

DMV11
M8053 M8064

DMV11 LINE UNIT DIAG 1
CVD MCAO

AH-F268A MC
FICHE 1 OF 2

MAY 98
COPYR 98
MADE IN USA



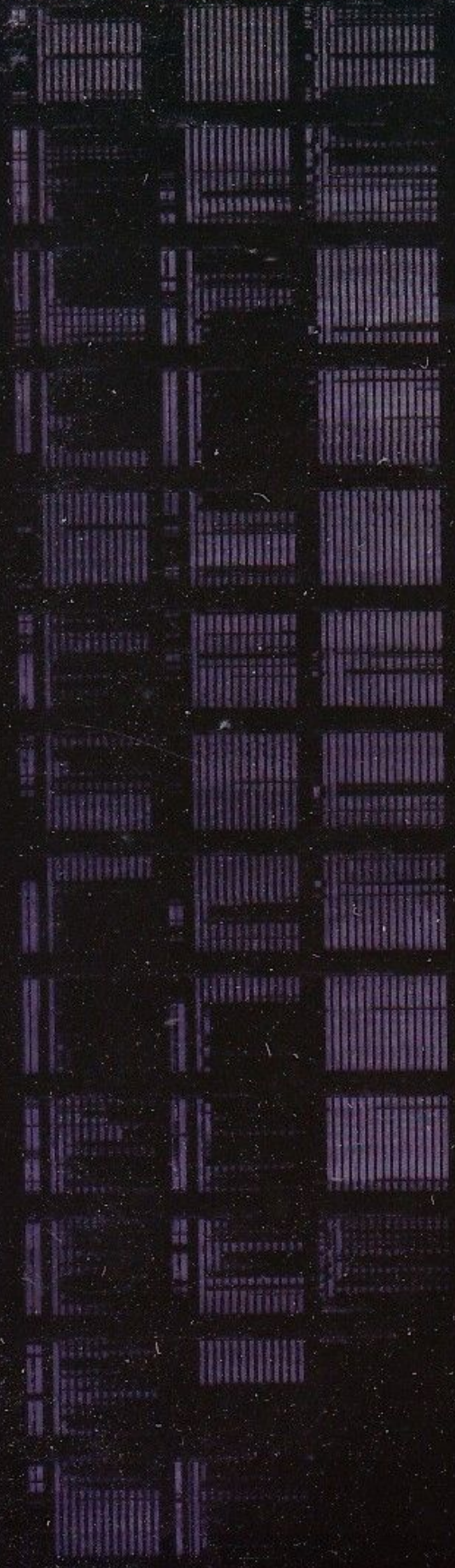
A microfiche card containing a grid of 100 frames (10 rows by 10 columns). Each frame contains a small, high-contrast image of a technical diagram or data table. The diagrams appear to be related to a line unit, possibly showing internal components, wiring, or test procedures. The text within the frames is too small to read clearly but seems to include labels and numerical data. The overall layout is a dense grid of technical information.

DMV11 MB053 MB064

DMV11 LINE UNIT DIAG 1
CVDMCA0

AH-F268A-MC
FICHE 2 OF 2

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
41
42
43

.TITLE CVDMAO DMV11 LINE UNIT DIAG1
.SBTTL PROGRAM DOCUMENT
.REM ^

IDENTIFICATION

PRODUCT CODE: AC-F267A-MC
PRODUCT NAME: CVDMAO DMV-11 LINE UNIT STATIC DIAGNOSTIC PART 1
PRODUCT DATE: JANUARY 1981
MAINTAINER: DIAGNOSTICS MERRIMACK CC:38P
AUTHORS: CHRIS BRIENEN
 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	

CVDMCA.P11

10-DEC-80 09:14

PROGRAM DOCUMENT

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

HISTORY

REV

DATE

REASON

0

14-JAN-81

INITIAL RELEASE

CVDMA.P11

10-DEC-80 09:14

PROGRAM DOCUMENT

CONTENTS

- 1.0 INTRODUCTION
- 2.0 HARDWARE REQUIREMENTS
- 3.0 PRELIMINARY PROGRAM REQUIREMENTS
- 4.0 GENERAL PROGRAM CONSIDERATIONS
 - 4.1 DIAGNOSTIC SUPERVISOR
 - 4.2 EXECUTION TIME
 - 4.3 XXDP+
 - 4.4 ACT/SLIDE
 - 4.5 APT
 - 4.6 MEMORY MANAGEMENT
 - 4.7 ERROR LOGGING
- 5.0 PROGRAM LOAD MEDIA
- 6.0 OPERATING INSTRUCTIONS
 - 6.1 LOADING AND STARTING PROCEDURES
 - 6.1.1 LOADING PROCEDURES
 - 6.1.2 STARTING PROCEDURES
 - 6.1.3 ** STEPS FOR QUICK AND SIMPLE EXECUTION **
 - 6.2 INITIAL DIALOGUE
 - 6.3 PROGRAM OPTIONS
 - 6.3.1 START COMMAND
 - 6.3.2 RESTART COMMAND
 - 6.3.3 CONTINUE COMMAND
 - 6.3.4 PROCEED COMMAND
 - 6.3.5 ADD COMMAND
 - 6.3.6 DROP COMMAND
 - 6.3.7 PRINT COMMAND
 - 6.3.8 DISPLAY COMMAND
 - 6.3.9 FLAGS COMMAND
 - 6.3.10 ZFLAGS COMMAND
 - 6.3.11 CONTROL CHARACTERS
 - 6.3.12 HARDWARE PARAMETERS
 - 6.3.13 SOFTWARE PARAMETERS
 - 6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE
- 7.0 TEST DESCRIPTIONS
- 8.0 ERROR INFORMATION
 - 8.1 ERROR REPORTING

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

CVDMA.P11

10-DEC-80 09:14

PROGRAM DOCUMENT

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
159
160
161

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 BASIC DIAGNOSTIC TESTING OF THE VIA, FIFO, AND USYRT (BCP/BOP MODES) ON THESE BOARDS. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: USYRT REGISTER ADDRESSING, USYRT REGISTER STATIC BIT INTERACTION AND READ/WRITE TESTING, AND BASIC BOP AND BCP TX TESTING (USING THE TSO STATUS BIT).

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

3.0 PRELIMINARY PROGRAM REQUIREMENTS

THIS PROGRAM (CVDMA) SHOULD BE THE THIRD OF THE FIVE DMV-11 STATIC DIAGNOSTICS TO BE RUN (CVDMA/CVDMB SHOULD BE RUN FIRST). ERRORS FOUND IN THIS PROGRAM SHOULD BE CORRECTED BEFORE RUNNING ANY OF THE OTHER LINE UNIT DIAGNOSTICS (CVDMD OR CVDME).

CVDMA.P11

10-DEC-80 09:14

PROGRAM DOCUMENT

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

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 FIVE SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP+

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

4.4 ACT/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

CVDMA.P11

10-DEC-80 09:14

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
CVDMA-A-0
DMV-11 LINE UNIT TESTS - PART 1 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-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

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

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

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

CVDMA.P11

10-DEC-80 09:14

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

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

ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

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

6.3.2.3 EFFECT OF RESTART COMMAND

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

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:<FLAG-LIST>

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

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

6.3.4.2 EFFECT OF PROCEED COMMAND

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

6.3.5 ADD COMMAND

ADD/UNITS:<UNIT-LIST>

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

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

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 SURPRESSED 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.

CVDMA.P11

10-DEC-80 09:14

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 (SELO) 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. SWITCH PACK # 1 (BOOT ADDRESS): (0) 0 ?

5. SWITCH PACK # 2 (DDCMP ADDRESS): (0) 0 ?

THESE REPRESENT THE TWO USER MODIFIABLE 8 POSITION DIP-SWITCHES. THE ALLOWABLE RANGES FOR BOTH ARE 000-377 (OCTAL), AND THE DEFAULTS ARE BOTH 0.

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

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

7. BAUD RATE (0=LOW (19.2K), 1=HIGH (56K)):

THIS IS THE SPEED AT WHICH THE DMV TRANSMITS AND RECEIVES DATA. IN THE UNIT IS AN M8064, THE ANSWER IS IGNORED (DEFAULTS TO 56K). IF THE UNIT IS AN M8053, THEN THE ANSWER SHOULD BE BASED ON THE 'SPEED SELECT SWITCH' LOCATED ON THE BOARD.

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 1 OF THE STATIC LOGIC TESTS.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

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

AS SOON AS THE QUESTION "# 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.

CVDMA.P11

10-DEC-80 09:14

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

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

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

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

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

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

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

CVDMA.P11

10-DEC-80 09:14

PROGRAM DOCUMENT

722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738

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

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

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
788
789
790
791
792
793
794

7.0 TEST DESCRIPTIONS

```

*****
* TEST 1 <TBMT MICROCODE INTERRUPT TEST>
*
* THIS TEST CHECKS THE OPERATION OF THE TBMT (IRQ) INTERRUPT.
* THIS IS DONE BY ISSUING THE 'SET MAINTENANCE INTERRUPT FLAG AND CLEAR
* INTERRUPT DISABLE IN PROCESSOR STATUS' COMMAND WHILE IN THE MAINTENANCE
* LOOP AND THEN CHECKING FOR BIT 7 OF BSEL3 TO BE SET (THE BIT IS SET
* BY THE MICROCODE WHEN THE TBMT INTERRUPT OCCURS).
*****

```

```

*****
* TEST 2 <SWITCH SETTING TEST>
*
* SUBTEST #1:
* THE TWO READABLE SWITCH PACKS WILL BE SAMPLED AND COMPARED AGAINST THE 2
* VALUES IN THE P-TABLE. AN ERROR IS REPORTED ON A MISMATCH.
*
* SUBTEST #2:
* THE SPEED SELECT SWITCH (SPDSEL) IS READ VIA THE VIAORA REGISTER (BIT PA4)
* AND COMPARED AGAINST THE BAUD RATE VALUE IN THE P-TABLE. IF A MISMATCH
* OCCURS IT WILL BE REPORTED. NOTE: THIS SUBROUTINE IS NOT RUN IF AN M8064
* BOARD IS BEING TESTED (IT ONLY RUNS 56K... MAKING A SPEED SWITCH USELESS).
*
* THIS TEST IS ONLY RUN ON THE FIRST PASS AFTER A 'START' OR 'RESTART'.
* ALL SUCCESSIVE PASSES WILL SKIP THIS TEST.
*****

```

```

*****
* TEST 3 <USYRT MASTER CLEAR TEST>
*
* ALL REGISTERS ARE LOADED WITH PATTERN E IN THE SAME SEQUENCE AS FOR
* PATTERN F BELOW. THE USYRT IS THEN CLEARED BY A MASTER CLEAR
* (BIT 6 OF BSEL 1). ALL REGISTERS ARE THEN CHECKED FOR THE PROPER CONTENTS.
* THE INITIALIZED STATE OF THE REGISTERS IS CHECKED AGAINST DATA PATTERN F.
*
* PATTERN E: 377, 377, 377, 377, 377, 377, 377, 366.
* PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110.
*
* SEQUENCE OF REGISTERS AS USED WITH PATTERNS E & F:
*
* RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
*****

```

```

*****
* TEST 4 <USYRT PROGRAM RESET TEST>
*
* ALL REGISTERS ARE LOADED WITH PATTERN E IN THE SAME SEQUENCE AS FOR
* PATTERN F BELOW. THE USYRT IS THEN RESET BY ASSERTING PROGRAM RESET

```

PROGRAM DOCUMENT

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
844
845
846
847
848
849
850

:* (BIT 0 @ A000) IN THE 6522 VIA. ALL REGISTERS ARE THEN CHECKED FOR
:* THE PROPER CONTENTS. THE INITIALIZED STATE OF THE REGISTERS IS CHECKED
:* AGAINST DATA PATTERN F.

:* PATTERN E: 377, 377, 377, 377, 377, 377, 377, 366.
:* PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110.

:* SEQUENCE OF REGISTERS AS USED WITH PATTERNS E & F:

:* RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG

:*****

:*****

:* TEST 5 <USYRT REGISTER ADDRESSING TEST>

:* FIRST, A MASTER CLEAR IS ISSUED, TO INITIALIZE THE USYRT REGS TO
:* PATTERN F. THEN, EACH REGISTER IS WRITTEN WITH A BYTE OF PATTERN J,
:* AND AFTER EACH IS WRITTEN, ALL ARE READ AND COMPARED TO THE CURRENT
:* EXPECTED VALUES. THIS IS PERFORMED FOR ALL REGISTERS — INCLUDING THE
:* READ ONLY REGS — IN ORDER TO MAKE SURE THAT EACH REGISTER ONLY RESPONDS
:* TO ITS OWN ADDRESS.

:* PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110
:* PATTERN J: 000, 000, 001, 002, 004, 020, 040, 010

:* SEQUENCE OF REGISTERS AS USED WITH PATTERNS F & J:

:* RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG

:*****

:*****

:* TEST 6 <R/W BIT TEST OF PCSAR HIGH BYTE>

:* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN G.

:* PATTERN G: 000, 001, 003, 004, 005, 007, 100, 101, 103, 104, 105,
:* 107, 000, 017, 027, 041, 200, 277, 103, 144, 115, 157, 000.

:*****

:*****

:* TEST 7 <R/W BIT TEST OF S/AR REGISTER>

:* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN H.

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

:*****

:*****

:* TEST 8 <R/W BIT TEST OF PCR REGISTER>

:* PATTERN I IS LOADED INTO PCR (HIGH) AND THE DATA READ BACK AND

PROGRAM DOCUMENT

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
900
901
902
903
904
905
906

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

:*****
:* TEST 9 <R/W BIT TEST OF TDSR REGISTER'S HIGH BYTE>
:*
:* PATTERN K IS LOADED INTO TDSR (HIGH) AND THE DATA READ BACK IS
:* COMPARED AGAINST PATTERN L. (UNPREDICTABLE BITS ARE MASKED OFF TO 0
:* WHEN READING FOR COMPARISON.)
:*
:* PATTERN K: 000, 377, 376, 375, 373, 376, 177, 377, 000, 001, 002,
:* 004, 010, 200, 125, 252, 000.
:*
:* PATTERN L: 000, 017, 016, 015, 013, 016, 017, 017, 000, 001, 002,
:* 004, 010, 000, 005, 012, 000.
:*
:* NOTE THAT THE UNDEFINED BITS (12, 13, & 14) ARE MASKED OFF TO 0'S
:* FOR THE COMPARISON. ALSO THAT BIT 15 IS A READ/ONLY BIT AND CAN'T BE
:* SET -- THEREFORE SHOULD ALWAYS BE READ AS A 0 BY THIS TEST.
:*****

:*****
:* TEST 10 <R/W BIT TEST OF TXDB REGISTER>
:*
:* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN H.
:*
:* PATTERN H: 000, 001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
:* 376, 375, 373, 367, 357, 337, 277, 177, 377, 000
:*****

:*****
:* TEST 11 <PSEUDO R/W BIT TEST OF RXDB>
:*
:* WRITE, READ (BUT NO COMPARE) OF EACH WORD IN DATA PATTERN H. THIS IS
:* PRIMARILY TO PROVIDE A SCOPE LOOP FUNCTION ON THIS REGISTER.
:*
:* PATTERN H: 000, 001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
:* 376, 375, 373, 367, 357, 337, 277, 177, 377, 000
:*****

:*****
:* TEST 12 <PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE>
:*
:* WRITE, READ (BUT NO COMPARE) OF EACH WORD IN DATA PATTERN H. THIS IS
:* PRIMARILY TO PROVIDE A SCOPE LOOP FUNCTION ON THIS REGISTER.
:*
:* PATTERN H: 000, 001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
:*****

CVDMCA.P11

10-DEC-80 09:14

PROGRAM DOCUMENT

907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962

:* 376, 375, 373, 367, 357, 337, 277, 177, 377, 000
:*****

:*****
:* TEST 13 <NULL CLOCK TEST>

:* FIRST, A MASTER CLEAR IS DONE TO INIT THE DMV. THEN, THE T1 TIMER ON THE
:* VIA CHIP IS PROGRAMMED FOR SQUARE WAVE CLOCK GENERATION ON PB7 (BIT 7
:* OF VIA OUTPUT REG B), WITH A BAUD RATE = 56 KBAUD. THIS IS THE MODE OF
:* VIA OPERATION WHICH IS USED TO GENERATE THE NULL CLOCK. THEN, THE PROGRAM
:* SCANS ORB REPEATEDLY TO MONITOR THE NULL CLOCK BIT, IN THE FOLLOWING
:* SEQUENCE :
:* - THE PROGRAM REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 1 STATE, AND
:* IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC (A GROSS TIMEOUT
:* INTERVAL), AN ERROR IS REPORTED. (AT 56 KBAUD, THE CLOCK SHOULD
:* HAVE A PERIOD OF ABOUT 18 MICRO-SEC.)
:* - THE PROGRAM NEXT REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 0 STATE,
:* AND IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC, AN ERROR IS
:* REPORTED.
:* - THE PROGRAM NEXT REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 1 STATE
:* AGAIN, AND IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC,
:* AN ERROR IS REPORTED.

:*****
:* TEST 14 <BCP TX RESET W/IDLE = 0>

:* THE USYRT IS INITIALIZED FOR 'BYTE-CONTROL PROTOCOL' (BCP) WITH IDLE
:* SET TO ZERO AND A 125 SYNC CHARACTER IS LOADED INTO S/AR. A 226 SYNC
:* CHARACTER IS LOADED INTO TXDB SO THAT THE SOURCE OF SYNC CHARACTERS
:* CAN BE LATER DETERMINED. THE VALID STATE OF THE USYRT REGISTERS IS
:* READ AND CHECKED. TXE IS ASSERTED TO ENABLE THE TRANSMITTER LOGIC.
:* THEN, TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE OBSERVING
:* TXA — IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
:* (TXBE SHOULD GO HIGH; AT THIS TIME THE S/AR'S SYNC CHARACTER SHOULD
:* BE LOADED INTO TXSO AND TSOM IS AGAIN SET — DRIVING TXBE LOW.)
:* THREE SYNC CHARACTERS ARE SENT/RECEIVED: THE FIRST TWO SYNCHRONIZE
:* THE RECEIVER, THE THIRD IS DIRECTLY READ (STRIP SYNC IS OFF) AND
:* COMPARED AGAINST 125 (THE S/AR SYNC CHARACTER).
:* IF VALUE READ IS 226, THEN TXDB PROVIDED THE SYNC (IE: ERROR).
:* THE USYRT IS THEN RESET AND REGISTERS ARE AGAIN READ AND CHECKED.
:* THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY
:* ONE MARK AND THREE SYNC CHARACTERS (FROM THE S/AR) IS TRANSMITTED.
:* ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
:* SEQUENCE.

:*****
:* TEST 15 <BCP TX RESET W/IDLE = 1>

:* THE USYRT IS INITIALIZED FOR 'BYTE-CONTROL PROTOCOL' (BCP) WITH IDLE
:* SET TO ONE AND A 226 SYNC CHARACTERS LOADED INTO S/AR AND TXDB.
:* THE VALID STATE OF THE USYRT REGISTERS IS READ AND CHECKED. TXE IS

963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018

```

: * ASSERTED TO ENABLE THE TRANSMITTER LOGIC.
: * THEN, TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE OBSERVING
: * TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
: * (TXBE SHOULD GO HIGH; AT THIS TIME THE TXDB SYNC CHARACTER SHOULD
: * BE LOADED INTO TXSO AND TSOM IS AGAIN SET -- DRIVING TXBE LOW.)
: * AFTER THE RECEIVER IS SYNCHRONIZED (TWO SYNC CHARACTERS), TXDB IS
: * LOADED WITH A 125 AND, WITH TSOM STILL = 1 (SYNC SOURCE = TXDB), THE
: * USYRT IS AGAIN CLOCKED.
: * AT THIS POINT, IF THE IDLE BIT WORKED, THE VALUE 125 WILL BE READ
: * BY THE RECEIVER. OTHERWISE A 226 WILL BE READ, INDICATING TXDB WASN'T
: * PROVIDING THE SYNC CHARACTERS.
: * WHEN TXBE GOES HIGH AGAIN, THE USYRT IS RESET.
: * ALL REGISTERS ARE AGAIN READ AND CHECKED.
: * THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY
: * ONE MARK AND THREE SYNCs (226,226,125 FROM TXDB) ARE TRANSMITTED.
: * ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
: * SEQUENCE.
: *****

```

```

: *****
: * TEST 16 <BCP TX UNDERRUN W/TSOM TERMINATION>
: *
: * THE USYRT IS INITIALIZED FOR BCP WITH IDLE = 1 AND TXC IS MANUALLY
: * CONTROLLED UNTIL TWO SYNC CHARACTERS AND ONE DATA CHARACTER (000)
: * HAVE BEEN TRANSMITTED. AT THIS TIME WHEN TXBE IS ASSERTED BY THE
: * USYRT, NO DATA IS LOADED INTO TXDB -- FORCING AN UNDER RUN
: * CONDITION. TXU AND TERR ARE CHECKED BOTH BEFORE AND AFTER THEIR
: * EXPECTED ASSERTIONS. AFTER THE FIRST NON-DATA CHARACTER (WHICH
: * SHOULD BE THE MARK CHARACTER) HAS BEEN STARTED, IDLE IS SET TO 0.
: * THIS SHOULD FORCE THE NEXT NON-DATA CHARACTER TO BE A SYNC CHARACTER
: * FROM S/AR. WHILE THIS SYNC CHARACTER IS BEING TRANSMITTED, TSOM IS
: * ASSERTED (CLEARING TXU AND TERR) -- IDLE IS LEFT AT 0. TXBE IS THEN
: * CYCLED THROUGH AT LEAST ONE MORE SYNC CHARACTER AND THE TEST IS
: * ABORTED. ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS
: * WITHIN THE SEQUENCE.
: * NOTE: BITS SHIFT OUT OF TX LSB FIRST.
: *****

```

```

: *****
: * TEST 17 <BCP TX UNDERRUN W/RESET TERMINATION>
: *
: * THE USYRT IS INITIALIZED FOR BCP WITH IDLE = 1 AND TXC IS MANUALLY
: * CONTROLLED UNTIL TWO SYNC CHARACTERS AND ONE DATA CHARACTER HAVE
: * BEEN TRANSMITTED. AT THIS TIME WHEN TXBE IS ASSERTED BY THE USYRT,
: * NO DATA IS LOADED INTO TXDB -- FORCING AN UNDER RUN CONDITION. TXU
: * AND TERR ARE CHECKED BOTH BEFORE AND AFTER THEIR EXPECTED
: * ASSERTIONS. AFTER THE FIRST NON-DATA CHARACTER (WHICH SHOULD BE THE
: * MARK CHARACTER) HAS BEEN STARTED, IDLE IS SET TO 0. THIS SHOULD
: * FORCE THE NEXT NON-DATA CHARACTER TO BE A SYNC CHARACTER.
: * IMMEDIATELY AFTER THIS SYNC CHARACTER HAS BEING TRANSMITTED, A
: * PROGRAM RESET IS ISSUED AND ALL REGISTERS ARE CHECKED. ERROR
: * LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
: * SEQUENCE.
: *****

```

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
 1067
 1068
 1069
 1070
 1071
 1072
 1073
 1074

```

*****
: * TEST 18 <BCP TX DISABLE TEST>
: *
: * THE USYRT IS INITIALIZED FOR BCP AND A MESSAGE IS STARTED. ONCE THE
: * SECOND DATA CHARACTER IS LOADED INTO TXDB TXE IS DROPPED. TXSO IS
: * WATCHED TO ASSURE THAT THE CHARACTER BEING TRANSMITTED IS COMPLETED.
: * WHEN IT IS, THE USYRT SHOULD DROP TXA AND STOP TRANSMITTING -- THE
: * LAST CHARACTER LOADED INTO TXDB SHOULD BE LOST.
: *
: * CHARACTERS LOADED: 125 252
: * CHARACTERS TRANSMITTED: 125
*****

```

```

*****
: * TEST 19 <FIFO STACKING CHARACTERS TEST>
: *
: * THE USYRT IS SETUP FOR BCP MODE WITH NO ERROR DETECTION.
: * THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS
: * TO FILL THE 8 CHARACTER RECEIVER FIFO WITH THE CHARACTERS:
: * 1/2(SYNCH),000,377,125,252,347,030,303,1/2(074).
: * THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. NOTE
: * THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE CHARACTERS SINCE THEY
: * ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
: * RDA = 0).
: * ALSO NOTE THAT DUE TO FIFO TIMING, TWO 'HALF CHARACTERS' ARE LOADED
: * INTO THE FIFO (THE 1ST AND LAST CHARACTERS).
: *
*****

```

```

*****
: * TEST 20 <BCP CHARACTER LENGTH TEST>
: *
: * THE USYRT IS INITIALIZED FOR BCP WITH NO ERROR CHECKING. TXC IS MANUALLY
: * CONTROLLED UNTIL TWO SYNC CHARACTERS HAVE BEEN TRANSMITTED. THEN 3
: * SUBTESTS FOLLOW, EACH ONE USING A DIFFERENT TRANSMIT CHARACTER LENGTH
: * STARTING AT FIVE (5) AND ENDING WITH SEVEN (7).
: *
: * TEST PATTERN: 111 222 333 044 155 266 377
*****

```

```

*****
: * TEST 21 <BOP TX TABORT/(IDLE = 0) TEST>
: *
: * THE USYRT IS INITIALIZED FOR 'BIT-ORIENTED PROTOCOL' (BOP) WITH IDLE
: * SET TO ZERO. TXE AND TSOH IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
: * OBSERVING TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
: * NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
: * LOADED INTO TXDB DRIVING TXBE LOW. THE TRANSMITTER IS CLOCKED THROUGH
: * ONE CHARACTER. WHEN TXBE GOES HIGH AGAIN, TABORT IS ASSERTED CAUSING
: * ABORT TO BE TRANSMITTED. ALL CHARACTERS ARE CHECKED AT TXSO.
: * THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY TWO

```

CVDMCA.P11

10-DEC-80 09:14

PROGRAM DOCUMENT

1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130

:* FLAGS, ONE ZERO CHARACTER, AND ONE ABORT CHARACTER IS SENT (INTO THE BIT
:* BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
:* THE SEQUENCE.
:*****

:*****
:* TEST 22 <BOP TX TABORT/(IDLE = 1) TEST>
:*
:* THE USYRT IS INITIALIZED FOR 'BIT-ORIENTED PROTOCOL' (BOP) WITH IDLE
:* SET TO ONE. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
:* OBSERVING TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
:* NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
:* LOADED INTO TXDB DRIVING TXBE LOW. THE TRANSMITTER IS CLOCKED THROUGH
:* ONE CHARACTER. WHEN TXBE GOES HIGH AGAIN, TABORT IS ASSERTED CAUSING
:* ABORT TO BE TRANSMITTED. ALL CHARACTERS ARE CHECKED AT TXSO.
:* THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY TWO
:* FLAGS, ONE ZERO CHARACTER, AND ONE FLAG CHARACTER IS SENT (INTO THE BIT
:* BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
:* THE SEQUENCE.
:*****

:*****
:* TEST 23 <BOP TX TXGA (TRANSMIT GO-AHEAD) TEST>
:*
:* THE USYRT IS INITIALIZED FOR BOP AND TXE IS ASSERTED. TSOM IS ASSERTED
:* AND TXA OBSERVED -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
:* NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
:* LOADED INTO TXDB DRIVING TXBE LOW. WHEN TXBE GOES HIGH AGAIN, TXGA
:* IS ASSERTED CAUSING GA TO BE TRANSMITTED. THE SEQUENCE OF EVENTS IS
:* CONTINUALLY MONITORED WHILE TXC IS MANUALLY CONTROLLED AND ALL
:* CHARACTERS ARE CHECKED AT TXSO. THIS TEST WILL GO NO FURTHER INTO
:* THE TRANSMIT SEQUENCE SO THAT ONLY TWO FLAGS, ONE ZERO CHARACTER
:* AND ONE GA CHARACTER IS SENT (INTO THE BIT BUCKET).
:* ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
:* THE SEQUENCE.
:*****

:*****
:* TEST 24 <BOP TX MESSAGE WITHOUT CRC>
:*
:* THE USYRT IS INITIALIZED FOR BOP MODE WITH NO ERROR DETECTION. TXC IS THEN
:* MANUALLY CONTROLLED UNTIL TWO FLAG CHARACTERS HAVE BEEN TRANSMITTED. THEN
:* A 5 CHARACTER MESSAGE IS TRANSMITTED, RECEIVED, CHECKED, AND TERMINATED
:* (WITH TEOM). A CHECK IS MADE TO ASCERTAIN THAT NO CRC OR VRC IS GENERATED
:* -- FLAG CHARACTERS SHOULD FOLLOW THE DATA.
:* (NOTE: NO BIT STUFFING OCCURS IN THIS TEST)

:* TEST MESSAGE: FLAG FLAG 000 307 125 252 201 FLAG
:*****

:*****
:* TEST 25 <BOP RX CHARACTER LENGTH TEST>

1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186

```

:*
:* THE USYRT IS INITIALIZED FOR BOP WITH CRC-CCITT PRESET TO 1'S. TXC
:* IS MANUALLY CONTROLLED UNTIL TWO FLAG CHARACTERS HAVE BEEN
:* TRANSMITTED. THEN 6 SUBTESTS FOLLOW, EACH ONE USING A DIFFERENT
:* TRANSMIT CHARACTER LENGTH STARTING AT TWO (2) AND ENDING WITH SEVEN
:* (7). IN EACH SUBTEST, TWO 8 BIT CHARACTERS WILL BE TRANSMITTED
:* BEFORE TXCL IS CHANGED TO THE CHARACTER LENGTH BEING TESTED. THIS
:* CORRESPONDS TO NORMAL USAGE WHERE EITHER:
:*
:* 1 -- A MESSAGE OF CHARACTERS WHICH ARE LESS THEN 8 BITS IS
:* SENT AS A STREAM OF 8 BIT CHARACTERS AND THE REMAINING
:* BITS ARE SENT AS A CHARACTER OF LESS THEN 8 BITS OR
:*
:* 2 -- A HEADER OF TWO 8 BIT CHARACTERS IS SENT FOLLOWED BY A
:* DATA STREAM OF DATA CHARACTERS WHICH MAY BE LESS THEN 8
:* BITS IN LENGTH (I.E. 2, 3, 4, 5, 6, OR 7 BIT
:* CHARACTERS).
:*
:* THE TEST PATTERN IS: 123 321 111 222 333 044 155 266 377

```

```

*****
:* TEST 26 <TX 'SPACING SEQUENCE'>
:*
:* THE TRANSMITTER IS INITIALIZED AND THE 'SPACING SEQUENCE' IS FORCED
:* BY ASSERTING BOTH TSON & TEOM AT THE SAME TIME -- CHECK THE BIT
:* STREAM FOR ACCURACY (SPACES) AND COMPLETNESS (16 OF THEM). WHEN TXBE
:* GOES HIGH (= 1) A SMALL MESSAGE IS SENT.

