEL6,#8. ;CHK FOR ERROR 8
BNE 40$ ;BR IF NOT
4976 024416 001006 ;REPORT TBM NOT SET
GEDF EM73,ERR12
4977
4978 024420 ; 'DEVICE FATAL' ERROR # 47
4979 TRAP C$ERDF
4980 024420 104455 .WORD 47
4981 024422 000057 .WORD EM73
4982 024424 014777 .WORD ERR12
4983 024426 020714
4984 024430 ESCAPE TST
4985 024430 104410 TRAP C$ESCAPE
4986 024432 000746 .WORD L10021-.
4987
4988 024434 127727 155772 000011 ;CHK FOR ERROR 9
40$: CMPB @BSEL6,#9. ;CHK FOR ERROR 9
BNE 42$ ;BR IF NOT
4989 024442 001006 ;REPORT RDA NOT SET
GEDF EM75,ERR12
4990
4991 024444 ; 'DEVICE FATAL' ERROR # 48
4992 TRAP C$ERDF
4993 024444 104455 .WORD 48
4994 024446 000060 .WORD EM75
4995 024450 015035 .WORD ERR12
4996 024452 020714
4997 024454 ESCAPE TST
4998 024454 104410 TRAP C$ESCAPE
4999 024456 000722 .WORD L10021-.
5000
5001 024460 127727 155746 000012 ;CHK FOR ERROR 10
42$: CMPB @BSEL6,#10. ;CHK FOR ERROR 10
BNE 44$ ;BR IF NOT
5002 024466 001017 ;REPORT RCV'D DATA MISCOMPARE ERROR
MOV #0,REGNUM ;SET REG NO. FOR PRINTOUT
5003 MOV #000,GDATA ;SET EXPECTED DATA
5004 024470 012737 000000 002336 MOV @BSEL7,BDATA ;SET ACTUAL DATA
5005 024476 112737 000000 002324 GEDF EM34,ERR10
5006 024504 117737 155724 002326
5007 024512 ; 'DEVICE FATAL' ERROR # 49
5008 TRAP C$ERDF
5009 024512 104455 .WORD 49
5010 024514 000061 .WORD EM34
5011 024516 014502 .WORD ERR10
5012 024520 020364
5013 024522 ESCAPE TST
5014 024522 104410 TRAP C$ESCAPE
5015 024524 000654 .WORD L10021-.
5016
5017 024526 127727 155700 000013 ;CHK FOR ERROR 11
44$: CMPB @BSEL6,#11. ;CHK FOR ERROR 11
BNE 46$ ;BR IF NOT
5018 024534 001006 ;REPORT RDA NOT SET
GEDF EM75,ERR12
5019
5020 024536 ; 'DEVICE FATAL' ERROR # 50
5021 TRAP C$ERDF
5022 024536 104455 .WORD 50
5023 024540 000062 .WORD EM75
5024 024542 015035

```


CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

5025	024544	020714							.WORD	ERR12
5026	024546				ESCAPE	TST				
5027	024546	104410							TRAP	C\$ESCAPE
5028	024550	000630							.WORD	L10021-.
5029										
5030	024552	127727	155654	000014	:CHK FOR ERROR 12					
5031	024560	001017			46\$: CMPB @BSEL6,#12.	:CHK FOR ERROR 12				
5032					BNE 48\$:BR IF NOT				
5033	024562	012737	000000	002336	:REPORT RCV'D DATA MISCOMPARE ERROR					
5034	024570	112737	000125	002324	MOV #0,REGNUM	:SET REG NO. FOR PRINTOUT				
5035	024576	117737	155632	002326	MOVB #125,GDATA	:SET EXPECTED DATA				
5036	024604				MOVB @BSEL7,BDATA	:SET ACTUAL DATA				
5037					GEDF EM34,ERR10					
5038	024604	104455				:	'DEVICE FATAL' ERROR # 51			
5039	024606	000063						TRAP	C\$ERDF	
5040	024610	014502						.WORD	51	
5041	024612	020364						.WORD	EM34	
5042	024614				ESCAPE	TST		.WORD	ERR10	
5043	024614	104410						TRAP	C\$ESCAPE	
5044	024616	000562						.WORD	L10021-.	
5045										
5046	024620	127727	155606	000015	:CHK FOR ERROR 13					
5047	024626	001006			48\$: CMPB @BSEL6,#13.	:CHK FOR ERROR 13				
5048					BNE 50\$:BR IF NOT				
5049	024630				:REPORT RDA NOT SET					
5050					GEDF EM75,ERR12					
5051	024630	104455				:	'DEVICE FATAL' ERROR # 52			
5052	024632	000064						TRAP	C\$ERDF	
5053	024634	015035						.WORD	52	
5054	024636	020714						.WORD	EM75	
5055	024640				ESCAPE	TST		.WORD	ERR12	
5056	024640	104410						TRAP	C\$ESCAPE	
5057	024642	000536						.WORD	L10021-.	
5058										
5059	024644	127727	155562	000016	:CHK FOR ERROR 14					
5060	024652	001017			50\$: CMPB @BSEL6,#14.	:CHK FOR ERROR 14				
5061					BNE 52\$:BR IF NOT				
5062	024654	012737	000000	002336	:REPORT RCV'D DATA MISCOMPARE ERROR					
5063	024662	112737	000252	002324	MOV #0,REGNUM	:SET REG NO. FOR PRINTOUT				
5064	024670	117737	155540	002326	MOVB #252,GDATA	:SET EXPECTED DATA				
5065	024676				MOVB @BSEL7,BDATA	:SET ACTUAL DATA				
5066					GEDF EM34,ERR10					
5067	024676	104455				:	'DEVICE FATAL' ERROR # 53			
5068	024700	000065						TRAP	C\$ERDF	
5069	024702	014502						.WORD	53	
5070	024704	020364						.WORD	EM34	
5071	024706				ESCAPE	TST		.WORD	ERR10	
5072	024706	104410						TRAP	C\$ESCAPE	
5073	024710	000470						.WORD	L10021-.	
5074										
5075	024712	127727	155514	000017	:CHK FOR ERROR 15					
5076	024720	001006			52\$: CMPB @BSEL6,#15.	:CHK FOR ERROR 15				
5077					BNE 54\$:BR IF NOT				
5078	024722				:REPORT RDA NOT SET					
5079					GEDF EM75,ERR12					
5080	024722	104455				:	'DEVICE FATAL' ERROR # 54	TRAP	C\$ERDF	

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

5137                                     ; 'DEVICE FATAL' ERROR # 58
5138 025106 104455                                     TRAP C$ERDF
5139 025110 000072                                     .WORD 58
5140 025112 014551                                     .WORD EM36
5141 025114 020364                                     .WORD ERR10
5142 025116                                     ESCAPE TST
5143 025116 104410                                     TRAP C$ESCAPE
5144 025120 000260                                     .WORD L10021-.
5145
5146 025122 127727 155304 000024 ;CHK FOR ERROR 20
5147 025130 001006 62$: CMPB @BSEL6,#20. ;CHK FOR ERROR 20
;REPORT RDA NOT SET ;BR IF NOT
5148                                     GEDF EM75,ERR12
5149 025132
5150                                     ; 'DEVICE FATAL' ERROR # 59
5151 025132 104455                                     TRAP C$ERDF
5152 025134 000073                                     .WORD 59
5153 025136 015035                                     .WORD EM75
5154 025140 020714                                     .WORD ERR12
5155 025142                                     ESCAPE TST
5156 025142 104410                                     TRAP C$ESCAPE
5157 025144 000234                                     .WORD L10021-.
5158
5159 025146 127727 155260 000025 ;CHK FOR ERROR 21
5160 025154 001017 64$: CMPB @BSEL6,#21. ;CHK FOR ERROR 21
;REPORT RCV'D DATA MISCOMPARE ERROR ;BR IF NOT
5161                                     MOV #0,REGNUM ;SET REG NO. FOR PRINTOUT
5162 025156 012737 000000 002336 ;MOV #160,GDATA ;SET EXPECTED DATA
5163 025164 112737 000160 002324 ;MOV @BSEL7,BDATA ;SET ACTUAL DATA
5164 025172 117737 155236 002326 ;GEDF EM34,ERR10
5165 025200
5166                                     ; 'DEVICE FATAL' ERROR # 60
5167 025200 104455                                     TRAP C$ERDF
5168 025202 000074                                     .WORD 60
5169 025204 014502                                     .WORD EM34
5170 025206 020364                                     .WORD ERR10
5171 025210                                     ESCAPE TST
5172 025210 104410                                     TRAP C$ESCAPE
5173 025212 000166                                     .WORD L10021-.
5174
5175 025214 127727 155212 000026 ;CHK FOR ERROR 22
5176 025222 001006 66$: CMPB @BSEL6,#22. ;CHK FOR ERROR 22
;REPORT RDA NOT SET ;BR IF NOT
5177                                     GEDF EM75,ERR12
5178 025224
5179                                     ; 'DEVICE FATAL' ERROR # 61
5180 025224 104455                                     TRAP C$ERDF
5181 025226 000075                                     .WORD 61
5182 025230 015035                                     .WORD EM75
5183 025232 020714                                     .WORD ERR12
5184 025234                                     ESCAPE TST
5185 025234 104410                                     TRAP C$ESCAPE
5186 025236 000142                                     .WORD L10021-.
5187
5188 025240 127727 155166 000027 ;CHK FOR ERROR 23
5189 025246 001017 68$: CMPB @BSEL6,#23. ;CHK FOR ERROR 23
;REPORT RCV'D DATA MISCOMPARE ERROR ;BR IF NOT
5190                                     MOV #0,REGNUM ;SET REG NO. FOR PRINTOUT
5191 025250 012737 000000 002336 ;MOV #034,GDATA ;SET EXPECTED DATA
5192 025256 112737 000034 002324

```

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

5193 025264 117737 155144 002326      MOVB  @BSEL7,BDATA ;SET ACTUAL DATA
5194 025272      GEDF  EM34,ERR10
5195      ; 'DEVICE FATAL' ERROR # 62
5196 025272 104455      TRAP  C$ERDF
5197 025274 000076      .WORD 62
5198 025276 014502      .WORD EM34
5199 025300 020364      .WORD ERR10
5200 025302      ESCAPE TST
5201 025302 104410      TRAP  C$ESCAPE
5202 025304 000074      .WORD L10021-.
5203      ;CHK FOR ERROR 24
5204 025306 127727 155120 000030 70$:  CMPB  @BSEL6,#2%. ;CHK FOR ERROR 24
5205 025314 001006      BNE   72$ ;BR IF NOT
5206      ;REPORT RDA NOT SET
5207 025316      GEDF  EM75,ERR12
5208      ; 'DEVICE FATAL' ERROR # 63
5209 025316 104455      TRAP  C$ERDF
5210 025320 000077      .WORD 63
5211 025322 015035      .WORD EM75
5212 025324 020714      .WORD ERR12
5213 025326      ESCAPE TST
5214 025326 104410      TRAP  C$ESCAPE
5215 025330 000050      .WORD L10021-.
5216      ;CHK FOR ERROR 25
5217 025332 127727 155074 000031 72$:  CMPB  @BSEL6,#25. ;CHK FOR ERROR 25
5218 025340 001011      BNE   74$ ;BR IF NOT
5219      ;REPORT CARRIER NOT CLEARED
5220 025342 012737 000001 002336  MOV  #1,REGNUM ;SET REG NO. FOR PRINTOUT
5221 025350      GEDF  EM81,ERR14
5222      ; 'DEVICE FATAL' ERROR # 64
5223 025350 104455      TRAP  C$ERDF
5224 025352 000100      .WORD 64
5225 025354 015201      .WORD EM81
5226 025356 021146      .WORD ERR14
5227 025360      ESCAPE TST
5228 025360 104410      TRAP  C$ESCAPE
5229 025362 000016      .WORD L10021-.
5230 025364 004737 004166 74$:  JSP  PC,GETWSR ;GET CSR'S FOR PRINTOUT
5231      ;REPORT INVALID ERROR CODE FROM 6502
5232 025370      GEDF  EM82,ERR3
5233      ; 'DEVICE FATAL' ERROR # 65
5234 025370 104455      TRAP  C$ERDF
5235 025372 000101      .WORD 65
5236 025374 015225      .WORD EM82
5237 025376 020072      .WORD ERR3
5238
5239 025400      90$:
5240 025400      ENDTST
5241 025400
5242 025400 104401      L10021: TRAP  C$ETST

```

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

5243 025402
5244
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297
5298

MCODE:

```

:LINE# LOC CODE LINE
:0001 0000 *=$0200 ;START OF MICROCODE FOR INTEGRAL
:0002 0200
:0003 0200
:0005 0200
:0006 0200
:0007 0200
:0008 0200
:0009 0200
:0010 0200
:0011 0200
:0012 0200
:0013 0200
:0014 0200
:0015 0200
:0016 0200
:0017 0200
:0018 0200
:0019 0200
:0020 0200
:0021 0200
:0022 0200
:0023 0200
:0024 0200
:0025 0200
:0026 0200
:0027 0200
:0028 0200
:0029 0200
:0030 0200
:0031 0200
:0032 0200
:0033 0200
:0034 0200
:0035 0200
:0036 0200
:0037 0200
:0038 0200
:0039 0200
:0040 0200
:0041 0200
:0042 0200
:0043 0200
:0044 0200
:0045 0200
:0046 0200
:0047 0200
:0048 0200
:0049 0200
:0050 0200
:0051 0200
:0052 0200
:0053 0200

```

: THIS IS THE 6502 MICROCODE WHICH IS LOADED INTO RAM AND EXECUTED FOR THE
: PURPOSE OF TESTING THE INTEGRAL MODEM ON THE M8064, AT 56K BAUD. AFTER TH
: LSI-11 PROGRAM DOES SOME INITIAL SETUP, IT TRANSFERS CONTROL TO THIS CODE
: IN RAM, AND WAITS FOR COMPLETION OF THE TEST, AS INDICATED BY MRDY SET.
: THIS CODE TRANSMITS, RECEIVES, AND CHECKS THE FOLLOWING CHARACTERS :
: 2 SYNCH CHARACTERS, 5 DATA CHARACTERS 000, 125, 252, 377, 000, 2 CRC-16
: CHARACTERS 160 AND 034, AND 2 TERMINATING SYNCHS. THE MESSAGE IS SENT USI
: CHARACTER (DDCMP) MODE, THE SYNCH CHARACTER USED IS 226, STRIP SYNCH AND
: IDLE MODES ARE SET, AND THE DATA CLOCK IS PROVIDED BY THE INTEGRAL MODEM.
: ALL DATA AND CRC CHARACTERS ARE CHECKED AS THEY ARE RECEIVED, AND THE CRC
: ERROR CHECK BIT IS CHECKED TO BE SET WITH RECEPTION OF THE LAST DATA
: CHARACTER (000).

```

: EQUATES FOR BIT DEFINITIONS
BIT0 =@1
BIT1 =@2
BIT2 =@4
BIT3 =@10
BIT4 =@20
BIT5 =@40
BIT6 =@100
BIT7 =@200
BIT8 =@400
BIT9 =@1000
BIT10 =@2000
BIT11 =@4000
BIT12 =@10000
BIT13 =@20000
BIT14 =@40000
BIT15 =@100000

```

```

: ADDRESS EQUATES FOR CSR REGISTERS
SEL0 = $10
BSEL0 = SEL0
BSEL1 = SEL0+1
SEL2 = SEL0+2
BSEL2 = SEL0+2
BSEL3 = SEL0+3
SEL4 = SEL0+4
BSEL4 = SEL0+4
BSEL5 = SEL0+5
SEL6 = SEL0+6
BSEL6 = SEL0+6
BSEL7 = SEL0+7

```

CVDMEA.P11

10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

5299 :0054 0200
5300 :0055 0200
5301 :0056 0200
5302 :0057 0200
5303 :0058 0200
5304 :0059 0200
5305 :0060 0200
5306 :0061 0200
5307 :0062 0200
5308 :0063 0200
5309 :0064 0200
5310 :0065 0200
5311 :0066 0200
5312 :0067 0200
5313 :0068 0200
5314 :0069 0200
5315 :0070 0200
5316 :0071 0200
5317 :0072 0200
5318 :0073 0200
5319 :0074 0200
5320 :0075 0200
5321 :0076 0200
5322 :0077 0200
5323 :0078 0200
5324 :0079 0200
5325 :0080 0200
5326 :0081 0200
5327 :0082 0200
5328 :0083 0200
5329 :0084 0200
5330 :0085 0200
5331 :0086 0200
5332 :0087 0200
5333 :0088 0200
5334 :0089 0200
5335 :0090 0200
5336 :0091 0200
5337 :0092 0200
5338 :0093 0200
5339 :0094 0200
5340 :0095 0200
5341 :0096 0200
5342 :0097 0200
5343 :0098 0200
5344 :0099 0200
5345 :0100 0200
5346 :0101 0200
5347 :0102 0200
5348 :0103 0200
5349 :0104 0200
5350 :0105 0200
5351 :0106 0200
5352 :0107 0200
5353 :0108 0200
5354 :0109 0200

```

```

;VERSATILE INTERFACE ADAPTER REGISTER EQUATES
OREGB = $A000 ;OUTPUT REGISTER B
OREGA = OREGB+1 ;OUTPUT REGISTER A
DDR8 = OREGB+2 ;DATA DIRECTION REGISTER B
DDRA = OREGB+3 ;DATA DIRECTION REGISTER A
T1LL = OREGB+6 ;TIMER 1 LATCH LOW BITS
T1LH = OREGB+7 ;TIMER 1 LATCH HIGH BITS
ACR = OREGB+$8 ;AUXILIARY CONTROL REGISTER
PCR = OREGB+$C ;PERIPHERAL CONTROL REGISTER

```

```

;VIA OUTPUT REGISTER B BIT EQUATES
NULCLK = BIT7
RXEN = BIT6
TXEN = BIT5
DTR = BIT4
RTSND = BIT3
HDX = BIT2
TTLOOP = BIT1
PRESET = BIT0

```

```

;VIA OUTPUT REGISTER A BIT EQUATES
RING = BIT7
CARRIER = BIT6
MDMRDY = BIT5
BDRATE = BIT4
CTS = BIT3
TM = BIT2
RCVDAT = BIT1
UMAIN = BIT0

```

```

;USYRT REGISTER ADDRESS EQUATES
RXDB = $A100
RDSR = RXDB+1
TXDB = RXDB+2
TDSR = RXDB+3
SAR = RXDB+4
PCSAR = RXDB+5
PCTLR = RXDB+7

```

```

;USYRT TDSR REGISTER BIT EQUATES
TEQM = BIT1
TSOM = BIT0

```

```

;USYRT RDSR BIT EQUATES

```

VDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

5355 :0110 0200 RERR = BIT7
5356 :0111 0200
5357 :0112 0200
5358 :0113 0200
5359 :0114 0200
5360 :0115 0200 ;USYRT STATUS REGISTER EQUATES
5361 :0116 0200 USTATR = $A400
5362 :0117 0200 RDA = BIT7
5363 :0118 0200 TBMT = BIT6
5364 :0119 0200 RXACT = BIT5
5365 :0120 0200 RSA = BIT4
5366 :0121 0200 TSO = BIT3
5367 :0122 0200 TXACT = BIT2
5368 :0123 0200 TXUERR = BIT1
5369 :0124 0200 SYNFLG = BIT0
5370 :0125 0200
5371 :0126 0200
5372 :0127 0200 ;MISCELLANEOUS EQUATES
5373 :0128 0200 SYNCH = @226
5374 :0129 0200
5375 :0130 0200
5376 :0131 0200 A0 30 LDY #0
5377 025402 240 000 .BYTE 240,000
5378 :0132 0202 84 16 STY BSEL6 ;CLEAR BSEL6
5379 025404 204 026 .BYTE 204,026
5380 :0133 0204 84 17 STY BSEL7 ;CLEAR BSEL7
5381 025406 204 027 .BYTE 204,027
5382 :0134 0206 ;TURN ON THE USYRT, CLOCK
5383 :0135 0206 A2 60 LDX #TXEN!RXEN ;ASSERT TXEN,RXEN,RTS,DTR
5384 025410 242 140 .BYTE 242,140
5385 :0136 0208 8E 00 A0 STX OREGB ; AND RELEASE INT MODEM RESET
5386 025412 216 000 240 .BYTE 216,000,240
5387 :0137 0208 A2 00 LDX #0 ;INIT TBMT TIME-OUT COUNTER
5388 025415 242 000 .BYTE 242,000
5389 :0138 020D 2C 00 A4 BIT USTATR ;SEE IF TBMT SET
5390 025417 054 000 244 .BYTE 054,000,244
5391 :0139 0210 70 08 BVS **10 ;BR IF TBMT SET
5392 025422 160 010 .BYTE 160,010
5393 :0140 0212 EB INX ;INCREMENT TIME-OUT COUNTER
5394 025424 350 .BYTE 350
5395 :0141 0213 D0 F8 BNE *-6 ;BR IF NO TIME-OUT
5396 025425 320 370 .BYTE 320,370
5397 :0142 0215 ; *** ERROR 1 ***
5398 :0143 0215 A0 01 LDY #1 ;SET CODE FOR TBMT TIME-OUT ERRO
5399 025427 240 001 .BYTE 240,001
5400 :0144 0217 4C 90 03 JMP A100 ;GO TAKE ERROR EXIT
5401 025431 114 220 003 .BYTE 114,220,003
5402 :0145 021A ;LOAD FIRST SYNCH CHAR INTO TRANSMITTER
5403 :0146 021A A2 96 LDX #SYNCH ;LOAD FIRST SYNCH CHAR
5404 025434 242 226 .BYTE 242,226
5405 :0147 021C 8E 02 A1 STX TXDB
5406 025436 216 002 241 .BYTE 216,002,241
5407 :0148 021F A2 00 LDX #0 ;INIT TBMT TIME-OUT COUNTER
5408 025441 242 000 .BYTE 242,000
5409 :0149 0221 2C 00 A4 BIT USTATR ;SEE IF TBMT SET
5410 025443 054 000 244 .BYTE 054,000,244

```

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

5411					:0150	0224 70 08	BVS	**+10		;BR IF TBMT SET
5412	025446	160	010			.BYTE 160,010				
5413					:0151	0226 E8	INX			;INCREM TIME-OUT COUNTER
5414	025450	350				.BYTE 350				
5415					:0152	0227 D0 F8	BNE	**--6		;BR IF NO TIME-OUT
5416	025451	320	370			.BYTE 320,370				
5417					:0153	0229			; *** ERROR 2 ***	
5418					:0154	0229 A0 02	LDY	#2		;SET CODE FOR TBMT TIME-OUT ERRO
5419	025453	240	002			.BYTE 240,002				
5420					:0155	0228 4C 90 03	JMP	A100		;GO TAKE ERROR EXIT
5421	025455	114	220	003		.BYTE 114,220,003				
5422					:0156	022E			;LOAD SECOND SYNCH CHAR INTO TRANSMITTER	
5423					:0157	022E A2 96	LDX	#SYNCH		;LOAD SECOND SYNCH CHAR
5424	025460	242	226			.BYTE 242,226				
5425					:0158	0230 8E 02 A1	STX	TXDB		
5426	025462	216	002	241		.BYTE 216,002,241				
5427					:0159	0233 A2 00	LDX	#0		;INIT TBMT TIME-OUT COUNTER
5428	025465	242	000			.BYTE 242,000				
5429					:0160	0235 2C 00 A4	BIT	USTATR		;SEE IF TBMT SET
5430	025467	054	000	244		.BYTE 054,000,244				
5431					:0161	0238 70 08	BVS	**+10		;BR IF TBMT SET
5432	025472	160	010			.BYTE 160,010				
5433					:0162	023A E8	INX			;INCREM TIME-OUT COUNTER
5434	025474	350				.BYTE 350				
5435					:0163	023B D0 F8	BNE	**--6		;BR IF NO TIME-OUT
5436	025475	320	370			.BYTE 320,370				
5437					:0164	023D			; *** ERROR 3 ***	
5438					:0165	023D A0 03	LDY	#3		;SET CODE FOR TBMT TIME-OUT ERRO
5439	025477	240	003			.BYTE 240,003				
5440					:0166	023F 4C 90 03	JMP	A100		;GO TAKE ERROR EXIT
5441	025501	114	220	003		.BYTE 114,220,003				
5442					:0167	0242			;CHECK FOR CARRIER SET	
5443					:0168	0242 2C 01 A0	BIT	OREGA		;SEE IF CARRIER SET YET
5444	025504	054	001	240		.BYTE 054,001,240				
5445					:0169	0245 70 05	BVS	**+7		;BR IF CARRIER SET
5446	025507	160	005			.BYTE 160,005				
5447					:0170	0247			; *** ERROR 4 ***	
5448					:0171	0247 A0 04	LDY	#4		;SET CODE FOR CARRIER NOT SET ER
5449	025511	240	004			.BYTE 240,004				
5450					:0172	0249 4C 90 03	JMP	A100		;GO TAKE ERROR EXIT
5451	025513	114	220	003		.BYTE 114,220,003				
5452					:0173	024C			;LOAD TRANSMITTER WITH 000 CHAR	
5453					:0174	024C A2 00	LDX	#000		;CLEAR TSOM
5454	025516	242	000			.BYTE 242,000				
5455					:0175	024E 8E 03 A1	STX	TDSR		
5456	025520	216	003	241		.BYTE 216,003,241				
5457					:0176	0251 8E 02 A1	STX	TXDB		;LOAD 000 CHAR
5458	025523	216	002	241		.BYTE 216,002,241				
5459					:0177	0254 A2 00	LDX	#0		;INIT TBMT TIME-OUT COUNTER
5460	025526	242	000			.BYTE 242,000				
5461					:0178	0256 2C 00 A4	BIT	USTATR		;SEE IF TBMT SET
5462	025530	054	000	244		.BYTE 054,000,244				
5463					:0179	0259 70 08	BVS	**+10		;BR IF TBMT SET
5464	025533	160	010			.BYTE 160,010				
5465					:0180	025B E8	INX			;INCREM TIME-OUT COUNTER
5466	025535	350				.BYTE 350				

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

5467      ;0181 025C D0 F8          BNE      *-6          ;BR IF NO TIME-OUT
5468 025536 320 370          .BYTE    320,370
5469      ;0182 025E          ; *** ERROR 5 ***
5470      ;0183 025E A0 05          LDY      #5          ;SET CODE FOR TBMT TIME-OUT ERRO
5471 025540 240 005          .BYTE    240,005
5472      ;0184 0260 4C 90 03          JMP      A100          ;GO TAKE ERROR EXIT
5473 025542 114 220 003      .BYTE    114,220,003
5474      ;0185 0263          ;LOAD TRANSMITTER WITH 125 CHAR
5475      ;0186 0263 A2 55          LDX      #0125          ;LOAD 125 CHAR
5476 025545 242 125          .BYTE    242,125
5477      ;0187 0265 8E 02 A1          STX      TXDB
5478 025547 216 002 241      .BYTE    216,002,241
5479      ;0188 0268 A2 00          LDX      #0          ;INIT TBMT TIME-OUT COUNTER
5480 025552 242 000          .BYTE    242,000
5481      ;0189 026A 2C 00 A4          BIT      USTATR          ;SEE IF TBMT SET
5482 025554 054 000 244      .BYTE    054,000,244
5483      ;0190 026D 70 08          BVS      **10          ;BR IF TBMT SET
5484 025557 160 010          .BYTE    160,010
5485      ;0191 026F E8          INX
5486 025561 350          .BYTE    350          ;INCREMENT TIME-OUT COUNTER
5487      ;0192 0270 D0 F8          BNE      *-6          ;BR IF NO TIME-OUT
5488 025562 320 370          .BYTE    320,370
5489      ;0193 0272          ; *** ERROR 6 ***
5490      ;0194 0272 A0 06          LDY      #6          ;SET CODE FOR TBMT TIME-OUT ERRO
5491 025564 240 006          .BYTE    240,006
5492      ;0195 0274 4C 90 03          JMP      A100          ;GO TAKE ERROR EXIT
5493 025566 114 220 003      .BYTE    114,220,003
5494      ;0196 C 377          ;LOAD TRANSMITTER WITH 252 CHAR
5495      ;0197 0277 A2 AA          LDX      #0252          ;LOAD 252 CHAR
5496 025571 242 252          .BYTE    242,252
5497      ;0198 0279 8E 02 A1          STX      TXDB
5498 025573 216 002 241      .BYTE    216,002,241
5499      ;0199 027C A2 00          LDX      #0          ;INIT TBMT TIME-OUT COUNTER
5500 025576 242 000          .BYTE    242,000
5501      ;0200 027E 2C 00 A4          BIT      USTATR          ;SEE IF TBMT SET
5502 025600 054 000 244      .BYTE    054,000,244
5503      ;0201 0281 70 08          BVS      **10          ;BR IF TBMT SET
5504 025603 160 010          .BYTE    160,010
5505      ;0202 0283 E8          INX          ;INCREMENT TIME-OUT COUNTER
5506 025605 350          .BYTE    350
5507      ;0203 0284 D0 F8          BNE      *-6          ;BR IF NO TIME-OUT
5508 025606 320 370          .BYTE    320,370
5509      ;0204 0286          ; *** ERROR 7 ***
5510      ;0205 0286 A0 07          LDY      #7          ;SET CODE FOR TBMT TIME-OUT ERRO
5511 025610 240 007          .BYTE    240,007
5512      ;0206 0288 4C 90 03          JMP      A100          ;GO TAKE ERROR EXIT
5513 025612 114 220 003      .BYTE    114,220,003
5514      ;0207 0288          ;LOAD TRANSMITTER WITH 377 CHAR AND END OF MESSAGE
5515      ;0208 028B A2 FF          LDX      #0377          ;LOAD 377 CHAR
5516 025615 242 377          .BYTE    242,377
5517      ;0209 028D 8E 02 A1          STX      TXDB
5518 025617 216 002 241      .BYTE    216,002,241
5519      ;0210 0290 A2 00          LDX      #0          ;INIT TBMT TIME-OUT COUNTER
5520 025622 242 000          .BYTE    242,000
5521      ;0211 0292 2C 00 A4          BIT      USTATR          ;SEE IF TBMT SET
5522 025624 054 000 244      .BYTE    054,000,244