```

```

*****
:* TEST 27 <FIFO OVERRUN INTEGRITY TEST>
:*
:* THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS TO FILL
:* THE 8 CHARACTER RECEIVER FIFO UNTIL RXOR WITH THE CHARACTERS:
:* (SYNCH) 000,377,125,252,347,030,303,074,125.
:* THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. OF IMPORTANCE
:* IS THE INTEGRITY OF THE LAST OVERRUN-CAUSING FIFO CHARACTER (IT SHOULD
:* REMAIN INTACT).
:* NOTE THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE CHARACTERS SINCE THEY
:* ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
:* RDA = 0).

```

```

*****
:* TEST 28 <BCP RX OVERRUN SET AND CLEAR TEST>
:*
:* THE USYRT IS INITIALIZED AND THREE SUBTESTS ARE PERFORMED.
:*
:* 1 -- AN OVERRUN CONDITION IS FORCED, RECEIVER STATUS REGISTER IS
:* READ TWICE: ONCE TO VERIFY ROR BIT = 1, AND AGAIN TO VERIFY
:* THAT THE FIRST READ CLEARED ROR .

```

1187
 1188
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1240
 1241
 1242

```

: *
: * 2 -- AN OVERRUN CONDITION IS FORCED (BY THE SAME TECHNIQUE USED IN
: * (2), THE USYRT IS RESET AND THE PROPER STATE OF ALL REGISTERS
: * IS VERIFIED.
: *
: * 3 -- AN OVERRUN CONDITION IS FORCED (AS ABOVE). RXE IS THEN DROPPED
: * AND A DELAY IS PROVIDED TO ALLOW TIME FOR THE FIFO TO FLUSH
: * (CAUSED BY RDA GOING LOW). RXE IS THEN RE-INITIALIZED AND ROR
: * IS CHECKED = 0. THE RECEIVER IS THEN RE-SYNCHED AND THE TEST IS
: * TERMINATED.
: *
: *****
: *****
: * TEST 29 <BCP RX SYNC CHARACTER RECOGNITION>
: *
: * THE FOLLOWING MESSAGE IS INITIATED WITHOUT ASSERTING RXE AND ONCE
: * THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):
: *
: * SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA
: * SYNC SYNC DATA DATA DATA SYNC SYNC
: *
: * THE RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE
: * THE NEXT TWO SYNC CHARACTERS FOR SYNCHRONIZATION, THEN PASS THE REST
: * OF THE MESSAGE (7 SYNC AND 6 DATA CHARACTERS) THROUGH RXDB REGISTER.
: *
: *****
: *****
: * TEST 30 <BCP RX STRIP-SYNC TEST>
: *
: * THE USYRT IS INITIALIZED WITH THE STRIP-SYNC CONTROL BIT ASSERTED.
: * THE FOLLOWING MESSAGE IS THEN INITIATED WITHOUT ASSERTING RXE AND
: * ONCE THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):
: *
: * SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA
: * SYNC SYNC DATA DATA DATA SYNC SYNC
: *
: * THE RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE
: * THE NEXT TWO SYNC CHARACTERS FOR SYNCHRONIZATION, IGNORE THE NEXT
: * THREE SYNC CHARACTERS, AND PASS THE REST OF THE MESSAGE (4 SYNC AND
: * 6 DATA CHARACTERS) THROUGH RXDB REGISTER.
: *
: *****
: *****
: * TEST 31 <BCP RX LOST RXE TEST>
: *
: * THE USYRT IS INITIALIZED (CRC16,STRIPS,BCP MODE) AND A MESSAGE IS STARTED.
: * WHILE IN THE MIDDLE OF TEXT, RXE IS DROPPED AND THE REACTION OF THE
: * RECEIVER IS MONITORED.
: *
: *****

```

CVDMCA.P11

10-DEC-80 09:14

PROGRAM DOCUMENT

1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284

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

CVDMA.P11 10-DEC-80 09:14

GENERAL EQUATES AND DS INVOCATION & SETUP

.SBTTL GENERAL EQUATES AND DS INVOCATION & SETUP

```

1285
1286
1287
1288      000000      HELP=0      : CONTROL LISTING OF HELP INFORMATION
1289
1290      .           : HELP=0    NO LIST
1291      .           : HELP=1    LIST
1292
1293      002000      .=2000
1294
1295      .MCALL      SVC
1296 002000      SVC      : INITIALIZE SUPERVISOR MACROS
1297
1298
1299 002000      BGNMOD  LU1MOD
1300
1301
1302      000001      $LSTIN= 1
1303      000001      $LSTTAG= 1
1304      000001      SVCINS= 1      : LIST INSTRUCTIONS, SHIFTED RIGHT
1305      000001      SVCTST= 1      : LIST TEST TAGS, SHIFTED RIGHT
1306      000001      SVCSUB= 1      : LIST SUBTEST TAGS, SHIFTED RIGHT
1307      000001      SVCGBL= 1      : LIST GLOBAL TAGS, SHIFTED RIGHT
1308      000001      SVCTAG= 1      : LIST OTHER TAGS, SHIFTED RIGHT
1309
1310      : CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1311      : TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
1312      : SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
1313      : CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.

```

CVDMCA.P11 10-DEC-80 09:14

PROGRAM HEADER

```

.SBTTL PROGRAM HEADER
:++
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:--
    
```

```

1314
1315
1316
1317
1318
1319
1320 002000
1321
1322
1323 002000
1324 002000
1325 002000 103
1326 002001 126
1327 002002 104
1328 002003 115
1329 002004 103
1330 002005 000
1331 002006 000
1332 002007 000
1333 002010
1334 002010 101
1335 002011
1336 002011 060
1337 002012
1338 002012 000000
1339 002014
1340 002014 000012
1341 002016
1342 002016 036640
1343 002020
1344 002020 000000
1345 002022
1346 002022 002224
1347 002024
1348 002024 000000
1349 002026
1350 002026 037426
1351 002030
1352 002030 000000
1353 002032
1354 002032 000000
1355 002034
1356 002034 000000
1357 002036
1358 002036 000000
1359 002040
1360 002040 002124
1361 002042
1362 002042 000000
1363 002044
1364 002044 000000
1365 002046
1366 002046 000000
1367 002050
1368 002050 003
1369 002051 003
    
```

POINTER BGNAU,BGNDU,ERRTBL

HEADER CVDMC,A,0,10.,0

```

LSNAME::
        .ASCII /C/
        .ASCII /V/
        .ASCII /D/
        .ASCII /M/
        .ASCII /C/
        .BYTE 0
        .BYTE 0
        .BYTE 0
LSREV::
        .ASCII /A/
LSDEPO::
        .ASCII /0/
LSUNIT::
        .WORD 0
LSTIML::
        .WORD 10.
LSHPCP::
        .WORD LSHARD
LSSPCP::
        .WORD 0
LSHPTP::
        .WORD LSHW
LSSPTP::
        .WORD 0
LSLADP::
        .WORD LSLAST
LSSTA::
        .WORD 0
LSCO::
        .WORD 0
LSDTYP::
        .WORD 0
LSAPT::
        .WORD 0
LSDTP::
        .WORD LSDISPATCH
LSPRIO::
        .WORD 0
LSENI::
        .WORD 0
LSEXP1::
        .WORD 0
LSMREV::
        .BYTE CSREVISION
        .BYTE C$EDIT
    
```

CVDMA.P11

10-DEC-80 09:14

PROGRAM HEADER

1370 002052
 1371 002052 000000
 1372 002054 000000
 1373 002056
 1374 002056 000000
 1375 002060
 1376 002060 003306
 1377 002062
 1378 002062 000000
 1379 002064
 1380 002064 000000
 1381 002066
 1382 002066 000000
 1383 002070
 1384 002070 021546
 1385 002072
 1386 002072 021542
 1387 002074
 1388 002074 000000
 1389 002076
 1390 002076 003326
 1391 002100
 1392 002100 104035
 1393 002102
 1394 002102 002246
 1395 002104
 1396 002104 021100
 1397 002106
 1398 002106 021540
 1399 002110
 1400 002110 021414
 1401 002112
 1402 002112 021072
 1403 002114
 1404 002114 000000
 1405 002116
 1406 002116 000000
 1407 002120
 1408 002120 000000
 1409
 1410
 1411

.EVEN

LSEF:: .WORD 0
 .WORD 0
 LSSPC:: .WORD 0
 LSDEVP:: .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
 LSPRT:: .WORD LSPROT
 LSTEST:: .WORD 0
 LSDLY:: .WORD 0
 LSHIME:: .WORD 0

CVDMCA.P11 10-DEC-80 09:14

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 31.

1412		
1413		
1414	002122	
1415		
1416		
1417		
1418	002122	
1419		
1420		
1421	002122	
1422	002122	000037
1423	002124	
1424	002124	021550
1425	002126	021730
1426	002130	022730
1427	002132	023114
1428	002134	023200
1429	002136	023424
1430	002140	023516
1431	002142	023610
1432	002144	023702
1433	002146	024032
1434	002150	024124
1435	002152	024200
1436	002154	024254
1437	002156	024576
1438	002160	024730
1439	002162	025112
1440	002164	025620
1441	002166	026220
1442	002170	026426
1443	002172	026704
1444	002174	027414
1445	002176	027610
1446	002200	030004
1447	002202	030200
1448	002204	030556
1449	002206	033500
1450	002210	034056
1451	002212	034442
1452	002214	035044
1453	002216	035522
1454	002220	036256
1455		

.WORD	31
LSDISPATCH::	
.WORD	T1
.WORD	T2
.WORD	T3
.WORD	T4
.WORD	T5
.WORD	T6
.WORD	T7
.WORD	T8
.WORD	T9
.WORD	T10
.WORD	T11
.WORD	T12
.WORD	T13
.WORD	T14
.WORD	T15
.WORD	T16
.WORD	T17
.WORD	T18
.WORD	T19
.WORD	T20
.WORD	T21
.WORD	T22
.WORD	T23
.WORD	T24
.WORD	T25
.WORD	T26
.WORD	T27
.WORD	T28
.WORD	T29
.WORD	T30
.WORD	T31

CVDMCA.P11 10-DEC-80 09:14

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

```

```

1482
1483
1484
1485
1486
1487
1488
1489 002244
1490 002244 000000
1491 002246
1492 002246
1493
1494 002246
1495 002246

```

BGNSW SFPTBL

```

                .WORD L10001-L$SW/2
L$SW::
SFPTBL::

```

ENDSW

L10001:

CVDMCA.P11 10-DEC-80 09:14

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 DEFINITIONS

BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100
BIT05== 40
BIT04== 20
BIT03== 10
BIT02== 4
BIT01== 2
BIT00== 1

BIT9== BIT09
BIT8== BIT08
BIT7== BIT07
BIT6== BIT06
BIT5== BIT05
BIT4== BIT04
BIT3== BIT03
BIT2== BIT02
BIT1== BIT01
BIT0== BIT00

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

EF.START== 32. : START COMMAND WAS ISSUED
EF.RESTART== 31. : RESTART COMMAND WAS ISSUED
EF.CONTINUE== 30. : CONTINUE COMMAND WAS ISSUED
EF.NEW== 29. : A NEW PASS HAS BEEN STARTED
EF.PWR== 28. : A POWER-FAIL/POWER-UP OCCURRED

: PRIORITY LEVEL DEFINITIONS

PRI07== 340
PRI06== 300
PRI05== 240
PRI04== 200

1496
1497
1498
1499
1500
1501
1502
1503
1504 002246
1505
1506
1507
1508 100000
1509 040000
1510 020000
1511 010000
1512 004000
1513 002000
1514 001000
1515 000400
1516 000200
1517 000100
1518 000040
1519 000020
1520 000010
1521 000004
1522 000002
1523 000001
1524
1525 001000
1526 000400
1527 000200
1528 000100
1529 000040
1530 000020
1531 000010
1532 000004
1533 000002
1534 000001
1535
1536
1537
1538
1539 000040
1540 000037
1541 000036
1542 000035
1543 000034
1544
1545
1546
1547
1548 000340
1549 000300
1550 000240
1551 000200

CVDMCA.P11 10-DEC-80 09:14

GLOBAL EQUATES SECTION -- BASIC EQUATES

1552	000140	PRI03== 140
1553	000100	PRI02== 100
1554	000040	PRI01== 40
1555	000000	PRI00== 0
1556		.
1557		.; OPERATOR FLAG BITS
1558		.
1559	000004	EVL== 4
1560	000010	LOT== 10
1561	000020	ADR== 20
1562	000040	IDU== 40
1563	000100	ISR== 100
1564	000200	UAM== 200
1565	000400	BOE== 400
1566	001000	PNT== 1000
1567	002000	PRI== 2000
1568	004000	IXE== 4000
1569	010000	IBE== 10000
1570	020000	IER== 20000
1571	040000	LOE== 40000
1572	100000	HOE== 100000

CVDMCA.P11 10-DEC-80 09:14

REGISTER DEFINITIONS -- MAINTENANCE REGISTERS -- SELN & BSELN

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

1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609

000020
000001

000200
000100
000001
000301

000200

000001
000002
000003
000004
000005
000007

```

:*****
:* 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 RQY STATE
MCLR     = BIT6      ;MASTER CLEAR
MREQ     = BIT0      ;M-LOOP ACCESS
STRMLOF= 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
WRILOC   = 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

```

CVDMCA.P11 10-DEC-80 09:14

REGISTER DEFINITIONS -- USYRT

.SBTTL REGISTER DEFINITIONS -- USYRT

1610
 1611
 1612
 1613 120400
 1614
 1615
 1616
 1617
 1618
 1619 120400
 1620
 1621
 1622
 1623
 1624
 1625 120401
 1626
 1627
 1628 000200
 1629 000160
 1630 000010
 1631 000004
 1632 000002
 1633 000001
 1634
 1635
 1636 100000
 1637 004000
 1638 002000
 1639 001000
 1640 000400
 1641
 1642 000001
 1643
 1644
 1645
 1646
 1647
 1648 120402
 1649
 1650
 1651
 1652
 1653
 1654 120403
 1655
 1656
 1657 000200
 1658 000010
 1659 000004
 1660 000002
 1661 000001
 1662
 1663
 1664 100000
 1665 004000

USYRT = 120400 ;USYRT BASE ADDRESS = A100 (HEX)
 ::*****
 :* USYRT 'RECEIVER DATA BUFFER' REGISTER -- READ ONLY
 ::*****
 RDSRL = 120400 ;ADDRESS OF THIS REG
 ::*****
 :* USYRT 'RECEIVER STATUS' REGISTER -- READ ONLY
 ::*****
 RDSRH = 120401 ;ADDRESS OF THIS REG
 :BIT DEFINITIONS ON BYTE BASIS :
 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
 :BIT DEFINITIONS ON WORD BASIS :
 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
 RERCHK = BIT0 ;FLAG TO INVOKE RERR CHK IN SUBROUTINE RXCHAR
 ::*****
 :* USYRT 'TRANSMITTER DATA BUFFER' REGISTER
 ::*****
 TDSRL = 120402 ;ADDRESS OF THIS REG
 ::*****
 :* USYRT 'TX STATUS AND CONTROL' REGISTER
 ::*****
 TDSRH = 120403 ;ADDRESS OF THIS REG
 :BIT DEFINITIONS ON BYTE BASIS :
 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
 :BIT DEFINITIONS ON WORD BASIS :
 TXERR = BIT15 ;TRANSMITTER UNDERRUN ERROR
 TXGA = BIT11 ;TRANSMIT GO AHEAD

CVDMCA.P11 10-DEC-80 09:14

REGISTER DEFINITIONS -- USYRT

1666 002000
 1667 001000
 1668 000400
 1669
 1670
 1671
 1672
 1673
 1674 120404
 1675 000226
 1676
 1677
 1678
 1679
 1680
 1681 120405
 1682
 1683
 1684
 1685 000200
 1686 000100
 1687 000040
 1688 000020
 1689 000010
 1690 000007
 1691
 1692
 1693
 1694 100000
 1695 040000
 1696 020000
 1697 010000
 1698 004000
 1699 001400
 1700 003400
 1701 002400
 1702 002000
 1703
 1704
 1705
 1706
 1707
 1708 120407
 1709
 1710
 1711
 1712 000340
 1713 000020
 1714 000010
 1715 000007
 1716
 1717
 1718
 1719
 1720 122000
 1721

TXAB = BIT10 ; TRANSMIT ABORT
 TXEOM = BIT9 ; TRANSMIT END-OF-MESSAGE
 TXSOM = BIT8 ; TRANSMIT START-OF-MESSAGE

::*****
 :* USYRT 'SYNC/SECONDARY ADDRESS' REGISTER
 ::*****

PCSARL = 120404 ; ADDRESS OF THIS REG
 SYNCH = 226 ; STANDARD SYNCH CHARACTER

::*****
 :* USYRT 'MODE CONTROL'
 ::*****

PCSARH = 120405 ; ADDRESS OF THIS REG

;BIT DEFINITIONS ON BYTE BASIS:

APA = BIT7 ; 'ALL PARTIES ADDRESS' ENABLE
 PROTO = BIT6 ; SPECIFIES BOP/CCP PROTOCOL -- 0 = BOP
 STRIP = BIT5 ; STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
 SECAD = BIT4 ; SECONDARY ADDRESS MODE -- BOP MODE ONLY
 IDLE = BIT3 ; IDLE & SYNC CHAR. TRANSMISSION CONTROL
 XYZ = BIT2!BIT1!BIT0 ; CRC/PARITY SELECTION CONTROL

;BIT DEFINITIONS ON WORD BASIS:

APAD = BIT15 ; 'ALL PARTIES ADDRESS' ENABLE
 DDCMP = BIT14 ; CODE FOR DDCMP MODE
 STRIPS = BIT13 ; STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
 SECADR = BIT12 ; SECONDARY ADDRESS MODE -- BOP MODE ONLY
 IDLES = BIT11 ; IDLE & SYNC CHAR. TRANSMISSION CONTROL
 CRC16 = BIT9!BIT8 ; CODE FOR CRC-16 SELECTION
 NOCHK = BIT10!BIT9!BIT8 ; CODE FOR NO ERROR CHECKING
 EVRC = BIT10!BIT8 ; CODE FOR VRC EVEN CHECK
 OVRC = BIT10 ; CODE FOR VRC ODD CHECK

::*****
 :* USYRT 'DATA LENGTH SELECT' REGISTER
 ::*****

PCR = 120407 ; ADDRESS OF THIS REG

;BIT DEFINITIONS:

TXDL = BIT7!BIT6!BIT5 ; TRANSMIT DATA LENGTH SELECTION
 EXADD = BIT4 ; EXTENDED ADDRESS FIELD -- NOT USED OR TESTED
 EXCON = BIT3 ; EXTENDED CONTROL FIELD -- NOT USED OR TESTED
 RXDL = BIT2!BIT1!BIT0 ; RECEIVER DATA LENGTH SELECTION

::*****
 :* USYRT STATUS REGISTER (ADDR. A400)
 ::*****

USTATR = 122000 ; USYRT STATUS REGISTER ADDRESS = A400 (HEX)

CVDMCA.P11

10-DEC-80 09:14

REGISTER DEFINITIONS -- USYRT

1722
1723
1724 000200
1725 000100
1726 000040
1727 000020
1728 000010
1729 000004
1730 000002
1731 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

CVDMCA.P11 10-DEC-80 09:14

REGISTER DEFINITIONS -- 6522 VIA CHIP

.SBTTL REGISTER DEFINITIONS -- 6522 VIA CHIP

1732
1733
1734 120000
1735
1736
1737
1738
1739
1740 120000
1741
1742 000200
1743 000100
1744 000040
1745 000020
1746 000010
1747 000004
1748 000002
1749 000001
1750 000000
1751
1752
1753
1754
1755
1756 120001
1757
1758 000200
1759 000100
1760 000040
1761 000020
1762 000010
1763 000004
1764 000002
1765 000001
1766
1767
1768
1769
1770
1771
1772 120002
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784 120003
1785
1786
1787

```

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

NULCLK   = 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' --
UMAIN    = BIT0           ; SELECT USYRT INT LOOPBACK **SELECT BIT**

:*****
:* DATA DIRECTION FOR PORT B -- 'DDR B' -- 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

```


CVDMCA.P11

10-DEC-80 09:14

REGISTER DEFINITIONS -- 6522 VIA CHIP

1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843

: 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
:*****

120004

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 T1MODE (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 T1MODE (IN VIAACR) = 3

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

CVDMCA.P11

10-DEC-80 09:14

REGISTER DEFINITIONS -- 6522 VIA CHIP

1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899

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

CVDMCA.P11

10-DEC-80 09:14

REGISTER DEFINITIONS -- 6522 VIA CHIP

```

1900           : 3   SHIFT IN UNDER CONTROL OF EXTERNAL INPUT PULSES
1901           : 4   SHIFT OUT -- FREE RUNNING -- RATE CONTROLLED BY T2
1902           : 5   SHIFT OUT -- RATE CONTROLLED BY T2 -- PULSES ON CB1
1903           : 6   SHIFT OUT -- SYS. CLOCK RATE -- PULSES ON CB1
1904           : 7   SHIFT OUT -- UNDER CONTROL OF PULSES APPLIED TO CB1

```

```

1905
1906           000002   PBLENB = BIT1           ;PB LATCH CONTROL -- 1 ENABLES LATCH
1907           000001   PALENB = BIT0          ;PA LATCH CONTROL -- 1 ENABLES LATCH
1908
1909
1910
1911

```

```

:*****
:* PERIPHERAL CONTROL REGISTER -- 'PCR' -- READ/WRITE
:*****

```

```

1912
1913
1914
1915
1916           120014   VIAPCR = 120014          ;ADDRESS OF THIS REGISTER -- HEX = A0XC
1917
1918           000340   CB2CTL = BIT7!BIT6!BIT5     ;CB2 MODE SELECT
1919           000020   CB1CTL = BIT4             ;CB1 MODE SELECT
1920           000016   CA2CTL = BIT3!BIT2!BIT1     ;CA2 MODE SELECT
1921           000001   CA1CTL = BIT0             ;CA1 MODE SELECT
1922
1923
1924

```

```

:*****
:* INTERRUPT FLAG REGISTER -- 'IFR' -- READ ONLY
:*****

```

```

1925
1926
1927
1928
1929           120015   VIAIFR = 120015          ;ADDRESS OF THIS REGISTER -- HEX = A0XD
1930
1931           000200   FLGIRQ = BIT7             ;SET WHEN A FLAG IN THIS REG. GOES HIGH AND
1932                                     ;ITS CORRESPONDING BIT IN VIAIER IS SET.
1933                                     ;(I.E. VIAIER IS THE ENABLE REGISTER FOR THE
1934                                     ;FOR THE SETTING OF IRQ AND THE ISSUANCE OF
1935                                     ;AN INTERRUPT TO THE 6502 WHEN IRQ IS SET.)
1936
1937           000100   FLGT1 = BIT6             ;TIMEOUT OF TIMER 1
1938           000040   FLGT2 = BIT5             ;TIMEOUT OF TIMER 2
1939           000020   FLGCB1 = BIT4            ;ACTIVE TRANSITION OF PIN 18 (CB1)
1940           000010   FLGCB2 = BIT3            ;ACTIVE TRANSITION OF PIN 19 (CB2)
1941           000004   FLGSR = BIT2             ;COMPLETION OF 8 SHIFTS
1942           000002   FLGCA1 = BIT1            ;ACTIVE TRANSITION OF PIN 40 (CA1)
1943           000001   FLGCA2 = BIT0            ;ACTIVE TRANSITION OF PIN 39 (CA2)
1944
1945
1946

```

```

:*****
:* INTERRUPT ENABLE REGISTER -- 'IER' -- READ/WRITE
:*****

```

```

1947
1948
1949
1950
1951           120016   VIAIER = 120016          ;ADDRESS OF THIS REGISTER -- HEX = A0XE
1952
1953           000200   INTSC = BIT7             ;CONTROLS THE SETTING OR CLEARING OF BITS IN
1954                                     ;THE REST OF IER. IF = 0 THE OTHER BITS IN
1955                                     ;THIS REG., IF SET, WILL CLEAR THEIR RESPECTIVE

```

CVDMCA.P11

10-DEC-80 09:14

REGISTER DEFINITIONS -- 6522 VIA CHIP

1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980

: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.

CVDMCA.P11 10-DEC-80 09:14

REGISTER DEFINITIONS -- MISC

.SBTTL REGISTER DEFINITIONS -- MISC

```

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

```

1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006

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)

```

CVDMCA.P11 10-DEC-80 09:14

GLOBAL DATA SECTION

.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

; STORAGE FOR DEVICE CSR REGISTERS

WSR0:
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.

2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018 002246
2019 002246
2020 002246 000000
2021 002250 000000
2022 002252 000000
2023 002254 000000
2024
2025
2026
2027
2028 002256
2029 002256 000000
2030 002260
2031 002260 000000
2032 002262
2033 002262 000000
2034 002264
2035 002264 000000
2036 002266
2037 002266 000000
2038 002270
2039 002270 000000
2040 002272
2041 002272 000000
2042 002274
2043 002274 000000
2044 002276 000000
2045 002300 000000
2046 002302 000000
2047 002304 000000
2048 002306 000000
2049 002310 000000
2050 002312 000000
2051 002314 000000
2052
2053 002316 000010
2054
2055
2056 002336 000020

CVDMCA.P11 10-DEC-80 09:14

GLOBAL DATA SECTION

```

2057 ;*****
2058 ;* MISCELLANEOUS STORAGE
2059 ;*****
2060 002376 000000 TDATA: .WORD 0 ;TEST DATA
2061 002400 000000 GDATA: .WORD 0 ;GOOD DATA
2062 002402 000000 BDATA: .WORD 0 ;BAD DATA
2063 002404 000000 XDATA: .WORD 0 ;EXCLUSIVE-OR BETWEEN GOOD AND BAD DATA
2064 002406 000000 SCRACH: .WORD 0 ;GEN'L PURPOSE SCRATCH WORD
2065 002410 000000 LOGDEV: .WORD 0 ;LOGICAL DEVICE NUMBER
2066 002412 000000 REGNUM: .WORD 0 ;CONTAINS A DEVICE REGISTER NUMBER
2067 002414 000000 PSTACK: .WORD 0 ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
2068 002416 000000 PRIOR: .WORD 0 ;CPU PRIORITY FOR PRINTOUT
2069 002420 000000 SUBRPC: .WORD 0 ;PC OF SUBR CALL FOR ERROR REPORTS
2070 002422 000000 INTFLG: .WORD 0 ;INTERRUPT RECEIVED FLAGS
2071 ; BIT 0 FOR TX, BIT 1 FOR RCV
2072 002424 000000 ERRFLG: .WORD 0 ;SUBROUTINE ERROR FLAG
2073 002426 000000 TIMFLG: .WORD 0 ;EVENT TIME-OUT FLAG
2074 002430 000000 RETADR: .WORD 0 ;SUBR ERROR RETURN ADDRESS
2075 002432 000000 REDBYT: .WORD 0 ;LO BYTE CONTAINS BYTE READ FROM LU REG
2076 002434 000000 WRIBYT: .WORD 0 ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
2077 002436 000000 LOADAT: .WORD 0 ;CONTAINS TEST DATA LOADED INTO REG
2078 002440 000000 GOODAT: .WORD 0 ;STORAGE FOR EXPECTED DATA
2079 002442 000000 BADDAT: .WORD 0 ;STORAGE FOR ACTUAL DATA
2080 002444 000000 FRSTIM: .WORD 0 ;FLAG=0 IF PROGRAM JUST LOADED
2081 002446 000000 SAVE4: .WORD 0 ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
2082 002450 000000 SAVE6: .WORD 0 ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
2083 002452 000000 ERROR1: .WORD 0 ;SUBR ERR. BIT FLAGS (DEF'D IN GLOBAL EQUATES)
2084 002454 000000 CHPTYP: .WORD 0 ;USYRT CHIP TYPE, =0 FOR SMC, ELSE =1
2085 002456 000000 SAVLEN: .WORD 0 ;SAVED TX AND RCV CHAR LENGTHS
2086 002460 000000 DEVMAP: .WORD 0 ;BIT MAP OF ACTIVE DEVICES
2087 002462 000000 DEVPTR: .WORD 0 ;DEVICE MAP BIT POINTER
2088 002464 000000 UNIT: .WORD 0 ;CONTAINS UNIT NO. (1 TO N)
2089 002466 000000 STARES: .WORD 0 ;FLAG TO SHOW NO. OF PASSES SINCE STA OR RES
2090 002470 000000 TSTNUM: .WORD 0 ;NO. OF CURRENT TEST (FOR SOME TESTS)
2091

```

CVDMA.P11 10-DEC-80 09:14

GLOBAL DATA SECTION

```

2092
2093 002472
2094 002472
2095 002472 160020
2096 002474 160021
2097 002476
2098 002476 160022
2099 002500 160023
2100 002502
2101 002502 160024
2102 002504 160025
2103 002506
2104 002506 160026
2105 002510 160027
2106 002512
2107 002512 160030
2108 002514 160031
2109 002516
2110 002516 160032
2111 002520 160033
2112 002522
2113 002522 160034
2114 002524 160035
2115 002526
2116 002526 160036
2117 002530 160037
2118
2119 002532 000300
2120 002534 000304
2121 002536 000240
2122 002540 000000
2123 002542 000000
2124 002544 000000
2125 002546 000000
2126 002550 000001
2127
2128

```

```

***** CURRENT DEVICE PARAMETERS *****
BSEL0:
SEL0:
MPCSR: .WORD 160020 :POINTER TO DMV11 CSR'S
BSEL1: .WORD 160021 :POINTER TO BSEL1
BSEL2:
SEL2: .WORD 160022 :POINTER TO SEL2
BSEL3: .WORD 160023 :POINTER TO BSEL3
BSEL4:
SEL4: .WORD 160024 :POINTER TO SEL4
BSEL5: .WORD 160025 :POINTER TO BSEL5
BSEL6:
SEL6: .WORD 160026 :POINTER TO SEL6
BSEL7: .WORD 160027 :POINTER TO BSEL7
BSEL10:
SEL10: .WORD 160030 :POINTER TO SEL10
BSEL11: .WORD 160031 :POINTER TO BSEL11
BSEL12:
SEL12: .WORD 160032 :POINTER TO SEL12
BSEL13: .WORD 160033 :POINTER TO BSEL13
BSEL14:
SEL14: .WORD 160034 :POINTER TO SEL14
BSEL15: .WORD 160035 :POINTER TO BSEL15
BSEL16:
SEL16: .WORD 160036 :POINTER TO SEL16
BSEL17: .WORD 160037 :POINTER TO BSEL17

MPIVEC: .WORD 300 :DMV11 INPUT INTERRUPT VECTOR
MPOVEC: .WORD 304 :DMV11 OUTPUT INTERRUPT VECTOR
MPRIOR: .WORD 240 :DMV11 DEVICE PRIORITY
LUSW1: .WORD 0 :LINE UNIT SWITCH PACK #1
LUSW2: .WORD 0 :LINE UNIT SWITCH PACK #2
BRDTYP: .WORD 0 :0=M8064, 1=M8053/V.35, 2=M8053/EIA
TSTCON: .WORD 0 :TEST CONNECTOR INDICATOR
BDRATE: .WORD 1 :BAUD RATE = 56 K
: 0 = 19.2 K
: 1 = 56 K

```


CVDMA.P11 10-DEC-80 09:14

GLOBAL DATA SECTION

```

2129
2130 002552 120400
2131 002554 120401
2132 002556 120402
2133 002560 120403
2134 002562 120404
2135 002564 120405
2136 002566 120407
2137 002570 122000
2138
2139
2140 002572 000010
2141
2142
2143 002602 000000
2144 002604 000000
2145 002606 000000
2146 002610 000000
2147 002612 000000
2148 002614 000000
2149 002616 000000
2150 002620 000000
2151
2152
2153 002622 000000
2154 002624 000000
2155 002626 000000
2156 002630 000000
2157 002632 000000
2158 002634 000000
2159 002636 000000
2160 002640 000000
2161
2162
2163 002642
2164 002642 377
2165 002643 000
2166 002644 000
2167 002645 360
2168 002646 000
2169 002647 000
2170 002650 347
2171
2172 002651 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

```

CVDMCA.P11 10-DEC-80 09:14

DATA TEST PATTERNS

2173		
2174		
2175	002652	
2176	002652	377
2177	002653	377
2178	002654	377
2179	002655	377
2180	002656	377
2181	002657	377
2182	002660	377
2183	002661	366
2184		
2185		
2186	002662	
2187	002662	000
2188	002663	000
2189	002664	000
2190	002665	000
2191	002666	000
2192	002667	000
2193	002670	000
2194	002671	110
2195		
2196		
2197	002672	
2198	002672	000
2199	002673	001
2200	002674	003
2201	002675	004
2202	002676	005
2203	002677	007
2204	002700	100
2205	002701	101
2206	002702	103
2207	002703	104
2208	002704	105
2209	002705	107
2210	002706	000
2211	002707	017
2212	002710	027
2213	002711	041
2214	002712	200
2215	002713	277
2216	002714	103
2217	002715	144
2218	002716	115
2219	002717	157
2220	002720	000
2221		
2222		
2223	002721	
2224	002721	125
2225	002722	252
2226	002723	000
2227	002724	377
2228	002725	000

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

.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 H *****
 PATH:

.BYTE	125
.BYTE	252
.BYTE	000
.BYTE	377
.BYTE	000

CVDMA.P11 10-DEC-80 09:14

DATA TEST PATTERNS

2229	002726	001	.BYTE	001
2230	002727	002	.BYTE	002
2231	002730	004	.BYTE	004
2232	002731	010	.BYTE	010
2233	002732	020	.BYTE	020
2234	002733	040	.BYTE	040
2235	002734	100	.BYTE	100
2236	002735	200	.BYTE	200
2237	002736	000	.BYTE	000
2238	002737	377	.BYTE	377
2239	002740	376	.BYTE	376
2240	002741	375	.BYTE	375
2241	002742	373	.BYTE	373
2242	002743	367	.BYTE	367
2243	002744	357	.BYTE	357
2244	002745	337	.BYTE	337
2245	002746	277	.BYTE	277
2246	002747	177	.BYTE	177
2247	002750	377	.BYTE	377
2248	002751	000	.BYTE	000
2249				
2250				
2251	002752			
2252	002752	000	.BYTE	000
2253	002753	041	.BYTE	041
2254	002754	102	.BYTE	102
2255	002755	143	.BYTE	143
2256	002756	204	.BYTE	204
2257	002757	245	.BYTE	245
2258	002760	306	.BYTE	306
2259	002761	347	.BYTE	347
2260	002762	000	.BYTE	000
2261	002763	001	.BYTE	001
2262	002764	002	.BYTE	002
2263	002765	004	.BYTE	004
2264	002766	040	.BYTE	040
2265	002767	100	.BYTE	100
2266	002770	200	.BYTE	200
2267	002771	000	.BYTE	000
2268	002772	346	.BYTE	346
2269	002773	345	.BYTE	345
2270	002774	343	.BYTE	343
2271	002775	307	.BYTE	307
2272	002776	247	.BYTE	247
2273	002777	147	.BYTE	147
2274	003000	347	.BYTE	347
2275	003001	242	.BYTE	242
2276	003002	105	.BYTE	105
2277	003003	347	.BYTE	347
2278	003004	010	.BYTE	010
2279	003005	020	.BYTE	020
2280	003006	367	.BYTE	367
2281	003007	357	.BYTE	357
2282	003010	030	.BYTE	030
2283	003011	027	.BYTE	027
2284	003012	377	.BYTE	377

***** DATA PATTERN I *****
 PATI:

CVDMCA.P11 10-DEC-80 09:14

DATA TEST PATTERNS

2285
 2286
 2287 003013
 2288 003013 000
 2289 003014 000
 2290 003015 001
 2291 003016 002
 2292 003017 004
 2293 003020 020
 2294 003021 040
 2295 003022 010
 2296
 2297
 2298 003023
 2299 003023 000
 2300 003024 377
 2301 003025 376
 2302 003026 375
 2303 003027 373
 2304 003030 376
 2305 003031 177
 2306 003032 377
 2307 003033 000
 2308 003034 001
 2309 003035 002
 2310 003036 004
 2311 003037 010
 2312 003040 200
 2313 003041 125
 2314 003042 252
 2315 003043 000
 2316
 2317
 2318 003044
 2319 003044 000
 2320 003045 017
 2321 003046 016
 2322 003047 015
 2323 003050 013
 2324 003051 016
 2325 003052 017
 2326 003053 017
 2327 003054 000
 2328 003055 001
 2329 003056 002
 2330 003057 004
 2331 003060 010
 2332 003061 000
 2333 003062 005
 2334 003063 012
 2335 003064 000

***** 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
 .BYTE 376
 .BYTE 177
 .BYTE 377
 .BYTE 000
 .BYTE 001
 .BYTE 002
 .BYTE 004
 .BYTE 010
 .BYTE 200
 .BYTE 125
 .BYTE 252
 .BYTE 000

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

.BYTE 000
 .BYTE 017
 .BYTE 016
 .BYTE 015
 .BYTE 013
 .BYTE 016
 .BYTE 017
 .BYTE 017
 .BYTE 000
 .BYTE 001
 .BYTE 002
 .BYTE 004
 .BYTE 010
 .BYTE 000
 .BYTE 005
 .BYTE 012
 .BYTE 000

CVDMCA.P11

10-DEC-80 09:14

DATA TEST PATTERNS

2336
 2337
 2338 003065 000
 2339 003066 003
 2340 003067 014
 2341 003070 060
 2342 003071 001
 2343 003072 007
 2344 003073 037
 2345 003074 177
 2346
 2347
 2348 003075 000
 2349 003076 140
 2350 003077 030
 2351 003100 006
 2352 003101 100
 2353 003102 160
 2354 003103 174
 2355 003104 177
 2356 003105
 2357 003106

:***** DATA PATTERN Q *****

PATQ: .BYTE 000
 .BYTE 003
 .BYTE 014
 .BYTE 060
 .BYTE 001
 .BYTE 007
 .BYTE 037
 .BYTE 177

:***** DATA PATTERN INVERTED Q *****

PATQB: .BYTE 000 :INVERTED 000 (7 BIT)
 .BYTE 140 :INVERTED 003 (7 BIT)
 .BYTE 030 :INVERTED 014 (7 BIT)
 .BYTE 006 :INVERTED 060 (7 BIT)
 .BYTE 100 :INVERTED 001 (7 BIT)
 .BYTE 160 :INVERTED 007 (7 BIT)
 .BYTE 174 :INVERTED 037 (7 BIT)
 .BYTE 177 :INVERTED 177 (7 BIT)

ENDPAT:
.EVEN

CVDMCA.P11

10-DEC-80 09:14

DATA TEST PATTERNS

2358
2359
2360
2361
2362
2363
2364
2365
2366

003106 000100

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

CVDMCA.P11 10-DEC-80 09:14

GLOBAL TEXT SECTION

2367
 2368
 2369
 2370
 2371
 2372
 2373
 2374
 2375
 2376
 2377
 2378 003306
 2379 003306
 2380 003306 034115 032460 020063
 2381 003314 051117 046440 030070
 2382 003322 032066 000
 2383 003326
 2384
 2385
 2386
 2387
 2388
 2389 000012
 2390 003326
 2391 003326
 2392 003326 046504 026526 030461
 2393 003334 046040 047111 020105
 2394 003342 047125 052111 052040
 2395 003350 051505 051524 026440
 2396 003356 050040 051101 020124
 2397 003364 020061 043117 031440
 2398 003372 000
 2399 003374
 2400 000010
 2401
 2402

.SBTTL GLOBAL TEXT SECTION
 :XXX
 :% THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
 :% MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
 :% MORE THAN ONE TEST.
 :XXX
 :*****
 :* NAMES OF DEVICES SUPPORTED BY PROGRAM
 :*****
 : DEVTYP <M8053 OR M8064>
 LSDVTYP::
 .ASCIZ /M8053 OR M8064/
 .EVEN
 :*****
 :* TITLE OF PROGRAM
 :*****
 .RADIX 10.
 DESCRIPT <DMV-11 LINE UNIT TESTS - PART 1 OF 3>
 L\$DESC::
 .ASCIZ /DMV-11 LINE UNI
 .EVEN
 .RADIX 8.

CVDPCA.P11 10-DEC-80 09:14

GLOBAL SUBROUTINE SECTION

.SBTTL GLOBAL SUBROUTINE SECTION

2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448

```

.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>
-----*****

```

```

2424 003374 112777 000301 177072 MSTCLR: MOVB  #RUN!MCLR!MREQ,@BSEL1 ;INITIATE M-LOOP
2425
2426 003402 010346          MOV    R3,-(SP)
2427 003404 012703 000030          MOV    #24,R3          ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
2428 003410 077301          1$:   SOB   R3,1$
2429 003412 012603          MOV    (SP)+,R3
2430
2431 003414 132777 000200 177054          BITB  #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
2432 003422 001023          BNE   5$             ;YES, GOOD. RETURN
2433 003424 004737 004210          JSR   PC,GETWSR     ;GET BYTE SELECT REGISTERS
2434 003430 012737 000301 002400          MOV   #RUN!MCLR!MREQ,GDATA ;IDENTIFY REQUESTED FUNCTION
2435 003436          GTDF  EM3,ERR4    ;'MRDY' TIMEOUT
2436
2437 003436 012737 000001 002246          ;
2438 003444 012737 000001 002250          MOV   #T.EDF,ERRTYP
2439 003452 012737 013211 002252          MOV   #1,ERRNBR
2440 003460 012737 017030 002254          MOV   #EM3,ERRMSG
2441 003466 000261          MOV   #ERR4,ERRBLK
2442 003470 000401          SEC
2443 003472 000241          5$:   BR    9$          ;SET CARRY TO INDICATE ERROR
2444 003474 000207          9$:   CLC          ;EXIT WITH THE 'ERROR' FLAG (CARRY BIT) SET
2445
2446
2447
2448          ;CLEAR C BIT FOR NO ERRORS
          ;RETURN

```


CVDMCA.P11 10-DEC-80 09:14

....M-LOOP -- READ

.SBTTLM-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>

2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466 003476 012577 177000
2467 003502 112777 000001 176766
2468
2469 003510 010346
2470 003512 012703 000050
2471 003516 077301
2472 003520 012603
2473
2474 003522 132777 000200 176746
2475 003530 001023
2476
2477 003532 004737 004210
2478 003536 012737 000001 002400
2479 003544
2480
2481 003544 012737 000001 002246
2482 003552 012737 000002 002250
2483 003560 012737 013235 002252
2484 003566 012737 017030 002254
2485 003574 000261
2486 003576 000401
2487
2488 003600 000241
2489 003602 117735 176700
2490 003606 000205
2491
2492
2493
2494

READ: MOV (R5)+,@SEL4 ;SETUP SOURCE POINTER
MOV #REDLOC,@BSEL2 ;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,@BSEL2 ;DID THE M-LOOP FINISH
BNE 5\$;YES, GOOD. RETURN

JSR PC,GETWSR ;GET BYTE SELECT REGISTERS
MOV #REDLOC,GDATA ;IDENTIFY REQUESTED FUNCTION
GDF EM4,ERR4 ;"MRDY" TIMEOUT
; QUEUE "DEVICE FATAL" ERROR # 2
MOV #T.EDF,ERRTYP
MOV #2,ERRNER
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\$: MOV @BSEL6,@(R5)+ ;PUT DATA WHERE CALLER WANTS IT
RTS R5 ;RETURN

CVDMCA.P11 10-DEC-80 09:14

....M-LOOP -- READ IMMEDIATE

.SBTTLM-LOOP -- READ IMMEDIATE

: READI - READ IMMEDIATE THE SPECIFIED ADDRESS WITHIN THE DMV-11 (M8053)

: 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>

2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512 003610
2513 003610 012577 176666
2514 003614 112777 000001 176654
2515
2516 003622 010346
2517 003624 012703 000050
2518 003630 077301
2519 003632 012603
2520
2521 003634 132777 000200 176634
2522 003642 001023
2523
2524 003644 004737 004210
2525 003650 012737 000001 002400
2526 003656
2527
2528 003656 012737 000001 002246
2529 003664 012737 000003 002250
2530 003672 012737 013235 002252
2531 003700 012737 017030 002254
2532 003706 000261
2533 003710 000401
2534
2535 003712 000241
2536 003714 017725 176566
2537 003720 000205
2538
2539
2540
2541

READI:

MOV (R5)+,@SEL4 ;SETUP SOURCE POINTER
MOVB #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

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

5\$: CLC ;INDICATE 'NO ERROR'
6\$: MOV @SEL6,(R5)+ ;PUT DATA WHERE CALLER WANTS IT
RTS R5 ;RETURN

MOV #T.EDF,ERRTYP
MOV #3,ERRNBR
MOV #EM4,ERRMSG
MOV #ERR4,ERRBLK

CVDMCA.P11 10-DEC-80 09:14

....M-LOOP -- WRITE

.SBTTLM-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
MOV @ (R5)+,@SEL6 :MAKE DATA AVAILABLE TO M-LOOP
BR MLWRI :THE REST OF THIS ROUTINE IS THE SAME AS 'WRITEI'

2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565

003722 012577 176554
003726 113577 176554
003732 000404

CVDMCA.P11 10-DEC-80 09:14

....M-LOOP -- WRITE IMMEDIATE

.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>

2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611

003734
003734 012577 176542
003740 012577 176542
003744 112777 000002 176524
003752 010346
003754 012703 000050
003760 077301
003762 012603
003764 132777 000200 176504
003772 001023
003774 004737 004210
004000 012737 000002 002400
004006
004006 012737 000001 002246
004014 012737 000004 002250
004022 012737 013235 002252
004030 012737 017030 002254
004036 000261
004040 000401
004042 000241
004044 000205

WRITEI:
MOV (R5)+,@SEL4 ;SETUP SOURCE POINTER
MOV (R5)+,@SEL6 ;MAKE DATA AVAILABLE TO M-LOOP
MLWRI: MOVB #WRILOC,@BSEL2 ;TELL M-LOOP TO WRITE THE DATA
MOV R3,-(SP)
MOV #4,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 #WRILOC,GDATA ;IDENTIFY REQUESTED FUNCTION
GTFD EM4,ERR4 ;"MRDY" TIMEOUT
; QUEUE "DEVICE FATAL" ERROR # 4
MOV #T.EDF,ERRTYP
MOV #4,ERRNBR
MOV #EM4,ERRMSG
MOV #ERR4,ERRBLK
SEC
BR 6\$;INDICATE AN ERROR HAS BEEN STACKED
;RETURN WITH THAT INDICATION
5\$: CLC
6\$: RTS R5 ;INDICATE 'NO ERROR'
;RETURN

CVDMCA.P11 10-DEC-80 09:14

....GETBSR -- GET BYTE SELECT REGISTERS

.SBTTLGETBSR -- 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 # # # # ##### # #

-----*****

2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628 004046 117737 176420 002256
2629 004054 117737 176414 002260
2630 004062 117737 176410 002262
2631 004070 117737 176404 002264
2632 004076 117737 176400 002266
2633 004104 117737 176374 002270
2634 004112 117737 176370 002272
2635 004120 117737 176364 002274
2636 004126 117737 176360 002276
2637 004134 117737 176354 002300
2638 004142 117737 176350 002302
2639 004150 117737 176344 002304
2640 004156 117737 176340 002306
2641 004164 117737 176334 002310
2642 004172 117737 176330 002312
2643 004200 117737 176324 002314
2644 004206 000207
2645
2646
2647
2648
2649 004210 017737 176256 002256
2650 004216 017737 176254 002260
2651 004224 017737 176252 002262
2652 004232 017737 176250 002264
2653 004240 017737 176246 002266
2654 004246 017737 176244 002270
2655 004254 017737 176242 002272
2656 004262 017737 176240 002274
2657 004270 000207

```

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

```

.SBTTLGETWSR -- 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

```

CVDMCA.P11 10-DEC-80 09:14

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

2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713

004272 010037 004306
004276 010037 004324
004302 004537 003722
004306 000000
004310 002376
004312 103431
004314 005037 002402
004320 004537 003476
004324 000000
004326 002402
004330 103422
004332 123737 002400 002402
004340 000241
004342 001415
004344
004344 012737 000001 002246
004352 012737 000005 002250
004360 012737 013366 002252
004366 012737 017252 002254
004374 000261
004376 000207

.SBTTLSTUREG -- 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
4\$: 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
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 ;INDICATE THAT AN ERROR WAS DETECTED
RTS PC

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

STALL: RTS PC

2714
2715

CVDMA.P11 10-DEC-80 09:14

```

2716
2717
2718
2719
2720
2721
2722
2723
2724
2725 004402 012737 002316 004444
2726 004410 012737 120400 004442
2727
2728 004416 005037 002334
2729 004422 004537 003476
2730 004426 122000
2731 004430 002334
2732
2733 004432 005077 000006
2734 004436 004537 003476
2735 004442 000000
2736 004444 000000
2737
2738 004446 005237 004442
2739 004452 023727 004442 120406
2740 004460 001772
2741
2742 004462 062737 000002 004444
2743 004470 023727 004442 120410
2744 004476 001355
2745
2746 004500 000207
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756 004502 012737 002336 004530
2757 004510 012737 120000 004526
2758 004516 005077 000006
2759 004522 004537 003476
2760 004526 000000
2761 004530 000000
2762 004532 005237 004526
2763 004536 062737 000002 004530
2764 004544 023727 004526 120020
2765 004552 001361
2766 004554 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
CLR UREGS+14. ;CLEAR STORAGE WORD
JSR R5,READ ;READ THE USYRT STATUS REGISTER
.WORD USTATR ;STATUS REGISTER'S ADDRESS WITHIN DMV-11
.WORD UREGS+14. ;ADDRESS ALLOCATED TO THAT REG. W/IN 'UREGS'
3$: CLR @5$ ;CLEAR STORAGE WORD
JSR R5,READ ;READ A LINE UNIT REG
4$: .WORD 0 ;REGISTER ADDRESS GOES HERE
5$: .WORD 0 ;STORAGE ADRS IN TABLE GOES HERE
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$: CLR @5$ ;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

```


CVDMCA.P11 10-DEC-80 09:14

....INITT1 -- INITIALIZE TIMER #1

2767
 2768
 2769
 2770
 2771
 2772
 2773
 2774
 2775
 2776
 2777
 2778
 2779
 2780
 2781
 2782
 2783
 2784
 2785
 2786
 2787
 2788
 2789
 2790
 2791
 2792
 2793
 2794 004556 010146
 2795 004560 012537 004702
 2796 004564 012537 004730
 2797 004570 111501
 2798 004572 143701 000077
 2799 004576 010137 004672
 2800 004602 112501
 2801
 2802 004604 106301
 2803 004606 106301
 2804
 2805
 2806
 2807
 2808 004610 143701 000177
 2809 004614 153701 000100
 2810 004620 010137 004632
 2811
 2812 004624 004537 003734
 2813 004630 120016
 2814 004632 000000
 2815
 2816 004634 004537 003610
 2817 004640 120013
 2818 004642 000000
 2819
 2820 004644 013701 004642
 2821 004650 143701 000300
 2822 004654 053701 004672

```

.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
MOV R1, R5 ;GET & PROCESS BITS FOR ACR 6 & 7
BICB 077, R1
MOV R1, 4$ ;SETUP CALL SET ACR'S BITS 6 & 7
MOV (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 & 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
;WRITE TO
;THE VIA'S IER
2$: VIAIER .WORD 0 ;INTERRUPT ENABLE/DISABLE INFORMATION
;READ THE CURRENT SETTING OF
;THE VIA'S ACR
3$: VIAACR .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

```

CVDMA.P11 10-DEC-80 09:14

....INITT1 -- INITIALIZE TIMER #1

```

2823 004660 010137 004672      MOV      R1,4$      ;PASS THE NEW REG. SETTING TO APPROPRIATE CALL
2824                               ;WRITE TO
2825 004664 004537 003734      JSR      R5,WRITEI ;THE VIA'S ACR
2826 004670 120013                VIAACR                ;THE NEW REGISTER SETTING
2827 004672 000000                .WORD  0
2828                               ;WRITE TO
2829 004674 004537 003734      JSR      R5,WRITEI ;LOW ORDER LATCH REGISTER (T1L-L)
2830 004700 120006                VIAT1C                ;THE VALUE PASSED
2831 004702 000000                .WORD  0
2832                               ;SETUP FOR AND
2833 004704 113737 004703 004720  MOVB     7$+1,8$    ;WRITE TO
2834 004712 004537 003734      JSR      R5,WRITEI ;HIGH ORDER LATCH REGISTER (T1L-H)
2835 004716 120007                VIAT1D                ;THE VALUE PASSED
2836 004720 000000                .WORD  0
2837                               ;WRITE TO
2838 004722 004537 003734      JSR      R5,WRITEI ;LOW ORDER LATCH & COUNTER (T1L-L & T1C-L)
2839 004726 120004                VIAT1A                ;THE VALUE PASSED
2840 004730 000000                .WORD  0
2841                               ;SETUP FOR AND
2842 004732 113737 004731 004746  MOVB     10$+1,11$  ;WRITE TO
2843 004740 004537 003734      JSR      R5,WRITEI ;HIGH ORDER COUNTER (T1C-H) <ALSO STARTS CTR>
2844 004744 120005                VIAT1B                ;THE VALUE PASSED
2845 004746 000000                .WORD  0
2846                               ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST RETURN!
2847
2848                               ;BUT FIRST RESTORE R1
2849 004750 012601                MOV      (SP)+,R1    ;AND PUT R5 BACK ON A WORD BOUNDARY (THE LAST
2850 004752 005205                INC      R5          ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2851
2852                               ;NOW, RETURN
2853 004754 000205                RTS      R5
2854
2855

```

CVDMCA.P11 10-DEC-80 09:14

....INITT2 -- INITIALIZE TIMER #2

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

004756	010146		
004760	012537	005100	
004764	111501		
004766	143701	000337	
004772	010137	005070	
004776	112501		
005000	106301		
005002	106301		
005004	106301		
005006	143701	000177	
005012	153701	000040	
005016	010137	005030	
005022	004537	003734	
005026	120016		
005030	000000		
005032	004537	003610	
005036	120013		
005040	000000		
005042	013701	005040	
005046	143701	000040	
005052	053701	005070	
005056	010137	005070	

```
.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
        MOVB   (R5), R1      ;GET & PROCESS BIT FOR ACR 5
        BICB   337, R1
        MOV     R1, 4$
        MOVB   (R5)+, R1     ;SETUP CALL TO SET OR CLEAR ACR'S BIT 5
                               ;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
                               ;WRITE TO
        JSR     R5, WRITEI   ;THE VIA'S IER
        VIAIER .WORD 0     ;INTERRUPT ENABLE/DISABLE INFORMATION
2$:
        JSR     R5, READI   ;READ THE CURRENT SETTING OF
        VIAACR .WORD 0     ;THE VIA'S ACR
3$:
        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
```

CVDMCA.P11 10-DEC-80 09:14

....INITT2 -- INITIALIZE TIMER #2

```

2912 005062 004537 003734      JSR    R5,WRITEI      :WRITE TO
2913 005066 120013              VIAACR                 :THE VIA'S ACR
2914 005070 000000      4$:  .WORD    0       :THE NEW REGISTER SETTING
2915
2916 005072 004537 003734      JSR    R5,WRITEI      :WRITE TO
2917 005076 120010              VIAT2A                 :LOW ORDER LATCH & COUNTER (T2L-L & T2C-L)
2918 005100 000000      10$: .WORD    0       :THE VALUE PASSED
2919
2920 005102 113737 005101 005116  MOVB   10$+1,11$      :SETUP FOR AND
2921 005110 004537 003734      JSR    R5,WRITEI      :WRITE TO
2922 005114 120011              VIAT2B                 :HIGH ORDER COUNTER (T2C-H) <ALSO STARTS CTR>
2923 005116 000000      11$: .WORD    0       :THE VALUE PASSED
2924
2925      : DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2926
2927 005120 012601      MOV    (SP)+,R1      :BUT FIRST RESTORE R1
2928 005122 005205      INC    R5            :AND PUT R5 BACK ON A WORD BOUNDARY (THE LAST
2929                                     :PASSED PARAM. WAS A BYTE, NOT A WORD!)
2930
2931 005124 000205      RTS    R5            :THEN RETURN
2932

```

CVDMCA.P11 10-DEC-80 09:14

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

.SBTTLRSTCHK -- 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:

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
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986

005126
005126 010146
005130 010246
005132 004537 003734
005136 120000
005140 000031
005142 004537 003734
005146 120000
005150 000030
005152 005001
005154 012702 002662
005160 016137 002552 005172 6\$:
005166 004537 003610
005172 000000 7\$:
005174 000000 8\$:
005176 123722 005174
005202 001432
005204 010137 002412
005210 006237 002412
005214 005037 002400
005220 116237 177777 002400
005226 013737 005174 002402
005234
005234 012737 000001 002246
005242 012737 000006 002250
005250 012737 013142 002252
005256 012737 017372 002254
005264 000261
005266 000406
005270 062701 000002 9\$:
005274 020127 000020
005300 002727
005302 000241
005304 012602 10\$:
005306 012601
005310 000205

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

CVDMA.P11 10-DEC-80 09:14

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

2987
 2988
 2989
 2990 005312 010146
 2991 005314 012701 000005
 2992 005320 077101
 2993 005322 012601
 2994 005324 000207
 2995
 2996
 2997
 2998
 2999
 3000
 3001
 3002
 3003
 3004
 3005
 3006
 3007
 3008
 3009
 3010
 3011
 3012
 3013 005326
 3014 005326 004537 003734
 3015 005332 120002
 3016 005334 000377
 3017 005336 004537 003734
 3018 005342 120003
 3019 005344 000001
 3020 005346 004537 003734
 3021 005352 120017
 3022 005354 000000
 3023 005356 004537 003734
 3024 005362 120000
 3025 005364 000030
 3026 005366 004537 003734
 3027 005372 120013
 3028 005374 000350
 3029 005376 004537 003734
 3030 005402 120014
 3031 005404 000022
 3032 005406 004537 003734
 3033 005412 120016
 3034 005414 000177
 3035 005416 000207
 3036
 3037

 : * 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
  
```

CVDMCA.P11 10-DEC-80 09:14

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

3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071
3072
3073
3074
3075
3076
3077
3078
3079
3080
3081
3082
3083
3084
3085

005420	004737	003374	
005424	004737	005326	
005430	000207		
005432			
005432	004537	003610	
005436	122000		
005440	000000		
005442	122537	005440	
005446	000241		
005450	001430		
005452	012737	000007	002412
005460	016537	177777	002400
005466	005037	002402	
005472	113737	005440	002402
005500			
005500	012737	000001	002246
005506	012737	000007	002250
005514	012737	013760	002252
005522	012737	017372	002254
005530	000261		
005532	005205		
005534	000205		

```
.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$,BDATA
;STACK 'USYRT STATUS INCORRECT' ERROR
        GTDF EM68,ERR10
; QUEUE 'DEVICE FATAL' ERROR # 7
        MOV #T.EDF,ERRTYP
        MOV #7,ERRNER
        MOV #EM68,ERRMSG
        MOV #ERR10,ERRBLK
2$: SEC ;SET C BIT FOR ERROR
        INC R5 ;INCREMENT R5 PAST ARGUMENT
        RTS R5 ;RETURN
```

CVDMCA.P11 10-DEC-80 09:14

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

3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096 005536
3097 005536 012737 000007 002412
3098 005544 004537 003610
3099 005550 122000
3100 005552 000000
3101 005554 032725 000001
3102 005560 001422
3103 005562 132737 000004 005552
3104 005570 001040
3105
3106 005572
3107
3108 005572 012737 000001 002246
3109 005600 012737 000010 002250
3110 005606 012737 014007 002252
3111 005614 012737 017722 002254
3112 005622 000261
3113 005624 000423
3114 005626 132737 000004 005552
3115 005634 001416
3116
3117 005636
3118
3119 005636 012737 000001 002246
3120 005644 012737 000011 002250
3121 005652 012737 014025 002252
3122 005660 012737 017722 002254
3123 005666 000261
3124 005670 000401
3125 005672 000241
3126 005674 000205
3127
3128
3129
3130

```

.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
GTFD 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
GTFD 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

```


CVDMCA.P11 10-DEC-80 09:14

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

3131
3132
3133
3134
3135
3136
3137
3138
3139
3140
3141 005676
3142 005676 012737 000007 002412
3143 005704 004537 003610
3144 005710 122000
3145 005712 000000
3146 005714 032725 000001
3147 005720 001422
3148 005722 132737 000040 005712
3149 005730 001040
3150
3151 005732
3152
3153 005732 012737 000001 002246
3154 005740 012737 000012 002250
3155 005746 012737 014047 002252
3156 005754 012737 017722 002254
3157 005762 000261
3158 005764 000423
3159 005766 132737 000040 005712
3160 005774 001416
3161
3162 005776
3163
3164 005776 012737 000001 002246
3165 006004 012737 000013 002250
3166 006012 012737 014065 002252
3167 006020 012737 017722 002254
3168 006026 000261
3169 006030 000401
3170 006032 000241
3171 006034 000205
3172
3173
3174
3175

```

.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
GTFD 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
GTFD 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
    
```