```

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

5523					:0212	0295 70 08	BVS	**10		;BR IF TBMT SFT
5524	025627	160	010			.BYTE 160,010				
5525					:0213	0297 E8	INX			;INCREMENT TIME-OUT COUNTER
5526	025631	350				.BYTE 350				
5527					:0214	0298 D0 F8	BNE	**6		;BR IF NO TIME-OUT
5528	025632	320	370			.BYTE 320,370				
5529					:0215	029A			; *** ERROR 8 ***	
5530					:0216	029A A0 08	LDY	#8		;SET CODE FOR TBMT TIME-OUT ERRO
5531	025634	240	010			.BYTE 240,010				
5532					:0217	029C 4C 90 03	JMP	A100		;GO TAKE ERROR EXIT
5533	025636	114	220	003		.BYTE 114,220,003				
5534					:0218	029F			;LOAD TRANSMITTER WITH 000 CHAR	
5535					:0219	029F A2 00	LDX	#000		;LOAD 000 CHAR
5536	025641	242	000			.BYTE 242,000				
5537					:0220	02A1 8E 02 A1	STX	TXDB		
5538	025643	216	002	241		.BYTE 216,002,241				
5539					:0221	02A4 A2 00	LDX	#0		;INIT RDA TIME-OUT COUNTER
5540	025646	242	000			.BYTE 242,000				
5541					:0222	02A6 2C 00 A4	BIT	USTATR		;SEE IF RDA SET
5542	025650	054	000	244		.BYTE 054,000,244				
5543					:0223	02A9 30 08	BMI	**10		;BR IF RDA SET
5544	025653	060	010			.BYTE 060,010				
5545					:0224	02AB E8	INX			;INCREMENT TIME-OUT COUNTER
5546	025655	350				.BYTE 350				
5547					:0225	02AC D0 F8	BNE	**6		;BR IF NO TIME-OUT
5548	025656	320	370			.BYTE 320,370				
5549					:0226	02AE			; *** ERROR 9 ***	
5550					:0227	02AE A0 09	LDY	#9		;SET CODE FOR RDA TIME-OUT ERROR
5551	025660	240	011			.BYTE 240,011				
5552					:0228	02B0 4C 90 03	JMP	A100		;GO TAKE ERROR EXIT
5553	025662	114	220	003		.BYTE 114,220,003				
5554					:0229	02B3			;READ AND CHECK 000 CHAR	
5555					:0230	02B3 AD 00 A1	LDA	RXDB		;READ RECEIVER BUFFER
5556	025665	255	000	241		.BYTE 255,000,241				
5557					:0231	02B6 C9 00	CMP	#000		;CHK FOR 000
5558	025670	311	000			.BYTE 311,000				
5559					:0232	02B8 F0 05	BEQ	**7		;BR IF 000
5560	025672	360	005			.BYTE 360,005				
5561					:0233	02BA			; *** ERROR 10 ***	
5562					:0234	02BA A0 0A	LDY	#10		;SET CODE FOR DATA MISCOMPARE ER
5563	025674	240	012			.BYTE 240,012				
5564					:0235	02BC 4C 90 03	JMP	A100		;GO TAKE ERROR EXIT
5565	025676	114	220	003		.BYTE 114,220,003				
5566					:0236	02BF A2 00	LDX	#0		;INIT RDA TIME-OUT COUNTER
5567	025701	242	000			.BYTE 242,000				
5568					:0237	02C1 2C 00 A4	BIT	USTATR		;SEE IF RDA SET
5569	025703	054	000	244		.BYTE 054,000,244				
5570					:0238	02C4 30 08	BMI	**10		;BR IF RDA SET
5571	025706	060	010			.BYTE 060,010				
5572					:0239	02C6 E8	INX			;INCREMENT TIME-OUT COUNTER
5573	025710	350				.BYTE 350				
5574					:0240	02C7 D0 F8	BNE	**6		;BR IF NO TIME-OUT
5575	025711	320	370			.BYTE 320,370				
5576					:0241	02C9			; *** ERROR 11 ***	
5577					:0242	02C9 A0 0B	LDY	#11		;SET CODE FOR RDA TIME-OUT ERROR
5578	025713	240	013			.BYTE 240,013				

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

5579					:0243	02CB 4C 90 03	JMP	A100		;GO TAKF ERROR EXIT
5580	025715	114	220	003		.BYTE 114,220,003				
5581					:0244	02CE A2 02	LDX	#TEOM		;SET TEOM TO TERMINATE MSG
5582	025720	242	002			.BYTE 242,002				
5583					:0245	02D0 8E 03 A1	STX	TDSR		
5584	025722	216	003	241		.BYTE 216,003,241				
5585					:0246	02D3			;READ AND CHECK 125 CHAR	
5586					:0247	02D3 AD 00 A1	LDA	RXDB		;READ RECEIVER BUFFER
5587	025725	255	000	241		.BYTE 255,000,241				
5588					:0248	02D6 C9 55	CMP	#@125		;CHK FOR 125
5589	025730	311	125			.BYTE 311,125				
5590					:0249	02D8 F0 05	BEQ	**7		;BR IF 125
5591	025732	360	005			.BYTE 360,005				
5592					:0250	02DA			; *** ERROR 12 ***	
5593					:0251	02DA A0 0C	LDY	#12		;SET CODE FOR DATA MISCOMPARE ER
5594	025734	240	014			.BYTE 240,014				
5595					:0252	02DC 4C 90 03	JMP	A100		;GO TAKE ERROR EXIT
5596	025736	114	220	003		.BYTE 114,220,003				
5597					:0253	02DF A2 00	LDX	#0		;INIT RDA TIME-OUT COUNTER
5598	025741	242	000			.BYTE 242,000				
5599					:0254	02E1 2C 00 A4	BIT	USTATR		;SEE IF RDA SET
5600	025743	054	000	244		.BYTE 054,000,244				
5601					:0255	02E4 30 08	BMI	**10		;BR IF RDA SET
5602	025746	060	010			.BYTE 060,010				
5603					:0256	02E6 E8	INX			;INCREMENT TIME-OUT COUNTER
5604	025750	350				.BYTE 350				
5605					:0257	02E7 D0 F8	BNE	**6		;BR IF NO TIME-OUT
5606	025751	320	370			.BYTE 320,370				
5607					:0258	02E9			; *** ERROR 13 ***	
5608					:0259	02E9 A0 0D	LDY	#13		;SET CODE FOR RDA TIME-OUT ERROR
5609	025753	240	015			.BYTE 240,015				
5610					:0260	02EB 4C 90 03	JMP	A100		;GO TAKE ERROR EXIT
5611	025755	114	220	003		.BYTE 114,220,003				
5612					:0261	02EE			;READ AND CHECK 252 CHAR	
5613					:0262	02EE AD 00 A1	LDA	RXDB		;READ RECEIVER BUFFER
5614	025760	255	000	241		.BYTE 255,000,241				
5615					:0263	02F1 C9 AA	CMP	#@252		;CHK FOR 252
5616	025763	311	252			.BYTE 311,252				
5617					:0264	02F3 F0 05	BEQ	**7		;BR IF 252
5618	025765	360	005			.BYTE 360,005				
5619					:0265	02F5			; *** ERROR 14 ***	
5620					:0266	02F5 A0 0E	LDY	#14		;SET CODE FOR DATA MISCOMPARE ER
5621	025767	240	016			.BYTE 240,016				
5622					:0267	02F7 4C 90 03	JMP	A100		;GO TAKE ERROR EXIT
5623	025771	114	220	003		.BYTE 114,220,003				
5624					:0268	02FA A2 00	LDX	#0		;INIT RDA TIME-OUT COUNTER
5625	025774	242	000			.BYTE 242,000				
5626					:0269	02FC 2C 00 A4	BIT	USTATR		;SEE IF RDA SET
5627	025776	054	000	244		.BYTE 054,000,244				
5628					:0270	02FF 30 08	BMI	**10		;BR IF RDA SET
5629	026001	060	010			.BYTE 060,010				
5630					:0271	0301 E8	INX			;INCREMENT TIME-OUT COUNTER
5631	026003	350				.BYTE 350				
5632					:0272	0302 D0 F8	BNE	**6		;BR IF NO TIME-OUT
5633	026004	320	370			.BYTE 320,370				
5634					:0273	0304			; *** ERROR 15 ***	

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

5635					:0274	0304 A0 0F	LDY #15	:SET CODE FOR RDA TIME-OUT ERROR
5636	026006	240	017			.BYTE 240,017		
5637					:0275	0306 4C 90 03	JMP A100	:GO TAKE ERROR EXIT
5638	026010	114	220	003		.BYTE 114,220,003		
5639					:0276	0309 AD 00 A1	;READ AND CHECK 377 CHAR	
5640					:0277	0309 AD 00 A1	LDA RXDB	:READ RECEIVER BUFFER
5641	026013	255	000	241		.BYTE 255,000,241		
5642					:0278	030C C9 FF	CMP #0377	:CHK FOR 377
5643	026016	311	377			.BYTE 311,377		
5644					:0279	030E F0 05	BEQ **7	:BR IF 377
5645	026020	360	005			.BYTE 360,005		
5646					:0280	0310	; *** ERROR 16 ***	
5647					:0281	0310 A0 10	LDY #16	:SET CODE FOR DATA MISCOMPARE ER
5648	026022	240	020			.BYTE 240,020		
5649					:0282	0312 4C 90 03	JMP A100	:GO TAKE ERROR EXIT
5650	026024	114	220	003		.BYTE 114,220,003		
5651					:0283	0315 A2 00	LDX #0	:INIT RDA TIME-OUT COUNTER
5652	026027	242	000			.BYTE 242,000		
5653					:0284	0317 2C 00 A4	BIT USTATR	:SEE IF RDA SET
5654	026031	054	000	244		.BYTE 054,000,244		
5655					:0285	031A 30 08	BMI **10	:BR IF RDA SET
5656	026034	060	010			.BYTE 060,010		
5657					:0286	031C E8	INX	:INCREMENT TIME-OUT COUNTER
5658	026036	350				.BYTE 350		
5659					:0287	031D D0 F8	BNE **6	:BR IF NO TIME-OUT
5660	026037	320	370			.BYTE 320,370		
5661					:0288	031F	; *** ERROR 17 ***	
5662					:0289	031F A0 11	LDY #17	:SET CODE FOR RDA TIME-OUT ERROR
5663	026041	240	021			.BYTE 240,021		
5664					:0290	0321 4C 90 03	JMP A100	:GO TAKE ERROR EXIT
5665	026043	114	220	003		.BYTE 114,220,003		
5666					:0291	0324	;READ AND CHECK 000 CHAR	
5667					:0292	0324 AD 00 A1	LDA RXDB	:READ RECEIVER BUFFER
5668	026046	255	000	241		.BYTE 255,000,241		
5669					:0293	0327 C9 00	CMP #0000	:CHK FOR 000
5670	026051	311	000			.BYTE 311,000		
5671					:0294	0329 F0 05	BEQ **7	:BR IF 000
5672	026053	360	005			.BYTE 360,005		
5673					:0295	032B	; *** ERROR 18 ***	
5674					:0296	032B A0 12	LDY #18	:SET CODE FOR DATA MISCOMPARE ER
5675	026055	240	022			.BYTE 240,022		
5676					:0297	032D 4C 90 03	JMP A100	:GO TAKE ERROR EXIT
5677	026057	114	220	003		.BYTE 114,220,003		
5678					:0298	0330 AE 01 A1	LDX RDSR	:CHECK FOR RERR BIT SET
5679	026062	256	001	241		.BYTE 256,001,241		
5680					:0299	0333 30 05	BMI **7	:BR IF RERR BIT SET (NO CRC ERRO
5681	026065	060	005			.BYTE 060,005		
5682					:0300	0335	; *** ERROR 19 ***	
5683					:0301	0335 A0 13	LDY #19	:SET CODE FOR DATA MISCOMPARE ER
5684	026067	240	023			.BYTE 240,023		
5685					:0302	0337 4C 90 03	JMP A100	:GO TAKE ERROR EXIT
5686	026071	114	220	003		.BYTE 114,220,003		
5687					:0303	033A A2 00	LDX #0	:INIT RDA TIME-OUT COUNTER
5688	026074	242	000			.BYTE 242,000		
5689					:0304	033C 2C 00 A4	BIT USTATR	:SEE IF RDA SET
5690	026076	054	000	244		.BYTE 054,000,244		

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

5691					:0305	033F 30 08	BMI	**+10	:BR IF RDA SET
5692	026101	060	010			.BYTE 060,010			
5693					:0306	0341 E8	INX		:INCREMENT TIME-OUT COUNTER
5694	026103	350				.BYTE 350			
5695					:0307	0342 D0 F8	BNE	**+6	:BR IF NO TIME-OUT
5696	026104	320	370			.BYTE 320,370			
5697					:0308	0344		: *** ERROR 20 ***	
5698					:0309	0344 A0 14	LDY	#20	:SET CODE FOR RDA TIME-OUT ERROR
5699	026106	240	024			.BYTE 240,024			
5700					:0310	0346 4C 90 03	JMP	A100	:GO TAKE ERROR EXIT
5701	026110	114	220	003		.BYTE 114,220,003			
5702					:0311	0349		:READ AND CHECK FIRST CRC CHAR (160)	
5703					:0312	0349 AD 00 A1	LDA	RXDB	:READ RECEIVER BUFFER
5704	026113	255	000	241		.BYTE 255,000,241			
5705					:0313	034C C9 70	CMP	#0160	:CHK FOR 160
5706	026116	311	160			.BYTE 311,160			
5707					:0314	034E F0 05	BEQ	**+7	:BR IF 160
5708	026120	360	005			.BYTE 360,005			
5709					:0315	0350		: *** ERROR 21 ***	
5710					:0316	0350 A0 15	LDY	#21	:SET CODE FOR DATA MISCOMPARE ER
5711	026122	240	025			.BYTE 240,025			
5712					:0317	0352 4C 90 03	JMP	A100	:GO TAKE ERROR EXIT
5713	026124	114	220	003		.BYTE 114,220,003			
5714					:0318	0355 A2 00	LDX	#0	:INIT RDA TIME-OUT COUNTER
5715	026127	242	000			.BYTE 242,000			
5716					:0319	0357 2C 00 A4	BIT	USTATR	:SEE IF RDA SET
5717	026131	054	000	244		.BYTE 054,000,244			
5718					:0320	035A 30 08	BMI	**+10	:BR IF RDA SET
5719	026134	060	010			.BYTE 060,010			
5720					:0321	035C E8	INX		:INCREMENT TIME-OUT COUNTER
5721	026136	350				.BYTE 350			
5722					:0322	035D D0 F8	BNE	**+6	:BR IF NO TIME-OUT
5723	026137	320	370			.BYTE 320,370			
5724					:0323	035F		: *** ERROR 22 ***	
5725					:0324	035F A0 16	LDY	#22	:SET CODE FOR RDA TIME-OUT ERROR
5726	026141	240	026			.BYTE 240,026			
5727					:0325	0361 4C 90 03	JMP	A100	:GO TAKE ERROR EXIT
5728	026143	114	220	003		.BYTE 114,220,003			
5729					:0326	0364		:READ AND CHECK 2ND CRC CHAR (034)	
5730					:0327	0364 AD 00 A1	LDA	RXDB	:READ RECEIVER BUFFER
5731	026146	255	000	241		.BYTE 255,000,241			
5732					:0328	0367 C9 1C	CMP	#0034	:CHK FOR 034
5733	026151	311	034			.BYTE 311,034			
5734					:0329	0369 F0 05	BEQ	**+7	:BR IF 034
5735	026153	360	005			.BYTE 360,005			
5736					:0330	036B		: *** ERROR 23 ***	
5737					:0331	036B A0 17	LDY	#23	:SET CODE FOR DATA MISCOMPARE ER
5738	026155	240	027			.BYTE 240,027			
5739					:0332	036D 4C 90 03	JMP	A100	:GO TAKE ERROR EXIT
5740	026157	114	220	003		.BYTE 114,220,003			
5741					:0333	0370 A2 00	LDX	#0	:INIT RDA TIME-OUT COUNTER
5742	026162	242	000			.BYTE 242,000			
5743					:0334	0372 2C 00 A4	BIT	USTATR	:SEE IF RDA SET
5744	026164	054	000	244		.BYTE 054,000,244			
5745					:0335	0375 30 08	BMI	**+10	:BR IF RDA SET
5746	026167	060	010			.BYTE 060,010			

CVDMEA.P11 10-DEC-80 09:16

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```

5747          ;0336 0377 E8          INX          ;INCREMENT TIME-OUT COUNTER
5748 026171   .BYTE 350
5749          ;0337 0378 D0 F8          BNE      *-6          ;BR IF NO TIME-OUT
5750 026172   .BYTE 320,370
5751          ;0338 037A          ; *** ERROR 24 ***
5752          ;0339 037A A0 18          LDY      #24          ;SET CODE FOR RDA TIME-OUT ERROR
5753 026174   .BYTE 240,030
5754          ;0340 037C 4C 90 03          JMP      A100          ;GO TAKE ERROR EXIT
5755 026176   .BYTE 114,220,003
5756          ;0341 037F          ;DROP RTS, CHECK FOR CARRIER TO DROP
5757          ;0342 037F A2 68          LDX      #TXEN!RXEN!RTSND ;DE-ASSERT RTS
5758 026201   .BYTE 242,150
5759          ;0343 0381 8E 00 A0          STX      OREGB
5760 026203   .BYTE 216,000,240
5761          ;0344 0384 A2 00          LDX      #0          ;INIT CARRIER DROP TIME-OUT COUN
5762 026206   .BYTE 242,000
5763          ;0345 0386 2C 01 A0          BIT      OREGA          ;SEE IF CARRIER CLEARED
5764 026210   .BYTE 054,001,240
5765          ;0346 0389 50 05          BVC      **7          ;BR IF CARRIER CLEARED
5766 026213   .BYTE 120,005
5767          ;0347 038B E8          INX          ;INCREMENT TIME-OUT COUNTER
5768 026215   .BYTE 350
5769          ;0348 038C D0 F8          BNE      *-6          ;BR IF NO TIME-OUT
5770 026216   .BYTE 320,370
5771          ;0349 038E          ; *** ERROR 25 ***
5772          ;0350 038E A0 19          LDY      #25          ;SET CODE FOR CARRIER DROP TIME-
5773 026220   .BYTE 240,031
5774          ;0351 0390          ;COME HERE FOR EXIT
5775          ;0352 0390 84 16          A100 STY      BSEL6          ;PUT ERROR NO. (IF ANY) INTO BSE
5776 026222   .BYTE 204,026
5777          ;0353 0392 85 17          STA      BSEL7          ;PUT BAD DATA (IF ANY) INTO BSEL
5778 026224   .BYTE 205,027
5779          ;0354 0394 60          RTS          ;RETURN CONTROL TO LSI-11 PROGRA
5780 026226   .BYTE 140
5781          ;0355 0395
5782          ;0356 0395
5783          ;
5784          ;
5785          ;ERRORS = 0000
5786          ;
5787 026227   ENDCOD:
5788          .EVEN

```

CVDMEA.P11 10-DEC-80 09:16

TEST 3 -- DATA TEST -- BCP XLB CRC-16

.SBTTL TEST 3 -- DATA TEST -- BCP XLB CRC-16

5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813 026230
5814 026230 012737 000003 002526
5815 026236 004737 005376
5816 026242 042737 001000 026320
5817 026250 012737 000011 027072
5818 026256 023727 002470 000000
5819 026264 001412
5820 026266 023727 002472 000004
5821 026274 001406
5822 026276 052737 001000 026320
5823 026304 052737 100000 027072
5824
5825 026312 004537 007234
5826 026316 065626
5827 026320 000000
5828 026322 103003
5829 026324
5830 026324 104460
5831 026326
5832 026326 104410
5833 026330 000554
5834
5835 026332 004537 010136
5836 026336 000226
5837 026340 000007
5838 026342 103003
5839 026344
5840 026344 104460
5841 026346
5842 026346 104410
5843 026350 000534
5844

```

*****
*
* TEST 3 -- DATA TEST -- BCP XLB CRC-16
*
* IF XLB IS SPECIFIED IN THE P-TABLE, THIS TEST WILL TRANSMIT &
* RECEIVE IN BCP MODE WITH CRC-16 ERROR DETECTION THE FOLLOWING
* MESSAGE:
*
* 125 252 000 377 001 002 004 010 020 040 100 200 376 375 373 367
* 357 337 277 177
*
* THIS MESSAGE WILL BE PRECEDED BY 3 SYNC CHARACTERS AND REPEATED
* THREE TIMES WITH CRC'S FOLLOWING EACH ONE. THE LAST TRANSMISSION OF
* THE CRC WILL BE FOLLOWED BY SEVERAL SYNC CHARACTERS BEFORE DROPPING
* TXE & RXE. 8-BIT CHARACTER LENGTHS ARE ALSO UTILIZED.
*
* IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
* USING INTERNAL LOOPBACK (TTLOOP=1).
*
*****

```

```

: BGNTST
:
: T3::
: MOV #3,REG0 ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
: JSR PC,INIDMV ;INIT DMV-11, ENTER MAINT LOOP
: BIC #NOLOOP,1$
: MOV #9,3$ ;*INIT ENTRAN COUNT/TTLOOP STATUS
: CMP BRDTYP,#0 ;IS THIS AN M8064?
: BEQ 2$ ; YES: USE TTLOOP (NOT XLB).
: CMP TSTCON,#4 ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
: BEQ 2$ ;BR IF NO
: BIS #NOLOOP,1$ ; YES: SPECIFY NO TTLOOP (INITRN)
: BIS #BIT15,3$ ; AND SET MSB OF ENTRAN STATUS (NOLOOP)
-----
2$: JSR R5,INITRN ;LOAD 1 SYNCH, CLK TX UNTIL ACTIVE
DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP, IDLE, CRC-16, SYNCH=226
1$: 0 ;USE 8 BIT CHARS
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP C$ERROR
TRAP C$ESCAPE
.WORD L10022-.
JSR R5,TXCHAR ;LOAD 2ND SYNCH, TX 1ST SYNCH
SYNCH
7.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP C$ERROR
TRAP C$ESCAPE
.WORD L10022-.

```

CVDMEA.P11 10-DEC-80 09:16

TEST 3 -- DATA TEST -- BCP XLB CRC-16

5845	026352	004537	010136	JSR	R5,TXCHAR	;LOAD 3RD SYNCH, TX 2ND SYNCH		
5846	026356	000226		SYNCH				
5847	026360	000010		8.				
5848	026362	103003		BCC	+.8.	;BR IF NO ERROR		
5849	026364			ERROR		;REPORT STACKED ERROR		
5850	026364	104460					TRAP	C\$ERROR
5851	026366			ESCAPE	TST	;SKIP TO END OF TEST		
5852	026366	104410					TRAP	C\$ESCAPE
5853	026370	000514					.WORD	L10022-.
5854								
5855	026372	004537	010250	JSR	R5,TXCTRL	;CLEAR TSOM		
5856	026376	000000		000				
5857	026400	000000		0				
5858								
5859	026402	004537	010136	JSR	R5,TXCHAR	;LOAD 125(DATA1), TX 3RD SYNCH		
5860	026406	000125		125				
5861	026410	000010		8.				
5862	026412	103003		BCC	+.8.	;BR IF NO ERROR		
5863	026414			ERROR		;REPORT STACKED ERROR		
5864	026414	104460					TRAP	C\$ERROR
5865	026416			ESCAPE	TST	;SKIP TO END OF TEST		
5866	026416	104410					TRAP	C\$ESCAPE
5867	026420	000464					.WORD	L10022-.
5868								
5869	026422	004537	010136	JSR	R5,TXCHAR	;LOAD 252(DATA2), TX 125(DATA1)		
5870	026426	000252		252				
5871	026430	000010		8.				
5872	026432	103003		BCC	+.8.	;BR IF NO ERROR		
5873	026434			ERROR		;REPORT STACKED ERROR		
5874	026434	104460					TRAP	C\$ERROR
5875	026436			ESCAPE	TST	;SKIP TO END OF TEST		
5876	026436	104410					TRAP	C\$ESCAPE
5877	026440	000444					.WORD	L10022-.
5878								
5879	026442	004537	010136	JSR	R5,TXCHAR	;LOAD 000(DATA3), TX 252(DATA2)		
5880	026446	000000		000				
5881	026450	000010		8.				
5882	026452	103003		BCC	+.8.	;BR IF NO ERROR		
5883	026454			ERROR		;REPORT STACKED ERROR		
5884	026454	104460					TRAP	C\$ERROR
5885	026456			ESCAPE	TST	;SKIP TO END OF TEST		
5886	026456	104410					TRAP	C\$ESCAPE
5887	026460	000424					.WORD	L10022-.
5888								
5889	026462	004537	010136	JSR	R5,TXCHAR	;LOAD 377(DATA4), TX 000(DATA3)		
5890	026466	000377		377				
5891	026470	000010		8.				
5892	026472	103003		BCC	+.8.	;BR IF NO ERROR		
5893	026474			ERROR		;REPORT STACKED ERROR		
5894	026474	104460					TRAP	C\$ERROR
5895	026476			ESCAPE	TST	;SKIP TO END OF TEST		
5896	026476	104410					TRAP	C\$ESCAPE
5897	026500	000404					.WORD	L10022-.
5898								
5899	026502	004537	010136	JSR	R5,TXCHAR	;LOAD 001(DATA5)		
5900	026506	000001		001				

CVDMEA.P11 10-DEC-80 09:16

TEST 3 -- DATA TEST -- BCP XLB CRC-16