CVDMCA.P11 10-DEC-80 09:14

....CKTBMT -- CHECK TRANSMIT BUFFER EMPTY

3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217
3218
3219
3220

006036 012737 000007 002412
006036 004537 003610
006044 004537 003610
006050 122000
006052 000000
006054 032725 000001
006060 001422
006062 132737 000100 006052
006070 001040
006072
006072 012737 000001 002246
006100 012737 000014 002250
006106 012737 014107 002252
006114 012737 017722 002254
006122 000261
006124 000423
006126 132737 000100 006052
006134 001416
006136
006136 012737 000001 002246
006144 012737 000015 002250
006152 012737 014124 002252
006160 012737 017722 002254
006166 000261
006170 000401
006172 000241
006174 000205

```
.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
GTFD 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
GTFD 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
```

CVDMCA.P11 10-DEC-80 09:14

....CKRDA -- CHECK RECEIVE DATA AVAILABLE

3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265

006176
006176 012737 000007 002412
006204 004537 003610
006210 122000
006212 000000
006214 032725 000001
006220 001422
006222 132737 000200 006212
006230 001040
006232
006232 012737 000001 002246
006240 012737 000016 002250
006246 012737 014145 002252
006254 012737 017722 002254
006262 000261
006264 000423
006266 132737 000200 006212
006274 001416
006276
006276 012737 000001 002246
006304 012737 000017 002250
006312 012737 014161 002252
006320 012737 017722 002254
006326 000261
006330 000401
006332 000241
006334 000205

```
.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
```

CVDMCA.P11 10-DEC-80 09:14

....CKRSA -- CHECK RECEIVER STATUS AVAILABLE

3266
3267
3268
3269
3270
3271
3272
3273
3274
3275
3276 006336
3277 006336 012737 000007 002412
3278 006344 004537 003610
3279 006350 122000
3280 006352 000000
3281 006354 032725 000001
3282 006360 001422
3283 006362 132737 000020 006352
3284 006370 001040
3285
3286 006372
3287
3288 006372 012737 000001 002246
3289 006400 012737 000020 002250
3290 006406 012737 014201 002252
3291 006414 012737 017722 002254
3292 006422 000261
3293 006424 000423
3294 006426 132737 000020 006352
3295 006434 001416
3296
3297 006436
3298
3299 006436 012737 000001 002246
3300 006444 012737 000021 002250
3301 006452 012737 014215 002252
3302 006460 012737 017722 002254
3303 006466 000261
3304 006470 000401
3305 006472 000241
3306 006474 000205
3307
3308

```

.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
GTFD 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
GTFD 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

```

CVDMCA.P11 10-DEC-80 09:14

....CKROR -- CHECK RECEIVER OVERRUN

3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352

```

.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 BIT 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
3$: SEC ;SET C BIT TO FLAG ERROR
BR 4$ ;TAKE ERROR EXIT
4$: CLC ;CLEAR C BIT FOR NO ERRORS
RTS R5 ;RETURN

```

CVDMCA.P11 10-DEC-80 09:14

....CKSEOM -- CHECK RSOM, REOM

3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408

006636 012737 000007 002412
006636 012737 000007 002412
006644 004537 003610
006650 120401
006652 000000
006654 032725 000001
006660 001422
006662 132737 000001 006652
006670 001040
006672
006672 012737 000001 002246
006700 012737 000024 002250
006706 012737 013520 002252
006714 012737 017722 002254
006722 000261
006724 000473
006726 132737 000001 006652
006734 001416
006736
006736 012737 000001 002246
006744 012737 000025 002250
006752 012737 013477 002252
006760 012737 017722 002254
006766 000261
006770 000451
006772 032765 000002 177776
007000 001422
007002 132737 000002 006652
007010 001040
007012
007012 012737 000001 002246
007020 012737 000026 002250
007026 012737 013556 002252
007034 012737 017722 002254
007042 000261
007044 000423
007046 132737 000002 006652

.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

CVDMCA.P11 10-DEC-80 09:14

....CKSEOM -- CHECK RSOM, REOM

```

3409 007054 001416
3410
3411 007056
3412
3413 007056 012737 000001 002246
3414 007064 012737 000027 002250
3415 007072 012737 013535 002252
3416 007100 012737 017722 002254
3417 007106 000261
3418 007110 000401
3419 007112 000241
3420 007114 000205
3421
3422

```

```

;STACK 'REOM NOT CLEARED' MSG
      BEQ      5$
      GTDF     EM30,ERR12
;
      SEC
      BR      6$
5$:   CLC
6$:   RTS     R5

```

```

;BR IF REOM = 0
;
;   QUEUE 'DEVICE FATAL' ERROR # 23
      MOV     #T,EDF,ERRTYP
      MOV     #23,ERRNBR
      MOV     #EM30,ERRMSG
      MOV     #ERR12,ERRBLK
;SET C BIT TO FLAG ERROR
;TAKE ERROR EXIT
;CLEAR C BIT FOR NO ERRORS
;RETURN

```

CVDMCA.P11 10-DEC-80 09:14

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

3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466

007116
007116 012737 000007 002412
007124 004537 003610
007130 122000
007132 000000
007134 032725 000001
007140 001422
007142 132737 000010 007132
007150 001040
007152
007152 012737 000001 002246
007160 012737 000030 002250
007166 012737 014425 002252
007174 012737 017722 002254
007202 000261
007204 000423
007206 132737 000010 007132
007214 001416
007216
007216 012737 000001 002246
007224 012737 000031 002250
007232 012737 014445 002252
007240 012737 017722 002254
007246 000261
007250 000401
007252 000241
007254 000205

```
.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
```


CVDMCA.P11 10-DEC-80 09:14

....SERIAL -- READ/CHECK TX CHARACTER VIA TSO BIT

3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479 007256
3480 007256 010146
3481 007260 010246
3482 007262 010346
3483
3484 007264 005001
3485 007266 012502
3486
3487 007270 006301
3488 007272 004537 011614
3489 007276 000001
3490
3491 007300 004537 007116
3492 007304 000001
3493 007306 103401
3494 007310 005201
3495 007312 077212
3496
3497 007314 012503
3498 007316 020103
3499 007320 001422
3500
3501 007322 010337 002400
3502 007326 010137 002402
3503
3504 007332
3505
3506 007332 012737 000001 002246
3507 007340 012737 000032 002250
3508 007346 012737 014471 002252
3509 007354 012737 020036 002254
3510 007362 000261
3511 007364 000401
3512
3513 007366 000241
3514 007370 012603
3515 007372 012602
3516 007374 012601
3517 007376 000205
3518
3519

```

.SBTTL ....SERIAL -- READ/CHECK TX CHARACTER VIA TSO BIT
*****
* SERIAL - THIS SUBROUTINE SERIALY READS/CLOCKS/CHECKS A CHARACTER FROM
* THE TRANSMIT SERIAL OUT (TSO) BIT OF THE USYRT STATUS REGISTER,
* AND STACKS MESSAGE/SETS 'C' BIT IF AN INCORRECT CHARACTER IS READ.
* NOTE: 'EXPECTED VALUE' ARGUMENT IS ALWAYS READ RIGHT-TO-LEFT.
*
* CALLING SEQUENCE :
* JSR R5,SERIAL
* .WORD <# OF BITS TO BE READ>
* .WORD <EXPECTED VALUE OF SERIAL BIT STREAM>
*****
SERIAL:
MOV R1,-(SP) ;SAVE R1
MOV R2,-(SP) ;SAVE R2 (TICKS)
MOV R3,-(SP) ;SAVE R3 (EXPECTED_WORD)

CLR R1 ;CLEAR ASSEMBLED_WORD
MOV (R5)+,R2 ;GET # OF TICKS

1$: ASL R1 ;SHIFT ASSEMBLED WORD
JSR R5,STEPLU ;CLOCK USYRT ONCE
1

JSR R5,CHKTSO ;CHECK FOR TSO=1
1

BCS 2$ ;BR IF TSO=0
INC R1 ;TSO=1: SET LSB OF ASSEMBLED_WORD
2$: SOB R2,1$ ;LOOP UNTIL NO MORE TICKS

MOV (R5)+,R3 ;GET EXPECTED WORD
CMP R1,R3 ;COMPARE EXPECTED AND ASSEMBLED_WORD
BEQ 3$ ;BR IF CORRECT VALUE READ

MOV R3,GDATA ;EXPECTED WORD => GDATA
MOV R1,BDATA ;ASSEMBLED WORD => BDATA
;*** STACK 'TRANSMISSION ERROR' MSG ***
GTDF EM102,ERR13
; QUEUE 'DEVICE FATAL' ERROR # 26
MOV #T.EDF,ERRTYP
MOV #26,ERRNBR
MOV #EM102,ERRMSG
MOV #ERR13,ERRBLK

SEC ;SET C BIT TO FLAG ERROR
BR .+4 ;TAKE ERROR EXIT

3$: CLC ;CLEAR C BIT FOR NO ERRORS
MOV (SP)+,R3 ;RESTORE REGISTERS
MOV (SP)+,R2
MOV (SP)+,R1

4$: RTS R5 ;RETURN
    
```

CVDMCA.P11 10-DEC-80 09:14

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

3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537 007400
3538 007400 010146
3539 007402 004537 003734
3540 007406 120000
3541 007410 000031
3542 007412 004537 003734
3543 007416 120000
3544 007420 000030
3545 007422 112537 007434
3546 007426 004537 003734
3547 007432 120404
3548 007434 000000
3549 007436 112537 007450
3550 007442 004537 003734
3551 007446 120405
3552 007450 000000
3553 007452 112537 007476
3554 007456 005037 002456
3555 007462 113737 007476 002456
3556 007470 004537 003734
3557 007474 120407
3558 007476 000000
3559 007500 004537 003734
3560 007504 120013
3561 007506 000200
3562 007510 004537 003734
3563 007514 120006
3564 007516 000300
3565 007520 004537 003734
3566 007524 120007
3567 007526 000000
3568 007530 004537 005432
3569 007534 000110
3570 007536 103454
3571
3572 007540 013737 007674 007560
3573 007546 142537 007560
3574
3575 007552 004537 003734

```

.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
1$:
MOVB (R5)+,1$ ;GET VALUE TO LOAD INTO USYRT PCSARL
JSR R5,WRITEI ;LOAD USYRT PCSARL
PCARL
2$:
.MWORD 0
MOVB (R5)+,2$ ;GET VALUE TO LOAD INTO PCSARH
JSR R5,WRITEI ;LOAD USYRT PCSARH
PCARH
3$:
.MWORD 0
MOVB (R5)+,3$ ;GET VALUE TO LOAD INTO PCR
CLR SAVLEN
MOVB 3$,SAVLEN ;SAVE CHAR LENGTH BITS
JSR R5,WRITEI ;LOAD USYRT PCR
PCR
.MWORD 0
JSR R5,WRITEI ;SET ACR FOR T1 ONE-SHOT MODE
VIAACR
200
JSR R5,WRITEI ;LOAD VIA T1L-L
VIAT1C
300
JSR R5,WRITEI ;LOAD VIA T1L-H
VIAT1D
000
JSR R5,CKUSTS ;CHK USYRT STATUS FOR INIT'D STATE
110 ; TBMT = 1, TSO = 1
BCS 7$ ;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

```

CVDMA.P11 10-DEC-80 09:14

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

```

3576 007556 120000
3577 007560 000142
3578
3579 007562 004537 003734
3580 007566 120403
3581 007570 000001
3582 007572 004537 003734
3583 007576 120402
3584 007600 000226
3585 007602 004537 006036
3586 007606 000000
3587 007610 103427
3588 007612 005001
3589 007614 004537 011614
3590 007620 000001
3591 007622 004537 003610
3592 007626 122000
3593 007630 000000
3594 007632 132737 000100 007630
3595 007640 001010
3596 007642 005201
3597 007644 020127 000003
3598 007650 002761
3599 007652 004537 006036
3600 007656 000001
3601 007660 103403
3602 007662 004537 005536
3603 007666 000001
3604 007670 012601
3605 007672 000205
3606
3607 007674 000142
3608
3609

```

```

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$: JSR R5,CKTACT ;CHK FOR TXACT = 1
      1
7$: 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

```

CVDMCA.P11 10-DEC-80 09:14

....TXCHAR -- TRANSMIT A CHARACTER

3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627 007676
3628 007676 010146
3629 007700 010246
3630 007702 012537 007714
3631 007706 004537 003734
3632 007712 120402
3633 007714 000000
3634 007716 005001
3635 007720 005002
3636 007722 112502
3637 007724 001425
3638 007726 004537 005536
3639 007732 000001
3640 007734 103421
3641 007736 020102
3642 007740 001414
3643
3644 007742 131527 000200
3645 007746 001004
3646
3647 007750 004537 006036
3648 007754 000000
3649 007756 103410
3650 007760 004537 011614
3651 007764 000001
3652 007766 005201
3653 007770 000756
3654 007772 004537 006036
3655 007776 000001
3656 010000 012602
3657 010002 012601
3658 010004 005205
3659 010006 000205
3660
3661
3662
3663

```

.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
BITB (R5),#NCTBMT ;* CHECK FOR 'TBMT=0 CHECK' DISABLE
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 SANE RETURN
RTS R5 ;RETURN (WITH C BIT = 1 IF ERROR)

```

CVDMCA.P11 10-DEC-80 09:14

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

3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680 010010
3681 010010 010146
3682 010012 010246
3683 010014 012537 010026
3684 010020 004537 003734
3685 010024 120403
3686 010026 000000
3687 010030 005001
3688 010032 012502
3689 010034 001422
3690 010036 004537 005536
3691 010042 000001
3692 010044 103416
3693 010046 020102
3694 010050 001411
3695 010052 004537 006036
3696 010056 000000
3697 010060 103410
3698 010062 004537 011614
3699 010066 000001
3700 010070 005201
3701 010072 000761
3702 010074 004537 006036
3703 010100 000001
3704 010102 012602
3705 010104 012601
3706 010106 000205
3707

```

.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)

```

CVDMCA.P11 10-DEC-80 09:14

....RXCHAR -- RECEIVE A CHARACTER

```

3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727 010110
3728 010110 010146
3729 010112 010246
3730 010114 004537 003610
3731 010120 120401
3732 010122 000000
3733 010124 004537 003610
3734 010130 120400
3735 010132 000000
3736 010134 111501
3737 010136 042701 177400
3738 010142 023727 002456 000347
3739 010150 001005
3740 010152 142737 000200 010132
3741 010160 142701 000200
3742 010164 123701 010132
3743 010170 001462
3744 010172 004537 003610
3745 010176 122000
3746 010200 000000
3747 010202 132737 000002 010200
3748 010210 001421
3749 010212 012737 000007 002412
3750
3751 010220
3752
3753 010220 012737 000001 002246
3754 010226 012737 000033 002250
3755 010234 012737 013717 002252
3756 010242 012737 017722 002254
3757 010250 000137 011350
3758 010254 005037 002412
3759 010260 005037 002400
3760 010264 110137 002400
3761 010270 005037 002402
3762 010274 113737 010132 002402
3763
    
```

```

.SBTTL ....RXCHAR -- 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
GTFD EMS4,ERR12
; QUEUE 'DEVICE FATAL' ERROR # 27
MOV #T.EDF,ERRTYP
MOV #27,ERRNBR
MOV #EMS4,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 ;SET ACTUAL DATA
CLR BDATA
MOVB 1$,BDATA
;STACK 'RCV'D DATA MISCOMPARE' ERROR
    
```

CVDMA.P11

10-DEC-80 09:14

....RXCHAR -- RECEIVE A CHARACTER

```

3764 010302          GTDF  EM34,ERR10          :
3765                                     :   QUEUE 'DEVICE FATAL' ERROR # 28
3766 010302 012737 000001 002246          :   MOV      #T.EDF,ERRTYP
3767 010310 012737 000034 002250          :   MOV      #28,ERRNBR
3768 010316 012737 013573 002252          :   MOV      #EM34,ERRMSG
3769 010324 012737 017372 002254          :   MOV      #ERR10,ERRBLK
3770 010332 000137 011350          :
3771 010336 116501 000001          6$:   JMP      20$          :TAKE ERROR EXIT
3772 010342 042701 177400          :   MOVB    1(R5),R1   :GET RDSRH
3773 010346 123701 010122          :   BIC     #177400,R1 :MASK OFF UNUSED BITS
3774 010352 001016          :   CMPB    2$,R1     :COMPARE RCV'D STATUS TO EXPECTED
3775 010354 000137 011234          :   BNE     7$        :BR IF MISMATCH
3776 010360 012737 000001 002412          :   JMP     17$        :CONTINUE
3777 010366 005037 002400          :   MOV     #1,REGNUM  :SET USYRT REG NO. FOR RDSRH
3778 010372 110137 002400          :   CLR     GDATA     :SET EXPECTED DATA
3779 010376 005037 002402          :   MOVB   R1,GDATA   :
3780 010402 113737 010122 002402          :   CLR     BDATA     :SET ACTUAL DATA
3781 010410 012737 000001 002412          :   MOVB   2$,BDATA   :
3782 010416 032765 000001 000002          7$:   MOV     #1,REGNUM  :SET REG NO. FOR PRINTOUT
3783 010424 001447          :   BIT     #RERRCHK,2(R5) :SEE IF RCV ERROR BIT SHOULD BE IGNORED
3784                                     :   BEQ     9$        :BR IF YES
3785 010426 132701 000200          :CHECK RERR BIT
3786 010432 001022          :   BITB   #RERR,R1   :SEE IF EXPECTED BIT = 1
3787 010434 132737 000200 010122          :   BNE     8$        :BR IF YES
3788 010442 001440          :   BITB   #RERR,2$   :SEE IF ACTUAL BIT = 0
3789                                     :   BEQ     9$        :BR IF YES
3790                                     :STACK 'RERR NOT CLEARED' MSG
3791                                     :   GTDF   EM35,ERR12
3792                                     :
3792 010444 012737 000001 002246          :   QUEUE 'DEVICE FATAL' ERROR # 29
3793 010452 012737 000035 002250          :   MOV      #T.EDF,ERRTYP
3794 010460 012737 013621 002252          :   MOV      #29,ERRNBR
3795 010466 012737 017722 002254          :   MOV      #EM35,ERRMSG
3796 010474 000137 011350          :   MOV      #ERR12,ERRBLK
3797 010500 132737 000200 010122          8$:   JMP      20$          :TAKE ERROR EXIT
3798 010506 001016          :   BITB   #RERR,2$   :SEE IF ACTUAL BIT = 1
3799                                     :   BNE     9$        :BR IF YES
3800                                     :STACK 'RERR NOT SET' MSG
3801                                     :   GTDF   EM36,ERR12
3802                                     :
3802 010510 012737 000001 002246          :   QUEUE 'DEVICE FATAL' ERROR # 30
3803 010516 012737 000036 002250          :   MOV      #T.EDF,ERRTYP
3804 010524 012737 013642 002252          :   MOV      #30,ERRNBR
3805 010532 012737 017722 002254          :   MOV      #EM36,ERRMSG
3806 010540 000137 011350          :   MOV      #ERR12,ERRBLK
3807                                     :
3807 010544 132701 000010          :CHECK ROR BIT
3808 010550 001022          9$:   JMP      20$          :TAKE ERROR EXIT
3809 010552 132737 000010 010122          :   BITB   #ROR,R1   :SEE IF EXPECTED BIT = 1
3810 010560 001440          :   BNE     10$       :BR IF YES
3811                                     :   BITB   #ROR,2$   :SEE IF ACTUAL BIT = 0
3812                                     :   BEQ     11$       :BR IF YES
3813                                     :STACK 'ROR NOT CLEARED' MSG
3814                                     :   GTDF   EM16,ERR12
3815                                     :
3815 010562 012737 000001 002246          :   QUEUE 'DEVICE FATAL' ERROR # 31
3816 010570 012737 000037 002250          :   MOV      #T.EDF,ERRTYP
3817 010576 012737 013346 002252          :   MOV      #31,ERRNBR
3818 010604 012737 017722 002254          :   MOV      #EM16,ERRMSG
3819 010612 000137 011350          :   MOV      #ERR12,ERRBLK
3819                                     :
3819                                     :   JMP     20$        :TAKE ERROR EXIT

```

CVDMA.P11 10-DEC-80 09:14

....RXCHAR -- RECEIVE A CHARACTER

```

3820 010616 132737 000010 010122 10$: BITB #ROR,2$ ;SEE IF ACTUAL BIT = 1
3821 010624 001016 ; BNE 11$ ;BR IF YES
3822 ;STACK 'ROR NOT SET' MSG
3823 010626 GTDF EM14,ERR12
3824 ; QUEUE 'DEVICE FATAL' ERROR # 32
3825 010626 012737 000001 002246 MOV #T,EDF,ERRTYP
3826 010634 012737 000040 002250 MOV #32,ERRNBR
3827 010642 012737 013332 002252 MOV #EM14,ERRMSG
3828 010650 012737 017722 002254 MOV #ERR12,ERRBLK
3829 010656 000137 011350 ;CHECK RABGA BIT
3830 ;CHECK RABGA BIT ;TAKE ERROR EXIT
3831 010662 132701 000004 11$: BITB #RABGA,R1 ;SEE IF EXPECTED BIT = 1
3832 010666 001022 ; BNE 12$ ;BR IF YES
3833 010670 132737 000004 010122 BITB #RABGA,2$ ;SEE IF ACTUAL BIT = 0
3834 010676 001440 ; BEQ 13$ ;BR IF YES
3835 ;STACK 'RABGA NOT CLEARED' MSG
3836 010700 GTDF EM39,ERR12
3837 ; QUEUE 'DEVICE FATAL' ERROR # 33
3838 010700 012737 000001 002246 MOV #T,EDF,ERRTYP
3839 010706 012737 000041 002250 MOV #33,ERRNBR
3840 010714 012737 013657 002252 MOV #EM39,ERRMSG
3841 010722 012737 017722 002254 MOV #ERR12,ERRBLK
3842 010730 000137 011350 ;CHECK RABGA BIT
3843 010734 132737 000004 010122 12$: BITB #RABGA,2$ ;TAKE ERROR EXIT
3844 010742 001016 ; BNE 13$ ;SEE IF ACTUAL BIT = 1
3845 ;STACK 'RABGA NOT SET' MSG ;BR IF YES
3846 010744 GTDF EM40,ERR12
3847 ; QUEUE 'DEVICE FATAL' ERROR # 34
3848 010744 012737 000001 002246 MOV #T,EDF,ERRTYP
3849 010752 012737 000042 002250 MOV #34,ERRNBR
3850 010760 012737 013701 002252 MOV #EM40,ERRMSG
3851 010766 012737 017722 002254 MOV #ERR12,ERRBLK
3852 010774 000137 011350 ;CHECK REOM BIT
3853 ;CHECK REOM BIT ;TAKE ERROR EXIT
3854 011000 132701 000002 13$: BITB #REOM,R1 ;SEE IF EXPECTED BIT = 1
3855 011004 001022 ; BNE 14$ ;BR IF YES
3856 011006 132737 000002 010122 BITB #REOM,2$ ;SEE IF ACTUAL BIT = 0
3857 011014 001440 ; BEQ 15$ ;BR IF YES
3858 ;STACK 'REOM NOT CLEARED' MSG
3859 011016 GTDF EM30,ERR12
3860 ; QUEUE 'DEVICE FATAL' ERROR # 35
3861 011016 012737 000001 002246 MOV #T,EDF,ERRTYP
3862 011024 012737 000043 002250 MOV #35,ERRNBR
3863 011032 012737 013535 002252 MOV #EM30,ERRMSG
3864 011040 012737 017722 002254 MOV #ERR12,ERRBLK
3865 011046 000137 011350 ;CHECK REOM BIT
3866 011052 132737 000002 010122 14$: BITB #REOM,2$ ;TAKE ERROR EXIT
3867 011060 001016 ; BNE 15$ ;SEE IF ACTUAL BIT = 1
3868 ;STACK 'REOM NOT SET' MSG ;BR IF YES
3869 011062 GTDF EM31,ERR12
3870 ; QUEUE 'DEVICE FATAL' ERROR # 36
3871 011062 012737 000001 002246 MOV #T,EDF,ERRTYP
3872 011070 012737 000044 002250 MOV #36,ERRNBR
3873 011076 012737 013556 002252 MOV #EM31,ERRMSG
3874 011104 012737 017722 002254 MOV #ERR12,ERRBLK
3875 011112 000137 011350 ;CHECK REOM BIT
; BNE 20$ ;TAKE ERROR EXIT

```


CVDMCA.P11 10-DEC-80 09:14

....RXCHAR -- RECEIVE A CHARACTER

```

3876
3877 011116 132701 000001
3878 011122 001022
3879 011124 132737 000001 010122
3880 011132 001440
3881
3882 011134
3883
3884 011134 012737 000001 002246
3885 011142 012737 000045 002250
3886 011150 012737 013477 002252
3887 011156 012737 017722 002254
3888 011164 000137 011350
3889 011170 132737 000001 010122
3890 011176 001016
3891
3892 011200
3893
3894 011200 012737 000001 002246
3895 011206 012737 000046 002250
3896 011214 012737 013520 002252
3897 011222 012737 017722 002254
3898 011230 000137 011350
3899
3900 011234 116502 000004
3901 011240 005001
3902
3903 011242 136527 000005 000040
3904 011250 001004
3905 011252 004537 005676
3906 011256 000001
3907 011260 103433
3908
3909 011262 020102
3910 011264 001415
3911
3912 011266 136527 000005 000200
3913 011274 001004
3914 011276 004537 006176
3915 011302 000000
3916 011304 103421
3917
3918 011306 004537 011614
3919 011312 000001
3920 011314 005201
3921 011316 000751
3922
3923 011320 136527 000005 000100
3924 011326 001004
3925 011330 004537 006176
3926 011334 000001
3927 011336 103404
3928
3929 011340 062705 000006
3930 011344 000241
3931 011346 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 # 37
      MOV #T,EDF,ERRTYP
      MOV #37,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 # 38
      MOV #T,EDF,ERRTYP
      MOV #38,ERRNBR
      MOV #EM29,ERRMSG
      MOV #ERR12,ERRBLK
;TAKE ERROR EXIT
17$: MOV 4(R5),R2 ;GET DESIRED NO. OF CYCLES
      CLR R1 ;INIT CYCLE COUNT
18$: BITB 5(R5),#BITS ;* IS RXACT CHECK TO BE DISABLED ?
      BNE 31$ ;* BR IF YES
      JSR R5,CKRACT ;CHK FOR RACT = 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
      BCS 20$ ;BR TO EXIT IF ERROR
22$: JSR R5,STEPLU ;CLOCK LU FOR 1 CYCLE
      INC R1 ;INCR CYCLE COUNT
      BR 18$ ;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
      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

```

CVDMCA.P11 10-DEC-80 09:14

....RXCHAR -- RECEIVE A CHARACTER

3932	011350	062705	000006
3933	011354	000261	
3934	011356	012602	
3935	011360	012601	
3936	011362	000205	

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

CVDMCA.P11 10-DEC-80 09:14

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

.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:

3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952 011364
3953 011364 010146
3954 011366 010246
3955 011370 005001
3956 011372 012502
3957 011374 062702 000003
3958 011400 004537 005676
3959 011404 000000
3960 011406 103446
3961 011410 004537 006176
3962 011414 000000
3963 011416 103442
3964 011420 004537 006636
3965 011424 000000
3966 011426 103436
3967 011430 004537 011614
3968 011434 000001
3969 011436 005201
3970 011440 004537 003610
3971 011444 122000
3972 011446 000000
3973 011450 132737 000200 011446
3974 011456 001006
3975 011460 020102
3976 011462 002762
3977 011464 004537 006176
3978 011470 000001
3979 011472 103414
3980 011474 020165 177776
3981 011500 002004
3982 011502 004537 006176
3983 011506 000000
3984 011510 103405
3985 011512 004537 005676
3986 011516 000001
3987 011520 103401
3988 011522 000241
3989 011524 012602
3990 011526 012601
3991 011530 000205

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,READI ;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
BGE 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)

CVDMCA.P11 10-DEC-80 09:14

....ENDTRN -- SHUT DOWN TRANSMITTER/RECEIVER

3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027

011532
011532 012537 011572
011536 004537 005536
011542 000001
011544 103422
011546 004537 005676
011552 000001
011554 103416
011556 004537 003734
011562 120000
011564 000002
011566 004537 011614
011572 000000
011574 004537 005536
011600 000000
011602 103403
011604 004537 005676
011610 000000
011612 000205

.SBTTLENDTRN -- SHUT DOWN TRANSMITTER/RECEIVER

* ENDTRN - 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.
* NOTE: THIS ROUTINE ASSUMES THAT TTLOOP MODE SHOULD BE ENABLED.
*
* CALLING SEQUENCE :
* JSR R5,ENDTRN
* <NO. OF CYCLES TO CLOCK>

ENDTRN:
MOV (R5)+,2\$;GET DESIRED NO. OF CYCLES TO CLOCK
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 ;** AND KEEP TTLOOP ENABLED **
TTLOOP
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

CVDMCA.P11 10-DEC-80 09:14

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

4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056

011614
011614 010146
011616 012501
011620 004537 003734
011624 120005
011626 000000
011630 005301
011632 001372
011634 012601
011636 000205

.SBTTLSTEPLU -- 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:
1\$: MOV R1, -(SP) ;SAVE R1
MOV (R5)+, R1 ;INIT CYCLE COUNTER
JSR R5, WRITEI ;LOAD 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

CVDMCA.P11 10-DEC-80 09:14

GLOBAL ERROR REPORT SECTION

4057
4058
4059
4060
4061
4062
4063
4064

.SBTTL GLOBAL ERROR REPORT SECTION

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

```

.NLIST BEX
011640 047045 047045 000 ENDEMB: .ASCIZ /%ZN%/
011645 045 000116 NEWLIN: .ASCIZ /%ZN/ ;USED TO TERMINATE ERROR MESSAGES

011650 047045 040445 040506 FMT2: .ASCIZ /%ZNAFAILING REG = %T%ASEL%01/
011705 045 022516 020101 FMT3: .ASCIZ /%ZNA EXPECTED: %03%A ACTUAL: %03%A XOR: %03/
011771 045 022516 052101 FMT4: .ASCIZ /%ZNA THE CONTENTS OF ALL %T%Z%T%/
012027 045 022516 030523 FMT4A: .ASCIZ /%ZNS1%03%5%03%5%03%5%03%/
012062 047045 052045 000 FMT4B: .ASCIZ /%ZNT%/
012067 045 022516 032523 FMT4C: .ASCIZ /%ZNS5%03%5%03%5%03%5%03%/
012122 047045 040445 020040 FMT5: .ASCIZ /%ZNA WHEN %03%A LOADED INTO BSEL1/
012165 045 022516 020101 FMT5A: .ASCIZ /%ZNA ATTEMPTING "M-LOOP" FUNCTION CODE %02%A (%T%A)/

012252 047045 040445 020040 FMT10: .ASCIZ /%ZNA EXPECTED:%08%A ACTUAL:%08%A XOR:%08/
012326 040445 020040 051514 FMT10A: .ASCIZ /%A LSI ADDR:%08/
012347 045 022516 034117 FMT11: .ASCIZ /%Z%08%08%08%08/
012366 040445 020040 042504 FMT15: .ASCIZ /%A DETECTED IN %T%T%A --/
012420 040445 020040 042504 FMT15A: .ASCIZ /%A DETECTED @ TEST PATTERN ELEMENT # %D2/

012472 047045 040445 042524 FMT19: .ASCIZ /%ZNA TEST %D2%A NOT RUN%N/
012523 045 022524 033117 FMT21: .ASCIZ /%T%06%N/
012533 045 022516 043101 FMT22: .ASCIZ /%ZNAFAILING REG: /
012555 045 042501 050130 FMT23: .ASCIZ /%AEXPECTED: %03%5%A ACTUAL: %03%5%A XOR: %03%N/
012634 047045 052045 047045 FMT24: .ASCIZ /%ZNT%Z%T%N/
012647 045 031517 051445 FMT25: .ASCIZ /%03%5%03%5%03%5%03%N/
012677 045 032123 047445 FMT26: .ASCIZ /%S4%03%5%03%5%03%5%03%N/
012732 052045 052045 047045 FMT27: .ASCIZ /%T%T%N/
012741 045 022524 000116 FMT29: .ASCIZ /%T%N/
012746 047045 040445 047506 FMT30: .ASCIZ /%ZNAFOR BAUD RATE SPECIFIED./
013003 045 022516 044501 FMT31: .ASCIZ /%ZNAIMPROPER CONNECTOR TYPE SPECIFIED/
013051 045 022516 043101 FMT32: .ASCIZ /%ZNAFOR OPTION SPECIFIED./

013103 122 043505 047040 EM1: .ASCIZ /REG NOT INITIALIZED BY MST CLR/
013142 051525 051131 020124 EM2: .ASCIZ /USVRT NOT INITIALIZED BY PROGRAM RESET/
013211 115 041511 047522 EM3: .ASCIZ /MICRO-DIAG. FAILURE/
013235 115 042122 020131 EM4: .ASCIZ /RDY TIMEOUT/
013252 052516 046114 041440 EM5: .ASCIZ /NULL CLK BIT STUCK AT 0/
013302 052516 046114 041440 EM6: .ASCIZ /NULL CLK BIT STUCK AT 1/
013332 047522 020122 047516 EM14: .ASCIZ /ROR NOT SET/
013346 047522 020122 047516 EM16: .ASCIZ /ROR NOT CLEARED/
013366 042522 042101 053457 EM25: .ASCIZ 'READ/WRITE DATA ERROR'
013414 047111 047503 051122 EM26: .ASCIZ /INCORRECT DATA CHAR RCV'D/
013446 047111 047503 051122 EM27: .ASCIZ /INCORRECT CRC BYTE RCV'D/
013477 122 047523 020115 EM28: .ASCIZ /RSM NOT CLEARED/
013520 051522 046517 047040 EM29: .ASCIZ /RSM NOT SET/
013535 122 047505 020115 EM30: .ASCIZ /REOM NOT CLEARED/
013556 042522 046517 047040 EM31: .ASCIZ /REOM NOT SET/
013573 122 053103 042047 EM34: .ASCIZ /RCV'D DATA MISCOMPARE/
013621 122 051105 020122 EM35: .ASCIZ /RERR NOT CLEARED/

```

CVDMA.P11 10-DEC-80 09:14

GLOBAL ERROR REPORT SECTION

013642	042522	051122	047040	EM36:	.ASCIZ	/RERR NOT SET/
013657	122	041101	040507	EM39:	.ASCIZ	/RABGA NOT CLEARED/
013701	122	041101	040507	EM40:	.ASCIZ	/RABGA NOT SET/
013717	124	020130	047125	EM54:	.ASCIZ	/TX UNDERRUN ERROR/
013741	122	043505	046440	EM66:	.ASCIZ	/REG MISCOMPARE/
013760	051525	051131	020124	EM68:	.ASCIZ	/USYRT STATUS INCORRECT/
014007	124	040530	052103	EM69:	.ASCIZ	/TXACT NOT SET/
014025	124	040530	052103	EM70:	.ASCIZ	/TXACT NOT CLEARED/
014047	122	040530	052103	EM71:	.ASCIZ	/RXACT NOT SET/
014065	122	040530	052103	EM72:	.ASCIZ	/RXACT NOT CLEARED/
014107	124	046502	020124	EM73:	.ASCIZ	/TBM NOT SET/
014124	041124	052115	047040	EM74:	.ASCIZ	/TBM NOT CLEARED/
014145	122	040504	047040	EM75:	.ASCIZ	/RDA NOT SET/
014161	122	040504	047040	EM76:	.ASCIZ	/RDA NOT CLEARED/
014201	122	040523	047040	EM77:	.ASCIZ	/RSA NOT SET/
014215	122	040523	047040	EM78:	.ASCIZ	/RSA NOT CLEARED/
014235	122	041505	044505	EM90:	.ASCIZ	/RECEIVER OVERRUN NOT SET/
014266	042522	042503	053111	EM91:	.ASCIZ	/RECEIVER OVERRUN NOT CLEARED/
014323	124	046502	020124	EM92:	.ASCIZ	/TBM INTERRUPT TEST FAILURE/
014357	124	051105	020122	EM98:	.ASCIZ	/TERR BIT NOT CLEARED/
014404	042524	051122	041040	EM99:	.ASCIZ	/TERR BIT NOT SET/
014425	124	047523	041040	EM100:	.ASCIZ	/TSO BIT NOT SET/
014445	124	047523	041040	EM101:	.ASCIZ	/TSO BIT NOT CLEARED/
014471	124	040522	051516	EM102:	.ASCIZ	/TRANSMISSION ERROR (AS READ BY TSO BIT)/

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

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

014541	102	042523	030114	TXT1:	.ASCIZ	/BSEL0 BSEL1 BSEL2 BSEL3/
014577	040	020040	041040	TXT2:	.ASCIZ	/ BSEL4 BSEL5 BSEL6 BSEL7/
014641	102	042523	030514	TXT2A:	.ASCIZ	/BSEL10 BSEL11 BSEL12 BSEL13/
014700	020040	020040	051502	TXT2B:	.ASCIZ	/ BSEL14 BSEL15 BSEL16 BSEL17/
014743	040	054502	042524	TXT3:	.ASCIZ	/ BYTE SELECT REG'S ARE:/
014773	040	020040	042523	TXT4:	.ASCIZ	/ SEL0 SEL2 SEL4 SEL6/
015033	040	020040	042523	TXT4A:	.ASCIZ	/ SEL10 SEL12 SEL14 SEL16/
015074	000102			TXT5:	.ASCIZ	/B/
015076	051440	046105	041505	TXT6:	.ASCIZ	/ SELECT REG'S ARE:/
015121	040	042522	044507	TXT7:	.ASCIZ	/ REGISTERS ORB ORA DDRB DDRA T1CL T1CH T1LL T1LH /
015211	040	020040	020040	TXT7A:	.ASCIZ	/ T2CL T2CH SR ACR PCR IFR IER ORA /
015301	040	054105	042520	TXT8:	.ASCIZ	/ EXPECTED: /
015321	040	041501	052524	TXT9:	.ASCIZ	/ ACTUAL: /
015341	040	047530	035122	TXT10:	.ASCIZ	/ XOR: /
015361	040	047040	020040	TXT11:	.ASCIZ	/ N P R R E G I S T E R S:/
015433	040	020040	020040	TXT11A:	.ASCIZ	/ CONTROL DATA/
015471	040	020040	020040	TXT11B:	.ASCIZ	/ OUT ADDR. IN ADDR./
015541	104	053105	041511	TXT12:	.ASCIZ	/DEVICE CSR ADDRESS : /
015567	125	054523	052122	TXT13:	.ASCIZ	/USYRT REGS :/
015604	042122	051123	020114	TXT14:	.ASCIZ	/RDSRL RDSRH TDSRL TDSRH/
015642	020040	020040	041520	TXT15:	.ASCIZ	/ PCSARL PCSARH PCR USTAT/
015704	044526	020101	042522	TXT16:	.ASCIZ	/VIA REGS :/
015717	117	041122	020040	TXT17:	.ASCIZ	/ORB ORA DDRB DDRA/
015754	020040	020040	030524	TXT18:	.ASCIZ	/ T1CL T1CH T1LL T1LH/

CVDMA.P11 10-DEC-80 09:14

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

```

016015      124  041462  020114  TXT19:  .ASCIZ  /T2CL  T2CH  SR  ACR/
016051      040  020040  050040  TXT20:  .ASCIZ  /  PCR  IFR  IER  ORA/

016111      021    000          TXTNUL:  .BYTE  21,0          ;CTL-Q -- THIS (WE HOPE) IS HARMLESS

016113      116  050117    000          TXTML0:  .ASCIZ  /NOP/
016117      122  040505  020104  TXTML1:  .ASCIZ  /READ 1 BYTE/
016133      127  044522  042524  TXTML2:  .ASCIZ  /WRITE 1 BYTE/
016150      050116  026522  052517  TXTML3:  .ASCIZ  /NPR-OUT 256 BYTES/
016172      050116  026522  047111  TXTML4:  .ASCIZ  /NPR-IN 256 BYTES/
016213      123  052105  046440  TXTML5:  .ASCIZ  /SET MICROPROCESSOR'S PC/
016243      125  042116  043105  TXTML6:  .ASCIZ  /UNDEFINED/
016255      101  046114  053517  TXTML7:  .ASCIZ  /A'LOW U-PROCESSOR INTERRUPTS/

016312      044526  020101  042522  TXTVR:   .ASCIZ  /VIA REGISTER /
016330      051117  000102          TXTVR0:  .ASCIZ  /ORB/
016334      051117  000101          TXTVR1:  .ASCIZ  /ORA/
016340      042104  041122    000          TXTVR2:  .ASCIZ  /DDRB/
016345      104  051104  000101  TXTVR3:  .ASCIZ  /DDRA/
016352      030524  046103    000          TXTVR4:  .ASCIZ  /T1CL/
016357      124  041461  000110  TXTVR5:  .ASCIZ  /T1CH/
016364      030524  046114    000          TXTVR6:  .ASCIZ  /T1LL/
016371      124  046061  000110  TXTVR7:  .ASCIZ  /T1LH/
016376      031124  046103    000          TXTVR8:  .ASCIZ  /T2CL/
016403      124  041462  000110  TXTVR9:  .ASCIZ  /T2CH/
016410      051123    000          TXTVRA:  .ASCIZ  /SR/
016413      101  051103    000          TXTVRB:  .ASCIZ  /ACR/
016417      120  051103    000          TXTVRC:  .ASCIZ  /PCR/
016423      111  051106    000          TXTVRD:  .ASCIZ  /IFR/
016427      111  051105    000          TXTVRE:  .ASCIZ  /IER/
016433      117  040522    000          TXTVRF:  .ASCIZ  /ORA/

016437      116  051120  000040  TXTNP:   .ASCIZ  /NPR /
016444      047503  052116  047522  TXTNP0:  .ASCIZ  /CONTROL/
016454      040504  040524  044040  TXTNP1:  .ASCIZ  /DATA HI/
016464      040504  040524  046040  TXTNP2:  .ASCIZ  /DATA LO/
016474      042101  051104  020056  TXTNP3:  .ASCIZ  /ADDR. OUT EX/
016511      101  042104  027122  TXTNP4:  .ASCIZ  /ADDR. OUT HI/
016526      042101  051104  020056  TXTNP5:  .ASCIZ  /ADDR. OUT LO/
016543      101  042104  027122  TXTNP6:  .ASCIZ  /ADDR. IN EX/
016557      101  042104  027122  TXTNP7:  .ASCIZ  /ADDR. IN HI/
016573      101  042104  027122  TXTNP8:  .ASCIZ  /ADDR. IN LO/

016607      125  054523  052122  TXTUR:   .ASCIZ  /USYRT REG /
016622      042122  051123  000114  TXTUR0:  .ASCIZ  /RDSRL/
016630      042122  051123  000110  TXTUR1:  .ASCIZ  /RDSRH/
016636      042124  051123  000114  TXTUR2:  .ASCIZ  /TDSRL/
016644      042124  051123  000110  TXTUR3:  .ASCIZ  /TDSRH/
016652      041520  040523  046122  TXTUR4:  .ASCIZ  /PCSARL/
016661      120  051503  051101  TXTUR5:  .ASCIZ  /PCSARH/
016670      041520  000122          TXTUR6:  .ASCIZ  /PCR/
016674      051525  040524  000124  TXTUR7:  .ASCIZ  /USTAT/
.LIST  BEX
.EVEN

```

4065
4066
4067

CVDMCA.P11 10-DEC-80 09:14

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

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

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

4068						
4069						
4070						
4071						
4072						
4073	016702	016113	016117	016133	TXTMLT: .WORD	TXTML0,TXTML1,TXTML2,TXTML3,TXTML4,TXTML5,TXTML6,TXTML7
4074	016710	016150	016172	016213		
4075	016716	016243	016255			
4076						
4077	016722	016312				
4078	016724	016330	016334	016340	TXTVRT: .WORD	TXTVR
4079	016732	016345	016352	016357		TXTVR0,TXTVR1,TXTVR2,TXTVR3,TXTVR4,TXTVR5,TXTVR6,TXTVR7
4080	016740	016364	016371			
4081	016744	016376	016403	016410	.WORD	TXTVR8,TXTVR9,TXTVRA,TXTVRB,TXTVRC,TXTVRD,TXTVRE,TXTVRF
4082	016752	016413	016417	016423		
4083	016760	016427	016433			
4084						
4085	016764	016437				
4086	016766	016444	016454	016464	TXTNPT: .WORD	TXTNP
4087	016774	016474	016511	016526		TXTNP0,TXTNP1,TXTNP2,TXTNP3,TXTNP4,TXTNP5,TXTNP6,TXTNP7,TXTNP8
4088	017002	016543	016557	016573		
4089	017010	016622	016630	016636	TXTURT: .WORD	TXTUR0,TXTUR1,TXTUR2,TXTUR3,TXTUR4,TXTUR5,TXTUR6,TXTUR7
4090	017016	016644	016652	016661		
4091	017024	016670	016674			
4092						
4093						

CVDMCA.P11 10-DEC-80 09:14

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

```

4094
4095
4096
4097 017030
4098 017030
4099 017030 105037 002401
4100 017034 010146
4101 017036 013701 002400
4102 017042 022701 000017
4103 017046 002012
4104 017050
4105 017050 010146
4106 017052 012746 012122
4107 017056 012746 000002
4108 017062 010600
4109 017064 104415
4110 017066 062706 000006
4111 017072 000424
4112
4113 017074 001001
4114 017076 005001
4115 017100 022701 000007
4116 017104 002002
4117 017106 012701 000006
4118 017112 006301
4119 017114
4120 017114 016146 016702
4121 017120 013746 002400
4122 017124 012746 012165
4123 017130 012746 000003
4124 017134 010600
4125 017136 104415
4126 017140 062706 000010
4127
4128 017144 012601
4129 017146 004737 020160
4130 017152
4131 017152
4132 017152 104423
4133
4134
4135
4136 017154
4137 017154
4138 017154
4139 017154 013746 002412
4140 017160 012746 015074
4141 017164 012746 011650
4142 017170 012746 000003
4143 017174 010600
4144 017176 104414
4145 017200 062706 000010
4146 017204 004737 020134
4147 017210
4148 017210 013746 002404
4149 017214 013746 002402

```

```

-----
:SBTTL ....ERROR HANDLER -- ERR4 -- M-LOOP TIMEOUT ERROR HANDLING
-----
      BGNMSG  ERR4
      CLRB    GDATA+1
      MOV     R1,-(SP)
      MOV     GDATA,R1
      CMP     #17,R1
      BGE     5$
      PRINTX  #FMT5,R1
      ERR4::
      :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   CSPNTX
      ADD     #6,SP
      BR      20$
5$:    BNE     6$
      CLR     R1
6$:    CMP     #7,R1
      BGE     7$
      MOV     #6,R1
7$:    ASL     R1
      PRINTX  #FMT5A,GDATA,TXTMLT(R1)
      :IF IT WAS A 17, THIS IS A 'NOP' AND
      : THE TEXT POINTER MUST SO REFLECT.
      :IS FUNCTION CODE > 7?
      :NO, THEN WE CAN HANDLE IT
      :YES, THAN IT'S UNDEFINED -- SAY SO
      :CONVERT TO A WORD OFFSET
      :REPORT THE FAILING FUNCTION
      MOV     TXTMLT(R1),-(SP)
      MOV     GDATA,-(SP)
      MOV     #FMT5A,-(SP)
      MOV     #3,-(SP)
      MOV     SP,R0
      TRAP   CSPNTX
      ADD     #10,SP
20$:   MOV     (SP)+,R1
      JSR     PC,ERR5$
      ENDMSG
      :RESTORE THE WORKING REGISTER
      :DUMP THE SELECT REGISTERS
      L10002: TRAP   CSMSG
-----
:SBTTL ....ERROR HANDLER -- ERR5 -- WORD SELECT REG. ERRORS
-----
      BGNMSG  ERR5
      PRINTB  #FMT2,#TXT5,REGNUM
      ERR5::
      MOV     REGNUM,-(SP)
      MOV     #TXT5,-(SP)
      MOV     #FMT2,-(SP)
      MOV     #3,-(SP)
      MOV     SP,R0
      TRAP   CSPNTB
      ADD     #10,SP
      JSR     PC,XORGB
      PRINTB  #FMT10,GDATA,BDATA,XDATA
      MOV     XDATA,-(SP)
      MOV     BDATA,-(SP)

```

CVDMA.P11 10-DEC-80 09:14

....ERROR HANDLER -- ERR5 -- WORD SELECT REG. ERRORS

```

4150 017220 013746 002400
4151 017224 012746 012252
4152 017230 012746 000004
4153 017234 010600
4154 017236 104414
4155 017240 062706 000012
4156 017244 004737 020160
4157 017250
4158 017250
4159 017250 104423
4160
4161
4162
4163 017252
4164 017252
4165 017252 113701 002412
4166 017256 006301
4167 017260
4168 017260 016146 017010
4169 017264 012746 016607
4170 017270 012746 012366
4171 017274 012746 000003
4172 017300 010600
4173 017302 104414
4174 017304 062706 000010
4175 017310 004737 020134
4176 017314
4177 017314 013746 002404
4178 017320 013746 002402
4179 017324 013746 002400
4180 017330 012746 011705
4181 017334 012746 000004
4182 017340 010600
4183 017342 104414
4184 017344 062706 000012
4185 017350
4186 017350 012746 011640
4187 017354 012746 000001
4188 017360 010600
4189 017362 104414
4190 017364 062706 000004
4191 017370
4192 017370
4193 017370 104423
4194
4195
4196
4197

```

```

JSR PC,ERR5$
ENDMSG

```

:DUMP THE SELECT REGISTERS

L10003:

TRAP CSMSG

:SBTTLERROR HANDLER -- ERR7A -- USYRT REGISTER ERRORS

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 CSMSG

:SBTTLERROR 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

```

```

4198 017372
4199 017372
4200 017372
4201 017372 013746 002472
4202 017376 012746 015541
4203 017402 012746 012523
4204 017406 012746 000003
4205 017412 010600

```

CVDMA.P11 10-DEC-80 09:14

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

```

4206 017414 104414
4207 017416 062706 000010
4208 017422 PRINTB #FMT22
4209 017422 012746 012533
4210 017426 012746 000001
4211 017432 010600
4212 017434 104414
4213 017436 062706 000004
4214 017442 013701 002412
4215 017446 006301
4216 017450
4217 017450 016146 017010
4218 017454 012746 016607
4219 017460 012746 012732
4220 017464 012746 000003
4221 017470 010600
4222 017472 104414
4223 017474 062706 000010
4224 017500 004737 020134
4225 017504
4226 017504 013746 002404
4227 017510 013746 002402
4228 017514 013746 002400
4229 017520 012746 012555
4230 017524 012746 000004
4231 017530 010600
4232 017532 104414
4233 017534 062706 000012
4234 017540 004737 020710
4235 017544
4236 017544
4237 017544 104423
4238
4239
4240
4241
4242
4243 017546
4244 017546
4245 017546
4246 017546 013746 002472
4247 017552 012746 015541
4248 017556 012746 012523
4249 017562 012746 000003
4250 017566 010600
4251 017570 104414
4252 017572 062706 000010
4253 017576
4254 017576 012746 012533
4255 017602 012746 000001
4256 017606 010600
4257 017610 104414
4258 017612 062706 000004
4259 017616 013701 002412
4260 017622 006301
4261 017624

TRAP C$PNTB
ADD #10,SP
MOV #FMT22,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
MOV REGNUM,R1
ASL R1
PRINTB #FMT27,#TXTUR,TXTURT(R1) ;GET PTR TO USYRT REG ASCII
MOV TXTURT(R1),-(SP)
MOV #TXTUR,-(SP)
MOV #FMT27,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #10,SP
JSR PC,XORGB ;COMPUTE XOR OF GOOD AND BAD DATA
PRINTB #FMT23,GDATA,BDATA,XDATA
MOV XDATA,-(SP)
MOV BDATA,-(SP)
MOV GDATA,-(SP)
MOV #FMT23,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #12,SP
JSR PC,ERR12$ ;GET & PRINT USYRT REGISTERS
ENDMSG
L10005:
TRAP C$MSG

```

:SBTTLERROR HANDLER -- ERR11 -- VIA REG ERROR (XOR, REG PRINTOUT)

```

4243 017546
4244 017546
4245 017546
4246 017546 013746 002472
4247 017552 012746 015541
4248 017556 012746 012523
4249 017562 012746 000003
4250 017566 010600
4251 017570 104414
4252 017572 062706 000010
4253 017576
4254 017576 012746 012533
4255 017602 012746 000001
4256 017606 010600
4257 017610 104414
4258 017612 062706 000004
4259 017616 013701 002412
4260 017622 006301
4261 017624

BGNMSG ERR11
PRINTB #FMT21,#TXT12,MPCSR
ERR11::
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
PRINTB #FMT27,#TXTVR,TXTVRT(R1) ;GET PTR TO VIA REG ASCII

```

CVDMCA.P11 10-DEC-80 09:14

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

4262	017624	016146	016724			MOV	TXTVRT(R1),-(SP)
4263	017630	012746	016312			MOV	#TXTVR, -(SP)
4264	017634	012746	012732			MOV	#FMT27, -(SP)
4265	017640	012746	000003			MOV	#3, -(SP)
4266	017644	010600				MOV	SP, R0
4267	017646	104414				TRAP	CSPNTB
4268	017650	062706	000010			ADD	#10, SP
4269	017654	004737	020134	JSR	PC, XORGB		
4270	017660			PRINTB	#FMT23, GDATA, BDATA, XDATA		
4271	017660	013746	002404			MOV	XDATA, -(SP)
4272	017664	013746	002402			MOV	BDATA, -(SP)
4273	017670	013746	002400			MOV	GDATA, -(SP)
4274	017674	012746	012555			MOV	#FMT23, -(SP)
4275	017700	012746	000004			MOV	#4, -(SP)
4276	017704	010600				MOV	SP, R0
4277	017706	104414				TRAP	CSPNTB
4278	017710	062706	000012			ADD	#12, SP
4279	017714	004737	020356	JSR	PC, ERR11\$		
4280	017720			ENDMSG			
4281	017720						
4282	017720	104423				L10006:	TRAP CSMSG

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

4288	017722			BGNMSG	ERR12		
4289	017722					ERR12::	
4290	017722			PRINTB	#FMT21, #TXT12, MPCR		
4291	017722	013746	002472			MOV	MPCR, -(SP)
4292	017726	012746	015541			MOV	#TXT12, -(SP)
4293	017732	012746	012523			MOV	#FMT21, -(SP)
4294	017736	012746	000003			MOV	#3, -(SP)
4295	017742	010600				MOV	SP, R0
4296	017744	104414				TRAP	CSPNTB
4297	017746	062706	000010			ADD	#10, SP
4298	017752			PRINTB	#FMT22		
4299	017752	012746	012533			MOV	#FMT22, -(SP)
4300	017756	012746	000001			MOV	#1, -(SP)
4301	017762	010600				MOV	SP, R0
4302	017764	104414				TRAP	CSPNTB
4303	017766	062706	000004			ADD	#4, SP
4304	017772	013701	002412	MOV	REGNUM, R1		
4305	017776	006301		ASL	R1		
4306	020000			PRINTB	#FMT27, #TXTUR, TXTURT(R1)		
4307	020000	016146	017010			MOV	TXTURT(R1), -(SP)
4308	020004	012746	016607			MOV	#TXTUR, -(SP)
4309	020010	012746	012732			MOV	#FMT27, -(SP)
4310	020014	012746	000003			MOV	#3, -(SP)
4311	020020	010600				MOV	SP, R0
4312	020022	104414				TRAP	CSPNTB
4313	020024	062706	000010			ADD	#10, SP
4314	020030	004737	020710	JSR	PC, ERR12\$		
4315	020034			ENDMSG			
4316	020034					L10007:	TRAP CSMSG
4317	020034	104423					

CVDMCA.P11 10-DEC-80 09:14

....ERROR HANDLER -- ERR12 -- USYRT REG ERROR (USYRT PRINTOUT)

4318
4319
4320
4321
4322
4323 020036
4324 020036
4325 020036
4326 020036 013746 002472
4327 020042 012746 015541
4328 020046 012746 012523
4329 020052 012746 000003
4330 020056 010600
4331 020060 104414
4332 020062 062706 000010
4333 020066 004737 020134
4334 020072
4335 020072 013746 002404
4336 020076 013746 002402
4337 020102 013746 002400
4338 020106 012746 012555
4339 020112 012746 000004
4340 020116 010600
4341 020120 104414
4342 020122 062706 000012
4343 020126 004737 020710
4344 020132
4345 020132
4346 020132 104423

```

-----
:SBTTL ....ERROR HANDLER -- ERR13 -- TRANSMISSION/TSO ERROR (XOR, REG PRINTOUT)
-----

```

BGNMSG ERR13

ERR13::

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

```

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,R0
TRAP C$PNTB
ADD #12,SP

```

JSR PC,ERR12\$;GET & PRINT USYRT REGISTERS

ENDMSG

L10010:

TRAP C\$MSG

```

:SBTTL ....ERROR 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

```

```

:SBTTL .....ERROR HANDLER SUBROUTINE -- ERR5$
-----

```

ERR5\$: COMMON ERROR SUBROUTINE TO PRINT SELECT REGISTERS

PRINTX #FMT4,#TXT6,#TXT4