```

5901 026510 000000      0
5902 026512 103003      BCC      .+8.      ;BR IF NO ERROR
5903 026514      ERROR      ;REPORT STACKED ERROR
5904 026514 104460      ESCAPE TST      ;SKIP TO END OF TEST      TRAP      C$ERROR
5905 026516      ;
5906 026516 104410      ;
5907 026520 000364      ;
5908 ;
5909 026522 004537 011624      JSR      R5,RCV1ST      ;CLOCK AND RCV 125
5910 026526 000000      0
5911 026530 103003      BCC      .+8.      ;BR IF NO ERROR
5912 026532      ERROR      ;REPORT STACKED ERROR
5913 026532 104460      ESCAPE TST      ;SKIP TO END OF TEST      TRAP      C$ERROR
5914 026534      ;
5915 026534 104410      ;
5916 026536 000346      ;
5917 ;
5918 026540 004537 010350      JSR      R5,RXCHAR      ;READ & CHK 125(DATA1), RCV 252(DATA2)
5919 026544 000125      125
5920 026546 000000      0
5921 026550 000010      8.
5922 026552 103003      BCC      .+8.      ;BR IF NO ERROR
5923 026554      ERROR      ;REPORT STACKED ERROR
5924 026554 104460      ESCAPE TST      ;SKIP TO END OF TEST      TRAP      C$ERROR
5925 026556      ;
5926 026556 104410      ;
5927 026560 000324      ;
5928 ;
5929 ;-----:
5930 ; TRANSMIT THE BULK OF DATA OUT OF TABLE 'PATX'
5931 ;-----:
5931 026562 012702 002646      MOV      #PATX+1,R2      ;SET UP TABLE POINTER
5932 026566 112237 026624      5$:      MOVB     (R2)+,20$      ;SET UP EXPECTED CHARACTER
5933 026572 116237 000003 026604      MOVB     3(R2),10$      ;SET UP TRANSMIT CHARACTER
5934 ;
5935 026600 004537 010136      JSR      R5,TXCHAR      ;LOAD A CHARACTER
5936 026604 000000      10$:      000      ;** HOLE FOR NEXT TX CHARACTER
5937 026606 000000      0
5938 026610 103003      BCC      .+8.      ;BR IF NO ERROR
5939 026612      ERROR      ;REPORT STACKED ERROR
5940 026612 104460      ESCAPE TST      ;SKIP TO END OF TEST      TRAP      C$ERROR
5941 026614      ;
5942 026614 104410      ;
5943 026616 000266      ;
5944 ;
5945 026620 004537 010350      JSR      R5,RXCHAR      ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
5946 026624 000000      20$:      000      ;** HOLE FOR EXPECTED CHARACTER
5947 026626 000000      0
5948 026630 000010      8.
5949 026632 103003      BCC      .+8.      ;BR IF NO ERROR
5950 026634      ERROR      ;REPORT STACKED ERROR
5951 026634 104460      ESCAPE TST      ;SKIP TO END OF TEST      TRAP      C$ERROR
5952 026636      ;
5953 026636 104410      ;
5954 026640 000244      ;
5955 ;
5956 026642 022702 002665      CMP      #PATX+16.,R2      ;CHECK FOR 20TH CHARACTER OF TABLE

```

CVDMEA.P11 10-DEC-80 09:16

TEST 3 -- DATA TEST -- BCP XLB CRC-16

Line	Address	Offset	Control	Op	Opnd	Comment	Trap	Trap Word
5957	026646	001347						
5958								
5959	026650	004537	010250	JSR	R5, TXCTRL	;LOAD 1ST TEOM		
5960	026654	000002		TEOM				
5961	026656	000000		0				
5962	026660	004537	010350	JSR	R5, RXCHAR	;READ/CHK 357(DATA17), RCV 337(DATA18)		
5963	026664	000357		357				
5964	026666	000000		0				
5965	026670	000010		8.				
5966	026672	103003		BCC	.+8.	;BR IF NO ERROR		
5967	026674			ERROR		;REPORT STACKED ERROR		
5968	026674	104460					TRAP	C\$ERROR
5969	026676			ESCAPE	TST	;SKIP TO END OF TEST		
5970	026676	104410					TRAP	C\$ESCAPE
5971	026700	000204					.WORD	L10022-
5972								
5973	026702	004537	010250	JSR	R5, TXCTRL	;LOAD 2ND TEOM		
5974	026706	000002		TEOM				
5975	026710	000000		0				
5976	026712	004537	010350	JSR	R5, RXCHAR	;READ/CHK 337(DATA18), RCV 277(DATA19)		
5977	026716	000337		337				
5978	026720	000000		0				
5979	026722	000010		8.				
5980	026724	103003		BCC	.+8.	;BR IF NO ERROR		
5981	026726			ERROR		;REPORT STACKED ERROR		
5982	026726	104460					TRAP	C\$ERROR
5983	026730			ESCAPE	TST	;SKIP TO END OF TEST		
5984	026730	104410					TRAP	C\$ESCAPE
5985	026732	000152					.WORD	L10022-
5986								
5987	026734	004537	010350	JSR	R5, RXCHAR	;READ/CHK 277(DATA19), RCV 177(DATA20)		
5988	026740	000277		277				
5989	026742	000000		0				
5990	026744	000010		8.				
5991	026746	103003		BCC	.+8.	;BR IF NO ERROR		
5992	026750			ERROR		;REPORT STACKED ERROR		
5993	026750	104460					TRAP	C\$ERROR
5994	026752			ESCAPE	TST	;SKIP TO END OF TEST		
5995	026752	104410					TRAP	C\$ESCAPE
5996	026754	000130					.WORD	L10022-
5997								
5998	026756	004537	010350	JSR	R5, RXCHAR	;READ/CHK 177(DATA20), RCV FIRST CRC BYTE		
5999	026762	100177		RXERR!177				
6000	026764	000001		RERCHK				
6001	026766	000010		8.				
6002	026770	103003		BCC	.+8.	;BR IF NO ERROR		
6003	026772			ERROR		;REPORT STACKED ERROR		
6004	026772	104460					TRAP	C\$ERROR
6005	026774			ESCAPE	TST	;SKIP TO END OF TEST		
6006	026774	104410					TRAP	C\$ESCAPE
6007	026776	000106					.WORD	L10022-
6008								
6009	027000	004537	010350	JSR	R5, RXCHAR	;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE		
6010	027004	000156		156				
6011	027006	000000		0				
6012	027010	000010		8.				

CVDMEA.P11 10-DEC-80 09:16

TEST 3 -- DATA TEST -- BCP XLB CRC-16

```

6013 027012 103003      BCC      .+8.      ;BR IF NO ERROR
6014 027014             ERROR      ;REPORT STACKED ERROR
6015 027014 104460             ESCAPE TST      ;SKIP TO END OF TEST
6016 027016             TRAP      C$ERROR
6017 027016 104410             TRAP      C$ESCAPE
6018 027020 000064             .WORD      L10022-.
6019
6020 027022 004537 010350     JSR      R5,RXCHAR ;READ & CHK 2ND CRC BYTE, RCV 1ST SYNCH
6021 027026 000236
6022 027030 000000
6023 027032 000010
6024 027034 103003      BCC      .+8.      ;BR IF NO ERROR
6025 027036             ERROR      ;REPORT STACKED ERROR
6026 027036 104460             TRAP      C$ERROR
6027 027040             ESCAPE TST      ;SKIP TO END OF TEST
6028 027040 104410             TRAP      C$ESCAPE
6029 027042 000042             .WORD      L10022-.
6030
-----
6031 027044 005337 002526     DEC      REG0      ;DECREMENT COUNT
6032 027050 001406             BEQ      40$      ;BR IF TRIPLE LOOP IS COMPLETED
6033
6034 027052 004537 010250     JSR      R5,TXCTRL ;CLEAR TEOM, SET TSOM
6035 027056 000001             TSOM
6036 027060 000001             1
6037 027062 000137 026312     JMP      2$      ;AND RUN TX/RX AGAIN
6038
6039 027066 004537 011772     40$: JSR      R5,ENTRAN ;SHUT DOWN TRANSMITTER, RECEIVER
6040 027072 000011             3$: 9.
6041 027074 103003      BCC      .+8.      ;BR IF NO ERROR
6042 027076             ERROR      ;REPORT STACKED ERROR
6043 027076 104460             TRAP      C$ERROR
6044 027100             ESCAPE TST      ;SKIP TO END OF TEST
6045 027100 104410             TRAP      C$ESCAPE
6046 027102 000002             .WORD      L10022-.
6047 027104             ENDTST
6048 027104
6049 027104 104401             L10022: TRAP      C$ETST

```

CVDMEA.P11 10-DEC-80 09:16

TEST 4 -- DATA TEST -- BCP XLB ODD VRC

.SBTTL TEST 4 -- DATA TEST -- BCP XLB ODD VRC

6050
6051
6052
6053
6054
6055
6056
6057
6058
6059
6060
6061
6062
6063
6064
6065
6066
6067
6068
6069
6070
6071
6072
6073
6074
6075
6076
6077
6078
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105

027106
027106 004737 005376
027112 042737 001000 027170
027120 012737 000011 027634
027126 023727 002470 000000
027134 001412
027136 023727 002472 000004
027144 001406
027146 052737 001000 027170
027154 052737 100000 027634

027162 004537 007234
027166 062226
027170 000347
027172 103003
027174 104460
027176
027176 104410
027200 000446

027202 004537 010136
027206 000226
027210 000007
027212 103003
027214 104460
027216
027216 104410
027220 000426

```
*****
*
* TEST 4 -- DATA TEST -- BCP XLB ODD VRC
*
* IF XLB IS SPECIFIED IN THE P-TABLE, THIS TEST WILL TRANSMIT &
* RECEIVE IN BCP MODE WITH ODD VRC ERROR DETECTION THE FOLLOWING
* MESSAGE:
*
* 125 252 000 377 001 002 004 010 020 040 100 200 376 375 373 367
* 357 337 277 177
*
* THIS MESSAGE WILL BE PRECEDED BY 3 SYNC CHARACTERS AND REPEATED
* THREE TIMES. AFTER THE LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
* SENT BEFORE DROPIING TXE & RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
* UTILIZED.
*
* IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
* USING INTERNAL LOOPBACK (TTLOOP=1).
*****
:
: BGNTST
:
: JSR PC,INIDMV ;INIT DMV-11, ENTER MAINT LOOP T4::
: BIC #NLOOP,1$
: MOV #9,3$
: CMP BRDTYP,#0 ;INIT ENTRAN COUNT/TTLOOP STATUS
: BEQ 2$ ;IS THIS AN M8064?
: CMP TSTCON,#4 ;YES: USE TTLOOP (NOT XLB).
: BEQ 2$ ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
: BIS #NLOOP,1$ ;BR IF NO
: BIS #BIT15,3$ ;YES: SPECIFY NO TTLOOP (INITRN)
: ; AND SPECIFY NLOOP IN ENTRAN
:-----
2$: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!STRIPS!OVRC!SYNCH ;SET DDCMP,STRIP SYNC,ODD VRC,SYNCH=226
1$: TXDL!RXDL ;USE 7 BIT CHARS FOR RX & TX
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP CSERROR
;WORD C$ESCAPE
;WORD L10023-.
:
: JSR R5,TXCHAR ;LOAD 2ND SYNCH, TX 1ST SYNCH
: SYNCH
: 7.
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
: ESCAPE TST ;SKIP TO END OF TEST TRAP CSERROR
;WORD C$ESCAPE
;WORD L10023-.
```

CVDMEA.P11 10-DEC-80 09:16

TEST 4 -- DATA TEST -- BCP XLB ODD VRC

6106	027222	004537	010136	JSR	R5,TXCHAR	;LOAD 3RD SYNCH, TX 2ND SYNCH		
6107	027226	000226		SYNCH				
6108	027230	000010		8.				
6109	027232	103003		BCC	.+8.	;BR IF NO ERROR		
6110	027234			ERROR		;REPORT STACKED ERROR		
6111	027234	104460					TRAP	C\$ERROR
6112	027236			ESCAPE	TST	;SKIP TO END OF TEST		
6113	027236	104410					TRAP	C\$ESCAPE
6114	027240	000406					.WORD	L10023-
6115								
6116	027242	004537	010250	JSR	R5,TXCTRL	;CLEAR TSOM		
6117	027246	000000		000				
6118	027250	000000		0				
6119								
6120	027252	004537	010136	JSR	R5,TXCHAR	;LOAD 125(DATA1), TX 3RD SYNCH		
6121	027256	000125		125				
6122	027260	000010		8.				
6123	027262	103003		BCC	.+8.	;BR IF NO ERROR		
6124	027264			ERROR		;REPORT STACKED ERROR		
6125	027264	104460					TRAP	C\$ERROR
6126	027266			ESCAPE	TST	;SKIP TO END OF TEST		
6127	027266	104410					TRAP	C\$ESCAPE
6128	027270	000356					.WORD	L10023-
6129								
6130	027272	004537	010136	JSR	R5,TXCHAR	;LOAD 252(DATA2), TX 125(DATA1)		
6131	027276	000252		252				
6132	027300	000010		8.				
6133	027302	103003		BCC	.+8.	;BR IF NO ERROR		
6134	027304			ERROR		;REPORT STACKED ERROR		
6135	027304	104460					TRAP	C\$ERROR
6136	027306			ESCAPE	TST	;SKIP TO END OF TEST		
6137	027306	104410					TRAP	C\$ESCAPE
6138	027310	000336					.WORD	L10023-
6139								
6140	027312	004537	010136	JSR	R5,TXCHAR	;LOAD 000(DATA3)		
6141	027316	000000		000				
6142	027320	000000		0				
6143	027322	103003		BCC	.+8.	;BR IF NO ERROR		
6144	027324			ERROR		;REPORT STACKED ERROR		
6145	027324	104460					TRAP	C\$ERROR
6146	027326			ESCAPE	TST	;SKIP TO END OF TEST		
6147	027326	104410					TRAP	C\$ESCAPE
6148	027330	000316					.WORD	L10023-
6149								
6150	027332	004537	011624	JSR	R5,RCV1ST	;CLOCK AND RCV 125(DATA1)		
6151	027336	000000		0				
6152	027340	103003		BCC	.+8.	;BR IF NO ERROR		
6153	027342			ERROR		;REPORT STACKED ERROR		
6154	027342	104460					TRAP	C\$ERROR
6155	027344			ESCAPE	TST	;SKIP TO END OF TEST		
6156	027344	104410					TRAP	C\$ESCAPE
6157	027346	000300					.WORD	L10023-
6158								
6159	027350	004537	010350	JSR	R5,RXCHAR	;READ & CHK 125(DATA1), RCV 252(DATA2)		
6160	027354	000125		125				
6161	027356	000001		RERCHK		; & CHECK RERR BIT=0 (GOOD VRC)		

CVDMEA.P11 10-DEC-80 09:16

TEST 4 -- DATA TEST -- BCP XLB ODD VRC

6162	027360	000010			8.				
6163	027362	103003			BCC	+.8.		:BR IF NO ERROR	
6164	027364				ERROR			:REPORT STACKED ERROR	
6165	027364	104460							TRAP C\$ERROR
6166	027366				ESCAPE	TST		:SKIP TO END OF TEST	
6167	027366	104410							TRAP C\$ESCAPE
6168	027370	000256							.WORD L10023-
6169									
6170					-----				
6171					: TRANSMIT THE BULK OF DATA OUT OF TABLE 'PATX'				
6172	027372	012702	002646		MOV	#PATX+1,R2		:SET UP TABLE POINTER	
6173	027376	112237	027434	5\$:	MOVB	(R2)+,20\$:SET UP EXPECTED CHARACTER	
6174	027402	116237	000001 027414		MOVB	1(R2),10\$:SET UP TRANSMIT CHARACTER	
6175									
6176	027410	004537	010136		JSR	R5,TXCHAR		:LOAD A CHARACTER	
6177	027414	000000		10\$:	000			:** HOLE FOR NEXT TX CHARACTER	
6178	027416	000000			0				
6179	027420	103003			BCC	+.8.		:BR IF NO ERROR	
6180	027422				ERROR			:REPORT STACKED ERROR	
6181	027422	104460							TRAP C\$ERROR
6182	027424				ESCAPE	TST		:SKIP TO END OF TEST	
6183	027424	104410							TRAP C\$ESCAPE
6184	027426	000220							.WORD L10023-
6185									
6186	027430	004537	010350		JSR	R5,RXCHAR		:CLK/RECEIVE/CHECK PREVIOUS CHARACTER	
6187	027434	000000		20\$:	000			:** HOLE FOR EXPECTED CHARACTER	
6188	027436	000001			RERCHK			: & CHECK RERR BIT=0 (GOOD VRC)	
6189	027440	000010			8.				
6190	027442	103003			BCC	+.8.		:BR IF NO ERROR	
6191	027444				ERROR			:REPORT STACKED ERROR	
6192	027444	104460							TRAP C\$ERROR
6193	027446				ESCAPE	TST		:SKIP TO END OF TEST	
6194	027446	104410							TRAP C\$ESCAPE
6195	027450	000176							.WORD L10023-
6196									
6197	027452	022702	002737		CMP	#EPATX-1,R2			
6198	027456	001347			BNE	5\$			
6199					-----				
6200	027460	004537	010250		JSR	R5,TXCTRL		:LOAD TSOM	
6201	027464	000001			TSOM				
6202	027466	000000			0				
6203	027470	103003			BCC	+.8.		:BR IF NO ERROR	
6204	027472				ERROR			:REPORT STACKED ERROR	
6205	027472	104460							TRAP C\$ERROR
6206	027474				ESCAPE	TST		:SKIP TO END OF TEST	
6207	027474	104410							TRAP C\$ESCAPE
6208	027476	000150							.WORD L10023-
6209									
6210	027500	004537	010350		JSR	R5,RXCHAR		:READ & CHK 277, RCV 177	
6211	027504	000277			277				
6212	027506	000001			RERCHK			: & CHECK RERR BIT=0 (GOOD VRC)	
6213	027510	000010			8.				
6214	027512	103003			BCC	+.8.		:BR IF NO ERROR	
6215	027514				ERROR			:REPORT STACKED ERROR	
6216	027514	104460							TRAP C\$ERROR
6217	027516				ESCAPE	TST		:SKIP TO END OF TEST	

CVDMEA.P11 10-DEC-80 09:16

TEST 4 -- DATA TEST -- BCP XLB ODD VRC

6218	027516	104410					TRAP	C\$ESCAPE
6219	027520	000126					.WORD	L10023-
6220								
6221	027522	004537	010250	JSR	R5,TXCTRL	;LOAD 2ND TSOM		
6222	027526	000001		TSOM				
6223	027530	000000		0				
6224	027532	103003		BCC	.+8.	;BR IF NO ERROR		
6225	027534			ERROR		;REPORT STACKED ERROR		
6226	027534	104460					TRAP	C\$ERROR
6227	027536			ESCAPE	TST	;SKIP TO END OF TEST		
6228	027536	104410					TRAP	C\$ESCAPE
6229	027540	000106					.WORD	L10023-
6230								
6231	027542	004537	010350	JSR	R5,RXCHAR	;READ & CHK 177, RCV FIRST SYNC		
6232	027546	000177		177				
6233	027550	000001		RERCHK		; & CHECK RERR BIT=0 (GOOD VRC)		
6234	027552	000010		8.				
6235	027554	103003		BCC	.+8.	;BR IF NO ERROR		
6236	027556			ERROR		;REPORT STACKED ERROR		
6237	027556	104460					TRAP	C\$ERROR
6238	027560			ESCAPE	TST	;SKIP TO END OF TEST		
6239	027560	104410					TRAP	C\$ESCAPE
6240	027562	000064					.WORD	L10023-
6241								
6242	027564	004537	010350	JSR	R5,RXCHAR	;READ & CHK 1ST SYNC, RCV SECOND SYNC		
6243	027570	000226		SYNCH				
6244	027572	000001		RERCHK		; & CHECK RERR BIT=0 (GOOD VRC)		
6245	027574	000010		8.				
6246	027576	103003		BCC	.+8.	;BR IF NO ERROR		
6247	027600			ERROR		;REPORT STACKED ERROR		
6248	027600	104460					TRAP	C\$ERROR
6249	027602			ESCAPE	TST	;SKIP TO END OF TEST		
6250	027602	104410					TRAP	C\$ESCAPE
6251	027604	000042					.WORD	L10023-
6252								
6253	027606	004537	010350	JSR	R5,RXCHAR	;READ & CHK 2ND SYNC, RCV NEXT ONE		
6254	027612	000226		SYNCH				
6255	027614	000001		RERCHK		; & CHECK RERR BIT=0 (GOOD VRC)		
6256	027616	000010		8.				
6257	027620	103003		BCC	.+8.	;BR IF NO ERROR		
6258	027622			ERROR		;REPORT STACKED ERROR		
6259	027622	104460					TRAP	C\$ERROR
6260	027624			ESCAPE	TST	;SKIP TO END OF TEST		
6261	027624	104410					TRAP	C\$ESCAPE
6262	027626	000020					.WORD	L10023-
6263								
6264	027630	004537	011772	JSR	R5,ENTRAN	;SHUT DOWN TRANSMITTER, RECEIVER		
6265	027634	000011		9.				
6266	027636	103003		BCC	.+8.	;BR IF NO ERROR		
6267	027640			ERROR		;REPORT STACKED ERROR		
6268	027640	104460					TRAP	C\$ERROR
6269	027642			ESCAPE	TST	;SKIP TO END OF TEST		
6270	027642	104410					TRAP	C\$ESCAPE
6271	027644	000002					.WORD	L10023-
6272	027646							
6273	027646							

ENDTST

L10023:

C 12

SEQ 145

CVDMEA.P11 10-DEC-80 09:16

TEST 4 -- DATA TEST -- BCP XLB ODD VRC

6274 027646 104401

TRAP CSETST

CVDMEA.P11 10-DEC-80 09:16

TEST 5 -- DATA TEST -- BCP XLB EVEN VRC

.SBTTL TEST 5 -- DATA TEST -- BCP XLB EVEN VRC

6275
6276
6277
6278
6279
6280
6281
6282
6283
6284
6285
6286
6287
6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299 027650
6300 027650 004737 005376
6301 027654 042737 001000 027732
6302 027662 012737 000011 030376
6303 027670 023727 002470 000000
6304 027676 001412
6305 027700 023727 002472 000004
6306 027706 001406
6307 027710 052737 001000 027732
6308 027716 053737 100000 030376
6309
6310 027724 004537 007234
6311 027730 062626
6312 027732 000347
6313 027734 103003
6314 027736
6315 027736 104460
6316 027740
6317 027740 104410
6318 027742 000446
6319
6320 027744 004537 010136
6321 027750 000226
6322 027752 000007
6323 027754 103003
6324 027756
6325 027756 104460
6326 027760
6327 027760 104410
6328 027762 000426
6329
6330 027764 004537 010136

```

*****
*
* TEST 5 -- DATA TEST -- BCP XLB EVEN VRC
*
* IF XLB IS SPECIFIED IN THE P-TABLE, THIS TEST WILL TRANSMIT &
* RECEIVE IN BCP MODE WITH EVEN VRC ERROR DETECTION THE FOLLOWING
* MESSAGE:
*
* 125 252 000 377 001 002 004 010 020 040 100 200 376 375 373 367
* 357 337 277 177
*
* THIS MESSAGE WILL BE PRECEDED BY 3 SYNC CHARACTERS AND REPEATED
* THREE TIMES. AFTER THE LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
* SENT BEFORE DROPPING TXE & RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
* UTILIZED.
*
* IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
* USING INTERNAL LOOPBACK (TTLOOP=1).
*****

```

```

*****
*
* BGNTST
*
* T5::
* JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP
* BIC #NOLOOP,1$
* MOV #9,3$ ;INIT ENTRAN COUNT/STATUS
* CMP BRDTYP,#0 ;IS BOARD TYPE M8064?
* BEQ 2$ ; YES: SPECIFY TTLOOP (NOT XLB)
* CMP TSTCON,#4 ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
* BEQ 2$ ;BR IF NO
* BIS #NOLOOP,1$ ; YES: SPECIFY NO TTLOOP (INITRN)
* BIS BIT15,3$ ; AND SPECIFY NOLOOP IN ENTRAN
*-----*
2$: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!STRIPS!EVRC!SYNCH ;SET DDCMP,STRIP SYNCH,EVEN VRC,SYNCH=226
1$: TXDL!RXDL ;USE 7 BIT CHARS FOR TX & RX
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP C$ERROR
.WORD L10024-.
*-----*
JSR R5,TXCHAR ;LOAD 2ND SYNCH, TX 1ST SYNCH
SYNCH
7.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP C$ERROR
.WORD L10024-.
*-----*
JSR R5,TXCHAR ;LOAD 3RD SYNCH, TX 2ND SYNCH

```


CVDMEA.P11 10-DEC-80 09:16

TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1

.SBTTL TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1

6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519 030412
6520 030412 012737 000003 002526
6521 030420 004737 005376
6522 030424 042737 001000 030466
6523 030432 023727 002470 000000
6524 030440 001407
6525 030442 023727 002472 000004
6526 030450 001403
6527 030452 052737 001000 030466
6528
6529 030460 004537 007234
6530 030464 000000
6531 030466 000000
6532 030470 103003
6533 030472
6534 030472 104460
6535 030474
6536 030474 104410
6537 030476 000374
6538
6539 030500 004537 010250
6540 030504 000001
6541 030506 000007
6542 030510 004537 010250
6543 030514 000001
6544 030516 000010
6545 030520 004537 010250
6546 030524 000001
6547 030526 000010
6548 030530 004537 010250
6549 030534 000000
6550 030536 000000
6551 030540 004537 010136
6552 030544 000125
6553 030546 000010
6554 030550 103003

```

*****
*
* TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1
*
* IF XLB IS SPECIFIED IN THE P-TABLE, THIS TEST WILL TRANSMIT &
* RECEIVE IN BOP MODE WITH CRC-CCITT-1 ERROR DETECTION THE FOLLOWING
* SHORT MESSAGE: 125 252 000 377 001
*
* THIS MESSAGE WILL BE PRECEDED BY FLAG CHARACTERS AND REPEATED
* THREE TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
* LENGTHS ARE ALSO UTILIZED.
*
* IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
* USING INTERNAL LOOPBACK (TTLOOP=1).
*
*****

```