```

MOV #TXT4,-(SP)
MOV #TXT6,-(SP)

```

4370 020160
4371 020160
4372 020160 012746 014773
4373 020164 012746 015076

CVDMCA.P11 10-DEC-80 09:14

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

4374 020170 012746 011771
 4375 020174 012746 000003
 4376 020200 010600
 4377 020202 104415
 4378 020204 062706 000010
 4379 020210
 4380 020210 013746 002264
 4381 020214 013746 002262
 4382 020220 013746 002260
 4383 020224 013746 002256
 4384 020230 012746 012347
 4385 020234 012746 000005
 4386 020240 010600
 4387 020242 104415
 4388 020244 062706 000014
 4389 020250
 4390 020250 012746 015033
 4391 020254 012746 012062
 4392 020260 012746 000002
 4393 020264 010600
 4394 020266 104415
 4395 020270 062706 000006
 4396 020274
 4397 020274 013746 002274
 4398 020300 013746 002272
 4399 020304 013746 002270
 4400 020310 013746 002266
 4401 020314 012746 012347
 4402 020320 012746 000005
 4403 020324 010600
 4404 020326 104415
 4405 020330 062706 000014
 4406 020334
 4407 020334 012746 011640
 4408 020340 012746 000001
 4409 020344 010600
 4410 020346 104414
 4411 020350 062706 000004
 4412 020354 000207

PRINTX #FMT11,WSR0,WSR2,WSR4,WSR6 ;DUMP THE SELECT REGISTERS

MOV #FMT4,-(SP)
 MOV #3,-(SP)
 MOV SP,R0
 TRAP C\$PNTX
 ADD #10,SP
 MOV WSR6,-(SP)
 MOV WSR4,-(SP)
 MOV WSR2,-(SP)
 MOV WSR0,-(SP)
 MOV #FMT11,-(SP)
 MOV #5,-(SP)
 MOV SP,R0
 TRAP C\$PNTX
 ADD #14,SP

PRINTX #FMT4B,#TXT4A

MOV #TXT4A,-(SP)
 MOV #FMT4B,-(SP)
 MOV #2,-(SP)
 MOV SP,R0
 TRAP C\$PNTX
 ADD #6,SP

PRINTX #FMT11,WSR10,WSR12,WSR14,WSR16 ;DUMP THE SELECT REGISTERS

MOV WSR16,-(SP)
 MOV WSR14,-(SP)
 MOV WSR12,-(SP)
 MOV WSR10,-(SP)
 MOV #FMT11,-(SP)
 MOV #5,-(SP)
 MOV SP,R0
 TRAP C\$PNTX
 ADD #14,SP

PRINTB #ENDEMB

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

RTS PC

:SBTTLERROR HANDLER SUBROUTINE -- ERR11\$

: COMMON ERROR SUBROUTINE TO GET/PRINT VIA REGISTERS

4419 020356 004737 004502
 4420 020362
 4421 020362 012746 015717
 4422 020366 012746 015704
 4423 020372 012746 012634
 4424 020376 012746 000003
 4425 020402 010600
 4426 020404 104415
 4427 020406 062706 000010
 4428 020412
 4429 020412 013746 002344

ERR11\$: JSR PC,GETVRS ;GET VIA REGS FOR PRINTOUT
 PRINTX #FMT24,#TXT16,#TXT17

MOV #TXT17,-(SP)
 MOV #TXT16,-(SP)
 MOV #FMT24,-(SP)
 MOV #3,-(SP)
 MOV SP,R0
 TRAP C\$PNTX
 ADD #10,SP
 MOV VREGS+6,-(SP)

PRINTX #FMT25,VREGS+0,VREGS+2,VREGS+4,VREGS+6

CVDMA.P11 10-DEC-80 09:14

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

4430	020416	013746	002342		MOV	VREGS+4,-(SP)
4431	020422	013746	002340		MOV	VREGS+2,-(SP)
4432	020426	013746	002336		MOV	VREGS+0,-(SP)
4433	020432	012746	012647		MOV	#FMT25,-(SP)
4434	020436	012746	000005		MOV	#5,-(SP)
4435	020442	010600			MOV	SP,R0
4436	020444	104415			TRAP	C\$PNTX
4437	020446	062706	000014		ADD	#14,SP
4438	020452			PRINTX	#FMT29	#TXT18
4439	020452	012746	015754		MOV	#TXT18,-(SP)
4440	020456	012746	012741		MOV	#FMT29,-(SP)
4441	020462	012746	000002		MOV	#2,-(SP)
4442	020466	010600			MOV	SP,R0
4443	020470	104415			TRAP	C\$PNTX
4444	020472	062706	000006		ADD	#6,SP
4445	020476			PRINTX	#FMT26,VREGS+8.,VREGS+10.,VREGS+12.,VREGS+14.	
4446	020476	013746	002354		MOV	VREGS+14,-(SP)
4447	020502	013746	002352		MOV	VREGS+12,-(SP)
4448	020506	013746	002350		MOV	VREGS+10,-(SP)
4449	020512	013746	002346		MOV	VREGS+8,-(SP)
4450	020516	012746	012677		MOV	#FMT26,-(SP)
4451	020522	012746	000005		MOV	#5,-(SP)
4452	020526	010600			MOV	SP,R0
4453	020530	104415			TRAP	C\$PNTX
4454	020532	062706	000014		ADD	#14,SP
4455	020536			PRINTX	#FMT29,#TXT19	
4456	020536	012746	016015		MOV	#TXT19,-(SP)
4457	020542	012746	012741		MOV	#FMT29,-(SP)
4458	020546	012746	000002		MOV	#2,-(SP)
4459	020552	010600			MOV	SP,R0
4460	020554	104415			TRAP	C\$PNTX
4461	020556	062706	000006		ADD	#6,SP
4462	020562			PRINTX	#FMT25,VREGS+16.,VREGS+18.,VREGS+20.,VREGS+22.	
4463	020562	013746	002364		MOV	VREGS+22,-(SP)
4464	020566	013746	002362		MOV	VREGS+20,-(SP)
4465	020572	013746	002360		MOV	VREGS+18,-(SP)
4466	020576	013746	002356		MOV	VREGS+16,-(SP)
4467	020602	012746	012647		MOV	#FMT25,-(SP)
4468	020606	012746	000005		MOV	#5,-(SP)
4469	020612	010600			MOV	SP,R0
4470	020614	104415			TRAP	C\$PNTX
4471	020616	062706	000014		ADD	#14,SP
4472	020622			PRINTX	#FMT29,#TXT20	
4473	020622	012746	016051		MOV	#TXT20,-(SP)
4474	020626	012746	012741		MOV	#FMT29,-(SP)
4475	020632	012746	000002		MOV	#2,-(SP)
4476	020636	010600			MOV	SP,R0
4477	020640	104415			TRAP	C\$PNTX
4478	020642	062706	000006		ADD	#6,SP
4479	020646			PRINTX	#FMT26,VREGS+24.,VREGS+26.,VREGS+28.,VREGS+30.	
4480	020646	013746	002374		MOV	VREGS+30,-(SP)
4481	020652	013746	002372		MOV	VREGS+28,-(SP)
4482	020656	013746	002370		MOV	VREGS+26,-(SP)
4483	020662	013746	002366		MOV	VREGS+24,-(SP)
4484	020666	012746	012677		MOV	#FMT26,-(SP)
4485	020672	012746	000005		MOV	#5,-(SP)

CVDMCA.P11 10-DEC-80 09:14

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

4486 020676 010600
 4487 020700 104415
 4488 020702 062706 000014
 4489 020706 000207
 4490
 4491
 4492
 4493
 4494
 4495
 4496 020710 004737 004402
 4497 020714
 4498 020714 012746 015604
 4499 020720 012746 015567
 4500 020724 012746 012634
 4501 020730 012746 000003
 4502 020734 010600
 4503 020736 104415
 4504 020740 062706 000010
 4505 020744
 4506 020744 013746 002324
 4507 020750 013746 002322
 4508 020754 013746 002320
 4509 020760 013746 002316
 4510 020764 012746 012647
 4511 020770 012746 000005
 4512 020774 010600
 4513 020776 104415
 4514 021000 062706 000014
 4515 021004
 4516 021004 012746 015642
 4517 021010 012746 012741
 4518 021014 012746 000002
 4519 021020 010600
 4520 021022 104415
 4521 021024 062706 000006
 4522 021030
 4523 021030 013746 002334
 4524 021034 013746 002332
 4525 021040 013746 002330
 4526 021044 013746 002326
 4527 021050 012746 012677
 4528 021054 012746 000005
 4529 021060 010600
 4530 021062 104415
 4531 021064 062706 000014
 4532 021070 000207
 4533
 4534

RTS PC

MOV SP,RO
TRAP C\$PNTX
ADD #14,SP

:-----
:SBTTLERROR 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,RO
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,RO
TRAP C\$PNTX
ADD #14,SP

PRINTX #FMT29,#TXT15

MOV #TXT15,-(SP)
MOV #FMT29,-(SP)
MOV #2,-(SP)
MOV SP,RO
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,RO
TRAP C\$PNTX
ADD #14,SP

RTS PC

CVDMCA.P11 10-DEC-80 09:14

LOAD DEVICE PROTECTION TABLE

```

4535
4536
4537
4538
4539
4540
4541
4542 021072
4543 021072
4544 021072 177777
4545 021074 177777
4546 021076 177777
4547 021100

```

.SBTTL LOAD DEVICE PROTECTION TABLE

```

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

```

BGNPROT

```

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

```

LSPROT::

CVDMCA.P11 10-DEC-80 09:14

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.
:////////////////////
    
```

```

4548
4549
4550
4551
4552
4553
4554
4555 021100          BGNINIT
4556 021100          LSINIT::
4557
4558 021100 010637 002414      MOV     SP,PSTACK      ;SAVE BASE-LEVEL STACK POINTER
4559 021104 005037 002420      CLR     SUBRPC         ;CLEAR SUBR CALL PC
4560 021110 005037 002454      CLR     CHPTYP        ;CLEAR USYRT CHIP TYPE INDICATOR
4561 021114 005037 002452      CLR     ERROR1        ;CLEAR ERROR FLAG
4562 021120 005037 002456      CLR     SAVLEN        ;CLEAR CHAR LENGTH FROM SETUP
4563 021124 005737 002444      TST     FRSTIM        ;SEE IF FIRST TIME THROUGH AFTER LOAD
4564 021130 001007          BNE     6$            ;BR IF NOT
4565 021132 013737 000004 002446  MOV     @#4,SAVE4      ;SAVE ERROR TRAP VECTOR
4566 021140 013737 000006 002450  MOV     @#6,SAVE6
4567 021146 000406          BR      9$
4568
4569 021150 013737 002446 000004 6$:  MOV     SAVE4,@#4      ;RESTORE ERROR TRAP VECTOR
4570 021156 013737 002450 000006  MOV     SAVE6,@#6
4571
4572 021164 012737 000001 002444 9$:  MOV     #1,FRSTIM     ;MARK FLAG FOR NEXT TIME THROUGH
4573
4574          ;SEE IF PROGRAM JUST STARTED, BR IF YES
4575 021172          READEF #EF.START
4576 021172 012700 000040          MOV     #EF.START,RO
4577 021176 104447          TRAP   CSREFG
4578 021200          BCOMPLETE      STARST
4579 021200 103415          BCS    STARST
4580
4581          ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
4582 021202          READEF #EF.RESTART
4583 021202 012700 000037          MOV     #EF.RESTART,RO
4584 021206 104447          TRAP   CSREFG
4585 021210          BCOMPLETE      STARST
4586 021210 103411          BCS    STARST
4587
4588          ;SEE IF THIS IS A NEW PASS, BR IF YES
4589 021212          READEF #EF.NEW
4590 021212 012700 000035          MOV     #EF.NEW,RO
4591 021216 104447          TRAP   CSREFG
4592 021220          BCOMPLETE      NEWST
4593 021220 103411          BCS    NEWST
4594
4595          ;SEE IF PROGRAM WAS JUST CONTINUED
4596 021222          READEF #EF.CONTINUE
4597 021222 012700 000036          MOV     #EF.CONTINUE,RO
4598 021226 104447          TRAP   CSREFG
4599 021230          BCOMPLETE      ENDIT
4600 021230 103470          BCS    ENDIT
4601 021232 000414          BR      GETPRM
4602
4603          STARST:
    
```

CVDMA.P11 10-DEC-80 09:14

INITIALIZE SECTION

```

4604 021234 005037 002466          CLR    STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
4605
4606          ;CLEAR DEVICE MAP
4607 021240 005037 002460          CLR    DEVMAP
4608 021244
4609 021244 012737 177777 002410  NEWST:  MOV    #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
4610 021252 005237 002466          INC    STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
4611 021256 012737 000001 002462          MOV    #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
4612
4613          ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
4614          ; CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
4615 021264
4616 021264 005237 002410  GETPRM:  INC    LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
4617 021270          GPHARD LOGDEV,R1      ;GET P-TABLE POINTER INTO R1
4618 021270 013700 002410          MOV    LOGDEV,R0
4619 021274 104442          TRAP  CSGPHRD
4620 021276 010001          MOV    R0,R1
4621 021300          BCOMPLETE 10$        ;BR IF DEVICE AVAILABLE
4622 021300 103403          BCS    10$
4623 021302 006337 002462          ASL    DEVPTR          ;SHIFT DEVICE POINTER
4624 021306 000766          BR     GETPRM         ;SKIP THIS DEVICE
4625 021310 053737 002462 002460 10$:  BIS    DEVPTR,DEVMAP    ;SET BIT FOR THIS DEVICE
4626 021316 006337 002462          ASL    DEVPTR          ;SHIFT BIT POINTER
4627
4628 021322 012102          MOV    (R1)+,R2        ;R2=CSR ADDR VALUE
4629 021324 012703 002472          MOV    #MPCSR,R3      ;R3=POINTER TO CSR ADDR STORAGE AREA
4630
4631 021330 010223 11$:  MOV    R2,(R3)+        ;PUT CSR ADDRESSES IN 'BSEL' AREA
4632 021332 005202          INC    R2              ;BUMP BSEL ADDR
4633 021334 022703 002532          CMP    #BSEL17+2,R3   ;ALL 16 ADDRESSES MOVED ?
4634 021340 001373          BNE    11$             ;NO: DO ANOTHER ADDRESS
4635          ;YES: CONTINUE
4636
4637 021342 011137 002532          MOV    (R1),MPIVEC     ;GET DMV11 INPUT INTRPT VECTOR
4638 021346 012137 002534          MOV    (R1)+,MPOVEC
4639 021352 062737 000004 002534          ADD    #4,MPOVEC      ;GET DMV11 OUTPUT INTRPT VECTOR
4640 021360 012137 002536          MOV    (R1)+,MPRIOR   ;GET DMV11 DEVICE PRIORITY
4641 021364 012137 002540          MOV    (R1)+,LUSWI1   ;GET LU SWITCH PACK #1
4642 021370 012137 002542          MOV    (R1)+,LUSWI2   ;GET LU SWITCH PACK #2
4643 021374 012137 002544          MOV    (R1)+,BRDTYP   ;GET DMV-11 BOARD TYPE
4644 021400 012137 002546          MOV    (R1)+,TSTCON   ;GET TEST CONNECTOR INDICATOR
4645 021404 011137 002550          MOV    (R1),BDRATE    ;GET BAUD RATE FOR THIS DEVICE
4646          ;ISSUE LSI BUS RESET, TO INIT DMV11
4647 021410          BRESET
4648 021410 104433          TRAP  CSRESET
4649 021412          ENDIT:
4650 021412          ENDINIT
4651 021412          L10012:
4652 021412 104411          TRAP  CSINIT

```

CVDMCA.P11 10-DEC-80 09:14

AUTO DROP UNIT SECTION

.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.

4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669 021414
4670 021414
4671
4672 021414
4673 021414 012746 000000
4674 021420 012746 021532
4675 021424 012746 000004
4676 021430 012746 000003
4677 021434 104437
4678 021436 062706 000010
4679 021442 005037 002622
4680 021446 012702 000001
4681 021452 013703 002472
4682
4683 021456 105723
4684 021460 006302
4685 021462 103375
4686
4687 021464 013703 002472
4688 021470 012702 000001
4689 021474 005723
4690 021476 006302
4691 021500 006302
4692 021502 103374
4693
4694 021504
4695 021504 012700 000004
4696 021510 104436
4697 021512 005737 002622
4698 021516 001403
4699 021520
4700 021520 013700 002410
4701 021524 104451
4702
4703 021526 000240
4704
4705 021530
4706 021530
4707 021530 104461
4708

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 CSVEC
ADD #10, SP

CLR TMO ;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\$

2\$: MOV BSEL0, R3 ;RE-INIT ADDRESS POINTER
MOV #1, R2 ;RE-INIT FLAG BIT
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 CSCVEC

TST TMO ;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 CSODDU

AD.OK: NOP ;(FOR PATCHING IN A HALT IF NECESSARY)

ENDAUTO

L10013:

TRAP CSAUTO

CVDMCA.P11 10-DEC-80 09:14

AUTO DROP UNIT SECTION

4709 021532 050237 002622
4710 021536 000002
4711

AD.HIT: BIS R2, TMP0
RTI

;FLAG THE HIT IF WE GET IT!
;RETURN

CVDMCA.P11 10-DEC-80 09:14

CLEANUP CODING SECTION

.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.

4712
4713
4714
4715
4716
4717
4718
4719 021540
4720 021540
4721
4722
4723 021540
4724 021540
4725 021540 104412

BGNCLN

L\$CLEAN::

ENDCLN

L10014: TRAP C\$CLEAN

CVDMCA.P11 10-DEC-80 09:14

DROP UNIT SECTION

.SBTTL DROP UNIT SECTION

```

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

```

```

4726
4727
4728
4729
4730
4731
4732
4733 021542
4734 021542
4735
4736 021542
4737 021542 104433
4738 021544
4739 021544
4740 021544 104453

```

```

          BGNDU
:ISSUE UNIBUS RESET TO CLEAN UP
          BRESET
          ENDDU

```

```

LSDU::
          TRAP  CSRESET
L10015:
          TRAP  CSDU

```


CVDMCA.P11

10-DEC-80 09:14

ADD UNIT SECTION

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

```

4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754

021546
021546
021546
021546
021546 104452

BGNAU
ENDAU

LSAU::
L10016: TRAP CSAU

CVDMCA.P11 10-DEC-80 09:14

TEST 1 -- TBMT MICROCODE INTERRUPT TEST

.SBTTL TEST 1 -- TBMT MICROCODE INTERRUPT TEST

```

:*****
:
:   TEST 1 -- TBMT MICROCODE INTERRUPT TEST
:
:   THIS TEST CHECKS THE OPERATION OF THE TBMT (IRQ) INTERRUPT.
:   THIS IS DONE BY ISSUING THE 'SET MAINTENANCE INTERRUPT FLAG AND CLEAR
:   INTERRUPT DISABLE IN PROCESSOR STATUS' COMMAND WHILE IN THE MAINTENANCE
:   LOOP AND THEN CHECKING FOR BIT 7 OF BSEL3 TO BE SET (THE BIT IS SET
:   BY THE MICROCODE WHEN THE TBMT INTERRUPT OCCURS).
:*****

```

```

4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770 021550
4771
4772 021550 004737 003374
4773 021554 103003
4774 021556
4775 021556 104460
4776 021560
4777 021560 104410
4778 021562 000144
4779
4780 021564 004537 005432
4781 021570 000110
4782 021572 103003
4783 021574
4784 021574 104460
4785 021576
4786 021576 104410
4787 021600 000126
4788
4789 021602 012777 000007 160666
4790
4791 021610 010346
4792 021612 012703 000050
4793 021616 077301
4794 021620 012603
4795
4796 021622 132777 000200 160646
4797 021630 001013
4798
4799 021632 004737 004210
4800 021636 012737 000007 002400
4801 021644
4802
4803 021644 104455
4804 021646 000047
4805 021650 013235
4806 021652 017030
4807 021654
4808 021654 104410
4809 021656 000050
4810

```

```

:
:   BGNTST
:
:   T1::
:
:   JSR    PC,MSTCLR    ;PUT THE MICROPROCESSOR IN THE MAINTENANCE LOOP
:   BCC    .+8.         ;IF NO ERROR, PROCEED
:   ERROR  ;ELSE, REPORT IT AND
:
:   ESCAPE TST         ;   EXIT THIS TEST
:
:   TRAP   CSERROR
:   .WORD L10017-.
:
:   JSR    R5,CKUSTS   ;CHK USYRT STATUS FOR INIT STATE
:   110
:   BCC    .+8.         ;TBMT=1, TSO=1
:   ERROR  ;IF ERROR, PRINT REPORT
:
:   ESCAPE TST         ; AND SKIP REMAINDER OF TEST
:
:   TRAP   CSERROR
:   .WORD L10017-.
:
:   MOV    #DOTBMT,@SEL2 ;ISSUE 'TBMT TEST' COMMAND
:
:   MOV    R3,-(SP)
:   MOV    #40,R3
:   SOB   R3,1$
:   MOV    (SP)+,R3
:   1$:
:
:   BITB  #MRDY,@BSEL2 ;'MRDY' SHOULD BE HIGH BY NOW.
:   BNE   $$           ;BR IF NO ERROR
:
:   JSR    PC,GETWSR   ;GET SELECT REGISTERS
:   MOV    #DOTBMT,GDATA ;IDENTIFY REQUESTED FUNCTION
:   GEDF  EM4,ERR4    ;REPORT 'MRDY' TIMEOUT ERROR...
:   ;   'DEVICE FATAL' ERROR # 39
:
:   TRAP   CSERDF
:   .WORD 39
:   .WORD EM4
:   .WORD ERR4
:
:   ESCAPE TST         ;AND EXIT TEST
:
:   TRAP   CSESCAPE
:   .WORD L10017-.

```


CVDMCA.P11 10-DEC-80 09:14

TEST 2 -- SWITCH SETTING TEST

.SBTTL TEST 2 -- SWITCH SETTING TEST

4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851 021730
4852 021730 004737 005420
4853
4854 021734
4855 021734
4856 021734 104402
4857 021736 004537 003610
4858 021742 121400
4859 021744 000000
4860 021746 004537 003610
4861 021752 121000
4862 021754 000000
4863
4864 021756 123737 021744 002540
4865 021764 001004
4866
4867 021766 123737 021754 002542
4868 021774 001405
4869
4870 021776
4871
4872 021776 104455
4873 022000 000051
4874 022002 022346
4875 022004 022116
4876 022006
4877 022006 104406
4878
4879 022010 000240
4880 022012
4881 022012
4882 022012 104403
4883
4884 022014
4885 022014
4886 022014 104402

```

:*****
:*
:*      TEST 2 -- SWITCH SETTING TEST
:*
:* SUBTEST #1:
:* THE TWO READABLE SWITCH PACKS WILL BE SAMPLED AND COMPARED AGAINST THE 2
:* VALUES IN THE P-TABLE. AN ERROR IS REPORTED ON A MISMATCH.
:*
:* SUBTEST #2:
:* THE SPEED SELECT SWITCH (SPDSEL) IS READ VIA THE VIAORA REGISTER (BIT PA4)
:* AND COMPARED AGAINST THE BAUD RATE VALUE IN THE P-TABLE. IF A MISMATCH
:* OCCURS IT WILL BE REPORTED. NOTE: THIS SUBROUTINE IS NOT RUN IF AN M8064
:* BOARD IS BEING TESTED (IT ONLY RUNS @56K... MAKING A SPEED SWITCH USELESS).
:*
:-----
:
:      BGNTST
:
:      JSR      PC,INIDMV      ;INIT DMV-11 (MAINT LOOP)      T2::
:-----
:      BGNSUB
:
:      T2.1:
:
:      JSR      R5,READI      ;GET 'DDCMP ADDRESS'
:      .WORD   SWPDDCMP
:      T3.SW1: .WORD   0      ; (IT WILL BE PUT HERE)
:      JSR      R5,READI      ;GET 'BOOT ADDRESS'
:      .WORD   SWPBOT
:      T3.SW2: .WORD   0      ; (IT WILL BE PUT HERE)
:
:      CMPB    T3.SW1,LUSWI1   ;DOES 'DDCMP ADDRESS' MATCH P-TABLE VALUE
:      BNE     T3.ERR         ;NO, REPORT ERROR
:
:      CMPB    T3.SW2,LUSWI2   ;DOES 'BOOT ADDRESS' MATCH P-TABLE VALUE
:      BEQ     T3.OK          ;NO, REPORT ERROR
:
:      T3.ERR: GEDF    T3.EHD,T3.EM1 ;REPORT SWITCH SETTINGS DON'T MATCH P-TABLE
:                      ; 'DEVICE FATAL' ERROR # 41
:
:                      TRAP    CSERDF
:                      .WORD   41
:                      .WORD   T3.EHD
:                      .WORD   T3.EM1
:
:      CKLOOP
:
:                      TRAP    CSCLP1
:
:      T3.OK:  NOP
:             ENDSUB
:
:
:                      L10021:
:                      TRAP    CSESUB
:-----
:      BGNSUB
:
:                      T2.2:
:                      TRAP    CSBSUB

```

CVDNCA.P11 10-DEC-80 09:14

TEST 2 -- SWITCH SETTING TEST

```

4887 022016 005737 002544      TST      BRDTYP      ;IS THIS AN M8064 ?
4888 022022 001433              BEQ      10$         ; IF YES: THEN SKIP THIS SUBROUTINE
4889 022024 004537 003610      JSR      R5,READI   ;GET VIAORA REGISTER
4890 022030 120017              VIAORA
4891 022032 000000      1$:      000              ;STATUS WORD GOES HERE
4892 022034 103003              BCC      .+8.
4893 022036              ERROR
4894 022036 104460              ESCAPE  SUB              TRAP      CSERROR
4895 022040              ESCAPE  SUB              TRAP      CSESCAPE
4896 022040 104410              ESCAPE  SUB              .WORD    L10022-.
4897 022042 000050              ESCAPE  SUB
4898 022044 142737 000357 022032  BICB    #357,1$      ;CLEAR ALL BUT SPEED SELECT BIT(PA4)
4899 022052 106237 022032      ASRB    1$          ;RIGHT JUSTIFY SPDSEL SWITCH BIT FOR
4900 022056 106237 022032      ASRB    1$          ; COMPARISON WITH OPERATOR'S REPLY
4901 022062 106237 022032      ASRB    1$
4902 022066 106237 022032      ASRB    1$
4903 022072 123737 022032 002550  CMPB    1$,BDRATE   ;IS THE SWITCH IN THE DESIRED POSITION?
4904 022100 001404              BEQ      10$         ; IF YES: END THE SUBROUTINE/TEST
4905
4906 022102              GEDF    T3.6,T3.EM2 ;REPORT ERROR
4907
4908 022102 104455              GEDF    T3.6,T3.EM2 ; 'DEVICE FATAL' ERROR # 42
4909 022104 000052              GEDF    T3.6,T3.EM2 ;
4910 022106 022644              GEDF    T3.6,T3.EM2 ;
4911 022110 022310              GEDF    T3.6,T3.EM2 ;
4912 022112              GEDF    T3.6,T3.EM2 ;
4913 022112              GEDF    T3.6,T3.EM2 ;
4914 022112 104403              GEDF    T3.6,T3.EM2 ;
4915
4916 022114              GEDF    T3.6,T3.EM2 ;
4917 022114              GEDF    T3.6,T3.EM2 ;
4918 022114 104401              GEDF    T3.6,T3.EM2 ;
4919
4920 022116              GEDF    T3.6,T3.EM2 ;
4921 022116              GEDF    T3.6,T3.EM2 ;
4922 022116              GEDF    T3.6,T3.EM2 ;
4923 022116 012746 022372      BGNMSG  T3.EM1      ;:PRINT ERROR MESSAGE
4924 022122 012746 000001      PRINTB  #T3.1
4925 022126 010600              PRINTB  #T3.1
4926 022130 104414              PRINTB  #T3.1
4927 022132 062706 000004      PRINTB  #T3.1
4928 022136              PRINTB  #T3.1
4929 022136 012746 022457      PRINTB  #T3.1
4930 022142 012746 000001      PRINTB  #T3.1
4931 022146 010600              PRINTB  #T3.1
4932 022150 104415              PRINTB  #T3.1
4933 022152 062706 000004      PRINTB  #T3.1
4934 022156              PRINTB  #T3.1
4935 022156 012746 022506      PRINTB  #T3.1
4936 022162 012746 000001      PRINTB  #T3.1
4937 022166 010600              PRINTB  #T3.1
4938 022170 104415              PRINTB  #T3.1
4939 022172 062706 000004      PRINTB  #T3.1
4940 022176              PRINTB  #T3.1
4941 022176 013746 002540      PRINTX  #T3.4,LUSWI2,LUSWI1
4942 022202 013746 002542      PRINTX  #T3.4,LUSWI2,LUSWI1

```

CVDMCA.P11 10-DEC-80 09:14

TEST 2 -- SWITCH SETTING TEST

```

4943 022206 012746 022537          MOV      #T3.4,-(SP)
4944 022212 012746 000003          MOV      #3,-(SP)
4945 022216 010600                   MOV      SP,R0
4946 022220 104415                   TRAP     C$PNTX
4947 022222 062706 000010          ADD      #10,SP
4948 022226 105037 021745          CLRB    T3.SW1+1      ;MAKE SURE BITS 8 THROUGH 15 AREN'T REPORTED
4949 022232 105037 021755          CLRB    T3.SW2+1      ; -- ESPECIALLY BIT 8!!
4950 022236                   PRINTX  #T3.5,T3.SW1,T3.SW2
4951 022236 013746 021754          MOV      T3.SW2,-(SP)
4952 022242 013746 021744          MOV      T3.SW1,-(SP)
4953 022246 012746 022602          MOV      #T3.5,-(SP)
4954 022252 012746 000003          MOV      #3,-(SP)
4955 022256 010600                   MOV      SP,R0
4956 022260 104415                   TRAP     C$PNTX
4957 022262 062706 000010          ADD      #10,SP
4958 022266                   PRINTB  #NEWLIN
4959 022266 012746 011645          MOV      #NEWLIN,-(SP)
4960 022272 012746 000001          MOV      #1,-(SP)
4961 022276 010600                   MOV      SP,R0
4962 022300 104414                   TRAP     C$PNTB
4963 022302 062706 000004          ADD      #4,SP
4964 022306                   ENDMSG
4965 022306                   L10023:
4966 022306 104423                   TRAP     C$MSG
4967
4968 022310                   BGNMSG  T3.EM2
4969 022310                   PRINTB  #FMT21,#TXT12,MPCSR
4970 022310                   T3.EM2::
4971 022310 013746 002472          MOV      MPCSR,-(SP)
4972 022314 012746 015541          MOV      #TXT12,-(SP)
4973 022320 012746 012523          MOV      #FMT21,-(SP)
4974 022324 012746 000003          MOV      #3,-(SP)
4975 022330 010600                   MOV      SP,R0
4976 022332 104414                   TRAP     C$PNTB
4977 022334 062706 000010          ADD      #10,SP
4978 022340 004737 020356          JSR     PC,ERR11$     ;GET AND PRINT VIA REGISTERS
4979 022344                   ENDMSG
4980 022344                   L10024:
4981 022344 104423                   TRAP     C$MSG
4982
4983 022346 053523 052111 044103 .NLIST BEX
022372 047045 040445 053523 T3.EHD: .ASCIZ 'SWITCH SETTING TEST'
022457 045 022516 031123 T3.1: .ASCIZ '/ZN$ASWITCH PACK SETTING DOES NOT MATCH P-TABLE ENTRY/'
022506 047045 051445 031062 T3.2: .ASCIZ '/ZN$S22%A DDCMP BOOT'
022537 045 022516 033123 T3.3: .ASCIZ '/ZN$S22%AADDRESS ADDRESS'
022602 047045 051445 022466 T3.4: .ASCIZ '/ZN$S6$AP-TABLE VALUES:%S1%03%S6%03'
022644 050123 051504 046105 T3.5: .ASCIZ '/ZN$S6$AREAD FROM DMV:%S2%03%S6%03'
T3.6: .ASCIZ '/SPDSEL SWITCH DOESN'T MATCH P-TABLE BAUD RATE ENTRY/'
.LIST BEX
.EVEN
4984

```

CVDMCA.P11 10-DEC-80 09:14

TEST 3 -- USYRT MASTER CLEAR TEST

.SBTTL TEST 3 -- USYRT MASTER CLEAR TEST

```

*****
*
* TEST 3 -- USYRT MASTER CLEAR TEST
*
* ALL REGISTERS ARE LOADED WITH PATTERN E IN THE SAME SEQUENCE AS FOR
* PATTERN F BELOW. THE USYRT IS THEN CLEARED BY A MASTER CLEAR
* (BIT 6 OF BSEL 1). ALL REGISTERS ARE THEN CHECKED FOR
* THE PROPER CONTENTS. THE INITIALIZED STATE OF THE REGISTERS IS CHECKED
* AGAINST DATA PATTERN F.
*
* PATTERN E: 377, 377, 377, 377, 377, 377, 377, 366.
* PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110.
*
* SEQUENCE OF REGISTERS AS USED WITH PATTERNS E & F:
*
* RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
*
*****

```

```

4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008 022730
5009 022730 004737 005420
5010 022734 005001
5011 022736 012702 002652
5012 022742 016137 002552 022760 1$:
5013 022750 112237 022762
5014 022754 004537 003734
5015 022760 000000 2$:
5016 022762 000000 3$:
5017 022764 062701 000002
5018 022770 020127 000020
5019 022774 002762
5020 022776 004737 005420
5021 023002 005001
5022 023004 012702 002662
5023 023010 016137 002552 023022 6$:
5024 023016 004537 003610
5025 023022 000000 7$:
5026 023024 000000 8$:
5027 023026 123722 023024
5028 023032 001422
5029 023034 010137 002412
5030 023040 006237 002412
5031 023044 005037 002400
5032 023050 116237 177777 002400
5033 023056 013737 023024 002402
5034
5035 023064
5036
5037 023064 104455
5038 023066 000053
5039 023070 013103
5040 023072 017372

```

```

: BGNST
:
: T3::
JSR PC,INIDMV ;INIT DMV-11
CLR R1 ;INIT USYRT REG ADRS POINTER
MOV #PATE,R2 ;INIT PATTERN E POINTER
MOV USYREG(R1),2$ ;GET USYRT REG ADRS
MOVB (R2)+,3$ ;GET A PATTERN BYTE
JSR R5,WRITEI ;WRITE A USYRT REG
;USYRT REG ADRS GOES HERE
2$: .WORD 0
3$: .WORD 0 ;DATA BYTE GOES HERE
ADD #2,R1 ;INCR REG ADRS PTR
CMP R1,#16. ;SEE IF ALL REGS WRITTEN YET
BLT 1$ ;BR IF NOT
JSR PC,INIDMV ;ISSUE MASTER CLEAR
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
;USYRT REG ADRS GOES HERE
7$: .WORD 0
8$: .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
;REPORT REG NOT CLEARED BY MASTER CLEAR
GEDF EM1,ERR10
; 'DEVICE FATAL' ERROR # 43
TRAP CSERDF
; .WORD 43
; .WORD EM1
; .WORD ERR10

```

CVDMCA.P11 10-DEC-80 09:14

TEST 3 -- USYRT MASTER CLEAR TEST

5041	023074		
5042	023074	104410	
5043	023076	000014	
5044	023100	062701	000002
5045	023104	020127	000020
5046	023110	002737	
5047	023112		
5048	023112		
5049	023112	104401	

ESCAPE TST

9\$: ADD #2,R1
 CMP R1,#16.
 BIT 6\$
 ENDTST

:INCR USYRT REG ADRS PTR
 :SEE IF ALL REGS READ YET
 :BR IF NOT

TRAP
 .WORD C\$ESCAPE
 L10025-

L10025:
 TRAP C\$ETST

CVDMCA.P11 10-DEC-80 09:14

TEST 4 -- USYRT PROGRAM RESET TEST

.SBTTL TEST 4 -- USYRT PROGRAM RESET TEST

TEST 4 -- USYRT PROGRAM RESET TEST

ALL REGISTERS ARE LOADED WITH PATTERN E IN THE SAME SEQUENCE AS FOR PATTERN F BELOW. THE USYRT IS THEN RESET BY ASSERTING PROGRAM RESET (BIT 0 @ A000) IN THE 6522 VIA. ALL REGISTERS ARE THEN CHECKED FOR THE PROPER CONTENTS. THE INITIALIZED STATE OF THE REGISTERS IS CHECKED AGAINST DATA PATTERN F.

PATTERN E: 377, 377, 377, 377, 377, 377, 377, 366.
PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110.

SEQUENCE OF REGISTERS AS USED WITH PATTERNS E & F:

RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG

5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5067
5068
5069
5070
5071
5072
5073 023114
5074 023114 004737 005420
5075 023120 005001
5076 023122 012702 002652
5077 023126 016137 002552 023144 1\$:
5078 023134 112237 023146
5079 023140 004537 003734
5080 023144 000000 2\$:
5081 023146 000000 3\$:
5082 023150 062701 000002
5083 023154 020127 000020
5084 023160 002762
5085
5086 023162 004537 005126
5087 023166 103003
5088 023170
5089 023170 104460
5090 023172
5091 023172 104410
5092 023174 000002
5093
5094
5095 023176
5096 023176
5097 023176 104401

BGNTST

JSR PC,INIDMV ;INIT DMV-11
CLR R1 ;INIT USYRT REG ADRS POINTER
MOV #PATE,R2 ;INIT PATTERN E POINTER
MOV USYREG(R1),2\$;GET USYRT REG ADRS
MOVB (R2)+,3\$;GET A PATTERN BYTE
JSR R5,WRITEI ;WRITE A USYRT REG
.WORD 0 ;USYRT REG ADRS GOES HERE
.WORD 0 ;DATA BYTE GOES HERE
ADD #2,R1 ;INCR REG ADRS PTR
CMP R1,#16. ;SEE IF ALL REGS WRITTEN YET
BLT 1\$;BR IF NOT

JSR R5,RSTCHK ;RESET USYRT/VERIFY SAME
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR

ESCAPE TST ;SKIP TO END OF TEST

ENDTST

TRAP C\$ERROR

TRAP C\$ESCAPE
.WORD L10026-

L10026:
TRAP C\$SETST

CVDMA.P11 10-DEC-80 09:14

TEST 5 -- USYRT REGISTER ADDRESSING TEST

.SBTTL TEST 5 -- USYRT REGISTER ADDRESSING TEST

```

*****
*
* TEST 5 -- USYRT REGISTER ADDRESSING TEST
*
* FIRST, A MASTER CLEAR IS ISSUED, TO INITIALIZE THE USYRT REGS TO
* PATTERN F. THEN, EACH REGISTER IS WRITTEN WITH A BYTE OF PATTERN J,
* AND AFTER EACH IS WRITTEN, ALL ARE READ AND COMPARED TO THE CURRENT
* EXPECTED VALUES. THIS IS PERFORMED FOR ALL REGISTERS -- INCLUDING THE
* READ ONLY REGS -- IN ORDER TO MAKE SURE THAT EACH REGISTER ONLY RESPONDS
* TO ITS OWN ADDRESS.
* PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110
* PATTERN J: 000, 000, 001, 002, 004, 020, 040, 010
*
* SEQUENCE OF REGISTERS AS USED WITH PATTERNS F & J:
* RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
*****

```

```

5098
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119 023200
5120 023200 004737 005420
5121 023204 012702 002662
5122 023210 012703 002572
5123 023214 112223
5124 023216 020227 002672
5125 023222 103774
5126 023224 005001
5127 023226 005002
5128 023230 016137 002552 023274
5129 023236 116237 003013 023276
5130 023244 113762 023276 002572
5131 023252 023727 023274 120402
5132 023260 001003
5133 023262 142737 000100 002601
5134 023270 004537 003734
5135 023274 000000
5136 023276 000000
5137 023300 005003
5138 023302 005004
5139 023304 016337 002552 023316
5140 023312 004537 003610
5141 023316 000000
5142 023320 000000
5143 023322 123764 023320 002572
5144 023330 001420
5145 023332 010437 002412
5146 023336 005037 002400
5147 023342 116437 002572 002400
5148 023350 013737 023320 002402
5149
5150 023356
5151
5152 023356 104455
5153 023360 000054

```

```

:
: BGNTST
:
: T5::
: JSR PC,INIDMV ;ISSUE MASTER CLEAR TO INIT DMV
: MOV #PATF,R2 ;INIT PATTERN I POINTER
: MOV #REDDAT,R3 ;INIT PTR TO EXPECTED DATA AREA
1$: MOVB (R2)+,(R3)+ ;MOV PATTERN F INTO REDDAT TABLE
: CMP R2,#PATG
: BLO 1$
: CLR R1 ;INIT USYRT REG ADRS PTR FOR WRITING
: CLR R2 ;INIT INDEX FOR WRITING
2$: MOV USYREG(R1),3$ ;SET USYRT REG ADRS
: MOVB PATJ(R2),4$ ;SET DATA FOR WRITE
: MOVB 4$,REDDAT(R2) ;SET EXPECTED DATA BYTE
: CMP 3$,#TDSRL ;SEE IF WRITING TDSRL
: BNE 10$ ;BR IF NOT
: BICB #100,REDDAT+7 ;FIX EXPECTED USTAT VALUE
10$: JSR R5,WRITEI ;WRITE BYTE INTO A USYRT REG
3$: .WORD 0 ;REG ADRS GOES HERE
4$: .WORD 0 ;DATA BYTE GOES HERE
: CLR R3 ;INIT USYRT REG ADRS PTR FOR READING
: CLR R4 ;INIT INDEX FOR READING
5$: MOV USYREG(R3),6$ ;SET USYRT REG ADRS
: JSR R5,READI ;READ A USYRT REG
6$: .WORD 0 ;REG ADRS GOES HERE
7$: .WORD 0 ;DATA BYTE READ IS RETURNED HERE
: CMPB 7$,REDDAT(R4) ;SEE IF BYTE READ MATCHES EXPECTED BYTE
: BEQ 8$ ;BR IF MATCH
: MOV R4,REGNUM ;SET FAILING REG NO. FOR ERROR REPORT
: CLR GDATA ;GET EXPECTED DATA
: MOVB REDDAT(R4),GDATA
: MOV 7$,BDATA ;GET ACTUAL DATA
: REPORT REGISTER MISCOMPARE ERROR
: GEDF EM66,ERR10
:
: 'DEVICE FATAL' ERROR # 44
: TRAP CSERDF
: .WORD 44

```

CVDMCA.P11 10-DEC-80 09:14

TEST 5 -- USYRT REGISTER ADDRESSING TEST

5154	023362	013741	
5155	023364	017372	
5156	023366		
5157	023366	104410	
5158	023370	000032	
5159	023372	062703	000002
5160	023376	005204	
5161	023400	020427	000010
5162	023404	002737	
5163	023406	062701	000002
5164	023412	005202	
5165	023414	020227	000010
5166	023420	002703	
5167	023422		
5168	023422		
5169	023422	104401	

ESCAPE TST

```

8$:  ADD    #2,R3
      INC    R4
      CMP    R4,#10
      BLT   5$
      ADD    #2,R1
      INC    R2
      CMP    R2,#10
      BLT   2$

```

ENDTST

```

:INCR USYRT REG ADRS PTR FOR READING
:INCR READ INDEX
:SEE IF ALL REGS READ YET
:BR IF NOT ALL READ YET
:INCR USYRT REG ADRS PTR FOR WRITING
:INCR WRITE INDEX
:SEE IF ALL REGS WRITTEN YET
:BR IF NOT ALL WRITTEN YET

```

```

.WORD EM66
.WORD ERR10
TRAP  C$ESCAPE
.WORD L10027-

```

```

L10027: TRAP C$ETST

```

CVDMCA.P11 10-DEC-80 09:14

TEST 6 -- R/W BIT TEST OF PCSAR HIGH BYTE

.SBTTL TEST 6 -- R/W BIT TEST OF PCSAR HIGH BYTE

```

:*****
:*
:*      TEST 6 -- R/W BIT TEST OF PCSAR HIGH BYTE
:*
:*      WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN G.
:*      PATTERN G: 000, 001, 003, 004, 005, 007, 100, 101, 103, 104, 105,
:*                  107, 000, 017, 027, 041, 200, 277, 103, 144, 115, 157, 000.
:*****

```

5170
5171
5172
5173
5174
5175
5176
5177
5178
5179
5180
5181
5182
5183
5184
5185
5186
5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221

023424
023424 004737 005420
023430 103003
023432 104460
023434 104410
023436 000056
023440 012701 002672
023444 012703 000025
023450 012737 000005 002412
023456
023456 104402
023460 111137 002376
023464 112137 002400
023470 012700 120405
023474 004737 004272
023500 103003
023502 104460
023504 104410
023506 000006
023510
023510 104403
023512 077317
023514
023514
023514 104401

```

:      BGNTST
:
:      JSR      PC,INIDMV      ;INIT DMV/VIA & START UP MAINT. LOOP
:      BCC     30$            ;IF AN ERROR OCCURED,
:      ERROR   ;REPORT IT &
:
:      ESCAPE  TST            ; EXIT
:
:      TRAP    C$ERROR
:
:      TRAP    C$ESCAPE
:      .WORD   L10030-
:
:      30$:   MOV     #PATG,R1      ;POINT TO PATTERN TABLE
:            MOV     #<PATH-PATG-2>,R3 ;GET # OF ENTRIES IN TABLE
:            MOV     #5,REGNUM     ;ERROR INDEX FOR PCSARH
:
:      T7.LP: BGNSUB           ;THE SUBTEST ONLY TESTS THE ONE PATTERN
:
:      T6.1: TRAP    C$BSUB
:
:      MOVB   (R1),TDATA        ;SETUP TEST DATA BYTE FOR 'STUREG'
:      MOVB   (R1)+,GDATA       ;SETUP EXPECTED DATA BYTE FOR 'STUREG'
:      MOV    #PCSARH,R0        ;SPECIFY THE REGISTER BEING TESTED
:      JSR    PC,STUREG         ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
:      BCC   10$                ;WAS AN ERROR FOUND?
:      ERROR   ;YES, REPORT IT AND
:
:      ESCAPE  TST            ; EXIT FROM THE TEST. 'CKLOOP' IS IMPLIED
:
:      TRAP    C$ERROR
:      TRAP    C$ESCAPE
:      .WORD   L10030-
:
:      10$:   ENDSUB
:
:      SOB    R3,T7.LP        ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
:
:      ;TEST IT. ELSE, FALL OUT OF LOOP AND TEST
:
:      L10031: TRAP    C$ESUB
:
:      L10030: TRAP    C$SETST

```

CVDMCA.P11 10-DEC-80 09:14

TEST 7 -- R/W BIT TEST OF S/AR REGISTER

.SBTTL TEST 7 -- R/W BIT TEST OF S/AR REGISTER

```

+++++*****
:
: * TEST 7 -- R/W BIT TEST OF S/AR REGISTER
: *
: * WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN H.
: *
: * PATTERN H: 125, 252, 000, 377, 000, 001, 002, 004, 010, 020, 040, 100,
: *             200, 000, 377, 376, 375, 373, 367, 357, 337, 277, 177, 377,
: *             000
: *
:-----*****

```

5222
5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243
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

023516
023516 004737 005420
023522 103003
023524 104460
023526 104410
023530 000056
023532 012701 002721
023536 012703 000027
023542 012737 000004 002412
023550
023550 104402
023552 111137 002376
023556 112137 002400
023562 012700 120404
023566 004737 004272
023572 103003
023574 104460
023576 104410
023600 000006
023602
023602 104403
023604 077317
023606
023606 104401

```

:
: BGNTST
:
: JSR PC,INIDMV ;INIT DMV & START UP THE MAINT. LOOP
: BCC 30$ ;IF AN ERROR OCCURED,
: ERROR ;REPORT IT &
:
: ESCAPE TST ; EXIT TRAP C$ERROR
:
: 30$: MOV #PATH,R1 ;POINT TO PATTERN TABLE
: MOV #<PATI-PATH-2>,R3 ;GET # OF ENTRIES IN TABLE
: MOV #4,REGNUM ;ERROR INDEX FOR S/AR BYTE
:
: T8.LP: BGNSUB ;THE SUBTEST ONLY TESTS THE ONE PATTERN
: T7.1: TRAP C$SUB
:
: MOVB (R1),TDATA ;SETUP TEST DATA BYTE FOR 'STUREG'
: MOVB (R1)+,GDATA ;SETUP EXPECTED DATA BYTE FOR 'STUREG'
: MOV #PC$ARL,R0 ;SPECIFY THE REGISTER BEING TESTED
: JSR PC,STUREG ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
: BCC 10$ ;WAS AN ERROR FOUND?
: ERROR ;YES, REPORT IT AND
:
: ESCAPE TST ; EXIT FROM THE TEST. 'CKLOOP' IS IMPLIED TRAP C$ERROR
: TRAP C$ESCAPE
: .WORD L10032-.
:
: 10$: ENDSUB
:
: SOB R3,T8.LP ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
: ;TEST IT. ELSE, FALL OUT OF LOOP AND TEST
:
: ENDTST
:
: L10033: TRAP C$ESUB
:
: L10032: TRAP C$ETST

```

CVDMCA.P11 10-DEC-80 09:14

TEST 8 -- R/W BIT TEST OF PCR REGISTER

.SBTTL TEST 8 -- R/W BIT TEST OF PCR REGISTER

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
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317
5318
5319
5320
5321
5322
5323
5324
5325
5326
5327

023610
023610 004737 005420
023614 103003
023616 104460
023620 104410
023622 000056
023624 012701 002752
023630 012703 000037
023634 012737 000006 002412
023642
023642 104402
023644 111137 002376
023650 112137 002400
023654 012700 120407
023660 004737 004272
023664 103003
023666 104460
023670 104410
023672 000006
023674
023674 104403
023676 077317
023700
023700
023700 104401

```

:*****
:*
:* TEST 8 -- R/W BIT TEST OF PCR REGISTER
:* PATTERN I IS LOADED INTO PCR (HIGH) AND THE DATA READ BACK AND
:* CHECKED.
:* PATTERN I: 000, 041, 102, 143, 204, 245, 306, 347, 000, 001, 002,
:* 004, 040, 100, 200, 000, 346, 345, 343, 307, 247, 147, 347, 242,
:* 105, 347, 010, 020, 357, 357, 030, 027, 377.
:*****
:
: BGNTST
:
: JSR PC,INIDMV ;INIT DMV/VIA & START UP MAINT. LOOP T8::
: BCC 30$ ;IF AN ERROR OCCURED,
: ERROR ;REPORT IT &
: ESCAPE TST ; EXIT TRAP C$ERROR
: .WORD L10034-.
:
: 30$: MOV #PATI,R1 ;POINT TO PATTERN TABLE
: MOV #<PATJ-PATI-2>,R3 ;GET # OF ENTRIES IN TABLE
: MOV #6,REGNUM ;ERROR INDEX FOR PCR REGISTER
:
: T9.LP: BGNSUB ;THE SUBTEST ONLY TESTS THE ONE PATTERN
: T8.1: TRAP C$BSUB
:
: MOVB (R1),TDATA ;SETUP TEST DATA BYTE FOR 'STUREG'
: MOVB (R1)+,GDATA ;SETUP EXPECTED DATA BYTE FOR 'STUREG'
: MOV #PCR,R0 ;SPECIFY THE REGISTER BEING TESTED
: JSR PC,STUREG ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
: BCC 10$ ;WAS AN ERROR FOUND?
: ERROR ;YES, REPORT IT AND
: ESCAPE TST ; EXIT FROM THE TEST. 'CKLOOP' IS IMPLIED TRAP C$ERROR
: .WORD L10034-.
:
: 10$: ENDSUB
:
: SOB R3,T9.LP ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
: ;TEST IT. ELSE, FALL OUT OF LOOP AND TEST
:
: ENDTST
:
: L10035: TRAP C$ESUB
:
: L10034: TRAP C$SETST

```

CVDMCA.P11 10-DEC-80 09:14

TEST 9 -- R/W BIT TEST OF TDSR REGISTER'S HIGH BYTE

.SBTTL TEST 9 -- R/W BIT TEST OF TDSR REGISTER'S HIGH BYTE

5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371
5372
5373
5374
5375
5376
5377
5378
5379
5380
5381
5382
5383

023702
023702 004737 005420
023706 103003
023710
023710 104460
023712
023712 104410
023714 000114
023716 012701 003023
023722 012703 000017
023726 012702 003044
023732
023732
023732 104402
023734 112137 002376
023740 112237 002400
023744 004537 003722
023750 120403
023752 002376
023754 004537 003476
023760 120403
023762 002402
023764 042737 177760 002402
023772 023737 002400 002402
024000 001411
024002 012737 000003 002412

```
*****
*
* TEST 9 -- R/W BIT TEST OF TDSR REGISTER'S HIGH BYTE
*
* PATTERN K IS LOADED INTO TDSR (HIGH) AND THE DATA READ BACK IS
* COMPARED AGAINST PATTERN L. (UNPREDICTABLE BITS ARE MASKED OFF TO 0
* WHEN READING FOR COMPARISON.)
*
* PATTERN K: 000, 377, 376, 375, 373, 376, 177, 377, 000, 001, 002,
* 004, 010, 200, 125, 252, 000.
*
* PATTERN L: 000, 017, 016, 015, 013, 016, 017, 017, 000, 001, 002,
* 004, 010, 000, 005, 012, 000.
*
* NOTE THAT THE UNDEFINED BITS (12, 13, & 14) ARE MASKED OFF TO 0'S
* FOR THE COMPARISON. ALSO THAT BIT 15 IS A READ/ONLY BIT AND CAN'T BE
* SET -- THEREFORE SHOULD ALWAYS BE READ AS A 0 BY THIS TEST.
*
*****
```

```

: BGNTST
:
: JSR PC,INIDMV ;INIT DMV/VIA & START UP MAINT. LOOP
: BCC 30$ ;IF AN ERROR OCCURED.
: ERROR ;REPORT IT &
: ESCAPE TST ; EXIT TRAP C$ERROR
: TRAP C$ESCAPE
: .WORD L10036-.
30$: MOV #PATK,R1 ;POINT TO PATTERN TABLE
MOV #<PATL-PATK-2>,R3 ;GET # OF ENTRIES IN TABLE
MOV #PATL,R2 ;POINT TO 'EXPECTED' DATA PATTERN TABLE
T10.LP: BGNSUB ;THE SUBTEST ONLY TESTS THE ONE PATTERN
: TRAP C$BSUB
: T9.1:
MOV (R1)+,TDATA ;SETUP TEST DATA BYTE
MOV (R2)+,GDATA ;SETUP EXPECTED DATA BYTE
JSR R5,WRITE ;WRITE TO DMV-11
: .WORD TDSRH ; DMV-11 ADDRESS WRITTEN TO
: .WORD TDATA ; LOCATION OF DATA WRITTEN
JSR R5,READ ;READ FROM DMV-11
: .WORD TDSRH ; DMV-11 ADDRESS READ FROM
: .WORD BDATA ; LOCATION WHERE READ DATA IS PUT
BIC #177760,BDATA ;MASK OUT 'DON'T CARE' BITS
CMP GDATA,BDATA ;READ DATA = EXPECTED DATA ?
BEQ 10$ ;WAS AN ERROR FOUND?
MOV #3,REGNUM ;YES: SET UP REGISTER NUMBER
```


CVDMCA.P11 10-DEC-80 09:14

TEST 10 -- R/W BIT TEST OF TXDB REGISTER

.SBTTL TEST 10 -- R/W BIT TEST OF TXDB REGISTER

```

*****
*
* TEST 10 -- R/W BIT TEST OF TXDB REGISTER
* WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN H.
* PATTERN H: 000, 001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
* 376, 375, 373, 367, 357, 337, 277, 177, 377, 000
*
*****

```

5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453

024032
024032 004737 005420
024036 103003
024040
024040 104460
024042
024042 104410
024044 000056

024046 012701 002721
024052 012703 000027
024056 012737 000002 002412

024064
024064
024064 104402

024066 111137 002376
024072 112137 002400
024076 012700 120402
024102 004737 004272
024106 103003
024110
024110 104460
024112
024112 104410
024114 000006

024116
024116
024116 104403

024120 077317

024122
024122
024122 104401

```

: BGNTST
:
: JSR PC,INIDMV ;INIT DMV/VIA & START UP MAINT. LOOP
: BCC 30$ ;IF AN ERROR OCCURED,
: ERROR ;REPORT IT &
: ESCAPE TST ; EXIT TRAP CSERROR
: ; TRAP C$ESCAPE
: ;.WORD L10040-.
:
30$: MOV #PATH,R1 ;POINT TO PATTERN TABLE
MOV #<PATI-PATH-2>,R3 ;GET # OF ENTRIES IN TABLE
MOV #2,REGNUM ;ERROR INDEX FOR TXDB REGISTER
:
T11.LP: BGNSUB ;THE SUBTEST ONLY TESTS THE ONE PATTERN
: T10.1: TRAP CSBSUB
:
MOV (R1),TDATA ;SETUP TEST DATA BYTE FOR 'STUREG'
MOV (R1)+,GDATA ;SETUP EXPECTED DATA BYTE FOR 'STUREG'
MOV #TDSRL,R0 ;SPECIFY THE REGISTER BEING TESTED
JSR PC,STUREG ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
BCC 10$ ;WAS AN ERROR FOUND?
ERROR ;YES, REPORT IT AND
: ESCAPE TST ; EXIT FROM THE TEST. 'CKLOOP' IS IMPLIED TRAP CSERROR
: ;.WORD L10040-. TRAP C$ESCAPE
:
10$: ENDSUB
:
SOB R3,T11.LP ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
:TEST IT. ELSE, FALL OUT OF LOOP AND TEST
:
ENDTST
: L10040: TRAP C$SETST

```

CVDNCA.P11 10-DEC-80 09:14

TEST 11 -- PSEUDO R/W BIT TEST OF RXDB

.SBTTL TEST 11 -- PSEUDO R/W BIT TEST OF RXDB

5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495

024124
024124 004737 005420
024130 103003
024132
024132 104460
024134
024134 104410
024136 000040
024140 012701 002721
024144 012703 000027
024150 012137 024162
024154 004537 003734
024160 120400
024162 000000
024164 004537 003610
024170 120400
024172 000000
024174 077313
024176
024176
024176 104401

*
* TEST 11 -- PSEUDO R/W BIT TEST OF RXDB
*
* WRITE, READ (BUT NO COMPARE) OF EACH WORD IN DATA PATTERN H. THIS IS
* PRIMARILY TO PROVIDE A SCOPE LOOP FUNCTION ON THIS REGISTER.
*
* PATTERN H: 000, 001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
* 376, 375, 373, 367, 357, 337, 277, 177, 377, 000
*

BGNTST
JSR PC,INIDMV ;INIT DMV/VIA & START UP MAINT. LOOP
BCC 30\$;IF AN ERROR OCCURED,
ERROR ;REPORT IT &
ESCAPE TST ; EXIT TRAP C\$ERROR
TRAP C\$ESCAPE
.WORD L10042-
T11::
30\$: MOV #PATH,R1 ;POINT TO PATTERN TABLE
MOV #<PATI-PATH-2>,R3 ;GET # OF ENTRIES IN TABLE
20\$: MOV (R1)+,2\$
JSR R5,WRITEI ;WRITE TO DMV-11
.WORD RDSRL ; DMV-11 ADDRESS WRITTEN TO
2\$: .WORD 0 ; ACTUAL DATA WRITTEN
JSR R5,READI ;READ FROM DMV-11
.WORD RDSRL ; DMV-11 ADDRESS READ FROM
.WORD 0 ; READ DATA IS PUT HERE
SOB R3,20\$;IF MORE DATA, LOOP BACK TO WRITE/READ IT,
; ELSE, FALL OUT OF LOOP AND TEST
ENDTST
L10042: TRAP C\$SETST

CVDMCA.P11 10-DEC-80 09:14

TEST 12 -- PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE

.SBTTL TEST 12 -- PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE

```

:*****
:*
:*      TEST 12 -- PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE
:*
:*  WRITE, READ (BUT NO COMPARE) OF EACH WORD IN DATA PATTERN H. THIS IS
:*  PRIMARILY TO PROVIDE A SCOPE LOOP FUNCTION ON THIS REGISTER.
:*
:*  PATTERN H: 000, 001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
:*             376, 375, 373, 367, 357, 337, 277, 177, 377, 000
:*
:-----*****

```

```

5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511 024200
5512 024200 004737 005420
5513 024204 103003
5514 024206
5515 024206 104460
5516 024210
5517 024210 104410
5518 024212 000040
5519
5520 024214 012701 002721
5521 024220 012703 000027
5522
5523 024224 012137 024236
5524
5525 024230 004537 003734
5526 024234 120401
5527 024236 000000
5528
5529 024240 004537 003610
5530 024244 120401
5531 024246 000000
5532
5533 024250 077313
5534
5535 024252
5536 024252
5537 024252 104401

```

```

:      BGNTST
:
:      JSR      PC,INIDMV      ;INIT DMV/VIA & START UP MAINT. LOOP
:      BCC     30$            ;IF AN ERROR OCCURED,
:      ERROR   ;REPORT IT &
:
:      ESCAPE  TST            ; EXIT
:
:                                     TRAP   CSERROR
:                                     .WORD  L10043-
:
:      30$:   MOV     #PATH,R1    ;POINT TO PATTERN TABLE
:      MOV     #<PATI-PATH-2>,R3 ;GET # OF ENTRIES IN TABLE
:
:      20$:   MOV     (R1)+,2$
:
:      JSR     R5,WRITEI        ;WRITE TO DMV-11
:      .WORD  RDSRH            ; DMV-11 ADDRESS WRITTEN TO
:      .WORD  0                ; ACTUAL DATA WRITTEN
:
:      JSR     R5,READI         ;READ FROM DMV-11
:      .WORD  RDSRH            ; DMV-11 ADDRESS READ FROM
:      .WORD  0                ; READ DATA IS PUT HERE
:
:      SOB    R3,20$           ;IF MORE DATA, LOOP BACK TO WRITE/READ IT,
:                               ; ELSE, FALL OUT OF LOOP AND TEST
:
:      ENDTST
:
:                                     L10043:
:                                     TRAP   CSETST

```

CVDMCA.P11 10-DEC-80 09:14

TEST 13 -- NULL CLOCK TEST

.SBTTL TEST 13 -- NULL CLOCK TEST

5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551
5552
5553
5554
5555
5556
5557
5558
5559
5560
5561
5562
5563
5564
5565
5566
5567
5568
5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593

024254
024254 004737 005420
024260 004537 003734
024264 120004
024266 000017
024270 004537 003734
024274 120005
024276 000000

024300 012737 170000 002602
024306 004537 003610
024312 120000
024314 000000
024316 132737 000200 024314
024324 001025
024326 005237 002602
024332 001365

024334 012737 000000 002412
024342 012737 000200 002400
024350 013737 024314 002402
024356 042737 000177 002402

024364

024364 104455
024366 000056
024370 013252

* TEST 13 -- NULL CLOCK TEST
* FIRST, A MASTER CLEAR IS DONE TO INIT THE DMV. THEN, THE T1 TIMER ON THE
* VIA CHIP IS PROGRAMMED FOR SQUARE WAVE CLOCK GENERATION ON PB7 (BIT 7
* OF VIA OUTPUT REG B), WITH A BAUD RATE = 56 KBAUD. THIS IS THE MODE OF
* VIA OPERATION WHICH IS USED TO GENERATE THE NULL CLOCK. THEN, THE PROGRAM
* SCANS ORB REPEATEDLY TO MONITOR THE NULL CLOCK BIT, IN THE FOLLOWING
* SEQUENCE :
* - THE PROGRAM REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 1 STATE, AND
* IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC (A GROSS TIMEOUT
* INTERVAL), AN ERROR IS REPORTED. (AT 56 KBAUD, THE CLOCK SHOULD
* HAVE A PERIOD OF ABOUT 18 MICRO-SEC.)
* - THE PROGRAM NEXT REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 0 STATE,
* AND IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC, AN ERROR IS
* REPORTED.
* - THE PROGRAM NEXT REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 1 STATE
* AGAIN, AND IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC,
* AN ERROR IS REPORTED.