```

: BGNTST
:
: T6::
:
: MOV #3,REGO ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
: JSR PC,INIDMV ;INIT DMV-11, ENTER MAINT LOOP
: BIC #NOLOOP,1$
: CMP BRDYP,#0 ;IS BOARD TYPE = M8064 ?
: BEQ 2$ ; YES: SPECIFY TTLOOP (NO XLB)
: CMP TSTCON,#4 ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
: BEQ 2$ ;BR IF NO
: BIS #NOLOOP,1$ ; YES: SPECIFY NO TTLOOP (INITRN)
-----
2$: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
0 ;SET BOP MODE,CRC-CCITT=>1'S CHECK
1$: 0 ;USE 8 BIT CHARS
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP C$ERROR
;WORD C$ESCAPE
L10025-
: JSR R5,TXCTRL ;LOAD 2ND FLAG, TX 1ST FLAG
: TSOM
: 7.
: JSR R5,TXCTRL ;LOAD 3RD FLAG, TX 2ND FLAG
: TSOM
: 8.
: JSR R5,TXCTRL ;LOAD 4TH FLAG, TX 3RD FLAG
: TSOM
: 8.
: JSR R5,TXCTRL ;CLEAR TSOM
: 000
: 0
: JSR R5,TXCHAR ;LOAD DATA1(125), TX 4TH FLAG
: 125
: 8.
: BCC .+8. ;BR IF NO ERROR

```

CVDMEA.P11 10-DEC-80 09:16

TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1

6555	030552			ERROR		;REPORT STACKED ERROR		
6556	030552	104460		ESCAPE	TST	;SKIP TO END OF TEST	TRAP	C\$ERROR
6557	030554							
6558	030554	104410					TRAP	C\$ESCAPE
6559	030556	000314					.WORD	L10025-.
6560								
6561	030560	004537	010136	JSR	R5,TXCHAR	;LOAD DATA2(252), TX DATA1(125)		
6562	030564	000252		252				
6563	030566	000010		8.				
6564	030570	103003		BCC	+.8.	;BR IF NO ERROR		
6565	030572			ERROR		;REPORT STACKED ERROR		
6566	030572	104460		ESCAPE	TST	;SKIP TO END OF TEST	TRAP	C\$ERROR
6567	030574							
6568	030574	104410					TRAP	C\$ESCAPE
6569	030576	000274					.WORD	L10025-.
6570								
6571	030600	004537	010136	JSR	R5,TXCHAR	;LOAD DATA3(000), TX DATA2(252)		
6572	030604	000000		000				
6573	030606	000010		8.				
6574	030610	103003		BCC	+.8.	;BR IF NO ERROR		
6575	030612			ERROR		;REPORT STACKED ERROR		
6576	030612	104460		ESCAPE	TST	;SKIP TO END OF TEST	TRAP	C\$ERROR
6577	030614							
6578	030614	104410					TRAP	C\$ESCAPE
6579	030616	000254					.WORD	L10025-.
6580								
6581	030620	004537	010136	JSR	R5,TXCHAR	;LOAD DATA4(377), TX DATA3(000)		
6582	030624	000377		377				
6583	030626	000010		8.				
6584	030630	103003		BCC	+.8.	;BR IF NO ERROR		
6585	030632			ERROR		;REPORT STACKED ERROR		
6586	030632	104460		ESCAPE	TST	;SKIP TO END OF TEST	TRAP	C\$ERROR
6587	030634							
6588	030634	104410					TRAP	C\$ESCAPE
6589	030636	000234					.WORD	L10025-.
6590								
6591	030640	004537	010136	JSR	R5,TXCHAR	;LOAD DATA5(001), TX DATA4(377)		
6592	030644	000001		001				
6593	030646	000011		9.				
6594	030650	103003		BCC	+.8.	;BR IF NO ERROR		
6595	030652			ERROR		;REPORT STACKED ERROR		
6596	030652	104460		ESCAPE	TST	;SKIP TO END OF TEST	TRAP	C\$ERROR
6597	030654							
6598	030654	104410					TRAP	C\$ESCAPE
6599	030656	000214					.WORD	L10025-.
6600								
6601	030660	004537	010250	JSR	R5,TXCTRL	;SET TEOM		
6602	030664	000002		TEOM				
6603	030666	000000		0				
6604								
6605	030670	004537	011624	JSR	R5,RCV1ST	;CLOCK AND RCV DATA1(125)		
6606	030674	000000		0				
6607	030676	103003		BCC	+.8.	;BR IF NO ERROR		
6608	030700			ERROR		;REPORT STACKED ERROR		
6609	030700	104460		ESCAPE	TST	;SKIP TO END OF TEST	TRAP	C\$ERROR
6610	030702							

CVDMEA.P11 10-DEC-80 09:16

TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1

6611	030702	104410					TRAP	C\$ESCAPE
6612	030704	000166					.WORD	L10025-
6613								
6614	030706	004537	010350	JSR	R5,RXCHAR			
6615	030712	000525		RXSOM!	125			
6616	030714	000000		0				
6617	030716	000010		8.				
6618	030720	103003		BCC	.+8.			
6619	030722			ERROR				
6620	030722	104460						
6621	030724			ESCAPE	TST			
6622	030724	104410						
6623	030726	000144						
6624								
6625	030730	004537	010250	JSR	R5,TXCTRL			
6626	030734	000002		TFQM				
6627	030736	000000		0				
6628								
6629	030740	004537	010350	JSR	R5,RXCHAR			
6630	030744	000252		252				
6631	030746	000000		0				
6632	030750	000010		8.				
6633	030752	103003		BCC	.+8.			
6634	030754			ERROR				
6635	030754	104460						
6636	030756			ESCAPE	TST			
6637	030756	104410						
6638	030760	000112						
6639								
6640	030762	004537	010350	JSR	R5,RXCHAR			
6641	030766	000000		000				
6642	030770	000000		0				
6643	030772	000010		8.				
6644	030774	103003		BCC	.+8.			
6645	030776			ERROR				
6646	030776	104460						
6647	031000			ESCAPE	TST			
6648	031000	104410						
6649	031002	000070						
6650								
6651	031004	004537	010350	JSR	R5,RXCHAR			
6652	031010	000377		377				
6653	031012	000000		0				
6654	031014	020010		NCRACK!	8.			
6655	031016	103003		BCC	.+8.			
6656	031020			ERROR				
6657	031020	104460						
6658	031022			ESCAPE	TST			
6659	031022	104410						
6660	031024	000046						
6661								
6662	031026	004537	010350	JSR	R5,RXCHAR			
6663	031032	001001		RXEOM!	001			
6664	031034	000001		RERCHK				
6665	031036	060000		NFCRDA!	NCRACK			
6666	031040	103003		BCC	.+8.			

CVDMEA.P11 10-DEC-80 09:16

TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1

6667	031042			ERROR		;REPORT STACKED ERROR		
6668	031042	104460					TRAP	C\$ERROR
6669	031044			ESCAPE TST		;SKIP TO END OF TEST		
6670	031044	104410					TRAP	C\$ESCAPE
6671	031046	000024					.WORD	L10025-
6672				;-----				
6673	031050	005337	002526	DEC	REG0	;DECREMENT COUNT		
6674	031054	001406		BEQ	40\$;BR IF TRIPLE LOOP IS COMPLETED		
6675								
6676	031056	004537	010250	JSR	R5,TXCTRL	;CLEAR TEOM, SET TSOM		
6677	031062	000001		TSOM				
6678	031064	000001		1				
6679	031066	000137	030460	JMP	2\$;AND RUN TX/RX AGAIN		
6680	031072							
6681	031072			40\$:				
6682	031072			ENDTST				
6683	031072	104401					L10025:	TRAP C\$ETST

CVDMEA.P11 10-DEC-80 09:16

TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0

.SBTTL TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0

6684
6685
6686
6687
6688
6689
6690
6691
6692
6693
6694
6695
6696
6697
6698
6699
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739

031074
031074 012737 000003 002526
031102 004737 005376
031106 042737 001000 031150
031114 023727 002470 000000
031122 001407
031124 023727 002472 000004
031132 001403
031134 052737 001000 031150

031142 004537 007234
031146 000400
031150 000000
031152 103003
031154 104460
031156 104410
031160 000374

031162 004537 010250
031166 000001
031170 000007
031172 004537 010250
031176 000001
031200 000010
031202 004537 010250
031206 000001
031210 000010
031212 004537 010250
031216 000000
031220 000000
031222 004537 010136
031226 000125
031230 000010
031232 103003

```
*****
*
* TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0
*
* IF XLB IS SPECIFIED IN THE P-TABLE THIS TEST WILL TRANSMIT &
* RECEIVE IN BOP MODE WITH CRC-CCITT-0 & OR DETECTION THE FOLLOWING
* SHORT MESSAGE: 125 252 000 377 001
*
* THIS MESSAGE WILL BE PRECEDED BY FLAG CHARACTERS AND REPEATED
* THREE TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
* LENGTHS ARE ALSO UTILIZED.
*
* IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
* USING INTERNAL LOOPBACK (TTLOOP=1).
*
*****
```

```

:          BGNTST
:
:          T7::
:          MOV #3,REG0          ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
:          JSR PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
:          BIC #NOLOOP,1$
:          CMP BRDTP,#0        ;BOARD TYPE = M8064 ?
:          BEQ 2$              ; YES: SPECIFY TTLOOP (NOT XLB)
:          CMP TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
:          BEQ 2$              ;BR IF NO
:          BIS #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
-----
2$: JSR R5,INITRN              ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
   CRCOS                      ;SET BOP MODE,CRC-CCITT=>0'S CHECK
1$: 0                          ;USE 8 BIT CHARS
   BCC .+8.                    ;BR IF NO ERROR
   ERROR                       ;REPORT STACKED ERROR
                                TRAP C$ERROR
   ESCAPE TST                  ;SKIP TO END OF TEST
                                TRAP C$ESCAPE
                                .WORD L10026-
:
:          JSR R5,TXCTRL        ;LOAD 2ND FLAG, TX 1ST FLAG
:          TSOM 7.
:          JSR R5,TXCTRL        ;LOAD 3RD FLAG, TX 2ND FLAG
:          TSOM 8.
:          JSR R5,TXCTRL        ;LOAD 4TH FLAG, TX 3RD FLAG
:          TSOM 8.
:          JSR R5,TXCTRL        ;CLEAR TSOM
:          000
:          0
:          JSR R5,TXCHAR        ;LOAD DATA1(125), TX 4TH FLAG
:          125
:          8.
:          BCC .+8.            ;BR IF NO ERROR
```

CVDMEA.P11 10-DEC-80 09:16

TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0

6740	031234			ERROR		;REPORT STACKED ERROR		
6741	031234	104460					TRAP	C\$ERROR
6742	031236			ESCAPE	TST	;SKIP TO END OF TEST		
6743	031236	104410					TRAP	C\$ESCAPE
6744	031240	000314					.WORD	L10026-
6745								
6746	031242	004537	010136	JSR	R5,TXCHAR	;LOAD DATA2(252), TX DATA1(125)		
6747	031246	000252		252				
6748	031250	000010		8.				
6749	031252	103003		BCC	.+8.	;BR IF NO ERROR		
6750	031254			ERROR		;REPORT STACKED ERROR		
6751	031254	104460					TRAP	C\$ERROR
6752	031256			ESCAPE	TST	;SKIP TO END OF TEST		
6753	031256	104410					TRAP	C\$ESCAPE
6754	031260	000274					.WORD	L10026-
6755								
6756	031262	004537	010136	JSR	R5,TXCHAR	;LOAD DATA3(000), TX DATA2(252)		
6757	031266	000000		000				
6758	031270	000010		8.				
6759	031272	103003		BCC	.+8.	;BR IF NO ERROR		
6760	031274			ERROR		;REPORT STACKED ERROR		
6761	031274	104460					TRAP	C\$ERROR
6762	031276			ESCAPE	TST	;SKIP TO END OF TEST		
6763	031276	104410					TRAP	C\$ESCAPE
6764	031300	000254					.WORD	L10026-
6765								
6766	031302	004537	010136	JSR	R5,TXCHAR	;LOAD DATA4(377), TX DATA3(000)		
6767	031306	000377		377				
6768	031310	000010		8.				
6769	031312	103003		BCC	.+8.	;BR IF NO ERROR		
6770	031314			ERROR		;REPORT STACKED ERROR		
6771	031314	104460					TRAP	C\$ERROR
6772	031316			ESCAPE	TST	;SKIP TO END OF TEST		
6773	031316	104410					TRAP	C\$ESCAPE
6774	031320	000234					.WORD	L10026-
6775								
6776	031322	004537	010136	JSR	R5,TXCHAR	;LOAD DATA5(001), TX DATA4(377)		
6777	031326	000001		001				
6778	031330	000011		9.				
6779	031332	103003		BCC	.+8.	;BR IF NO ERROR		
6780	031334			ERROR		;REPORT STACKED ERROR		
6781	031334	104460					TRAP	C\$ERROR
6782	031336			ESCAPE	TST	;SKIP TO END OF TEST		
6783	031336	104410					TRAP	C\$ESCAPE
6784	031340	000214					.WORD	L10026-
6785								
6786	031342	004537	010250	JSR	R5,TXCTRL	;SET TEOM		
6787	031346	000002		TEOM				
6788	031350	000000		0				
6789	031352	004537	011624	JSR	R5,RCV1ST	;CLOCK AND RCV DATA1(125)		
6790	031356	000000		0				
6791	031360	103003		BCC	.+8.	;BR IF NO ERROR		
6792	031362			ERROR		;REPORT STACKED ERROR		
6793	031362	104460					TRAP	C\$ERROR
6794	031364			ESCAPE	TST	;SKIP TO END OF TEST		
6795	031364	104410					TRAP	C\$ESCAPE

CVDMEA.P11 10-DEC-80 09:16

TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0

6796	031366	000166				.WORD	L10026-
6797							
6798	031370	004537	010350	JSR R5,RXCHAR	:READ/CHK DATA1(125), RCV DATA2(252)		
6799	031374	000525		RXSOM!125	: & CHECK RSOM=1		
6800	031376	000000		0			
6801	031400	000010		8.			
6802	031402	103003		BCC .+8.	:BR IF NO ERROR		
6803	031404			ERROR	:REPORT STACKED ERROR		
6804	031404	104460				TRAP	C\$ERROR
6805	031406			ESCAPE TST	:SKIP TO END OF TEST		
6806	031406	104410				TRAP	C\$ESCAPE
6807	031410	000144				.WORD	L10026-
6808							
6809	031412	004537	010250	JSR R5, TXCTRL	:SET TEOM		
6810	031416	000002		TEOM			
6811	031420	000000		0			
6812	031422	004537	010350	JSR R5,RXCHAR	:READ/CHK DATA2(252), RCV DATA3(000)		
6813	031426	000252		252			
6814	031430	000000		0			
6815	031432	000010		8.			
6816	031434	103003		BCC .+8.	:BR IF NO ERROR		
6817	031436			ERROR	:REPORT STACKED ERROR		
6818	031436	104460				TRAP	C\$ERROR
6819	031440			ESCAPE TST	:SKIP TO END OF TEST		
6820	031440	104410				TRAP	C\$ESCAPE
6821	031442	000112				.WORD	L10026-
6822							
6823	031444	004537	010350	JSR R5,RXCHAR	:READ/CHK DATA3(000), RCV DATA4(377)		
6824	031450	000000		000			
6825	031452	000000		0			
6826	031454	000010		8.			
6827	031456	103003		BCC .+8.	:BR IF NO ERROR		
6828	031460			ERROR	:REPORT STACKED ERROR		
6829	031460	104460				TRAP	C\$ERROR
6830	031462			ESCAPE TST	:SKIP TO END OF TEST		
6831	031462	104410				TRAP	C\$ESCAPE
6832	031464	000070				.WORD	L10026-
6833							
6834	031466	004537	010350	JSR R5,RXCHAR	:READ/CHK DATA4(377), RCV DATA5(001)		
6835	031472	000377		377			
6836	031474	000000		0			
6837	031476	020010		NCRCT!8.	:DON'T CHECK FOR FINAL PXACT=1		
6838	031500	103003		BCC .+8.	:BR IF NO ERROR		
6839	031502			ERROR	:REPORT STACKED ERROR		
6840	031502	104460				TRAP	C\$ERROR
6841	031504			ESCAPE TST	:SKIP TO END OF TEST		
6842	031504	104410				TRAP	C\$ESCAPE
6843	031506	000046				.WORD	L10026-
6844							
6845	031510	004537	010350	JSR R5,RXCHAR	:READ/CHK DATA5(001), RCV FIRST FLAG		
6846	031514	001001		RXEOM!001	: & CHECK REOM		
6847	031516	000001		RERCHK	: & CHECK RERR BIT=0 (GOOD CRC)		
6848	031520	060000		NFCRDA!NCRCT	:DON'T CHECK FOR FINAL RDA=RXACT=1		
6849	031522	103003		BCC .+8.	:BR IF NO ERROR		
6850	031524			ERROR	:REPORT STACKED ERROR		
6851	031524	104460				TRAP	C\$ERROR

CVDMEA.P11 10-DEC-80 09:16

TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0

6852 031526
 6853 031526 104410
 6854 031530 000024
 6855
 6856 031532 005337 002526
 6857 031536 001406
 6858
 6859 031540 004537 010250
 6860 031544 000001
 6861 031546 000001
 6862 031550 000137 031142
 6863 031554
 6864 031554
 6865 031554
 6866 031554 104401

ESCAPE TST ;SKIP TO END OF TEST
 TRAP C\$ESCAPE
 .WORD L10026-.

DEC REGO ;DECREMENT COUNT
 BEQ 40\$;BR IF TRIPLE LOOP IS COMPLETED
 JSR R5,TXCTRL ;CLEAR TEOM, SET TSOM
 TSOM
 1
 JMP 2\$;AND RUN TX/RX AGAIN
 40\$:
 ENDTST
 L10026: TRAP C\$ETST

CVDMEA.P11 10-DEC-80 09:16

TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST

6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893 031556
6894 031556 012737 000010 002414
6895 031564 004537 007532
6896 031570 100000
6897 031572 103002
6898 031574
6899 031574 104432
6900 031576 000722
6901 031600
6902 031600
6903 031600
6904 031600 104402
6905
6906 031602 004737 003352
6907 031606 004537 003712
6908 031612 120002
6909 031614 000377
6910 031616 004537 003712
6911 031622 120003
6912 031624 000000
6913 031626 004537 003712
6914 031632 120000
6915 031634 000002
6916 031636 012737 000001 002336
6917
6918 031644 004537 003566
6919 031650 120017
6920 031652 000000
6921 031654 142737 000023 031652
6922 031662 123727 031652 000354

.SBTTL TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST
:*****
*
* TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST
*
* FIRST, THE DMV-11 IS INITIALIZED. THEN, TTL LOOPBACK IS SELECTED,
* AND THE FOLLOWING CHECKS ARE PERFORMED INVOLVING THE MODEM STATUS
* REGISTER :
* - RING, CARRIER, MODEM READY, TEST MODE, CTS ARE CHECKED FOR 1 STATE.
* - RTS IS DE-ASSERTED AND CTS IS CHECKED FOR 0.
* - RTS IS ASSERTED AND CTS IS CHECKED FOR 1.
*
* NEXT, IF THE OPTION IS AN M8053 WITH AN H3254 TEST CONNECTOR INSTALLED,
* THE DMV-11 IN INITIALIZED AGAIN, (TTL LOOPBACK IS CLEARED), AND
* THE FOLLOWING CHECKS ARE PERFORMED :
* - RING (IF EIA), CARRIER, MODEM READY, CTS ARE CHECKED FOR 1, TEST
* MODE IS CHECKED FOR 0.
* - RTS IS DE-ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 0.
* - RTS IS ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 1.
* - DTR IS DE-ASSERTED, AND MODEM READY IS CHECKED FOR 0.
* - DTR IS ASSERTED, AND MODEM READY IS CHECKED FOR 1.
*
*-----*****

```

:      BGNTST
:                                          T8::
:      MOV      #8, TSTNUM      ;SET TEST NO. FOR POSSIBLE PRINTOUT
:      JSR      R5, CKLPBK     ;CHK FOR H3254/5 INSTALLED
:      TCCHEK
:      BCC      2$            ;BR IF YES, TO RUN TEST
:      EXIT     TST           ;NO TEST CONNECTOR, SKIP TEST
:
:      2$:
:      BGNSUB
:
:                                          T8.1:
:                                          TRAP    CSBSUB
:      :INIT DMV, SET TTL LOOPBACK, CHK MODEM STATUS
:      JSR      PC, MSTCLR     ;PERFORM MASTER CLEAR TO INIT DMV11
:      JSR      R5, WRITEI     ;SET PORT B FOR OUTPUT MODE
:      VIADPB
:      377
:      JSR      R5, WRITEI     ;SET PORT A FOR INPUT MODE
:      VIADPA
:      000
:      JSR      R5, WRITEI     ;SET TTL LOOPBACK
:      VIAORB
:      TTLOOP
:      MOV      #1, REGNUM     ;SET REG NO. FOR PRINTOUT
:      :CHK FOR RING, CARRIER, MODEM RDY, CTS, TEST MODE, = 1
:      JSR      R5, READI      ;READ MODEM STATUS
:
:      4$:
:      .WORD    0
:      BICB    #SPEED!RCVDAT!UMAIN, 4$ ;CLEAR UNNEEDED BITS
:      CMPB    4$, #RING!CARRIER!MDMRDY!CTS!TM ;CHK FOR BITS SET

```

CVDMEA.P11 10-DEC-80 09:16

TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST

```

6923 031670 001414      BEQ      8$      ;BR IF ALL BITS SET
6924 031672 012737 000354 002324      MOV      #RING!CARRIER!MDM!RDY!CTS!TM,GDATA ;SET EXPECTED DATA
6925 031700 013737 031652 002326      MOV      4$,BDATA ;SET ACTUAL DATA
6926      ;REPORT 'MODEM STATUS INCORRECT'
6927 031706      GEDF      EM83,ERR11
6928      ;
6929 031706 104455      ;
6930 031710 000102      TRAP     C$ERDF
6931 031712 015262      .WORD   66
6932 031714 020540      .WORD   EM83
6933 031716      .WORD   ERR11
6934 031716 104410      ESCAPE  SUB
6935 031720 000106      TRAP     C$ESCAPE
6936      .WORD   L10030-.
6937 031722 004537 003712      ;DE-ASSERT RTS, CHK FOR CTS = 0
6938 031726 120000      8$:     JSR      R5,WRITEI ;DE-ASSERT RTS
6939 031730 000012      VIAORB
6940 031732 004537 003566      RTSND!TTLOOP
6941 031736 120017      JSR      R5,READI ;READ MODEM STATUS
6942 031740 000000      VIAORA
6943 031742 132737 000010 031740      10$:    .WORD   0
6944 031750 001406      BITB    #CTS,10$ ;CHK FOR CTS = 0
6945      BEQ      12$      ;BR IF YES
6946 031752      ;REPORT CTS NOT CLEARED
6947      GEDF      EM84,ERR14
6948      ;
6949 031752 104455      ;
6950 031754 000103      TRAP     C$ERDF
6951 031756 015311      .WORD   67
6952 031760 021146      .WORD   EM84
6953 031762      .WORD   ERR14
6954 031762 104410      ESCAPE  SUB
6955 031764 000042      TRAP     C$ESCAPE
6956 031766 004537 003712      ;ASSERT RTS, CHK FOR CTS = 1
6957 031772 120000      12$:    JSR      R5,WRITEI ;ASSERT RTS
6958 031774 000002      VIAORB
6959 031776 004537 003566      TTLOOP
6960 032002 120017      JSR      R5,READI ;READ MODEM STATUS
6961 032004 000000      VIAORA
6962 032006 132737 000010 032004      14$:    .WORD   0
6963 032014 001004      BITB    #CTS,14$ ;CHK FOR CTS = 1
6964      BNE      15$      ;BR IF YES
6965 032016      ;REPORT CTS NOT SET
6966      GEDF      EM85,ERR14
6967 032016 104455      ;
6968 032020 000104      TRAP     C$ERDF
6969 032022 015326      .WORD   68
6970 032024 021146      .WORD   EM85
6971 032026      .WORD   ERR14
6972 032026      ENDSUB
6973 032026      L10030:
6974 032026 104403      TRAP     C$ESUB
6975      ;SEE IF BOARD IS M8053 WITH H3254 INSTALLED
6976 032030 005737 002470      TST     BRDTYP ;SEE IF M8053
6977 032034 001002      BNE     17$      ;BR IF YES
6978 032036 000137 032520      16$:    JMP      A1 ;SKIP THIS SECTION OF CODE

```

CVDMEA.P11 10-DEC-80 09:16

TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST

```

6979 032042 023727 002472 000000 17$:  CMP      TSTCON,#0      ;SEE IF H3254 INSTALLED
6980 032050 001372                BNE      16$          ;BR IF NOT, TO SKIP CODE
6981 032052                BGNSUB
6982 032052                T8.2:
6983 032052 104402                TRAP     CSBSUB
6984                ;INIT DMV, (TTL LOOPBACK IS CLEARED), CHK MODEM STATUS
6985 032054 004737 003352        JSR      PC,MSTCLR    ;PERFORM MASTER CLEAR TO INIT DMV11
6986 032060 004537 003712        JSR      R5,WRITEI   ;SET PORT B FOR OUTPUT MODE
6987 032064 120002                VIADPB
6988 032066 000377                377
6989 032070 004537 003712        JSR      R5,WRITEI   ;SET PORT A FOR INPUT MODE
6990 032074 120003                VIADPA
6991 032076 000000                OC 0
6992 032100 004537 003712        JSR      R5,WRITEI   ;DISABLE TTLOOP
6993 032104 120000                VIAORB
6994 032106 000000                000
6995 032110 012737 000001 002336    MOV      #1,REGNUM   ;SET REG NO. FOR PRINTOUT
6996
6997                -----
6998                ;CHK FOR RING (IF EIA), CARRIER, MODEM RDY, CTS = 1, TEST MODE, = 0
6999                -----
6999 032116 004537 003566        JSR      R5,READI    ;READ MODEM STATUS
7000 032122 120017                VIAORA
7001 032124 000000                .WORD   0
7002
7003 032126 023727 002470 000001        CMP      BRDYP,#1    ;IS V.35 THE SELECTED I/F ?
7004 032134 001013                BNE      21$          ; NO: BR TO DO CHECK WITH RING
7005                ;YES: REMOVE RING BEFORE CHECKING
7006 032136 142737 000223 032124        BICB    #RING!SPEED!RCVDAT!UMAIN,18$ ;CLEAR UNNEEDED BITS
7007 032144 123727 032124 000150        CMPB    18$,#CARRIER!MDMRDY!CTS ;CHK FOR CORRECT STATUS
7008 032152 001427                BEQ     20$          ;BR IF STATUS CORRECT
7009 032154 012737 000150 002324        MOV     #CARRIER!MDMRDY!CTS,GDATA ;SET EXPECTED DATA
7010 032162 000412                BR      19$
7011                ; DO CHECK WITH RING....
7012 032164 142737 000023 032124 21$:  BICB    #SPEED!RCVDAT!UMAIN,18$ ;CLEAR UNNEEDED BITS
7013 032172 123727 032124 000350        CMPB    18$,#RING!CARRIER!MDMRDY!CTS ;CHK FOR CORRECT STATUS
7014 032200 001414                BEQ     20$          ;BR IF STATUS CORRECT
7015 032202 012737 000350 002324        MOV     #RING!CARRIER!MDMRDY!CTS,GDATA ;SET EXPECTED DATA
7016
7017                ;REPORT 'MODEM STATUS INCORRECT'
7018 032210 013737 032124 002326 19$:  MOV     18$,BDATA    ;SET ACTUAL DATA
7019 032216                GDFD    EM83,ERR11
7020                ; 'DEVICE FATAL' ERROR # 69
7021 032216 104455                TRAP    CSERDF
7022 032220 000105                .WORD  69
7023 032222 015262                .WORD  EM83
7024 032224 020540                .WORD  ERR11
7025 032226                ESCAPE  SUB
7026 032226 104410                TRAP    C$ESCAPE
7027 032230 000266                .WORD  L10031-
7028                ;DE-ASSERT RTS, CHK FOR CTS,CARRIER = 0
7029 032232 004537 003712 20$:  JSR      R5,WRITEI   ;DE-ASSERT RTS
7030 032236 120000                VIAORB
7031 032240 000010                RTSND
7032 032242 004537 003566        JSR      R5,READI    ;READ MODEM STATUS
7033 032246 120017                VIAORA
7034 032250 000000                .WORD   0

```

CVDMEA.P11 10-DEC-80 09:16

TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST

```

7035 032252 132737 000010 032250      BITB  #CTS,22$      ;CHK FOR CTS = 0
7036 032260 001406                    BEQ    24$          ;BR IF YES
7037                                     ;REPORT CTS NOT CLEARED
7038 032262                                GEDF  EM84,ERR14
7039                                     ;          'DEVICE FATAL' ERROR # 70
7040 032262 104455                                TRAP  C$ERDF
7041 032264 000106                                .WORD 70
7042 032266 015311                                .WORD EM84
7043 032270 021146                                .WORD ERR14
7044 032272                                ESCAPE SUB
7045 032272 104410                                TRAP  C$ESCAPE
7046 032274 000222                                .WORD L10031-.
7047 032276 132737 000100 032250 24$:  BITB  #CARRIER,22$ ;CHK FOR CARRIER = 0
7048 032304 001406                    BEQ    26$          ;BR IF YES
7049                                     ;REPORT CARRIER NOT CLEARED
7050 032306                                GEDF  EM86,ERR14
7051                                     ;          'DEVICE FATAL' ERROR # 71
7052 032306 104455                                TRAP  C$ERDF
7053 032310 000107                                .WORD 71
7054 032312 015342                                .WORD EM86
7055 032314 021146                                .WORD ERR14
7056 032316                                ESCAPE SUB
7057 032316 104410                                TRAP  C$ESCAPE
7058 032320 000176                                .WORD L10031-.
7059                                     ;ASSERT RTS, CHK FOR CTS,CARRIER = 1
7060 032322 004537 003712 26$:  JSR   R5,WRITEI ;ASSERT RTS
7061 032326 120000                    VIAORB
7062 032330 000000                    000
7063 032332 004537 003566                    JSR   R5,READI ;READ MODEM STATUS
7064 032336 120017                    VIAORA
7065 032340 000000                    .WORD 0
7066 032342 132737 000010 032340 28$:  BITB  #CTS,28$      ;CHK FOR CTS = 1
7067 032350 001006                    BNE   30$          ;BR IF YES
7068                                     ;REPORT CTS NOT SET
7069 032352                                GEDF  EM85,ERR14
7070                                     ;          'DEVICE FATAL' ERROR # 72
7071 032352 104455                                TRAP  C$ERDF
7072 032354 000110                                .WORD 72
7073 032356 015326                                .WORD EM85
7074 032360 021146                                .WORD ERR14
7075 032362                                ESCAPE SUB
7076 032362 104410                                TRAP  C$ESCAPE
7077 032364 000132                                .WORD L10031-.
7078 032366 132737 000100 032340 30$:  BITB  #CARRIER,28$ ;CHK FOR CARRIER = 1
7079 032374 001006                    BNE   32$          ;BR IF YES
7080                                     ;REPORT CARRIER NOT SET
7081 032376                                GEDF  EM87,ERR14
7082                                     ;          'DEVICE FATAL' ERROR # 73
7083 032376 104455                                TRAP  C$ERDF
7084 032400 000111                                .WORD 73
7085 032402 015363                                .WORD EM87
7086 032404 021146                                .WORD ERR14
7087 032406                                ESCAPE SUB
7088 032406 104410                                TRAP  C$ESCAPE
7089 032410 000106                                .WORD L10031-.
7090                                     ;DE-ASSERT DTR, CHK FOR MODEM READY = 0

```


CV0MEA.P11 10-DEC-80 09:16

TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST

7091 032412 004537 003712
 7092 032416 120000
 7093 032420 000020
 7094 032422 004537 003566
 7095 032426 120017
 7096 032430 000000
 7097 032432 132737 000040 032430
 7098 032440 001406
 7099
 7100 032442
 7101
 7102 032442 104455
 7103 032444 000112
 7104 032446 015403
 7105 032450 021146
 7106 032452
 7107 032452 104410
 7108 032454 000042
 7109
 7110 032456 004537 003712
 7111 032462 120000
 7112 032464 000000
 7113 032466 004537 003566
 7114 032472 120017
 7115 032474 000000
 7116 032476 132737 000040 032474
 7117 032504 001004
 7118
 7119 032506
 7120
 7121 032506 104455
 7122 032510 000113
 7123 032512 015426
 7124 032514 021146
 7125 032516
 7126 032516
 7127 032516
 7128 032516 104403
 7129 032520
 7130 032520
 7131 032520
 7132 032520 104401

```

32$: JSR R5,WRITEI ;DE-ASSERT DTR
      VIAORB
      DTR
      JSR R5,READI ;READ MODEM STATUS
      VIAORA
34$: .WORD 0
      BITB #MDMRDY,34$ ;CHK FOR MODEM READY = 0
      BEQ 36$ ;BR IF YES
;REPORT MODEM READY NOT CLEARED
      GEDF EM88,ERR14
; 'DEVICE FATAL' ERROR # 74
; TRAP C$ERDF
; .WORD 74
; .WORD EM88
; .WORD ERR14
      ESCAPE SUB
; TRAP C$ESCAPE
; .WORD L10031-.
;ASSERT DTR, CHK FOR MODEM READY = 1
36$: JSR R5,WRITEI ;ASSERT DTR
      VIAORB
      000
      JSR R5,READI ;READ MODEM STATUS
      VIAORA
38$: .WORD 0
      BITB #MDMRDY,38$ ;CHK FOR MODEM READY = 1
      BNE 40$ ;BR IF YES
;REPORT MODEM READY NOT SET
      GEDF EM89,ERR14
; 'DEVICE FATAL' ERROR # 75
; TRAP C$ERDF
; .WORD 75
; .WORD EM89
; .WORD ERR14
40$: ENDSUB
; L10031: TRAP C$ESUB
A1:
ENDTST
; L10027: TRAP C$ETST

```

CVDMEA.P11 10-DEC-80 09:16

TEST 9 -- DDCMP MESSAGE TEST

```

7133
7134
7135
7136
7137
7138
7139
7140
7141
7142
7143
7144
7145
7146
7147
7148
7149
7150
7151
7152
7153
7154
7155
7156
7157
7158
7159
7160 032522
7161 032522 012737 000003 002526
7162 032530 004737 005376
7163 032534 042737 001000 032576
7164 032542 023727 002470 000000
7165 032550 001407
7166 032552 023727 002472 000004
7167 032560 001403
7168 032562 052737 001000 032576
7169
7170 032570 004537 007234
7171 032574 065626
7172 032576 000000
7173 032600 103003
7174 032602
7175 032602 104460
7176 032604
7177 032604 104410
7178 032606 001032
7179
7180 032610 004537 010250
7181 032614 000001
7182 032616 000007
7183 032620 004537 010250
7184 032624 000001
7185 032626 000010
7186 032630 004537 010250
7187 032634 000000
7188 032636 000000

```

```

.SBTTL TEST 9 -- DDCMP MESSAGE TEST
*****
*
* TEST 9 -- DDCMP MESSAGE TEST
*
* THIS TEST WILL USE XLB IF IT IS ENABLED -- OTHERWISE TTL LOOPBACK
* WILL BE UTILIZED. THIS ASSURES THAT IT CAN ALWAYS BE RUN AS A
* GENERAL 'RINGOUT' OF THE M8053.
*
* INITIALIZATION: BCP MODE, CRC-16, IDLE = 0, SYNC (S/AR) = 226 OCT.
* (96 HEX.), RXCL & TXCL = 0 (CHAR. LENGTH = 8).
*
* THE FOLLOWING SAMPLE DDCMP MESSAGE IS TRANSMITTED & RECEIVED AND ALL
* DATA AND CRC CHARACTERS ARE CHECKED FOR ERRORS:
*
* ----- HEADER ----- DATA (PATTERN K) -----
* SYNC SYNC 201 000 075 003 002 001 CRC CRC 000 377 ... 252 000 CRC CRC
*
* THE ATTEMPT HERE IS TO PROVIDE A TEST JUST BELOW THE LEVEL OF THE
* FUNCTIONAL DIAGNOSTIC. THE USYRT WILL BE RESPONSIBLE FOR ALL CRC
* GENERATION AND VERIFICATION BUT THE CRC'S WILL ALSO BE VERIFIED BY
* SOFTWARE.
*****
*
* BGNSTST
*
* T9::
* MOV #3,REG0 ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
* JSR PC,INIDMV ;INIT DMV-11, ENTER MAINT LOOP
* BIC #NOLoop,1$
* CMP BRDTYP,#0 ;BOARD TYPE = M8064 ?
* BEQ 2$ ; YES: SPECIFY TTLOOP (NOT XLB)
* CMP TSTCON,#4 ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
* BEQ 2$ ;BR IF NO
* BIS #NOLoop,1$ ; YES: SPECIFY NO TTLOOP (INITRN)
*-----
2$: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP, IDLE, CRC-16, SYNCH=226
1$: 0 ;USE 8 BIT CHARS
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
*
* ESCAPE TST ;SKIP TO END OF TEST TRAP C$ERROR
*
* .WORD L10032-. TRAP C$ESCAPE
*
* JSR R5,TXCTRL ;SET TSOM, TX 1ST SYNCH
* TSOM
* 7.
* JSR R5,TXCTRL ;TX 2ND SYNCH
* TSOM
* 8.
* JSR R5,TXCTRL ;CLEAR TSOM
* 000
* 0

```

CVDMEA.P11 10-DEC-80 09:16

TEST 9 -- DDCMP MESSAGE TEST

7189	032640	004537	010136	JSR	R5,TXCHAR	:LOAD 201(HEADR1), TX 3RD SYNCH		
7190	032644	000201		201				
7191	032646	000010		8.				
7192	032650	103003		BCC	+.8.	:BR IF NO ERROR		
7193	032652			ERROR		:REPORT STACKED ERROR		
7194	032652	104460					TRAP	C\$ERROR
7195	032654			ESCAPE	TST	:SKIP TO END OF TEST		
7196	032654	104410					TRAP	C\$ESCAPE
7197	032656	000762					.WORD	L10032-.
7198								
7199	032660	004537	010136	JSR	R5,TXCHAR	:LOAD 000(HEADR2), TX HEADR1		
7200	032664	000000		000				
7201	032666	000010		8.				
7202	032670	103003		BCC	+.8.	:BR IF NO ERROR		
7203	032672			ERROR		:REPORT STACKED ERROR		
7204	032672	104460					TRAP	C\$ERROR
7205	032674			ESCAPE	TST	:SKIP TO END OF TEST		
7206	032674	104410					TRAP	C\$ESCAPE
7207	032676	000742					.WORD	L10032-.
7208								
7209	032700	004537	010136	JSR	R5,TXCHAR	:LOAD 075(HEADR3), TX HEADR2		
7210	032704	000075		075				
7211	032706	000010		8.				
7212	032710	103003		BCC	+.8.	:BR IF NO ERROR		
7213	032712			ERROR		:REPORT STACKED ERROR		
7214	032712	104460					TRAP	C\$ERROR
7215	032714			ESCAPE	TST	:SKIP TO END OF TEST		
7216	032714	104410					TRAP	C\$ESCAPE
7217	032716	000722					.WORD	L10032-.
7218								
7219	032720	004537	010136	JSR	R5,TXCHAR	:LOAD 003(HEADR4), TX HEADR3		
7220	032724	000003		003				
7221	032726	000010		8.				
7222	032730	103003		BCC	+.8.	:BR IF NO ERROR		
7223	032732			ERROR		:REPORT STACKED ERROR		
7224	032732	104460					TRAP	C\$ERROR
7225	032734			ESCAPE	TST	:SKIP TO END OF TEST		
7226	032734	104410					TRAP	C\$ESCAPE
7227	032736	000702					.WORD	L10032-.
7228								
7229	032740	004537	010136	JSR	R5,TXCHAR	:LOAD 002(HEADR5)		
7230	032744	000002		002				
7231	032746	000000		0				
7232	032750	103003		BCC	+.8.	:BR IF NO ERROR		
7233	032752			ERROR		:REPORT STACKED ERROR		
7234	032752	104460					TRAP	C\$ERROR
7235	032754			ESCAPE	TST	:SKIP TO END OF TEST		
7236	032754	104410					TRAP	C\$ESCAPE
7237	032756	000662					.WORD	L10032-.
7238								
7239	032760	004537	011624	JSR	R5,RCV1ST	:CLOCK AND RCV 201		
7240	032764	000000		0				
7241	032766	103003		BCC	+.8.	:BR IF NO ERROR		
7242	032770			ERROR		:REPORT STACKED ERROR		
7243	032770	104460					TRAP	C\$ERROR
7244	032772			ESCAPE	TST	:SKIP TO END OF TEST		

CVDMEA.P11 10-DEC-80 09:16

TEST 9 -- DDCMP MESSAGE TEST

7245	032772	104410					TRAP	C\$ESCAPE
7246	032774	000644					.WORD	L10032-.
7247								
7248	032776	004537	010350	JSR	R5,RXCHAR	;READ & CHK 201(HEADR1), RCV HEADR2		
7249	033002	000201		201				
7250	033004	000000		0				
7251	033006	000010		8.				
7252	033010	103003		BCC	.+8.	;BR IF NO ERROR		
7253	033012			ERROR		;REPORT STACKED ERROR		
7254	033012	104460					TRAP	C\$ERROR
7255	033014			ESCAPE	TST	;SKIP TO END OF TEST		
7256	033014	104410					TRAP	C\$ESCAPE
7257	033016	000622					.WORD	L10032-.
7258								
7259	033020	004537	010136	JSR	R5,TXCHAR	;LOAD 001(HEADR6)		
7260	033024	000001		001				
7261	033026	000000		0				
7262	033030	103003		BCC	.+8.	;BR IF NO ERROR		
7263	033032			ERROR		;REPORT STACKED ERROR		
7264	033032	104460					TRAP	C\$ERROR
7265	033034			ESCAPE	TST	;SKIP TO END OF TEST		
7266	033034	104410					TRAP	C\$ESCAPE
7267	033036	000602					.WORD	L10032-.
7268								
7269	033040	004537	010350	JSR	R5,RXCHAR	;READ & CHK 000(HEADR2), RCV HEADR3		
7270	033044	000000		000				
7271	033046	000000		0				
7272	033050	000010		8.				
7273	033052	103003		BCC	.+8.	;BR IF NO ERROR		
7274	033054			ERROR		;REPORT STACKED ERROR		
7275	033054	104460					TRAP	C\$ERROR
7276	033056			ESCAPE	TST	;SKIP TO END OF TEST		
7277	033056	104410					TRAP	C\$ESCAPE
7278	033060	000560					.WORD	L10032-.
7279								
7280	033062	004537	010250	JSR	R5,TXCTRL	;SET TEOM		
7281	033066	000002		TEOM		;(STARTS CRC-16 CHARACTER)		
7282	033070	000000		0				
7283	033072	004537	010350	JSR	R5,RXCHAR	;READ & CHK 075(HEADR3), RCV HEADR4		
7284	033076	000075		075				
7285	033100	000000		0				
7286	033102	000010		8.				
7287	033104	103003		BCC	.+8.	;BR IF NO ERROR		
7288	033106			ERROR		;REPORT STACKED ERROR		
7289	033106	104460					TRAP	C\$ERROR
7290	033110			ESCAPE	TST	;SKIP TO END OF TEST		
7291	033110	104410					TRAP	C\$ESCAPE
7292	033112	000526					.WORD	L10032-.
7293								
7294	033114	004537	010350	JSR	R5,RXCHAR	;READ & CHK 003(HEADR4), RCV HEADR5		
7295	033120	000003		003				
7296	033122	000000		0				
7297	033124	000010		8.				
7298	033126	103003		BCC	.+8.	;BR IF NO ERROR		
7299	033130			ERROR		;REPORT STACKED ERROR		
7300	033130	104460					TRAP	C\$ERROR

CVDMEA.P11 10-DEC-80 09:16

TEST 9 -- DDCMP MESSAGE TEST

7301	033132			ESCAPE TST	:SKIP TO END OF TEST		
7302	033132	104410				TRAP	C\$ESCAPE
7303	033134	000504				.WORD	L10032-.
7304							
7305	033136	004537	010250	JSR R5,TXCTRL	:CLEAR TEOM		
7306	033142	000000		000			
7307	033144	000000		0			
7308							
7309	033146	004537	010136	JSR R5,TXCHAR	:LOAD 000(DATA1)		
7310	033152	000000		000			
7311	033154	000000		0			
7312	033156	103003		BCC .+8.	:BR IF NO ERROR		
7313	033160			ERROR	:REPORT STACKED ERROR		
7314	033160	104460				TRAP	C\$ERROR
7315	033162			ESCAPE TST	:SKIP TO END OF TEST		
7316	033162	104410				TRAP	C\$ESCAPE
7317	033164	000454				.WORD	L10032-.
7318							
7319	033166	004537	010350	JSR R5,RXCHAR	:READ & CHK 002(HEADR5), RCV HEADR6		
7320	033172	000002		002			
7321	033174	000000		0			
7322	033176	000010		8.			
7323	033200	103003		BCC .+8.	:BR IF NO ERROR		
7324	033202			ERROR	:REPORT STACKED ERROR		
7325	033202	104460				TRAP	C\$ERROR
7326	033204			ESCAPE TST	:SKIP TO END OF TEST		
7327	033204	104410				TRAP	C\$ESCAPE
7328	033206	000432				.WORD	L10032-.
7329							
7330	033210	004537	010136	JSR R5,TXCHAR	:LOAD 377(DATA2)		
7331	033214	000377		377			
7332	033216	000000		0			
7333	033220	103003		BCC .+8.	:BR IF NO ERROR		
7334	033222			ERROR	:REPORT STACKED ERROR		
7335	033222	104460				TRAP	C\$ERROR
7336	033224			ESCAPE TST	:SKIP TO END OF TEST		
7337	033224	104410				TRAP	C\$ESCAPE
7338	033226	000412				.WORD	L10032-.
7339							
7340	033230	004537	010350	JSR R5,RXCHAR	:READ/CHK 001(HEADR6), 'RCV' FIRST CRC BYTE		
7341	033234	100001		RXERR!001			
7342	033236	000001		RERCHK			
7343	033240	000010		8.			
7344	033242	103003		BCC .+8.	:BR IF NO ERROR		
7345	033244			ERROR	:REPORT STACKED ERROR		
7346	033244	104460				TRAP	C\$ERROR
7347	033246			ESCAPE TST	:SKIP TO END OF TEST		
7348	033246	104410				TRAP	C\$ESCAPE
7349	033250	000370				.WORD	L10032-.
7350							
7351	033252	004537	010136	JSR R5,TXCHAR	:LOAD 376(DATA3)		
7352	033256	000376		376			
7353	033260	000000		0			
7354	033262	103003		BCC .+8.	:BR IF NO ERROR		
7355	033264			ERROR	:REPORT STACKED ERROR		
7356	033264	104460				TRAP	C\$ERROR

CVDMEA.P11 10-DEC-80 09:16

TEST 9 -- DDCMP MESSAGE TEST

```

7357 033266          ESCAPE TST          ;SKIP TO END OF TEST
7358 033266 104410          TRAP      C$ESCAPE
7359 033270 000350          .WORD      L10032-.
7360
7361 033272 004537 010350  JSR      R5,RXCHAR      ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
7362 033276 000043          043
7363 033300 000000          0
7364 033302 000010          8.
7365 033304 103003          BCC     .+8.          ;BR IF NO ERROR
7366 033306          ERROR          ;REPORT STACKED ERROR
7367 033306 104460          TRAP      C$ERROR
7368 033310          ESCAPE TST          ;SKIP TO END OF TEST
7369 033310 104410          TRAP      C$ESCAPE
7370 033312 000326          .WORD      L10032-.
7371
7372 033314 004537 010136  JSR      R5, TXCHAR     ;LOAD 375(DATA4)
7373 033320 000375          375
7374 033322 000000          0
7375 033324 103003          BCC     .+8.          ;BR IF NO ERROR
7376 033326          ERROR          ;REPORT STACKED ERROR
7377 033326 104460          TRAP      C$ERROR
7378 033330          ESCAPE TST          ;SKIP TO END OF TEST
7379 033330 104410          TRAP      C$ESCAPE
7380 033332 000306          .WORD      L10032-.
7381
7382 033334 004537 010350  JSR      R5,RXCHAR     ;READ & CHK SECOND CRC BYTE; RCV DATA1
7383 033340 000035          035
7384 033342 000000          0
7385 033344 000010          8.
7386 033346 103003          BCC     .+8.          ;BR IF NO ERROR
7387 033350          ERROR          ;REPORT STACKED ERROR
7388 033350 104460          TRAP      C$ERROR
7389 033352          ESCAPE TST          ;SKIP TO END OF TEST
7390 033352 104410          TRAP      C$ESCAPE
7391 033354 000264          .WORD      L10032-.
7392
7393
7394
7395
-----
; TRANSMIT THE BULK OF DATA OUT OF TABLE 'PATK'
-----
7396 033356 012702 003012  MOV     #PATK,R2      ;SET UP TABLE POINTER
7397 033362 112237 033420  5$:    MOVB   (R2)+,20$    ;SET UP EXPECTED RX CHARACTER
7398 033366 116237 000003 033400  MOVB   3(R2),10$     ;SET UP TRANSMIT CHARACTER
7399
7400 033374 004537 010136  JSR     R5, TXCHAR     ;LOAD A CHARACTER
7401 033400 000000          10$:   000          ;** HOLE FOR NEXT TX CHARACTER
7402 033402 000000          0
7403 033404 103003          BCC     .+8.          ;BR IF NO ERROR
7404 033406          ERROR          ;REPORT STACKED ERROR
7405 033406 104460          TRAP      C$ERROR
7406 033410          ESCAPE TST          ;SKIP TO END OF TEST
7407 033410 104410          TRAP      C$ESCAPE
7408 033412 000226          .WORD      L10032-.
7409
7410 033414 004537 010350  JSR     R5,RXCHAR     ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
7411 033420 000000          20$:   000          ;** HOLE FOR EXPECTED CHARACTER
7412 033422 000000          0

```

CVDMEA.P11 10-DEC-80 09:16

TEST 9 -- DDCMP MESSAGE TEST

7413	033424	000010		8.					
7414	033426	103003		BCC	.+8.				
7415	033430			ERROR					
7416	033430	104460						TRAP	C\$ERROR
7417	033432			ESCAPE	TST				
7418	033432	104410						TRAP	C\$ESCAPE
7419	033434	000204						.WORD	L10032-
7420									
7421	033436	022702	003027	CMP	#PATK+13.,R2				
7422	033442	001347		BNE	5\$				
7423									
7424	033444	004537	010250	JSR	R5,TXCTRL				
7425	033450	000002		TEOM					
7426	033452	000000		0					
7427	033454	004537	010350	JSR	R5,RXCHAR				
7428	033460	000200		200					
7429	033462	000000		0					
7430	033464	000010		8.					
7431	033466	103003		BCC	.+8.				
7432	033470			ERROR					
7433	033470	104460						TRAP	C\$ERROR
7434	033472			ESCAPE	TST				
7435	033472	104410						TRAP	C\$ESCAPE
7436	033474	000144						.WORD	L10032-
7437									
7438	033476	004537	010250	JSR	R5,TXCTRL				
7439	033502	000002		TEOM					
7440	033504	000000		0					
7441	033506	004537	010350	JSR	R5,RXCHAR				
7442	033512	000125		125					
7443	033514	000000		0					
7444	033516	000010		8.					
7445	033520	103003		BCC	.+8.				
7446	033522			ERROR					
7447	033522	104460						TRAP	C\$ERROR
7448	033524			ESCAPE	TST				
7449	033524	104410						TRAP	C\$ESCAPE
7450	033526	000112						.WORD	L10032-
7451									
7452	033530	004537	010350	JSR	R5,RXCHAR				
7453	033534	000252		252					
7454	033536	000000		0					
7455	033540	000010		8.					
7456	033542	103003		BCC	.+8.				
7457	033544			ERROR					
7458	033544	104460						TRAP	C\$ERROR
7459	033546			ESCAPE	TST				
7460	033546	104410						TRAP	C\$ESCAPE
7461	033550	000070						.WORD	L10032-
7462									
7463	033552	004537	010350	JSR	R5,RXCHAR				
7464	033556	100000		RXERR!000					
7465	033560	000001		RERCHK					
7466	033562	000010		8.					
7467	033564	103003		BCC	.+8.				
7468	033566			ERROR					

CVDMEA.P11 10-DEC-80 09:16

TEST 9 -- DDCMP MESSAGE TEST

```

7469 033566 104460
7470 033570          ESCAPE TST          ;SKIF TO END OF TEST          TRAP C$ERROR
7471 033570 104410
7472 033572 000046          .WORD C$ESCAPE
7473                                     L10032-.
7474 033574 004537 010350 JSR R5,RXCHAR          ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
7475 033600 000231
7476 033602 000000
7477 033604 000010
7478 033606 103003 BCC .+8.          ;BR IF NO ERROR
7479 033610          ERROR          ;REPORT STACKED ERROR
7480 033610 104460          TRAP C$ERROR
7481 033612          ESCAPE TST          ;SKIP TO END OF TEST
7482 033612 104410          TRAP C$ESCAPE
7483 033614 000024          .WORD L10032-.
7484
7485 033616 004537 010350 JSR R5,RXCHAR          ;READ & CHK 2ND CRC BYTE, RCV 1ST SYNCH
7486 033622 000176
7487 033624 000000
7488 033626 000010
7489 033630 103003 BCC .+8.          ;BR IF NO ERROR
7490 033632          ERROR          ;REPORT STACKED ERROR
7491 033632 104460          TRAP C$ERROR
7492 033634          ESCAPE TST          ;SKIP TO END OF TEST
7493 033634 104410          TRAP C$ESCAPE
7494 033636 000002          .WORD L10032-.
7495          ENDTST
7496 033640
7497 033640 104401          L10032: TRAP C$ETST
    
```


CVDMEA.P11 10-DEC-80 09:16

HARDWARE PARAMETER CODING SECTION

.SBTTL HARDWARE PARAMETER CODING SECTION

7498
7499
7500
7501
7502
7503
7504
7505
7506
7507
7508
7509
7510
7511
7512
7513
7514
7515
7516
7517
7518
7519
7520
7521
7522
7523
7524
7525
7526
7527
7528
7529
7530
7531
7532
7533
7534
7535
7536
7537
7538
7539
7540
7541
7542
7543
7544
7545
7546
7547
7548

033642
033642 000027
033644
033644
033644 000031
033646 033722
033650 160020
033652 177776
033654
033654 001031
033656 033750
033660 000000
033662 000674
033664
033664 002032
033666 034001
033670 007000
033672 000004
033674 000007
033676
033676 005032
033700 034032
033702 000007
033704 000000
033706 000002
033710
033710 006032
033712 034115
033714 000007
033716 000000
033720 000004

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

BGNHRD

.WORD L10033-LSHARD/2
LSHARD::

GPRMA ADDRES,0,0,160020,177776,YES

.WORD TSCODE
.WORD ADDRES
.WORD TSLOLIM
.WORD TSHILIM

GPRMA VECTOR,2,0,0,674,YES

.WORD TSCODE
.WORD VECTOR
.WORD TSLOLIM
.WORD TSHILIM

GPRMD PRIRTY,4,0,7000,4,7,YES

.WORD TSCODE
.WORD PRIRTY
.WORD 7000
.WORD TSLOLIM
.WORD TSHILIM

GPRMD BDTY.M,12,0,7,0,2,YES

.WORD TSCODE
.WORD BDTY.M
.WORD 7
.WORD TSLOLIM
.WORD TSHILIM

GPRMD TCON.M,14,0,7,0,4,YES

.WORD TSCODE
.WORD TCON.M
.WORD 7
.WORD TSLOLIM
.WORD TSHILIM

ENDHRD

.EVEN
L10033:

.NLIST BEX
033722 042504 044526 042503 ADDRESS: .ASCIZ /DEVICE CSR ADDRESS : /
033750 042504 044526 042503 VECTOR: .ASCIZ /DEVICE VECTOR ADDRESS : /
034001 104 053105 041511 PRIRTY: .ASCIZ /DEVICE PRIORITY LEVEL : /
034032 047502 051101 020104 BDTY.M: .ASCIZ /BOARD TYPE (0=M8064, 1=M8053-V.35, 2=M8053-EIA) : /
034115 124 051125 040516 TCON.M: .ASCII /TURNAROUND CONNECTOR TYPE -/<15><12>

CVDMEA.P11 10-DEC-80 09:16

HARDWARE PARAMETER CODING SECTION

034152	030050	044075	031063
034240	031440	053075	031456

```
.ASCII / (0=H3254&H3255, 1=INTEGRAL MODEM CABLE, 2=EIA CABLE, /<15><12>
.ASCIIZ / 3=V.35 CABLE, 4=NONE) : /
BEX
```

7549

.LIST
.EVEN

CVDMEA.P11 10-DEC-80 09:16

SOFTWARE PARAMETER CODING SECTION

.SBTTL SOFTWARE PARAMETER CODING SECTION

7550
7551
7552
7553
7554
7555
7556
7557
7558
7559
7560
7561
7562
7563
7564
7565
7566
7567
7568

034272
034272 000000
034274
034274
034274
034274

:///
:// 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

.WORD L10034-L\$SOFT/2
L\$SOFT::

ENDSFT

L10034: .EVEN

CVDMEA.P11 10-DEC-80 09:16

```

7569
7570
7571 034274
7572          034474
7573 034474 000240
7574 034476 000240
7575 034500 000240
7576
7577
7578
7579
7580 034502
7581
7582
7583 034502
7584
7585 034502 000000
7586 034504 000000
7587 034506
7588
7589          000001

```

```

***** PATCH AREA FOR DEBUG *****
.SBTTL ***** PATCH AREA FOR DEBUG *****
PATCH:
      .=+200
      NOP
      NOP
      NOP
;*****
.SBTTL 'ENDMOD' STATEMENT
      ENDMOD
.SBTTL 'LASTAD' STATEMENT & END OF PROGRAM
      LASTAD
L$LAST::
.END

```

```

.EYEN
.WORD 0
.WORD 0

```


CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- USER SYMBOLS

CSRESE=	000033	955#	4505	4594						
CSREVI=	000003	955#	1026							
CSRFLA=	000021	955#								
CSRPT =	000025	955#								
CSSEFG=	000046	955#								
CSSPRI=	000041	955#								
CSSVEC=	000037	955#	4534							
CSTPRI=	000013	955#								
DDCMP =	040000	1331#	4632	4824	5826	6087	6311	7171		
DEVMAP	002404	1723#	4464*	4482*						
DEVPTR	002406	1724#	4468*	4480*	4482	4483*				
DFPTBL	002150	G 1103#								
DIAGMC=	000000	955								
DOTBMT=	000007	1243#								
DTR =	000020	1382#	2614	2617	2690	3152	3155	4817	4820	7093
DTRL =	000000	1387#								
D.BUG =	000000	1644#								
EF.COM=	000036	G 1177#	4454							
EF.NEW=	000035	G 1178#	4447							
EF.PWR=	000034	G 1179#								
EF.RES=	000037	G 1176#	4440							
EF.STA=	000040	G 1175#	4433							
EIAV35=	000002	1632#	3258							
EM100	015536	3112	3812#							
EM101	015556	3124	3812#							
EM14	014324	3569	3812#							
EM16	014340	3559	3812#							
EM2	014214	2638	3812#							
EM25	014360	2366	3812#							
EM28	014406	3056	3628	3812#						
EM29	014427	3045	3638	3812#						
EM3	014263	2105	3812#							
EM30	014444	3080	3605	3812#						
EM31	014465	3069	3615	3812#						
EM34	014502	3510	3812#	5011	5040	5069	5098	5127	5169	5198
EM35	014530	3536	3812#							
EM36	014551	3546	3812#	5140						
EM39	014566	3582	3812#							
EM4	014307	2149	2196	2267	3812#	4870				
EM40	014610	3592	3812#							
EM54	014626	3497	3812#							
EM68	014650	2742	3812#							
EM69	014677	2775	3812#							
EM70	014715	2786	3812#							
EM71	014737	2820	3812#							
EM72	014755	2831	3812#							
EM73	014777	2865	3812#	4890	4903	4916	4943	4956	4969	4982
EM74	015014	2876	3812#							
EM75	015035	2910	3812#	4995	5024	5053	5082	5111	5153	5182
EM76	015051	2921	3812#							
EM77	015071	2955	3812#							
EM78	015105	2966	3812#							
EM79	015125	3812#	4805							
EM80	015161	3812#	4930							
EM81	015201	3812#	5225							
EM82	015225	3812#	5236							

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- USER SYMBOLS

	7076	7088	7107	7127	7131	7160	7177	7196	7206	7216	7226	7236	7245
	7256	7266	7277	7291	7302	7316	7327	7337	7348	7358	7369	7379	7390
	7407	7418	7435	7449	7460	7471	7482	7493	7496	7512	7563	7581	
F\$CLEA= 000007	955#	4577	4581										
F\$DU = 000016	955#	4591	4596										
F\$END = 000041	955#	958	3858	3897	3931	3975	4020	4055	4090	4125	4510	4565	4583
	4598	4611	4627	4638	4648	4670	4682	4695	4705	4718	4729	4740	4743
	4745	4764	4771	4808	4835	4849	4873	4893	4906	4919	4933	4946	4959
	4972	4985	4998	5014	5027	5043	5056	5072	5085	5101	5114	5130	5143
	5156	5172	5185	5201	5214	5228	5241	5243	5813	5832	5842	5852	5866
	5876	5886	5896	5906	5915	5926	5942	5953	5970	5984	5995	6006	6017
	6028	6045	6048	6050	6074	6093	6103	6113	6127	6137	6147	6156	6167
	6183	6194	6207	6218	6228	6239	6250	6261	6270	6273	6275	6299	6317
	6327	6337	6351	6361	6371	6380	6391	6407	6418	6431	6442	6452	6463
	6474	6485	6494	6497	6499	6519	6536	6558	6568	6578	6588	6598	6611
	6622	6637	6648	6659	6670	6682	6684	6704	6721	6743	6753	6763	6773
	6783	6795	6806	6820	6831	6842	6853	6865	6867	6893	6899	6903	6934
	6953	6973	6975	6982	7026	7045	7057	7076	7088	7107	7127	7129	7131
	7133	7160	7177	7196	7206	7216	7226	7236	7245	7256	7266	7277	7291
	7302	7316	7327	7337	7348	7358	7369	7379	7390	7407	7418	7435	7449
	7460	7471	7482	7493	7496	7498	7547	7569	7581				
F\$HARD= 000004	955#	7512	7545										
F\$HW = 000013	955#	1101	1117										
F\$INIT= 000006	955#	4413	4508										
F\$JMP = 000050	955#	4771	6899										
F\$MOD = 000000	955#	958	7581										
F\$MSG = 000011	955#	3847	3856	3862	3895	3901	3929	3936	3973	3981	4018	4026	4053
	4061	4088	4096	4123									
F\$PROT= 000021	955#	4400	4405										
F\$PWR = 000017	955#												
F\$RPT = 000012	955#												
F\$SEG = 000003	955#												
F\$SOFT= 000005	955#	7563	7567										
F\$SRV = 000010	955#												
F\$SUB = 000002	955#	6904	6973	6983	7127								
F\$SW = 000014	955#	1126	1131										
F\$TEST= 000001	955#	4628	4743	4765	5241	5814	6048	6075	6273	6300	6497	6520	6682
	6705	6865	6894	7131	7161	7496							
GDATA 002324	1698#	2100*	2144*	2191*	2262*	2358	2630*	2631*	2734*	3501*	3502*	3519*	3520*
	3863*	3865	3885	3916	3965	4010	4081	4141	4796*	4797*	4864*	4880*	5005*
	5034*	5063*	5092*	5121*	5163*	5192*	6924*	7009*	7015*				
GETBSR 004024	2294#												
GETPRM 022740	4458	4472#	4481										
GETURS 004360	2390#	4353											
GETVRS 004460	2421#	4276											
GETWSR 004166	2099	2143	2190	2261	2315#	4863	5230						
GOODAT 002364	1715#												
G\$CNTD= 000200	955#												
G\$DELM= 000372	955#												
G\$DISP= 000003	955#												
G\$EXCP= 000400	955#												
G\$HILI= 000002	955#												
G\$LOLI= 000001	955#												
G\$ND = 000000	955#												
G\$OFFS= 000400	955#	7516	7521	7526	7532	7538							
G\$OF SI= 000376	955#	7516	7521	7526	7532	7538							

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- USER SYMBOLS

SEL16	002452	1753#	2322										
SEL2	002422	1735#	2316										
SEL4	002426	1738#	2132*	2179*	2225*	2250*	2317	4833*	4854*				
SEL6	002432	1741#	2202	2226*	2251*	2318							
SETVIA	005304	2678#	2713										
SFPTBL	002172 G	1128#											
SFR =	000001	1368#											
SPEED =	000020	1398#	6921	7006	7012								
SRMODE =	000034	1532#											
STALL	004356	2377#											
STARES	002412	1726#	3298	3309	3320	3337	4461*	4467*					
STARST	022710	4436	4443	4460#									
STEPLU	012072	3200	3391	3439	3660	3710	3766	3789#	4685				
STRIP =	000040	1323#											
STRIPS =	020000	1332#	4632	4824	5826	6087	6311	7171					
STRTML =	000301	1228#											
STUREG	004250	2344#											
SUBRPC	002344	1706#	4416*										
SVCGBL =	000000	955#	958	965#	982	991	993	995	997	999	1001	1003	1005
		1009	1011	1013	1015	1017	1019	1021	1023	1025	1028	1031	1033
		1037	1039	1041	1043	1045	1047	1049	1051	1053	1055	1057	1059
		1063	1065	1081	1102	1103	1127	1128	1656	2045	2057	3847	3862
		3936	3981	4026	4061	4096	4400	4413	4527	4577	4591	4607	7513
		7587#	7588										7564
SVCINS =	000001	955#	962#	983	984	985	986	987	988	989	990	992	994
		998	1000	1002	1004	1006	1008	1010	1012	1014	1016	1018	1020
		1024	1026	1027	1029	1030	1032	1034	1036	1038	1040	1042	1044
		1048	1050	1052	1054	1056	1058	1060	1062	1064	1066	1080	1082
		1084	1085	1086	1087	1088	1089	1090	1101	1126	2046	2049	2058
		2103	2104	2105	2106	2147	2148	2149	2150	2194	2195	2196	2197
		2266	2267	2268	2364	2365	2366	2367	2636	2637	2638	2639	2740
		2742	2743	2773	2774	2775	2776	2784	2785	2786	2787	2818	2819
		2821	2829	2830	2831	2832	2863	2864	2865	2866	2874	2875	2876
		2908	2909	2910	2911	2919	2920	2921	2922	2953	2954	2955	2877
		2965	2966	2967	2996	2997	2998	2999	3007	3008	3009	3010	2956
		3045	3046	3054	3055	3056	3057	3067	3068	3069	3070	3078	2964
		3081	3110	3111	3112	3113	3122	3123	3124	3125	3301	3302	3043
		3305	3312	3313	3314	3315	3316	3323	3324	3325	3326	3327	3079
		3342	3343	3344	3345	3495	3496	3497	3498	3508	3509	3510	3080
		3535	3536	3537	3544	3545	3546	3547	3557	3558	3559	3560	3303
		3569	3570	3580	3581	3582	3583	3590	3591	3592	3593	3603	3304
		3606	3613	3614	3615	3616	3626	3627	3628	3629	3636	3637	3340
		3850	3851	3852	3853	3854	3857	3869	3870	3871	3872	3873	3511
		3885	3886	3887	3888	3889	3890	3896	3905	3906	3907	3908	3534
		3911	3914	3915	3916	3917	3918	3919	3920	3921	3923	3924	3567
		3927	3930	3938	3939	3940	3941	3942	3943	3944	3946	3947	3568
		3950	3954	3955	3956	3957	3958	3959	3960	3963	3964	3965	3604
		3968	3969	3970	3974	3983	3984	3985	3986	3987	3988	3989	3605
		3993	3994	3995	3999	4000	4001	4002	4003	4004	4005	4008	3638
		4011	4012	4013	4014	4015	4019	4028	4029	4030	4031	4032	3639
		4036	4037	4038	4039	4040	4044	4045	4046	4047	4048	4049	3850
		4063	4064	4065	4066	4067	4068	4069	4071	4072	4073	4074	3851
		4079	4080	4081	4082	4083	4084	4085	4086	4089	4098	4099	3852
		4102	4103	4104	4106	4107	4108	4109	4110	4114	4115	4116	3853
		4119	4120	4124	4153	4154	4155	4156	4157	4158	4159	4161	3854
		4164	4165	4166	4167	4168	4169	4171	4172	4173	4174	4175	3855

CVDMEA.P11

10-DEC-80 09:16

CROSS REFERENCE TABLE -- USER SYMBOLS

4179	4180	4181	4182	4183	4184	4185	4186	4188	4189	4190	4191	4192
4193	4195	4196	4197	4198	4199	4200	4201	4202	4203	4205	4206	4207
4208	4209	4210	4212	4213	4214	4215	4216	4217	4218	4219	4220	4229
4230	4231	4232	4233	4234	4235	4237	4238	4239	4240	4241	4242	4243
4244	4245	4247	4248	4249	4250	4251	4252	4254	4255	4256	4257	4258
4259	4260	4261	4262	4264	4265	4266	4267	4268	4278	4279	4280	4281
4282	4283	4284	4286	4287	4288	4289	4290	4291	4292	4293	4294	4296
4297	4298	4299	4300	4301	4303	4304	4305	4306	4307	4308	4309	4310
4311	4313	4314	4315	4316	4317	4318	4320	4321	4322	4323	4324	4325
4326	4327	4328	4330	4331	4332	4333	4334	4335	4337	4338	4339	4340
4341	4342	4343	4344	4345	4355	4356	4357	4358	4359	4360	4361	4363
4364	4365	4366	4367	4368	4369	4370	4371	4373	4374	4375	4376	4377
4378	4380	4381	4382	4383	4384	4385	4386	4387	4388	4433	4434	4436
4440	4441	4443	4447	4448	4450	4454	4455	4457	4475	4476	4477	4479
4505	4509	4530	4531	4532	4533	4534	4535	4552	4553	4557	4558	4564
4582	4594	4597	4610	4636	4638	4639	4646	4648	4649	4668	4670	4671
4680	4682	4683	4693	4695	4696	4702	4705	4706	4716	4718	4719	4727
4729	4730	4738	4740	4741	4744	4771	4772	4803	4804	4805	4806	4808
4809	4833	4835	4836	4847	4849	4850	4868	4869	4870	4871	4873	4874
4888	4889	4890	4891	4893	4894	4901	4902	4903	4904	4906	4907	4914
4915	4916	4917	4919	4920	4928	4929	4930	4931	4933	4934	4941	4942
4943	4944	4946	4947	4954	4955	4956	4957	4959	4960	4967	4968	4969
4970	4972	4973	4980	4981	4982	4983	4985	4986	4993	4994	4995	4996
4998	4999	5009	5010	5011	5012	5014	5015	5022	5023	5024	5025	5027
5028	5038	5039	5040	5041	5043	5044	5051	5052	5053	5054	5056	5057
5067	5068	5069	5070	5072	5073	5080	5081	5082	5083	5085	5086	5096
5097	5098	5099	5101	5102	5109	5110	5111	5112	5114	5115	5125	5126
5127	5128	5130	5131	5138	5139	5140	5141	5143	5144	5151	5152	5153
5154	5156	5157	5167	5168	5169	5170	5172	5173	5180	5181	5182	5183
5185	5186	5196	5197	5198	5199	5201	5202	5209	5210	5211	5212	5214
5215	5223	5224	5225	5226	5228	5229	5234	5235	5236	5237	5242	5830
5832	5833	5840	5842	5843	5850	5852	5853	5864	5866	5867	5874	5876
5877	5884	5886	5887	5894	5896	5897	5904	5906	5907	5913	5915	5916
5924	5926	5927	5940	5942	5943	5951	5953	5954	5968	5970	5971	5982
5984	5985	5993	5995	5996	6004	6006	6007	6015	6017	6018	6026	6028
6029	6043	6045	6046	6049	6091	6093	6094	6101	6103	6104	6111	6113
6114	6125	6127	6128	6135	6137	6138	6145	6147	6148	6154	6156	6157
6165	6167	6168	6181	6183	6184	6192	6194	6195	6205	6207	6208	6216
6218	6219	6226	6228	6229	6237	6239	6240	6248	6250	6251	6259	6261
6262	6268	6270	6271	6274	6315	6317	6318	6325	6327	6328	6335	6337
6338	6349	6351	6352	6359	6361	6362	6369	6371	6372	6378	6380	6381
6389	6391	6392	6405	6407	6408	6416	6418	6419	6429	6431	6432	6440
6442	6443	6450	6452	6453	6461	6463	6464	6472	6474	6475	6483	6485
6486	6492	6494	6495	6498	6534	6536	6537	6556	6558	6559	6566	6568
6569	6576	6578	6579	6586	6588	6589	6596	6598	6599	6609	6611	6612
6620	6622	6623	6635	6637	6638	6646	6648	6649	6657	6659	6660	6668
6670	6671	6683	6719	6721	6722	6741	6743	6744	6751	6753	6754	6761
6763	6764	6771	6773	6774	6781	6783	6784	6793	6795	6796	6804	6806
6807	6818	6820	6821	6829	6831	6832	6840	6842	6843	6851	6853	6854
6866	6899	6900	6904	6929	6930	6931	6932	6934	6935	6948	6949	6950
6951	6953	6954	6967	6968	6969	6970	6974	6983	6983	7021	7023	7024
7026	7027	7040	7041	7042	7043	7045	7046	7052	7053	7054	7055	7057
7058	7071	7072	7073	7074	7076	7077	7083	7084	7085	7086	7088	7089
7102	7103	7104	7105	7107	7108	7121	7122	7123	7124	7128	7132	7175
7177	7178	7194	7196	7197	7204	7206	7207	7214	7216	7217	7224	7226
7227	7234	7236	7237	7243	7245	7246	7254	7256	7257	7264	7266	7267

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- USER SYMBOLS

	7275	7277	7278	7289	7291	7292	7300	7302	7303	7314	7316	7317	7325
	7327	7328	7335	7337	7338	7346	7348	7349	7356	7358	7359	7367	7369
	7370	7377	7379	7380	7388	7390	7391	7405	7407	7408	7416	7418	7419
	7433	7435	7436	7447	7449	7450	7458	7460	7461	7469	7471	7472	7480
	7482	7483	7491	7493	7494	7497	7512	7516	7517	7518	7519	7521	7522
	7523	7524	7526	7527	7528	7529	7530	7532	7533	7534	7535	7536	7538
	7539	7540	7541	7542	7545	7563	7567	7584	7585	7586			
SVCSUB= 000001	955#	964#	6903	6982									
SVCTAG= 000001	955#	966#	1073	1077	1117	1131	3856	3895	3929	3973	4018	4053	4088
	4123	4508	4563	4581	4596	4609	4743	5241	6048	6273	6497	6682	6865
	6973	7127	7131	7496	7546	7568							
SVCTST= 000001	955#	963#	4627	4764	5813	5074	6299	6519	6704	6893	7160		
SWPBOT= 121000	1624#												
SWPDDC= 121400	1625#												
SYNCH = 000226	1311#	3195	4632	5826	5836	5846	6087	6097	6107	6243	6254	6311	6321
	6331	6467	6478	7171									
SSLSYM= 010000	955#	1118#	1132#	3857#	3896#	3930#	3974#	4019#	4054#	4089#	4124#	4509#	4564#
	4582#	4597#	4610#	4744#	5242#	6049#	6274#	6498#	6683#	6866#	6974#	7128#	7132#
	7497#	7547#	7569#										
TAB = 000004	1295#												
TBMT = 000100	1362#	2858	2869	3205									
TCCHK= 100000	1629#	3243	6896										
TCON.M 034115	7539	7548#											
TDATA 002322	1697#	2349											
TDSRH = 120403	1290#	3191	3426	4838									
TDSRL = 120402	1284#	3194	3373	4841									
TDSRNR 002575	1809#												
TEOM = 000002	1296#	5960	5974	6602	6626	6787	6810	7281	7425	7439			
TERR = 000200	1293#												
TGA = 000010	1294#												
TIMFLG 002352	1710#												
TM = 000004	1400#	6922	6924										
TMPO 002546	1790#	4536*	4554	4566*									
TMP1 002550	1791#												
TMP2 002552	1792#												
TMP3 002554	1793#												
TMP4 002556	1794#												
TMP5 002560	1795#												
TMP6 002562	1796#												
TMP7 002564	1797#												
TSO = 000010	1365#	3105	3117										
TSOM = 000001	1297#	3192	4652	4656	4839	6035	6201	6222	6425	6446	6540	6543	6546
	6677	6725	6728	6731	6860	7181	7184						
TSTCON 002472	1762#	3245	3250	3266	3268	3277	3280	3286	4501*	5820	6080	6305	6525
	6710	6979	7166										
TSTNUM 002414	1727#	3340	4765*	6894*									
TTLOOP= 000002	1385#	3188	3218	3752	3765	4677	4690	6915	6939	6958			
TXAB = 002000	1302#												
TXACT = 000004	1366#	2768	2779										
TXCHAR 010136	3368#	4663	5835	5845	5859	5869	5879	5889	5899	5935	6096	6106	6120
	6130	6140	6176	6320	6330	6344	6354	6364	6400	6551	6561	6571	6581
	6591	6736	6746	6756	6766	6776	7189	7199	7209	7219	7229	7259	7309
	7330	7351	7372	7400									
TXCTRL 010250	3421#	4651	4655	4659	5855	5959	5973	6034	6116	6200	6221	6340	6424
	6445	6539	6542	6545	6548	6601	6625	6676	6724	6727	6730	6733	6786
	6809	6859	7180	7183	7186	7280	7305	7424	7438				

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- USER SYMBOLS

TXDL = 000340	1349#	3480	6088	6312	
TXEN = 000040	1381#	3188	3218	4677	4690
TXEOM = 001000	1303#				
TXERR = 100000	1300#				
TXGA = 004000	1301#				
TXSOM = 000400	1304#				
TXTMLT 017744	3821#	3884			
TXTML0 017154	3812#	3821			
TXTML1 017160	3812#	3821			
TXTML2 017174	3812#	3821			
TXTML3 017211	3812#	3821			
TXTML4 017233	3812#	3821			
TXTML5 017254	3812#	3821			
TXTML6 017304	3812#	3821			
TXTML7 017316	3812#	3821			
TXTNP 017500	3812#	3833			
TXTNPT 020030	3834#				
TXTNP0 017505	3812#	3834			
TXTNP1 017515	3812#	3834			
TXTNP2 017525	3812#	3834			
TXTNP3 017535	3812#	3834			
TXTNP4 017552	3812#	3834			
TXTNP5 017567	3812#	3834			
TXTNP6 017604	3812#	3834			
TXTNP7 017620	3812#	3834			
TXTNP8 017634	3812#	3834			
TXTNUL 017152	3812#				
TXTUR 017650	3812#	3906	3955	4045	
TXTURT 020052	3837#	3905	3954	4044	
TXTUR0 017663	3812#	3837			
TXTUR1 017671	3812#	3837			
TXTUR2 017677	3812#	3837			
TXTUR3 017705	3812#	3837			
TXTUR4 017713	3812#	3837			
TXTUR5 017722	3812#	3837			
TXTUR6 017731	3812#	3837			
TXTUR7 017735	3812#	3837			
TXTVR 017353	3812#	3825	4000	4115	
TXTVRA 017451	3812#	3829			
TXTVRB 017454	3812#	3829			
TXTVRC 017460	3812#	3829			
TXTVRD 017464	3812#	3829			
TXTVRE 017470	3812#	3829			
TXTVRF 017474	3812#	3829			
TXTVRT 017766	3826#	3999	4114		
TXTVR0 017371	3812#	3826			
TXTVR1 017375	3812#	3826			
TXTVR2 017401	3812#	3826			
TXTVR3 017406	3812#	3826			
TXTVR4 017413	3812#	3826			
TXTVR5 017420	3812#	3826			
TXTVR6 017425	3812#	3826			
TXTVR7 017432	3812#	3826			
TXTVR8 017437	3812#	3829			
TXTVR9 017444	3812#	3829			
TXT1 015602	3812#	4153			

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- USER SYMBOLS

TSLSYM= 010000	955#	1118	1132	3857	3896	3930	3974	4019	4054	4089	4124	4509	4564
	4582	4597	4610	4744	5242	6049	6274	6498	6683	6866	6974	7128	7132
	7497	7547	7569										
TSLTNO= 000011	7588#												
TSNEST= 177777	955#	958#	1101#	1117#	1126#	1131#	3847#	3856#	3862#	3895#	3901#	3929#	3936#
	3973#	3981#	4018#	4026#	4053#	4061#	4088#	4096#	4123#	4400#	4405#	4413#	4508#
	4527#	4563#	4577#	4581#	4591#	4596#	4607#	4609#	4628#	4743#	4765#	5241#	5814#
	6048#	6075#	6273#	6300#	6497#	6520#	6682#	6705#	6865#	6894#	6904#	6973#	6983#
	7127#	7131#	7161#	7496#	7512#	7545#	7563#	7567#	7581#				
TSNS0 = 000000	958#	7581											
TSNS1 = 000005	1101#	1117	1126#	1131	3847#	3856	3862#	3895	3901#	3929	3936#	3973	3981#
	4018	4026#	4053	4061#	4088	4096#	4123	4400#	4405	4413#	4508	4527#	4563
	4577#	4581	4591#	4596	4607#	4609	4628#	4743	4765#	5241	5814#	6048	6075#
	6273	6300#	6497	6520#	6682	6705#	6865	6894#	7131	7161#	7496	7512#	7545
	7563#	7567											
	6904#	6973	6983#	7127									
TSNS2 = 000002	955#												
TSPTNU= 000000	955#												
TSSAVL= 177777	955#												
TSSEGL= 177777	955#												
TSSUBN= 000000	955#	4627#	4764#	5813#	6074#	6299#	6519#	6704#	6893#	6903#	6982#	7160#	
TSTAGL= 177777	955#												
TSTAGN= 010035	955#	1101#	1126#	3847#	3862#	3901#	3936#	3981#	4026#	4061#	4096#	4400#	4413#
	4527#	4577#	4591#	4607#	4628#	4765#	5814#	6075#	6300#	6520#	6705#	6894#	6904#
	6983#	7161#	7512#	7563#									
TSTEMP= 000000	1073#	1077#	1082#	1083#	1084#	1085#	1086#	1087#	1088#	1089#	1090#	1091#	1117#
	1131#	3856#	3895#	3929#	3973#	4018#	4053#	4088#	4123#	4405#	4508#	4563#	4581#
	4596#	4609#	4638#	4639	4648#	4649	4670#	4671	4682#	4683	4695#	4696	4705#
	4706	4718#	4719	4729#	4730	4740#	4741	4743#	4771#	4772	4808#	4809	4835#
	4836	4849#	4850	4873#	4874	4893#	4894	4906#	4907	4919#	4920	4933#	4934
	4946#	4947	4959#	4960	4972#	4973	4985#	4986	4998#	4999	5014#	5015	5027#
	5028	5043#	5044	5056#	5057	5072#	5073	5085#	5086	5101#	5102	5114#	5115
	5130#	5131	5143#	5144	5156#	5157	5172#	5173	5185#	5186	5201#	5202	5214#
	5215	5228#	5229	5241#	5832#	5833	5842#	5843	5852#	5853	5866#	5867	5876#
	5877	5886#	5887	5896#	5897	5906#	5907	5915#	5916	5926#	5927	5942#	5943
	5953#	5954	5970#	5971	5984#	5985	5995#	5996	6006#	6007	6017#	6018	6028#
	6029	6045#	6046	6048#	6093#	6094	6103#	6104	6113#	6114	6127#	6128	6137#
	6138	6147#	6148	6156#	6157	6167#	6168	6183#	6184	6194#	6195	6207#	6208
	6218#	6219	6228#	6229	6239#	6240	6250#	6251	6261#	6262	6270#	6271	6273#
	6317#	6318	6327#	6328	6337#	6338	6351#	6352	6361#	6362	6371#	6372	6380#
	6381	6391#	6392	6407#	6408	6418#	6419	6431#	6432	6442#	6443	6452#	6453
	6463#	6464	6474#	6475	6485#	6486	6494#	6495	6497#	6536#	6537	6558#	6559
	6568#	6569	6578#	6579	6588#	6589	6598#	6599	6611#	6612	6622#	6623	6637#
	6638	6648#	6649	6659#	6660	6670#	6671	6682#	6721#	6722	6743#	6744	6753#
	6754	6763#	6764	6773#	6774	6783#	6784	6795#	6796	6806#	6807	6820#	6821
	6831#	6832	6842#	6843	6853#	6854	6865#	6899#	6900	6934#	6935	6953#	6954
	6973#	7026#	7027	7045#	7046	7057#	7058	7076#	7077	7088#	7089	7107#	7108
	7127#	7131#	7177#	7178	7196#	7197	7206#	7207	7216#	7217	7226#	7227	7236#
	7237	7245#	7246	7256#	7257	7266#	7267	7277#	7278	7291#	7292	7302#	7303
	7316#	7317	7327#	7328	7337#	7338	7348#	7349	7358#	7359	7369#	7370	7379#
	7380	7390#	7391	7407#	7408	7418#	7419	7435#	7436	7449#	7450	7460#	7461
	7471#	7472	7482#	7483	7493#	7494	7496#	7516#	7521#	7526#	7532#	7538#	7545#
	7567#	7581#											
TSTEST= 000011	955#	4627#	4764#	5813#	6074#	6299#	6519#	6704#	6893#	6903	6982	7160#	7588
TSTSTM= 177777	955#	3304	3315	3326	3344	3853	3857	3873	3889	3896	3910	3920	3926
	3930	3943	3949	3959	3969	3974	3988	3994	4004	4014	4019	4033	4039
	4049	4054	4068	4075	4085	4089	4103	4109	4119	4124	4158	4168	4175

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- USER SYMBOLS

	2874	2908	2919	2953	2964	2996	3007	3043	3054	3067	3078	3110	3122
	3495	3508	3534	3544	3557	3567	3580	3590	3603	3613	3626	3636	
T.EHRD= 000002	1644#												
T.ESF = 000000	1644#												
T.ESFT= 000003	1644#												
T1 023224 G	1082	4627#											
T1MODE= 000300	1517#												
T2 023524 G	1083	4764#											
T2MODE= 000040	1527#												
T3 026230 G	1084	5813#											
T4 027106 G	1085	6074#											
T5 027650 G	1086	6299#											
T6 030412 G	1087	6519#											
T7 031074 G	1088	6704#											
T8 031556 G	1089	6893#											
T8.1 031600	6903#												
T8.2 032052	6982#												
T9 032522 G	1090	7160#											
UAM = 000200 G	1200#												
UMAINT= 000001	1402#	6921	7006	7012									
UNIT 002410	1725#												
UPBITS 002566	1800#												
UREGS 002242	1690#	2390	2393*	2396	4363	4364	4365	4366	4380	4381	4382	4383	
USTATR= 122000	1357#	2395	2728	2764	2809	2854	2899	2944	3101	3203	3487	3714	
USYREG 002476	1767#	2621											
USYRT = 120400	1249#	2391	2404	2408									
VECTOR 033750	7522	7548#											
VIA = 120000	1371#	2422	2429										
VIAACR= 120013	1515#	2482	2491	2569	2578	2692	3171	4642					
VIADPA= 120003	1421#	2683	6911	6990									
VIADPB= 120002	1409#	2680	6908	6987									
VIAIER= 120016	1588#	2478	2565	2698									
VIAIFR= 120015	1566#												
VIAIS = 120001	1393#												
VIAORA= 120017	1608#	2686	6919	6941	6960	7000	7033	7064	7095	7114			
VIAORB= 120000	1377#	2613	2616	2689	3151	3154	3187	3764	4676	4689	4816	4819	6914
	6938	6957	6993	7030	7061	7092	7111						
VIAPCR= 120014	1553#	2695											
VIASR = 120012	1505#	4699											
VIAT1A= 120004	1435#	2504											
VIAT1B= 120005	1447#	2509	3793										
VIAT1C= 120006	1460#	2495	3174										
VIAT1D= 120007	1471#	2500	3177										
VIAT2A= 120010	1482#	2582											
VIAT2B= 120011	1494#	2587											
VREGS 002262	1693#	2421	4286	4287	4288	4289	4303	4304	4305	4306	4320	4321	4322
	4323	4337	4338	4339	4340								
WAIT50 005270	2655#												
WRIBYT 002360	1713#												
WRILOC= 000002	1239#	2252	2262										
WRIPAG= 000004	1241#												
WRITE 003700	2225#	2347											
WRITEI 003712	2249#	2477	2490	2494	2499	2503	2508	2564	2577	2581	2586	2612	2615
	2679	2682	2685	2688	2691	2694	2697	3150	3153	3157	3161	3167	3170
	3173	3176	3186	3190	3193	3372	3425	3763	3792	4641	4675	4688	4779
	4815	4818	4821	4826	4837	4840	6907	6910	6913	6937	6956	6986	6989

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- USER SYMBOLS

WSR0	002202	6992	7029	7060	7091	7110								
WSR10	002212	1665#	2315*	4240										
WSR12	002214	1673#	2319*	4257										
WSR14	002216	1675#	2320*	4256										
WSR16	002220	1677#	2321*	4255										
WSR2	002204	1679#	2322*	4254										
WSR4	002206	1667#	2316*	4239										
WSR6	002210	1665#	2317*	4238										
XDATA	002330	1671#	2318*	4237										
XORGB	021262	1700#	3914	3963	4008	4079	4142*	4143*						
XYZ =	000007	3912	3961	4006	4077	4140#								
X\$ALWA=	000000	1326#												
X\$FALS=	000040	955#												
X\$OFFS=	000400	955#												
X\$TRUE=	000020	955#												
\$E =	000113	1644#	2102#	2146#	2193#	2264#	2363#	2635#	2739#	2772#	2783#	2817#	2828#	2862#
		2873#	2907#	2918#	2952#	2963#	2995#	3006#	3042#	3053#	3066#	3077#	3109#	3121#
		3494#	3517#	3533#	3543#	3556#	3566#	3579#	3589#	3602#	3612#	3625#	3635#	4802#
		4867#	4887#	4900#	4913#	4927#	4940#	4953#	4966#	4979#	4992#	5008#	5021#	5037#
		5050#	5066#	5079#	5095#	5108#	5124#	5137#	5150#	5166#	5179#	5195#	5208#	5222#
		5233#	6928#	6947#	6966#	7020#	7039#	7051#	7070#	7082#	7101#	7120#		
\$LSTIN=	000001	960#												
\$LSTTA=	000001	961#												
\$T =	000011	1644#	4611#	4745#	5789#	6050#	6275#	6499#	6684#	6867#	7133#			
. =	034506	951#	1690#	1693#	1777#	2028#	2049#	2065#	3813#	4634	4639	4644	4649	4666
		4671	4678	4683	4691	4696	4701	4706	4714	4719	4725	4730	4736	4741
		4772	4809	4831	4836	4845	4850	4874	4894	4907	4920	4934	4947	4960
		4973	4986	4999	5015	5028	5044	5057	5073	5086	5102	5115	5131	5144
		5157	5173	5186	5202	5215	5229	5788#	5828	5833	5838	5843	5848	5853
		5862	5867	5872	5877	5882	5887	5892	5897	5902	5907	5911	5916	5922
		5927	5938	5943	5949	5954	5966	5971	5980	5985	5991	5996	6002	6007
		6013	6018	6024	6029	6041	6046	6089	6094	6099	6104	6109	6114	6123
		6128	6133	6138	6143	6148	6152	6157	6163	6168	6179	6184	6190	6195
		6203	6208	6214	6219	6224	6229	6235	6240	6246	6251	6257	6262	6266
		6271	6313	6318	6323	6328	6333	6338	6347	6352	6357	6362	6367	6372
		6376	6381	6387	6392	6403	6408	6414	6419	6427	6432	6438	6443	6448
		6453	6459	6464	6470	6475	6481	6486	6490	6495	6532	6537	6554	6559
		6564	6569	6574	6579	6584	6589	6594	6599	6607	6612	6618	6623	6633
		6638	6644	6649	6655	6660	6666	6671	6717	6722	6739	6744	6749	6754
		6759	6764	6769	6774	6779	6784	6791	6796	6802	6807	6816	6821	6827
		6832	6838	6843	6849	6854	6900	6935	6954	7027	7046	7058	7077	7089
		7108	7173	7178	7192	7197	7202	7207	7212	7217	7222	7227	7232	7237
		7241	7246	7252	7257	7262	7267	7273	7278	7287	7292	7298	7303	7312
		7317	7323	7328	7333	7338	7344	7349	7354	7359	7365	7370	7375	7380
		7386	7391	7403	7408	7414	74 9	7431	7436	7445	7450	7456	7461	7467
		7472	7478	7483	7489	7494	7572#							

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- MACRO NAMES

IOSTAR	1#	955#														
KT11	1#	955#														
LASTAD	1#	955#	7583													
MANUAL	1#	955#														
MEMORY	1#	955#														
MSG	4611#	4617#	4745#	4751	5789#	5795	6050#	6056	6275#	6281	6499#	6505	6684#	6690	6867#	
	6873	7133#	7139													
MSBYTE	1#	955#	982#	988	989	990										
MSCHEC	1#	955#	4771#	6899#												
MSCNTO	1#	955#	7516#	7521#	7526#	7532#	7538#									
MSCOUN	1#	955#	3301#	3312#	3323#	3340#	3850#	3869#	3884#	3905#	3914#	3923#	3938#	3946#	3954#	
	3963#	3983#	3991#	3999#	4008#	4028#	4036#	4044#	4063#	4071#	4079#	4098#	4106#	4114#	4153#	
	4161#	4171#	4178#	4188#	4195#	4205#	4212#	4229#	4237#	4247#	4254#	4264#	4278#	4286#	4296#	
	4303#	4313#	4320#	4330#	4337#	4355#	4363#	4373#	4380#							
MSDATA	1#	955#	982#	991	993	995	997	999	1001	1003	1005	1007	1009	1011	1013	
	1015	1017	1019	1021#	1023	1025	1028	1031	1033	1035	1037	1039	1041	1043	1045	
	1047	1049	1051	1053	1055	1057	1059	1061	1063	1065	2045#	2057#				
MSDECR	1#	955#	1117#	1131#	3856#	3895#	3929#	3973#	4018#	4053#	4088#	4123#	4508#	4563#	4563#	
	4581#	4596#	4609#	4743#	5241#	6048#	6273#	6497#	6682#	6865#	6973#	7127#	7131#	7131#	7545#	
	7567#	7581#														
MSDEFA	1#	955#	7516#	7521#	7526#	7532#	7538#									
MSENDE	1#	955#	1117#	1131#	3856#	3895#	3929#	3973#	4018#	4053#	4088#	4123#	4508#	4563#	4581#	
	4596#	4609#	4743#	5241#	6048#	6273#	6497#	6682#	6865#	6973#	7127#	7131#	7496#	7545#	7567#	
	7581#															
MSERRI	1#	955#	4803#	4868#	4888#	4901#	4914#	4928#	4941#	4954#	4967#	4980#	4993#	5009#	5022#	
	5038#	5051#	5067#	5080#	5096#	5109#	5125#	5138#	5151#	5167#	5180#	5196#	5209#	5223#	5234#	
	6929#	6948#	6967#	7021#	7040#	7052#	7071#	7083#	7102#	7121#						
MSESCA	1#	955#	4638#	4639	4648#	4649	4670#	4671	4682#	4683	4695#	4696	4705#	4706	4718#	
	4719	4729#	4730	4740#	4741	4808#	4809	4835#	4836	4849#	4850	4873#	4874	4893#	4894	
	4906#	4907	4919#	4920	4933#	4934	4946#	4947	4959#	4960	4972#	4973	4985#	4986	4998#	
	4999	5014#	5015	5027#	5028	5043#	5044	5056#	5057	5072#	5073	5085#	5086	5101#	5102	
	5114#	5115	5130#	5131	5143#	5144	5156#	5157	5172#	5173	5185#	5186	5201#	5202	5214#	
	5215	5228#	5229	5832#	5833	5842#	5843	5852#	5853	5866#	5867	5877	5886#	5887		
	5896#	5897	5906#	5907	5915#	5916	5926#	5927	5942#	5943	5953#	5954	5970#	5971	5984#	
	5985	5995#	5996	6006#	6007	6017#	6018	6028#	6029	6045#	6046	6093#	6094	6103#	6104	
	6113#	6114	6127#	6128	6137#	6138	6147#	6148	6156#	6157	6167#	6168	6183#	6184	6194#	
	6195	6207#	6208	6218#	6219	6228#	6229	6239#	6240	6250#	6251	6261#	6262	6270#	6271	
	6317#	6318	6327#	6328	6337#	6338	6351#	6352	6361#	6362	6371#	6372	6380#	6381	6391#	
	6392	6407#	6408	6418#	6419	6431#	6432	6442#	6443	6452#	6453	6463#	6464	6474#	6475	
	6485#	6486	6494#	6495	6536#	6537	6558#	6559	6568#	6569	6578#	6579	6588#	6589	6598#	
	6599	6611#	6612	6622#	6623	6637#	6638	6648#	6649	6659#	6660	6670#	6671	6721#	6722	
	6743#	6744	6753#	6754	6763#	6764	6773#	6774	6783#	6784	6795#	6796	6806#	6807	6820#	
	6821	6831#	6832	6842#	6843	6853#	6854	6934#	6935	6953#	6954	7026#	7027	7045#	7046	
	7057#	7058	7076#	7077	7088#	7089	7107#	7108	7177#	7178	7196#	7197	7206#	7207	7216#	
	7217	7226#	7227	7236#	7237	7245#	7246	7256#	7257	7266#	7267	7277#	7278	7291#	7292	
	7302#	7303	7316#	7317	7327#	7328	7337#	7338	7348#	7349	7358#	7359	7369#	7370	7379#	
	7380	7390#	7391	7407#	7408	7418#	7419	7435#	7436	7449#	7450	7460#	7461	7471#	7472	
	7482#	7483	7493#	7494												
MSESCS	1#	955#	4638#	4648#	4670#	4682#	4695#	4705#	4718#	4729#	4740#	4808#	4835#	4849#	4873#	
	4893#	4906#	4919#	4933#	4946#	4959#	4972#	4985#	4998#	5014#	5027#	5043#	5056#	5072#	5085#	
	5101#	5114#	5130#	5143#	5156#	5172#	5185#	5201#	5214#	5228#	5832#	5842#	5852#	5866#	5876#	
	5886#	5896#	5906#	5915#	5926#	5942#	5953#	5970#	5984#	5995#	6006#	6017#	6028#	6045#	6093#	
	6103#	6113#	6127#	6137#	6147#	6156#	6167#	6183#	6194#	6207#	6218#	6228#	6239#	6250#	6261#	
	6270#	6317#	6327#	6337#	6351#	6361#	6371#	6380#	6391#	6407#	6418#	6431#	6442#	6452#	6463#	
	6474#	6485#	6494#	6536#	6558#	6568#	6578#	6588#	6598#	6611#	6622#	6637#	6648#	6659#	6670#	
	6721#	6743#	6753#	6763#	6773#	6783#	6795#	6806#	6820#	6831#	6842#	6853#	6934#	6953#	7026#	

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- MACRO NAMES

	7045#	7057#	7076#	7088#	7107#	7177#	7196#	7206#	7216#	7226#	7236#	7245#	7256#	7266#	7277#
	7291#	7302#	7316#	7327#	7337#	7348#	7358#	7369#	7379#	7390#	7407#	7418#	7435#	7449#	7460#
	7471#	7482#	7493#												
MSEXCP	1#	955#	7516#	7521#	7526#	7532#	7538#								
MSEXIT	1#	955#	4771#	4772	6899#	6900									
MSEXSE	1#	955#	4771#	6899#											
MSEXTJ	1#	955#	4771#	6899#											
MSGEN	1#	955#	958#	982#	991#	993#	995#	997#	999#	1001#	1003#	1005#	1007#	1009#	1011#
	1013#	1015#	1017#	1019#	1021#	1023#	1025#	1028#	1031#	1033#	1035#	1037#	1039#	1041#	1043#
	1045#	1047#	1049#	1051#	1053#	1055#	1057#	1059#	1061#	1063#	1065#	1081#	1102#	1103#	1117#
	1127#	1128#	1131#	1656#	2045#	2057#	3847#	3856#	3862#	3895#	3901#	3929#	3936#	3973#	3981#
	4018#	4026#	4053#	4061#	4088#	4096#	4123#	4400#	4413#	4508#	4527#	4563#	4577#	4581#	4591#
	4596#	4607#	4609#	4627#	4743#	4764#	5241#	5813#	6048#	6074#	6273#	6299#	6497#	6519#	6682#
	6704#	6865#	6893#	6903#	6973#	6982#	7127#	7131#	7160#	7496#	7513#	7546#	7564#	7568#	7587#
MSGENB	1#	955#													
MSGETS	1#	955#	1117#	1131#	3856#	3895#	3929#	3973#	4018#	4053#	4088#	4123#	4405#	4508#	4563#
	4581#	4596#	4609#	4743#	5241#	6048#	6273#	6497#	6682#	6865#	6973#	7127#	7131#	7496#	7545#
	7567#	7581#													
MSGETT	1#	955#	4638#	4648#	4670#	4682#	4695#	4705#	4718#	4729#	4740#	4771#	4808#	4835#	4849#
	4873#	4893#	4906#	4919#	4933#	4946#	4959#	4972#	4985#	4998#	5014#	5027#	5043#	5056#	5072#
	5085#	5101#	5114#	5130#	5143#	5156#	5172#	5185#	5201#	5214#	5228#	5832#	5842#	5852#	5866#
	5876#	5886#	5896#	5906#	5915#	5926#	5942#	5953#	5970#	5984#	5995#	6006#	6017#	6028#	6045#
	6093#	6103#	6113#	6127#	6137#	6147#	6156#	6167#	6183#	6194#	6207#	6218#	6228#	6239#	6250#
	6261#	6270#	6317#	6327#	6337#	6351#	6361#	6371#	6380#	6391#	6407#	6418#	6431#	6442#	6452#
	6463#	6474#	6485#	6494#	6536#	6558#	6568#	6578#	6588#	6598#	6611#	6622#	6637#	6648#	6659#
	6670#	6721#	6743#	6753#	6763#	6773#	6783#	6795#	6806#	6820#	6831#	6842#	6853#	6899#	6934#
	6953#	7026#	7045#	7057#	7076#	7088#	7107#	7177#	7196#	7206#	7216#	7226#	7236#	7245#	7256#
	7266#	7277#	7291#	7302#	7316#	7327#	7337#	7348#	7358#	7369#	7379#	7390#	7407#	7418#	7435#
	7449#	7460#	7471#	7482#	7493#										
MSGNGB	1#	955#	958#	982#	991#	993#	995#	997#	999#	1001#	1003#	1005#	1007#	1009#	1011#
	1013#	1015#	1017#	1019#	1021#	1023#	1025#	1028#	1031#	1033#	1035#	1037#	1039#	1041#	1043#
	1045#	1047#	1049#	1051#	1053#	1055#	1057#	1059#	1061#	1063#	1065#	1080#	1081	1101#	1102
	1103	1126#	1127	1128	1656#	2045#	2057#	3847#	3862#	3901#	3936#	3981#	4026#	4061#	4096#
	4400#	4413#	4527#	4577#	4591#	4607#	7512#	7513	7563#	7564	7584#	7587			
MSGNIN	1#	955#	982#	983	984	985	986	987	988#	989#	990#	991#	992	993#	994
	995#	996	997#	998	999#	1000	1001#	1002	1003#	1004	1005#	1006	1007#	1008	1009#
	1010	1011#	1012	1013#	1014	1015#	1016	1017#	1018	1019#	1020	1021#	1022	1023#	1024
	1025#	1026	1027	1028#	1029	1030#	1031#	1032	1033#	1034	1035#	1036	1037#	1038	1039#
	1040	1041#	1042	1043#	1044	1045#	1046	1047#	1048	1049#	1050	1051#	1052	1053#	1054
	1055#	1056	1057#	1058	1059#	1060	1061#	1062	1063#	1064	1065#	1066	1080#	1082#	1083#
	1084#	1085#	1086#	1087#	1088#	1089#	1090#	1101#	1126#	2045#	2046	2049	2057#	2058	2065
	2103	2104	2105	2106	2147	2148	2149	2150	2194	2195	2196	2197	2265	2266	2267
	2268	2364	2365	2366	2367	2636	2637	2638	2639	2740	2741	2742	2743	2773	2774
	2775	2776	2784	2785	2786	2787	2818	2819	2820	2821	2829	2830	2831	2832	2863
	2864	2865	2866	2874	2875	2876	2877	2908	2909	2910	2911	2919	2920	2921	2922
	2953	2954	2955	2956	2964	2965	2966	2967	2996	2997	2998	2999	3007	3008	3009
	3010	3043	3044	3045	3046	3054	3055	3056	3057	3067	3068	3069	3070	3078	3079
	3080	3081	3110	3111	3112	3113	3122	3123	3124	3125	3301#	3302#	3303	3304#	3305
	3312#	3313#	3314	3315#	3316	3323#	3324#	3325	3326#	3327	3340#	3341#	3342#	3343	3344#
	3345	3495	3496	3497	3498	3508	3509	3510	3511	3534	3535	3536	3537	3544	3545
	3546	3547	3557	3558	3559	3560	3567	3568	3569	3570	3580	3581	3582	3583	3590
	3591	3592	3593	3603	3604	3605	3606	3613	3614	3615	3616	3626	3627	3628	3629
	3636	3637	3638	3639	3850#	3851#	3852	3853#	3854	3857#	3869#	3870#	3871#	3872	3873#
	3874	3884#	3885#	3886#	3887#	3888	3889#	3890	3896#	3905#	3906#	3907#	3908#	3909	3910#
	3911	3914#	3915#	3916#	3917#	3918#	3919	3920#	3921	3923#	3924#	3925	3926#	3927	3930#
	3938#	3939#	3940#	3941#	3942	3943#	3944	3946#	3947#	3948	3949#	3950	3954#	3955#	3956#

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- MACRO NAMES

3957#	3958	3959#	3960	3963#	3964#	3965#	3966#	3967#	3968	3969#	3970	3974#	3983#	3984#
3985#	3986#	3987	3988#	3989	3991#	3992#	3993	3994#	3995	3999#	4000#	4001#	4002#	4003
4004#	4005	4008#	4009#	4010#	4011#	4012#	4013	4014#	4015	4019#	4028#	4029#	4030#	4031#
4032	4033#	4034	4036#	4037#	4038	4039#	4040	4044#	4045#	4046#	4047#	4048	4049#	4050
4054#	4063#	4064#	4065#	4066#	4067	4068#	4069	4071#	4072#	4073#	4074	4075#	4076	4079#
4080#	4081#	4082#	4083#	4084	4085#	4086	4089#	4098#	4099#	4100#	4101#	4102	4103#	4104
4106#	4107#	4108	4109#	4110	4114#	4115#	4116#	4117#	4118	4119#	4120	4124#	4153#	4154#
4155#	4156#	4157	4158#	4159	4161#	4162#	4163#	4164#	4165#	4166#	4167	4168#	4169	4171#
4172#	4173#	4174	4175#	4176	4178#	4179#	4180#	4181#	4182#	4183#	4184	4185#	4186	4188#
4189#	4190#	4191	4192#	4193	4195#	4196#	4197#	4198#	4199#	4200#	4201	4202#	4203	4205#
4206#	4207#	4208	4209#	4210	4212#	4213#	4214#	4215#	4216#	4217#	4218	4219#	4220	4229#
4230#	4231#	4232#	4233	4234#	4235	4237#	4238#	4239#	4240#	4241#	4242#	4243	4244#	4245
4247#	4248#	4249#	4250	4251#	4252	4254#	4255#	4256#	4257#	4258#	4259#	4260	4261#	4262
4264#	4265#	4266	4267#	4268	4278#	4279#	4280#	4281#	4282	4283#	4284	4284#	4287#	4288#
4289#	4290#	4291#	4292	4293#	4294	4296#	4297#	4298#	4299	4300#	4301	4303#	4304#	4305#
4306#	4307#	4308#	4309	4310#	4311	4313#	4314#	4315#	4316	4317#	4318	4320#	4321#	4322#
4323#	4324#	4325#	4326	4327#	4328	4330#	4331#	4332#	4333	4334#	4335	4337#	4338#	4339#
4340#	4341#	4342#	4343	4344#	4345	4355#	4356#	4357#	4358#	4359	4360#	4361	4363#	4364#
4365#	4366#	4367#	4368#	4369	4370#	4371	4373#	4374#	4375#	4376	4377#	4378	4380#	4381#
4382#	4383#	4384#	4385#	4386	4387#	4388	4433#	4434#	4436#	4440#	4441#	4443#	4447#	4448#
4450#	4454#	4455#	4457#	4475#	4476#	4477#	4479#	4505#	4509#	4530#	4531#	4532#	4533#	4534#
4535	4552#	4553#	4557#	4558#	4564#	4582#	4594#	4597#	4610#	4636#	4638#	4639#	4646#	4648#
4649#	4668#	4670#	4671#	4680#	4682#	4683#	4693#	4695#	4696#	4703#	4705#	4706#	4716#	4718#
4719#	4727#	4729#	4730#	4738#	4740#	4741#	4744#	4771#	4772#	4803#	4804#	4805#	4806#	4808#
4809#	4833#	4835#	4836#	4847#	4849#	4850#	4868#	4869#	4870#	4871#	4873#	4874#	4888#	4889#
4890#	4891#	4893#	4894#	4901#	4902#	4903#	4904#	4906#	4907#	4914#	4915#	4916#	4917#	4919#
4920#	4928#	4929#	4930#	4931#	4933#	4934#	4941#	4942#	4943#	4944#	4946#	4947#	4954#	4955#
4956#	4957#	4959#	4960#	4967#	4968#	4969#	4970#	4972#	4973#	4980#	4981#	4982#	4983#	4985#
4986#	4993#	4994#	4995#	4996#	4998#	4999#	5009#	5010#	5011#	5012#	5014#	5015#	5022#	5023#
5024#	5025#	5027#	5028#	5038#	5039#	5040#	5041#	5043#	5044#	5051#	5052#	5053#	5054#	5056#
5057#	5067#	5068#	5069#	5070#	5072#	5073#	5080#	5081#	5082#	5083#	5085#	5086#	5096#	5097#
5098#	5099#	5101#	5102#	5109#	5110#	5111#	5112#	5114#	5115#	5125#	5126#	5127#	5128#	5130#
5131#	5138#	5139#	5140#	5141#	5143#	5144#	5151#	5152#	5153#	5154#	5156#	5157#	5167#	5168#
5169#	5170#	5172#	5173#	5180#	5181#	5182#	5183#	5185#	5186#	5196#	5197#	5198#	5199#	5201#
5202#	5209#	5210#	5211#	5212#	5214#	5215#	5223#	5224#	5225#	5226#	5228#	5229#	5234#	5235#
5236#	5237#	5242#	5830#	5832#	5833#	5840#	5842#	5843#	5850#	5852#	5853#	5864#	5866#	5867#
5874#	5876#	5877#	5884#	5886#	5887#	5894#	5896#	5897#	5904#	5906#	5907#	5913#	5915#	5916#
5924#	5926#	5927#	5940#	5942#	5943#	5951#	5953#	5954#	5968#	5970#	5971#	5982#	5984#	5985#
5993#	5995#	5996#	6004#	6006#	6007#	6015#	6017#	6018#	6026#	6028#	6029#	6043#	6045#	6046#
6049#	6091#	6093#	6094#	6101#	6103#	6104#	6111#	6113#	6114#	6125#	6127#	6128#	6135#	6137#
6138#	6145#	6147#	6148#	6154#	6156#	6157#	6165#	6167#	6168#	6181#	6183#	6184#	6192#	6194#
6195#	6205#	6207#	6208#	6216#	6218#	6219#	6226#	6228#	6229#	6237#	6239#	6240#	6248#	6250#
6251#	6259#	6261#	6262#	6268#	6270#	6271#	6274#	6315#	6317#	6318#	6325#	6327#	6328#	6335#
6337#	6338#	6349#	6351#	6352#	6359#	6361#	6362#	6369#	6371#	6372#	6378#	6380#	6381#	6389#
6391#	6392#	6405#	6407#	6408#	6416#	6418#	6419#	6429#	6431#	6432#	6440#	6442#	6443#	6450#
6452#	6453#	6461#	6463#	6464#	6472#	6474#	6475#	6483#	6485#	6486#	6492#	6494#	6495#	6498#
6534#	6536#	6537#	6556#	6558#	6559#	6566#	6568#	6569#	6576#	6578#	6579#	6586#	6587#	6589#
6596#	6598#	6599#	6609#	6611#	6612#	6620#	6622#	6623#	6635#	6637#	6638#	6646#	6648#	6649#
6657#	6659#	6660#	6668#	6670#	6671#	6683#	6719#	6721#	6722#	6741#	6743#	6744#	6751#	6753#
6754#	6761#	6763#	6764#	6771#	6773#	6774#	6781#	6783#	6784#	6793#	6795#	6796#	6804#	6806#
6807#	6818#	6820#	6821#	6829#	6831#	6832#	6840#	6842#	6843#	6851#	6853#	6854#	6866#	6899#
6900#	6904#	6929#	6930#	6931#	6932#	6934#	6935#	6948#	6949#	6950#	6953#	6954#	6954#	6967#
6968#	6969#	6970#	6974#	6983#	7021#	7022#	7023#	7024#	7026#	7027#	7027#	7041#	7042#	7043#
7045#	7046#	7052#	7053#	7054#	7055#	7057#	7058#	7071#	7072#	7073#	7073#	7076#	7077#	7083#
7084#	7085#	7086#	7088#	7089#	7102#	7103#	7104#	7105#	7107#	7108#	7121#	7122#	7123#	7124#
7128#	7132#	7175#	7177#	7178#	7194#	7196#	7197#	7204#	7206#	7207#	7214#	7216#	7217#	7224#

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- MACRO NAMES

	7226#	7227#	7234#	7236#	7237#	7243#	7245#	7246#	7254#	7256#	7257#	7264#	7266#	7267#	7275#
	7277#	7278#	7289#	7291#	7292#	7300#	7302#	7303#	7314#	7316#	7317#	7325#	7327#	7328#	7335#
	7337#	7338#	7346#	7348#	7349#	7356#	7358#	7359#	7367#	7369#	7370#	7377#	7379#	7380#	7388#
	7390#	7391#	7405#	7407#	7408#	7416#	7418#	7419#	7433#	7435#	7436#	7447#	7449#	7450#	7458#
	7460#	7461#	7469#	7471#	7472#	7480#	7482#	7483#	7491#	7493#	7494#	7497#	7512#	7516#	7517
	7518	7519	7521#	7522	7523	7524	7526#	7527	7528	7529	7530	7532#	7533	7534	7535
	7536	7538#	7539	7540	7541	7542	7545#	7563#	7567#	7584#	7585#	7586#			
MSGNLS	1#	955#													
MSGNSU	1#	955#	6903#	6982#											
MSGNTA	1#	955#	1117#	1131#	3856#	3895#	3929#	3973#	4018#	4053#	4088#	4123#	4508#	4563#	4581#
	4596#	4609#	4743#	5241#	6048#	6273#	6497#	6682#	6865#	6973#	7127#	7131#	7496#	7545#	7546
	7567#	7568													
MSGNTE	1#	955#	4627#	4764#	5813#	6074#	6299#	6519#	6704#	6893#	7160#				
MSHAPT	1#	955#	982#												
MSHINAP	1#	955#	982#	1021											
MSINCR	1#	955#	958#	1101#	1126#	3304#	3315#	3326#	3344#	3847#	3853#	3857#	3862#	3873#	3889#
	3896#	3901#	3910#	3920#	3926#	3930#	3936#	3943#	3949#	3959#	3969#	3974#	3981#	3988#	3994#
	4004#	4014#	4019#	4026#	4033#	4039#	4049#	4054#	4061#	4068#	4075#	4085#	4089#	4096#	4103#
	4109#	4119#	4124#	4158#	4168#	4175#	4185#	4192#	4202#	4209#	4219#	4234#	4244#	4251#	4261#
	4267#	4283#	4293#	4300#	4310#	4317#	4327#	4334#	4344#	4360#	4370#	4377#	4387#	4400#	4413#
	4434#	4441#	4448#	4455#	4476#	4505#	4509#	4527#	4534#	4553#	4558#	4564#	4577#	4582#	4591#
	4594#	4597#	4607#	4610#	4627#	4628#	4636#	4638#	4646#	4648#	4668#	4670#	4680#	4682#	4693#
	4695#	4703#	4705#	4716#	4718#	4727#	4729#	4738#	4740#	4744#	4764#	4765#	4771#	4803#	4808#
	4833#	4835#	4847#	4849#	4868#	4873#	4888#	4893#	4901#	4906#	4914#	4919#	4928#	4933#	4941#
	4946#	4954#	4959#	4967#	4972#	4980#	4985#	4993#	4998#	5009#	5014#	5022#	5027#	5038#	5043#
	5051#	5056#	5067#	5072#	5080#	5085#	5096#	5101#	5109#	5114#	5125#	5130#	5138#	5143#	5151#
	5156#	5167#	5172#	5180#	5185#	5196#	5201#	5209#	5214#	5223#	5228#	5234#	5242#	5813#	5814#
	5830#	5832#	5840#	5842#	5850#	5852#	5864#	5866#	5874#	5876#	5884#	5886#	5894#	5896#	5904#
	5906#	5913#	5915#	5924#	5926#	5940#	5942#	5951#	5953#	5968#	5970#	5982#	5984#	5993#	5995#
	6004#	6006#	6015#	6017#	6026#	6028#	6043#	6045#	6049#	6074#	6075#	6091#	6093#	6101#	6103#
	6111#	6113#	6125#	6127#	6135#	6137#	6145#	6147#	6154#	6156#	6165#	6167#	6181#	6183#	6192#
	6194#	6205#	6207#	6216#	6218#	6226#	6228#	6237#	6239#	6248#	6250#	6259#	6261#	6268#	6270#
	6274#	6299#	6300#	6315#	6317#	6325#	6327#	6335#	6337#	6349#	6351#	6359#	6361#	6369#	6371#
	6378#	6380#	6389#	6391#	6405#	6407#	6416#	6418#	6429#	6431#	6440#	6442#	6450#	6452#	6461#
	6463#	6472#	6474#	6483#	6485#	6492#	6494#	6498#	6519#	6520#	6534#	6536#	6556#	6558#	6566#
	6568#	6576#	6578#	6586#	6588#	6596#	6598#	6609#	6611#	6620#	6622#	6635#	6637#	6646#	6648#
	6657#	6659#	6668#	6670#	6683#	6704#	6705#	6719#	6721#	6741#	6743#	6751#	6753#	6761#	6763#
	6771#	6773#	6781#	6783#	6793#	6795#	6804#	6806#	6818#	6820#	6829#	6831#	6840#	6842#	6851#
	6853#	6866#	6893#	6894#	6899#	6903#	6904#	6929#	6934#	6948#	6953#	6967#	6974#	6982#	6983#
	7021#	7026#	7040#	7045#	7052#	7057#	7071#	7076#	7083#	7088#	7102#	7107#	7121#	7128#	7132#
	7160#	7161#	7175#	7177#	7194#	7196#	7204#	7206	7214#	7216#	7224#	7226#	7234#	7236#	7243#
	7245#	7254#	7256#	7264#	7266#	7275#	7277#	72P' #	7291#	7300#	7302#	7314#	7316#	7325#	7327#
	7335#	7337#	7346#	7348#	7356#	7358#	7367#	7369#	7377#	7379#	7388#	7390#	7405#	7407#	7416#
	7418#	7433#	7435#	7447#	7449#	7458#	7460#	7469#	7471#	7480#	7482#	7491#	7493#	7497#	7512#
	7563#														
MSIOSE	1#	955#													
MSLDRO	1#	955#	4433#	4440#	4447#	4454#	4475#	4552#	4557#						
MSMASK	1#	955#													
MSMCHI	1#	955#													
MSMCLO	1#	955#													
MSMSK1	1#	955#													
MSPOP	1#	955#	1117#	1131#	3856#	3895#	3929#	3973#	4018#	4053#	4088#	4123#	4405#	4508#	4563#
	4581#	4596#	4609#	4743#	5241#	6048#	6273#	6497#	6682#	6865#	6973#	7127#	7131#	7496#	7545#
	7567#	7581#													
MSPRIN	1#	955#	3301#	3312#	3323#	3340#	3850#	3869#	3884#	3905#	3914#	3923#	3938#	3946#	3954#
	3963#	3983#	3991#	3999#	4008#	4028#	4036#	4044#	4063#	4071#	4079#	4098#	4106#	4114#	4153#

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- MACRO NAMES

	4161#	4171#	4178#	4188#	4195#	4205#	4212#	4229#	4237#	4247#	4254#	4264#	4278#	4286#	4296#
	4303#	4313#	4320#	4330#	4337#	4355#	4363#	4373#	4380#						
MSRUSH	1#	955#	958#	1101#	1126#	3847#	3862#	3901#	3936#	3981#	4026#	4061#	4096#	4400#	4413#
	4527#	4577#	4591#	4607#	4627#	4628	4764#	4765	5813#	5814	6074#	6075	6299#	6300	6519#
	6520	6704#	6705	6893#	6894	6903#	6904	6982#	6983	7160#	7161	7512#	7563#		
MSRPUT	1#	955#	3301#	3312#	3323#	3340#	3850#	3869#	3884#	3905#	3914#	3923#	3938#	3946#	3954#
	3963#	3983#	3991#	3999#	4008#	4028#	4036#	4044#	4063#	4071#	4079#	4098#	4106#	4114#	4153#
	4161#	4171#	4178#	4188#	4195#	4205#	4212#	4229#	4237#	4247#	4254#	4264#	4278#	4286#	4296#
	4303#	4313#	4320#	4330#	4337#	4355#	4363#	4373#	4380#	4530#					
MSRPUT1	1#	955#	3301#	3302	3312#	3313	3323#	3324	3340#	3341	3342	3850#	3851	3869#	3870
	3871	3884#	3885	3886	3887	3905#	3906	3907	3908	3914#	3915	3916	3917	3918	3923#
	3924	3938#	3939	3940	3941	3946#	3947	3954#	3955	3956	3957	3963#	3964	3965	3966
	3967	3983#	3984	3985	3986	3991#	3992	3999#	4000	4001	4002	4008#	4009	4010	4011
	4012	4028#	4029	4030	4031	4036#	4037	4044#	4045	4046	4047	4063#	4064	4065	4066
	4071#	4072	4073	4079#	4080	4081	4082	4083	4098#	4099	4100	4101	4106#	4107	4114#
	4115	4116	4117	4153#	4154	4155	4156	4161#	4162	4163	4164	4165	4166	4171#	4172
	4173	4178#	4179	4180	4181	4182	4183	4188#	4189	4190	4195#	4196	4197	4198	4199
	4200	4205#	4206	4207	4212#	4213	4214	4215	4216	4217	4229#	4230	4231	4232	4237#
	4238	4239	4240	4241	4242	4247#	4248	4249	4254#	4255	4256	4257	4258	4259	4264#
	4265	4278#	4279	4280	4281	4286#	4287	4288	4289	4290	4291	4296#	4297	4298	4303#
	4304	4305	4306	4307	4308	4313#	4314	4315	4320#	4321	4322	4323	4324	4325	4330#
	4331	4332	4337#	4338	4339	4340	4341	4342	4355#	4356	4357	4358	4363#	4364	4365
	4366	4367	4368	4373#	4374	4375	4380#	4381	4382	4383	4384	4385	4530#	4531	4532
	4533														
MSRADI	1#	955#	7516#	7521#	7526#	7532#	7538#								
MSRPRO	1#	955#													
MSRPRO	1#	955#	4475#	4477											
MSSETS	1#	955#	958#	1101#	1126#	3847#	3862#	3901#	3936#	3981#	4026#	4061#	4096#	4400#	4413#
	4527#	4577#	4591#	4607#	4628#	4765#	5814#	6075#	6300#	6520#	6705#	6894#	6904#	6983#	7161#
	7512#	7563#													
MSSTAR	1#	955#													
MSVC	1#	955#	3301#	3304	3312#	3315	3323#	3326	3340#	3344	3850#	3853	3856#	3857	3869#
	3873	3884#	3889	3895#	3896	3905#	3910	3914#	3920	3923#	3926	3929#	3930	3938#	3943
	3946#	3949	3954#	3959	3963#	3969	3973#	3974	3983#	3988	3991#	3994	3999#	4004	4008#
	4014	4018#	4019	4028#	4033	4036#	4039	4044#	4049	4053#	4054	4063#	4068	4071#	4075
	4079#	4085	4088#	4089	4098#	4103	4106#	4109	4114#	4119	4123#	4124	4153#	4158	4161#
	4168	4171#	4175	4178#	4185	4188#	4192	4195#	4202	4205#	4209	4212#	4219	4229#	4234
	4237#	4244	4247#	4251	4254#	4261	4264#	4267	4278#	4283	4286#	4293	4296#	4300	4303#
	4310	4313#	4317	4320#	4327	4330#	4334	4337#	4344	4355#	4360	4363#	4370	4373#	4377
	4380#	4387	4433#	4434	4440#	4441	4447#	4448	4454#	4455	4475#	4476	4505#	4508#	4509
	4530#	4534	4552#	4553	4557#	4558	4563#	4564	4581#	4582	4594#	4596#	4597	4609#	4610
	4636#	4638#	4646#	4648#	4668#	4670#	4680#	4682#	4693#	4695#	4703#	4705#	4716#	4718#	4727#
	4729#	4738#	4740#	4743#	4744	4771#	4803	4808#	4833#	4835#	4847#	4849#	4868	4873#	4888
	4893#	4901	4906#	4914	4919#	4928	4933#	4941	4946#	4954	4959#	4967	4972#	4980	4985#
	4993	4998#	5009	5014#	5022	5027#	5038	5043#	5051	5056#	5067	5072#	5080	5085#	5096
	5101#	5109	5114#	5125	5130#	5138	5143#	5151	5156#	5167	5172#	5180	5185#	5196	5201#
	5209	5214#	5223	5228#	5234	5241#	5242	5830#	5832#	5840#	5842#	5850#	5852#	5864#	5866#
	5874#	5876#	5884#	5886#	5894#	5896#	5904#	5906#	5913#	5915#	5924#	5926#	5940#	5942#	5951#
	5953#	5968#	5970#	5982#	5984#	5993#	5995#	6004#	6006#	6015#	6017#	6026#	6028#	6043#	6045#
	6048#	6049	6091#	6093#	6101#	6103#	6111#	6113#	6125#	6127#	6135#	6137#	6145#	6147#	6154#
	6156#	6165#	6167#	6181#	6183#	6192#	6194#	6205#	6207#	6216#	6218#	6226#	6228#	6237#	6239#
	6248#	6250#	6259#	6261#	6268#	6270#	6273#	6274	6315#	6317#	6325#	6327#	6335#	6337#	6349#
	6351#	6359#	6361#	6369#	6371#	6378#	6380#	6389#	6391#	6405#	6407#	6416#	6418#	6429#	6431#
	6440#	6442#	6450#	6452#	6461#	6463#	6472#	6474#	6483#	6485#	6492#	6494#	6497#	6498	6534#
	6536#	6556#	6558#	6566#	6568#	6576#	6578#	6586#	6588#	6596#	6598#	6609#	6611#	6620#	6622#
	6635#	6637#	6646#	6648#	6657#	6659#	6668#	6670#	6682#	6683	6719#	6721#	6741#	6743#	6751#

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- MACRO NAMES

6753#	6761#	6763#	6771#	6773#	6781#	6783#	6793#	6795#	6804#	6806#	6818#	6820#	6829#	6831#	
6840#	6842#	6851#	6853#	6865#	6866	6899#	6903#	6904	6929	6934#	6948	6953#	6967	6973#	
6974	6982#	6983	7021	7026#	7040	7045#	7052	7057#	7071	7076#	7083	7088#	7102	7107#	
7121	7127#	7128	7131#	7132	7175#	7177#	7194#	7196#	7204#	7206#	7214#	7216#	7224#	7226#	
7234#	7236#	7243#	7245#	7254#	7256#	7264#	7266#	7275#	7277#	7289#	7291#	7300#	7302#	7314#	
7316#	7325#	7327#	7335#	7337#	7346#	7348#	7356#	7358#	7367#	7369#	7377#	7379#	7388#	7390#	
7405#	7407#	7416#	7418#	7433#	7435#	7447#	7449#	7458#	7460#	7469#	7471#	7480#	7482#	7491#	
7493#	7496#	7497													
MSTLAB	1#	955#	3304#	3315#	3326#	3344#	3853#	3857#	3873#	3889#	3896#	3910#	3920#	3926#	3930#
3943#	3949#	3959#	3969#	3974#	3988#	3994#	4004#	4014#	4019#	4033#	4039#	4049#	4054#	4068#	
4075#	4085#	4089#	4103#	4109#	4119#	4124#	4158#	4168#	4175#	4185#	4192#	4202#	4209#	4219#	
4234#	4244#	4251#	4261#	4267#	4283#	4293#	4300#	4310#	4317#	4327#	4334#	4344#	4360#	4370#	
4377#	4387#	4434#	4441#	4448#	4455#	4476#	4505#	4509#	4534#	4553#	4558#	4564#	4582#	4594#	
4597#	4610#	4636#	4638#	4646#	4648#	4668#	4670#	4680#	4682#	4693#	4695#	4703#	4705#	4716#	
4718#	4727#	4729#	4738#	4740#	4744#	4771#	4803#	4808#	4833#	4835#	4847#	4849#	4868#	4873#	
4888#	4893#	4901#	4906#	4914#	4919#	4928#	4933#	4941#	4946#	4954#	4959#	4967#	4972#	4980#	
4985#	4993#	4998#	5009#	5014#	5022#	5027#	5038#	5043#	5051#	5056#	5067#	5072#	5080#	5085#	
5096#	5101#	5109#	5114#	5125#	5130#	5138#	5143#	5151#	5156#	5167#	5172#	5180#	5185#	5196#	
5201#	5209#	5214#	5223#	5228#	5234#	5242#	5830#	5832#	5840#	5842#	5850#	5852#	5864#	5866#	
5874#	5876#	5884#	5886#	5894#	5896#	5904#	5906#	5913#	5915#	5924#	5926#	5940#	5942#	5951#	
5953#	5968#	5970#	5982#	5984#	5993#	5995#	6004#	6006#	6015#	6017#	6026#	6028#	6043#	6045#	
6049#	6091#	6093#	6101#	6103#	6111#	6113#	6125#	6127#	6135#	6137#	6145#	6147#	6154#	6156#	
6165#	6167#	6181#	6183#	6192#	6194#	6205#	6207#	6216#	6218#	6226#	6228#	6237#	6239#	6248#	
6250#	6259#	6261#	6268#	6270#	6274#	6315#	6317#	6325#	6327#	6335#	6337#	6349#	6351#	6359#	
6361#	6369#	6371#	6378#	6380#	6389#	6391#	6405#	6407#	6416#	6418#	6429#	6431#	6440#	6442#	
6450#	6452#	6461#	6463#	6472#	6474#	6483#	6485#	6492#	6494#	6498#	6534#	6536#	6556#	6558#	
6566#	6568#	6576#	6578#	6586#	6588#	6596#	6598#	6609#	6611#	6620#	6622#	6635#	6637#	6646#	
6648#	6657#	6659#	6668#	6670#	6683#	6719#	6721#	6741#	6743#	6751#	6753#	6761#	6763#	6771#	
6773#	6781#	6783#	6793#	6795#	6804#	6806#	6818#	6820#	6829#	6831#	6840#	6842#	6851#	6853#	
6866#	6899#	6904#	6929#	6934#	6948#	6953#	6967#	6974#	6983#	7021#	7026#	7040#	7045#	7052#	
7057#	7071#	7076#	7083#	7088#	7102#	7107#	7121#	7128#	7132#	7175#	7177#	7194#	7196#	7204#	
7206#	7214#	7216#	7224#	7226#	7234#	7236#	7243#	7245#	7254#	7256#	7264#	7266#	7275#	7277#	
7289#	7291#	7300#	7302#	7314#	7316#	7325#	7327#	7335#	7337#	7346#	7348#	7356#	7358#	7367#	
7369#	7377#	7379#	7388#	7390#	7405#	7407#	7416#	7418#	7433#	7435#	7447#	7449#	7458#	7460#	
7469#	7471#	7480#	7482#	7491#	7493#	7497#									
MSTSTL	1#	955#	3304#	3315#	3326#	3344#	3853#	3857#	3873#	3889#	3896#	3910#	3920#	3926#	3930#
3943#	3949#	3959#	3969#	3974#	3988#	3994#	4004#	4014#	4019#	4033#	4039#	4049#	4054#	4068#	
4075#	4085#	4089#	4103#	4109#	4119#	4124#	4158#	4168#	4175#	4185#	4192#	4202#	4209#	4219#	
4234#	4244#	4251#	4261#	4267#	4283#	4293#	4300#	4310#	4317#	4327#	4334#	4344#	4360#	4370#	
4377#	4387#	4434#	4441#	4448#	4455#	4476#	4505#	4509#	4534#	4553#	4558#	4564#	4582#	4594#	
4597#	4610#	4636#	4638#	4646#	4648#	4668#	4670#	4680#	4682#	4693#	4695#	4703#	4705#	4716#	
4718#	4727#	4729#	4738#	4740#	4744#	4771#	4803#	4808#	4833#	4835#	4847#	4849#	4868#	4873#	
4888#	4893#	4901#	4906#	4914#	4919#	4928#	4933#	4941#	4946#	4954#	4959#	4967#	4972#	4980#	
4985#	4993#	4998#	5009#	5014#	5022#	5027#	5038#	5043#	5051#	5056#	5067#	5072#	5080#	5085#	
5096#	5101#	5109#	5114#	5125#	5130#	5138#	5143#	5151#	5156#	5167#	5172#	5180#	5185#	5196#	
5201#	5209#	5214#	5223#	5228#	5234#	5242#	5830#	5832#	5840#	5842#	5850#	5852#	5864#	5866#	
5874#	5876#	5884#	5886#	5894#	5896#	5904#	5906#	5913#	5915#	5924#	5926#	5940#	5942#	5951#	
5953#	5968#	5970#	5982#	5984#	5993#	5995#	6004#	6006#	6015#	6017#	6026#	6028#	6043#	6045#	
6049#	6091#	6093#	6101#	6103#	6111#	6113#	6125#	6127#	6135#	6137#	6145#	6147#	6154#	6156#	
6165#	6167#	6181#	6183#	6192#	6194#	6205#	6207#	6216#	6218#	6226#	6228#	6237#	6239#	6248#	
6250#	6259#	6261#	6268#	6270#	6274#	6315#	6317#	6325#	6327#	6335#	6337#	6349#	6351#	6359#	
6361#	6369#	6371#	6378#	6380#	6389#	6391#	6405#	6407#	6416#	6418#	6429#	6431#	6440#	6442#	
6450#	6452#	6461#	6463#	6472#	6474#	6483#	6485#	6492#	6494#	6498#	6534#	6536#	6556#	6558#	
6566#	6568#	6576#	6578#	6586#	6588#	6596#	6598#	6609#	6611#	6620#	6622#	6635#	6637#	6646#	
6648#	6657#	6659#	6668#	6670#	6683#	6719#	6721#	6741#	6743#	6751#	6753#	6761#	6763#	6771#	
6773#	6781#	6783#	6793#	6795#	6804#	6806#	6818#	6820#	6829#	6831#	6840#	6842#	6851#	6853#	

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- MACRO NAMES

	6866#	6899#	6904#	6929#	6934#	6948#	6953#	6967#	6974#	6983#	7021#	7026#	7040#	7045#	7052#
	7057#	7071#	7076#	7083#	7088#	7102#	7107#	7121#	7128#	7132#	7175#	7177#	7194#	7196#	7204#
	7206#	7214#	7216#	7224#	7226#	7234#	7236#	7243#	7245#	7254#	7256#	7264#	7266#	7275#	7277#
	7289#	7291#	7300#	7302#	7314#	7316#	7325#	7327#	7335#	7337#	7346#	7348#	7356#	7358#	7367#
	7369#	7377#	7379#	7388#	7390#	7405#	7407#	7416#	7418#	7433#	7435#	7447#	7449#	7458#	7460#
	7469#	7471#	7480#	7482#	7491#	7493#	7497#								
MSWORD	1#	955#	1021#	1030	1080#	1082	1083	1084	1085	1086	1087	1088	1089	1090	4771#
	4803#	4804	4805	4806	4868#	4869	4870	4871	4888#	4889	4890	4891	4901#	4902	4903
	4904	4914#	4915	4916	4917	4928#	4929	4930	4931	4941#	4942	4943	4944	4954#	4955
	4956	4957	4967#	4968	4969	4970	4980#	4981	4982	4983	4993#	4994	4995	4996	5009#
	5010	5011	5012	5022#	5023	5024	5025	5038#	5039	5040	5041	5051#	5052	5053	5054
	5067#	5068	5069	5070	5080#	5081	5082	5083	5096#	5097	5098	5099	5109#	5110	5111
	5112	5125#	5126	5127	5128	5138#	5139	5140	5141	5151#	5152	5153	5154	5167#	5168
	5169	5170	5180#	5181	5182	5183	5196#	5197	5198	5199	5209#	5210	5211	5212	5223#
	5224	5225	5226	5234#	5235	5236	5237	6899#	6929#	6930	6931	6932	6948#	6949	6950
	6951	6967#	6968	6969	6970	7021#	7022	7023	7024	7040#	7041	7042	7043	7052#	7053
	7054	7055	7071#	7072	7073	7074	7083#	7084	7085	7086	7102#	7103	7104	7105	7121#
	7122	7123	7124	7516#	7521#	7526#	7532#	7538#	7585	7586					
MSXFER	1#	955#													
NEWTST	1644#	4611	4745	5789	6050	6275	6499	6684	6867	7133					
NTST	1644#	4611	4745	5789	6050	6275	6499	6684	6867	7133					
OPEN	1#	955#													
POINTE	1#	955#	978												
PRINTB	1#	955#	3849	3904	3913	3922	3937	3945	3953	3962	3982	3990	3998	4007	4027
	4035	4043	4062	4070	4078	4097	4105	4113	4263						
PRINTF	1#	955#	3300	3311	3322	3339									
PRINTS	1#	955#													
PRINTX	1#	955#	3868	3883	4152	4160	4170	4177	4187	4194	4204	4211	4228	4236	4246
	4253	4277	4285	4295	4302	4312	4319	4329	4336	4354	4362	4372	4379		
READBU	1#	955#													
READEF	1#	955#	4432	4439	4446	4453									
RFLAGS	1#	955#													
SETDF	1644#	2103	2147	2194	2265	2364	2636	2740	2773	2784	2818	2829	2863	2874	2908
	2919	2953	2964	2996	3007	3043	3054	3067	3078	3110	3122	3495	3508	3534	3544
	3557	3567	3580	3590	3603	3613	3626	3636							
SETHRD	1644#														
SETPRI	1#	955#													
SETSF	1644#														
SETSFT	1644#														
SETVEC	1#	955#	4529												
SLASH	1#	955#	1072	1076											
STARS	1#	955#													
SVC	1#	953#	954												
T\$GEN	1644#	2103	2147	2194	2265	2364	2636	2740	2773	2784	2818	2829	2863	2874	2908
	2919	2953	2964	2996	3007	3043	3054	3067	3078	3110	3122	3495	3508	3534	3544
	3557	3567	3580	3590	3603	3613	3626	3636							
XFER	1#	955#	4771#	4799#											
XFERF	1#	955#													
XFERT	1#	955#													
\$GEDF	1644#	4802	4867	4887	4900	4913	4927	4940	4953	4966	4979	4992	5008	5021	5037
	5050	5066	5079	5095	5108	5124	5137	5150	5166	5179	5195	5208	5222	5233	6928
	6947	6966	7020	7039	7051	7070	7082	7101	7120						
\$GEHRD	1644#														
\$GESF	1644#														
\$GESFT	1644#														
\$GTDF	1644#	2102	2146	2193	2264	2363	2635	2739	2772	2783	2817	2828	2862	2873	2907

CVDMEA.P11 10-DEC-80 09:16

CROSS REFERENCE TABLE -- MACRO NAMES

	2918	2952	2963	2995	3006	3042	3053	3066	3077	3109	3121	3494	3507	3533	3543
	3556	3566	3579	3589	3602	3612	3625	3635							
SGTHRD	1644#														
SGTSF	1644#														
SGTSFT	1644#														

. ABS. 034506 000

ERRORS DETECTED: 0

CVDMEA.BIN,CVDMEA.SEQ/CRF/SOL=SVC34R.MAC,CVDMEA.P11
 RUN-TIME: 32 41 4 SECONDS
 RUN-TIME RATIO: 164/78=2.0
 CORE USED: 21K (41 PAGES)