```

          BGNTST
          T13::
5564      JSR      PC,INIDMV      ;INIT DMV11
5565      JSR      R5,WRITEI     ;LOAD T1C-L, T1L-L FOR 56K BAUD
5566      VIAT1A
5567      15.
5568      JSR      R5,WRITEI     ;LOAD T1C-H, T1L-H, START CLOCK
5569      VIAT1B
5570      000
-----
:WAIT FOR NULCLK BIT TO BE SET TO 1
-----
5575      MOV      #170000,REG0   ;INIT PROGRAM TIMER
5576      JSR      R5,READI     ;READ VIA ORB
5577      VIAORB
5578      .WORD    0
5579      BITB     #NULCLK,2$    ;SEE IF CLOCK BIT SET
5580      BNE     3$            ;BR IF SET
5581      INC     REG0         ;INCR TIMER
5582      BNE     1$            ;BR IF TIMER DID NOT TIME OUT
5583      ; (TIME OUT = SEVERAL HUNDRED MICRO-SEC)
5584      MOV     #0,REGNUM     ;SET VIA REG NO. FOR PRINTOUT
5585      MOV     #NULCLK,GDATA ;GET EXPECTED DATA
5586      MOV     2$,BDATA     ;GET ACTUAL DATA
5587      BIC     #177,BDATA    ;CLEAR UNUSED BITS
5588      ;REPORT NULL CLK STUCK AT 0
5589      GEDF     EM5,ERR11
5590      ;
5591      'DEVICE FATAL' ERROR # 46
5592      TRAP    CSERDF
5593      .WORD   46
          .WORD   EM5
```

CVDMCA.P11 10-DEC-80 09:14

TEST 13 -- NULL CLOCK TEST

5594 024372 017546
5595 024374
5596 024374 104410
5597 024376 000176

ESCAPE TST

.WORD ERR11
TRAP C\$ESCAPE
.WORD L10044-

:WAIT FOR NULCLK BIT TO BE CLEARED TO 0

5601 024400 012737 170000 002602
5602 024406 004537 003610
5603 024412 120000
5604 024414 000000
5605 024416 132737 000200 024414
5606 024424 001425
5607 024426 005237 002602
5608 024432 001365
5609
5610 024434 012737 000000 002412
5611 024442 012737 000000 002400
5612 024450 013737 024414 002402
5613 024456 042737 000177 002402

3\$: MOV #170000,REG0 ;INIT PROGRAM TIMER
4\$: JSR R5,READI ;READ VIA ORB
VIAORB
5\$: .WORD 0
BITB #NULCLK,5\$;SEE IF CLOCK BIT CLEARED
BEQ 6\$;BR IF CLEARED
INC REG0 ;INCR TIMER
BNE 4\$;BR IF TIMER DID NOT TIME OUT
;(TIME OUT = SEVERAL HUNDRED MICRO-SEC)
MOV #0,REGNUM ;SET VIA REG NO. FOR PRINTOUT
MOV #000,GDATA ;GET EXPECTED DATA
MOV 5\$,BDATA ;GET ACTUAL DATA
BIC #177,BDATA ;CLEAR UNUSED BITS
:REPORT NULL CLK STUCK AT 1
GEDF EM6,ERR11

; 'DEVICE FATAL' ERROR # 47
TRAP CSERDF
.WORD 47
.WORD EM6
.WORD ERR11

5614 024464
5615
5616
5617 024464 104455
5618 024466 000057
5619 024470 013302
5620 024472 017546
5621 024474
5622 024474 104410
5623 024476 000076

ESCAPE TST

TRAP C\$ESCAPE
.WORD L10044-

:WAIT FOR NULCLK BIT TO BE SET TO 1 AGAIN

5624
5625
5626
5627 024500 012737 170000 002602
5628 024506 004537 003610
5629 024512 120000
5630 024514 000000
5631 024516 132737 000200 024514
5632 024524 001023
5633 024526 005237 002602
5634 024532 001365
5635
5636 024534 012737 000000 002412
5637 024542 012737 000200 002400
5638 024550 013737 024514 002402
5639 024556 042737 000177 002402

6\$: MOV #170000,REG0 ;INIT PROGRAM TIMER
7\$: JSR R5,READI ;READ VIA ORB
VIAORB
8\$: .WORD 0
BITB #NULCLK,8\$;SEE IF CLOCK BIT SET
BNE 9\$;BR IF SET
INC REG0 ;INCR TIMER
BNE 7\$;BR IF TIMER DID NOT TIME OUT
;(TIME OUT = SEVERAL HUNDRED MICRO-SEC)
MOV #0,REGNUM ;SET VIA REG NO. FOR PRINTOUT
MOV #NULCLK,GDATA ;GET EXPECTED DATA
MOV 8\$,BDATA ;GET ACTUAL DATA
BIC #177,BDATA ;CLEAR UNUSED BITS
:REPORT NULL CLK STUCK AT 0
GEDF EM5,ERR11

; 'DEVICE FATAL' ERROR # 48
TRAP CSERDF
.WORD 48
.WORD EM5
.WORD ERR11

5640 024564
5641
5642
5643 024564 104455
5644 024566 000060
5645 024570 013252
5646 024572 017546
5647 024574
5648 024574
5649 024574

9\$:
ENDTST

L10044:

CVDMCA.P11 10-DEC-80 09:14

TEST 13 -- NULL CLOCK TEST

5650 024574 104401

TRAP CSETST

CVDMCA.P11 10-DEC-80 09:14

TEST 14 -- BCP TX RESET W/IDLE = 0

.SBTTL TEST 14 -- BCP TX RESET W/IDLE = 0

```

*****
*
* TEST 14 -- BCP TX RESET W/IDLE = 0
*
* THE USYRT IS INITIALIZED FOR 'BYTE-CONTROL PROTOCOL' (BCP) WITH IDLE
* SET TO ZERO AND A 125 SYNC CHARACTER IS LOADED INTO S/AR. A 226 SYNC
* CHARACTER IS LOADED INTO TXDB SO THAT THE SOURCE OF SYNC CHARACTERS
* CAN BE LATER DETERMINED. THE VALID STATE OF THE USYRT REGISTERS IS
* READ AND CHECKED. TXE IS ASSERTED TO ENABLE THE TRANSMITTER LOGIC.
* THEN, TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE OBSERVING
* TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
* (TXBE SHOULD GO HIGH; AT THIS TIME THE S/AR'S SYNC CHARACTER SHOULD
* BE LOADED INTO TXSO AND TSOM IS AGAIN SET -- DRIVING TXBE LOW.)
* THREE SYNC CHARACTERS ARE SENT/RECEIVED: THE FIRST TWO SYNCHRONIZE
* THE RECEIVER, THE THIRD IS DIRECTLY READ (STRIP SYNC IS OFF) AND
* COMPARED AGAINST 125 (THE S/AR SYNC CHARACTER).
* IF VALUE READ IS 226, THEN TXDB PROVIDED THE SYNC (IE: ERROR).
* THE USYRT IS THEN RESET AND REGISTERS ARE AGAIN READ AND CHECKED.
* THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY
* ONE MARK AND THREE SYNC CHARACTERS (FROM THE S/AR) IS TRANSMITTED.
* ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
* SEQUENCE.
*****

```

5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706

024576
024576 004737 005420
024602 004537 007400
024606 043525
024610 000000
024612 103003
024614 104460
024614 104410
024620 000106
024622 004537 010010
024626 000001
024630 000007
024632 004537 010010
024636 000001
024640 000010
024642 004537 010010
024646 000001
024650 000010
024652 004537 011364
024656 000000

```

BGNTST
T14::
JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP
JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!NOCHK!125 ;SET DDCMP,NO CHECK,S/AR(SYNC)=125
0 ;USE 8 BIT CHARS
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST
TRAP CSERROR
TRAP CSESCAPE
.WORD L10045-.

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 ;OUTPUT 3RD SYNC CHARACTER (125)
TSOM ;AND KNOCK DOWN TBMT
8.

JSR R5,RCV1ST ;CLOCK AND RCV TSOM
0

```

CVDMA.P11 10-DEC-80 09:14

TEST 14 -- BCP TX RESET W/IDLE = 0

5707	024660	103003		BCC	+.8.		;BR IF NO ERROR		
5708	024662			ERROR			;REPORT STACKED ERROR		
5709	024662	104460						TRAP	C\$ERROR
5710	024664			ESCAPE	TST		;SKIP TO END OF TEST		
5711	024664	104410						TRAP	C\$ESCAPE
5712	024666	000040						.WORD	L10045-
5713									
5714	024670	004537	010110	JSR	R5,RXCHAR		;READ AND CHECK FOR S/AR 125 SYNC		
5715	024674	000125		125			; AND CLOCK IN NEXT ONE.		
5716	024676	000000		0					
5717	024700	000010		8.					
5718	024702	103003		BCC	+.8.		;BR IF NO ERROR		
5719	024704			ERROR			;REPORT STACKED ERROR		
5720	024704	104460						TRAP	C\$ERROR
5721	024706			ESCAPE	TST		;SKIP TO END OF TEST		
5722	024706	104410						TRAP	C\$ESCAPE
5723	024710	000016						.WORD	L10045-
5724									
5725	024712	004537	005126	JSR	R5,RSTCHK		;RESET USYRT/VERIFY SAME		
5726	024716	103003		BCC	+.8.		;BR IF NO ERROR		
5727	024720			ERROR			;REPORT STACKED ERROR		
5728	024720	104460						TRAP	C\$ERROR
5729	024722			ESCAPE	TST				
5730	024722	104410						TRAP	C\$ESCAPE
5731	024724	000002						.WORD	L10045-
5732									
5733	024726								
5734	024726								
5735	024726	104401						L10045:	TRAP C\$ETST

ENDTST

CVDMCA.P11 10-DEC-80 09:14

TEST 15 -- BCP TX RESET W/IDLE = 1

.SBTTL TEST 15 -- BCP TX RESET W/IDLE = 1

5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791

024730
024730 004737 005420
024734 004537 007400
024740 047626
024742 000000
024744 103003
024746 104460
024750 104410
024752 000136
024754 004537 007676
024760 000226
024762 000007
024764 103003
024766 104460
024770 104410
024772 000116
024774 004537 007676
025000 000125

```
*****
*
* TEST 15 -- BCP TX RESET W/IDLE = 1
*
* THE USYRT IS INITIALIZED FOR 'BYTE-CONTROL PROTOCOL' (BCP) WITH IDLE
* SET TO ONE AND A 226 SYNC CHARACTERS LOADED INTO S/AR AND TXDB.
* THE VALID STATE OF THE USYRT REGISTERS IS READ AND CHECKED. TXE IS
* ASSERTED TO ENABLE THE TRANSMITTER LOGIC.
* THEN, TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE OBSERVING
* TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
* (TXBE SHOULD GO HIGH; AT THIS TIME THE TXDB SYNC CHARACTER SHOULD
* BE LOADED INTO TXSO AND TSOM IS AGAIN SET -- DRIVING TXBE LOW.)
* AFTER THE RECEIVER IS SYNCHRONIZED (TWO SYNC CHARACTERS), TXDB IS
* LOADED WITH A 125 AND, WITH TSOM STILL = 1 (SYNC SOURCE = TXDB), THE
* USYRT IS AGAIN CLOCKED.
* AT THIS POINT, IF THE IDLE BIT WORKED, THE VALUE 125 WILL BE READ
* BY THE RECEIVER. OTHERWISE A 226 WILL BE READ, INDICATING TXDB WASN'T
* PROVIDING THE SYNC CHARACTERS.
* WHEN TXBE GOES HIGH AGAIN, THE USYRT IS RESET.
* ALL REGISTERS ARE AGAIN READ AND CHECKED.
* THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY
* ONE MARK AND THREE SYNC'S (226,226,125 FROM TXDB) ARE TRANSMITTED.
* ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
* SEQUENCE.
*****
```

BGNTST

T15::

```
JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP
JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!IDLES!NOCHK!SYNCH ;SET DDCMP,STRIP,IDLE,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 CSERROR
;WORD C$ESCAPE
L10046-.

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
L10046-.

JSR R5,TXCHAR ;LOAD 125, TX 2ND SYNCH
125
```

CVDMA.P11 10-DEC-80 09:14

TEST 15 -- BCP TX RESET W/IDLE = 1

5792	025002	000010		8.				
5793	025004	103003		BCC	+.8.	:BR IF NO ERROR		
5794	025006			ERROR		:REPORT STACKED ERROR		
5795	025006	104460					TRAP	C\$ERROR
5796	025010			ESCAPE	TST	:SKIP TO END OF TEST		
5797	025010	104410					TRAP	C\$ESCAPE
5798	025012	000076					.WORD	L10046-
5799								
5800	025014	004537	007676	JSR	R5,TXCHAR	:LOAD 377, TX 125		
5801	025020	000377		377				
5802	025022	000010		8.				
5803	025024	103003		BCC	+.8.	:BR IF NO ERROR		
5804	025026			ERROR		:REPORT STACKED ERROR		
5805	025026	104460					TRAP	C\$ERROR
5806	025030			ESCAPE	TST	:SKIP TO END OF TEST		
5807	025030	104410					TRAP	C\$ESCAPE
5808	025032	000056					.WORD	L10046-
5809								
5810	025034	004537	011364	JSR	R5,RCV1ST	:CLOCK AND RCV 125		
5811	025040	000000		0				
5812	025042	103003		BCC	+.8.	:BR IF NO ERROR		
5813	025044			ERROR		:REPORT STACKED ERROR		
5814	025044	104460					TRAP	C\$ERROR
5815	025046			ESCAPE	TST	:SKIP TO END OF TEST		
5816	025046	104410					TRAP	C\$ESCAPE
5817	025050	000040					.WORD	L10046-
5818								
5819	025052	004537	010110	JSR	R5,RXCHAR	:READ AND CHECK FOR TXDB 125 SYNC		
5820	025056	000125		125				
5821	025060	000000		0				
5822	025062	000010		8.				
5823	025064	103003		BCC	+.8.	:BR IF NO ERROR		
5824	025066			ERROR		:REPORT STACKED ERROR		
5825	025066	104460					TRAP	C\$ERROR
5826	025070			ESCAPE	TST	:SKIP TO END OF TEST		
5827	025070	104410					TRAP	C\$ESCAPE
5828	025072	000016					.WORD	L10046-
5829								
5830	025074	004537	005126	JSR	R5,RSTCHK	:RESET USYRT/VERIFY SAME		
5831	025100	103003		BCC	+.8.	:BR IF NO ERROR		
5832	025102			ERROR		:REPORT STACKED ERROR		
5833	025102	104460					TRAP	C\$ERROR
5834	025104			ESCAPE	TST	:SKIP TO END OF TEST		
5835	025104	104410					TRAP	C\$ESCAPE
5836	025106	000002					.WORD	L10046-
5837	025110							
5838	025110			ENDTST				
5839	025110	104401					L10046:	
5840							TRAP	C\$ETST

CVDMCA.P11 10-DEC-80 09:14

TEST 16 -- BCP TX UNDERRUN W/T SOM TERMINATION

.SBTTL TEST 16 -- BCP TX UNDERRUN W/T SOM TERMINATION

TEST 16 -- BCP TX UNDERRUN W/T SOM TERMINATION

THE USYRT IS INITIALIZED FOR BCP WITH IDLE = 1 AND TXC IS MANUALLY CONTROLLED UNTIL TWO SYNC CHARACTERS AND ONE DATA CHARACTER (000) HAVE BEEN TRANSMITTED. AT THIS TIME WHEN TXBE IS ASSERTED BY THE USYRT, NO DATA IS LOADED INTO TXDB -- FORCING AN UNDER RUN CONDITION. TXU AND TERR ARE CHECKED BOTH BEFORE AND AFTER THEIR EXPECTED ASSERTIONS. AFTER THE FIRST NON-DATA CHARACTER (WHICH SHOULD BE THE MARK CHARACTER) HAS BEEN STARTED, IDLE IS SET TO 0. THIS SHOULD FORCE THE NEXT NON-DATA CHARACTER TO BE A SYNC CHARACTER FROM S/AR. WHILE THIS SYNC CHARACTER IS BEING TRANSMITTED, TSOM IS ASSERTED (CLEARING TXU AND TERR) -- IDLE IS LEFT AT 0. TXBE IS THEN CYCLED THROUGH AT LEAST ONE MORE SYNC CHARACTER AND THE TEST IS ABORTED. ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE SEQUENCE.

NOTE: BITS SHIFT OUT OF TX LSB FIRST.

BGNTST

T16::

5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851
5852
5853
5854
5855
5856
5857
5858
5859
5860
5861
5862
5863
5864
5865 025112
5866 025112 004737 005420
5867
5868 025116 004537 007400
5869 025122 047626
5870 025124 040000
5871 025126 103003
5872 025130
5873 025130 104460
5874 025132
5875 025132 104410
5876 025134 000462
5877
5878 025136 004537 007116
5879 025142 000000
5880 025144 103003
5881 025146
5882 025146 104460
5883 025150
5884 025150 104410
5885 025152 000444
5886
5887 025154 004537 007256
5888 025160 000007
5889 025162 000151
5890 025164 103003
5891 025166
5892 025166 104460
5893 025170
5894 025170 104410
5895 025172 000424
5896

```

JSR    PC,INIDMV      ;INIT DMV-11, ENTER M-LOOP
JSR    R5,INITRN     ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!IDLES!NOCHK!SYNCH ;SET DDCMP,STRIP,IDLE,NO CHECK,SYNCH=226
NORXEN ;USE 8 BIT CHARS, RECEIVER DISABLED
BCC    .+8.          ;BR IF NO ERROR
ERROR  ;REPORT STACKED ERROR
ESCAPE TST           ;SKIP TO END OF TEST
TRAP   CSERROR
JSR    R5,CHKTSO     ;CHECK 1ST BIT OF EXPECTED 'SYNCH'
0      ; CHARACTER (SHOULD BE 0)
BCC    .+8.          ;BR IF NO ERROR
ERROR  ;REPORT STACKED ERROR
ESCAPE TST           ;AND EXIT TEST
TRAP   CSERROR
JSR    R5,SERIAL     ;READ REMAINING 7 BITS OF 'SYNCH' CHARACTER
7      ; (OFF OF TSO BIT)
151    ; EXPECTED BIT SEQUENCE (0010110)
BCC    .+8.          ;BR IF NO ERROR
ERROR  ;REPORT STACKED ERROR
ESCAPE TST           ;AND EXIT TEST
TRAP   CSERROR

```

CVDMA.P11 10-DEC-80 09:14

TEST 16 -- BCP TX UNDERRUN W/TSON TERMINATION

```

5897 025174 004537 003734      JSR    R5,WRITEI      ;LOAD 1ST DATA CHARACTER (000)
5898 025200 120402              TDSRL
5899 025202 000000              000
5900
5901 025204 004537 010010      JSR    R5,TXCTRL     ;CLEAR TSON
5902 025210 000000              000
5903 025212 000000              0
5904
5905 025214 004537 007256      JSR    R5,SERIAL     ;READ 2ND SYNCH CHARACTER VIA TSO
5906 025220 000010              8.          ; 8 BIT CHAR/CLOCK TICKS
5907 025222 000151              151        ; EXPECTED BIT SEQUENCE (010010110)
5908 025224 103003              BCC      .+8.       ;BR IF NO ERROR
5909 025226 104460              ERROR      ;REPORT STACKED ERROR
5910
5911 025230 104460              ESCAPE TST          ;AND EXIT TEST
5912 025230 104410
5913 025232 000364
5914
5915
5916
5917 025234 004537 005432      JSR    R5,CKUSTS     ;CHECK USYRT STATUS
5918 025240 000114              TXACT!TBMT!TSO     ; FOR TRANSMITTER ACTIVE (NO TXU YET!)
5919 025242 103003              BCC      .+8.       ;BR IF NO ERROR
5920 025244 104460              ERROR      ;REPORT STACKED ERROR
5921 025244 104460
5922 025246 104410              ESCAPE TST          ;SKIP TO END OF TEST
5923 025246 104410
5924 025250 000346
5925
5926 025252 004537 003610      JSR    R5,READI     ;GET TX ERROR BIT (TO SEE IF SET)
5927 025256 120403              TDSRH        ; TERR BIT IS IN TDSRH
5928 025260 000000              000         ; TDSRH STORED HERE
5929 025262 133727 025260 000200 10$: BITB    10$,#BIT7   ;CHECK TERR BIT
5930 025270 001406              BEQ      15$      ; BR IF TERR NOT SET
5931 025272
5932
5933 025272 104455              GEDF    EM98,ERR12 ;REPORT ERROR: "TERR NOT CLEAR"
5934 025274 000061
5935 025276 014357
5936 025300 017722
5937 025302
5938 025302 104410
5939 025304 000312
5940
5941
5942
5943 025306 004537 007256      JSR    R5,SERIAL     ;READ DATA CHARACTER (000) VIA TSO
5944 025312 000010              8.          ; 8 BIT CHAR/CLOCK TICKS
5945 025314 000000              000        ; EXPECTED BIT SEQUENCE (00000000)
5946 025316 103003              BCC      .+8.       ;BR IF NO ERROR
5947 025320 104460              ERROR      ;REPORT STACKED ERROR
5948 025320 104460
5949 025322 104410              ESCAPE TST          ;AND EXIT TEST
5950 025322 104410
5951 025324 000272
5952

```

```

-----
: VERIFY THAT TXU AND TERR BITS ARE ZERO AT THIS POINT
-----

```

```

-----
: NOW READ 1ST DATA CHARACTER (AND CHECK TERR=1, TXU=1)
-----

```

CVDMCA.P11 10-DEC-80 09:14

TEST 16 -- BCP TX UNDERRUN W/TSON TERMINATION

```

5953 025326 004537 011614      JSR    R5,STEPLU      ;GENERATE 1 TICK TO UNDERRUN TX
5954 025332 000001              1
5955
5956 025334 004537 005432      JSR    R5,CKUSTS      ;CHECK USYRT STATUS
5957 025340 000116      TBMT!TSO!TXACT!TXU    ; FOR TBMT AND TX UNDERRUN
5958 025342 103003      BCC    .+8.           ;BR IF NO ERROR
5959 025344              ERROR              ;REPORT STACKED ERROR
5960 025344 104460              TRAP   C$ERROR
5961 025346              ESCAPE TST           ;SKIP TO END OF TEST
5962 025346 104410              TRAP   C$ESCAPE
5963 025350 000246              .WORD  L10047-.
5964
5965 025352 004537 003610      JSR    R5,READI      ;GET TX ERROR BIT (TO SEE IF SET)
5966 025356 120403      TDSRH              ; TERR BIT IS IN TDSRH
5967 025360 000000      000              ; TDSRH STORED HERE
5968 025362 133727 025360 000200 20$:  BITB   20$,#BIT7    ;CHECK TERR BIT
5969 025370 001006      BNE    25$           ; BR IF TERR IS SET
5970 025372              GEDF   EM99,ERR12   ;REPORT ERROR: 'TERR NOT SET'
5971                          ; 'DEVICE FATAL' ERROR # 50
5972 025372 104455              TRAP   C$ERDF
5973 025374 000062              .WORD  50
5974 025376 014404              .WORD  EM99
5975 025400 017722              .WORD  ERR12
5976 025402              ESCAPE TST           ;AND EXIT TEST
5977 025402 104410              TRAP   C$ESCAPE
5978 025404 000212              .WORD  L10047-.
5979
5980      ;-----
5981      ; NOW CHECK MARK CHARACTER (RESULT OF UNDERRUNNING TRANSMITTER)
5982      ;-----
5982 025406 004537 007116 25$:  JSR    R5,CHKTSO    ;CHECK 1ST BIT OF EXPECTED 'MARK'
5983 025412 000001              1              ; CHARACTER (BIT SHOULD=1)
5984 025414 103003      BCC    .+8.           ;BR IF NO ERROR
5985 025416              ERROR              ;REPORT STACKED ERROR
5986 025416 104460              TRAP   C$ERROR
5987 025420              ESCAPE TST           ;AND EXIT TEST
5988 025420 104410              TRAP   C$ESCAPE
5989 025422 000174              .WORD  L10047-.
5990
5991 025424 004537 007256      JSR    R5,SERIAL     ;READ MARK CHARACTER VIA TSO
5992 025430 000007              7              ; 7 BIT CHAR/CLOCK TICKS
5993 025432 000177      177              ; EXPECTED BIT SEQUENCE (1111111)
5994 025434 103003      BCC    .+8.           ;BR IF NO ERROR
5995 025436              ERROR              ;REPORT STACKED ERROR
5996 025436 104460              TRAP   C$ERROR
5997 025440              ESCAPE TST           ;AND EXIT TEST
5998 025440 104410              TRAP   C$ESCAPE
5999 025442 000154              .WORD  L10047-.
6000
6001      ;-----
6002      ; SET IDLE=0, VERIFY SYNCH CHARACTER IS OUTPUT
6003      ;-----
6003 025444 004537 003734      JSR    R5,WRITEI    ;SET IDLE BIT=0
6004 025450 120405      PCSARH
6005 025452 000107      PROTO!XYZ
6006
6007 025454 004537 007256      JSR    R5,SERIAL     ;READ MARK CHARACTER VIA TSO
6008 025460 000010              8.             ; 8 BIT CHAR/CLOCK TICKS
    
```

CVDMCA.P11 10-DEC-80 09:14

TEST 16 -- BCP TX UNDERRUN W/T SOM TERMINATION

```

6009 025462 000377          377          ; EXPECTED BIT SEQUENCE (11111111)
6010 025464 103003          BCC          .+8.          ; BR IF NO ERROR
6011 025466                ERROR          ; REPORT STACKED ERROR
6012 025466 104460                ESCAPE TST          ; AND EXIT TEST
6013 025470                TRAP          CSERROR
6014 025470 104410                TRAP          CSESCAPE
6015 025472 000124                .WORD          L10047-.
6016
6017 025474 004537 007256          JSR          R5,SERIAL      ; READ SYNCH CHARACTER VIA TSO
6018 025500 000010          8.          ; 8 BIT CHAR/CLOCK TICKS
6019 025502 000151          151         ; EXPECTED BIT SEQUENCE (010010110)
6020 025504 103003          BCC          .+8.          ; BR IF NO ERROR
6021 025506                ERROR          ; REPORT STACKED ERROR
6022 025506 104460                ESCAPE TST          ; AND EXIT TEST
6023 025510                TRAP          CSERROR
6024 025510 104410                TRAP          CSESCAPE
6025 025512 000104                .WORD          L10047-.
6026
6027 025514 004537 010010          JSR          R5, TXCTRL     ; * ASSERT TSOM (SHOULD CLEAR TXU, TERR)
6028 025520 000001          TSOM        ; *
6029 025522 000000          0           ; *
6030
-----
6031 ; VERIFY THAT TXU AND TERR BITS ARE ZERO AT THIS POINT
-----
6032
6033 025524 004537 005432          JSR          R5, CKUSTS     ; CHECK USYRT STATUS
6034 025530 000014          TXACT!TSO   ; FOR TRANSMITTER ACTIVE (NO TXU YET!)
6035 025532 103003          BCC          .+8.          ; BR IF NO ERROR
6036 025534                ERROR          ; REPORT STACKED ERROR
6037 025534 104460                ESCAPE TST          ; SKIP TO END OF TEST
6038 025536                TRAP          CSERROR
6039 025536 104410                TRAP          CSESCAPE
6040 025540 000056                .WORD          L10047-.
6041
6042 025542 004537 003610          JSR          R5, READI      ; GET TX ERROR BIT (TO SEE IF SET)
6043 025546 120403          TDSRH       ; TERR BIT IS IN TDSRH
6044 025550 000000          000        ; TDSRH STORED HERE
6045 025552 133727 025550 000200 17$: BITB       17$, #BIT7     ; CHECK TERR BIT
6046 025560 001406          BEQ          18$          ; BR IF TERR NOT SET
6047 025562                GEDF          EM98,ERR12   ; REPORT ERROR: "TERR NOT CLEAR"
6048 ; "DEVICE FATAL" ERROR # 51
6049 025562 104455                TRAP          CSERDF
6050 025564 000063                .WORD          51
6051 025566 014357                .WORD          EM98
6052 025570 017722                .WORD          ERR12
6053 025572                ESCAPE TST          ; AND EXIT TEST
6054 025572 104410                TRAP          CSESCAPE
6055 025574 000022                .WORD          L10047-.
6056
-----
6057 ; READ/CHECK FOR SYNCH CHARACTER
-----
6058
6059 025576 004537 007256          18$: JSR          R5,SERIAL      ; READ !SYNCH! CHARACTER VIA TSO
6060 025602 000010          8.          ; 8 BIT CHAR/CLOCK TICKS
6061 025604 000151          151         ; EXPECTED BIT SEQUENCE (10010110)
6062 025606 103003          BCC          .+8.          ; BR IF NO ERROR
6063 025610                ERROR          ; REPORT STACKED ERROR
6064 025610 104460                TRAP          CSERROR

```

CVDMCA.P11 10-DEC-80 09:14

TEST 16 -- BCP TX UNDERRUN W/TSDM TERMINATION

6065 025612
6066 025612 104410
6067 025614 000002
6068 025616
6069 025616
6070 025616 104401

ESCAPE TST

;AND EXIT TEST

ENDTST

TRAP C\$ESCAPE
.WORD L10047-

L10047:

TRAP C\$ETST

CVDMCA.P11 10-DEC-80 09:14

TEST 17 -- BCP TX UNDERRUN W/RESET TERMINATION

.SBTTL TEST 17 -- BCP TX UNDERRUN W/RESET TERMINATION

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
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126

025620
025620 004737 005420
025624 004537 007400
025630 047626
025632 040000
025634 103003
025636
025636 104460
025640
025640 104410
025642 000354
025644 004537 007116
025650 000000
025652 103003
025654
025654 104460
025656
025656 104410
025660 000336
025662 004537 007256
025666 000007
025670 000151
025672 103003
025674
025674 104460
025676
025676 104410
025700 000316
025702 004537 003734
025706 120402

```
*****
*
* TEST 17 -- BCP TX UNDERRUN W/RESET TERMINATION
*
* THE USYRT IS INITIALIZED FOR BCP WITH IDLE = 1 AND TXC IS MANUALLY
* CONTROLLED UNTIL TWO SYNC CHARACTERS AND ONE DATA CHARACTER HAVE
* BEEN TRANSMITTED. AT THIS TIME WHEN TXBE IS ASSERTED BY THE USYRT,
* NO DATA IS LOADED INTO TXDB -- FORCING AN UNDER RUN CONDITION. TXU
* AND TERR ARE CHECKED BOTH BEFORE AND AFTER THEIR EXPECTED
* ASSERTIONS. AFTER THE FIRST NON-DATA CHARACTER (WHICH SHOULD BE THE
* MARK CHARACTER) HAS BEEN STARTED, IDLE IS SET TO 0. THIS SHOULD
* FORCE THE NEXT NON-DATA CHARACTER TO BE A SYNC CHARACTER.
* IMMEDIATELY AFTER THIS SYNC CHARACTER HAS BEING TRANSMITTED, A
* PROGRAM RESET IS ISSUED AND ALL REGISTERS ARE CHECKED. ERROR
* LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
* SEQUENCE.
*****
```

```

      BGNSTST
      JSR      PC,INIDMV      ;INIT DMV-11, ENTER M-LOOP      T17::
      JSR      R5,INITRN     ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
      DDCMP!IDLES!NOCHK!SYNCH ;SET DDCMP,STRIP,IDLE,NO CHECK,SYNCH=226
      NORXEN   ;USE 8 BIT CHARS, RECEIVER DISABLED
      BCC      .+8.          ;BR IF NO ERROR
      ERROR    ;REPORT STACKED ERROR
      ESCAPE   TST           ;SKIP TO END OF TEST              TRAP      CSERROR
      .WORD    L10050-.
      JSR      R5,CHKTSO     ;CHECK 1ST BIT OF EXPECTED 'SYNCH'
      0        ; CHARACTER (SHOULD BE 0)
      BCC      .+8.          ;BR IF NO ERROR
      ERROR    ;REPORT STACKED ERROR
      ESCAPE   TST           ;AND EXIT TEST                    TRAP      CSERROR
      .WORD    L10050-.
      JSR      R5,SERIAL     ;READ REMAINING 7 BITS OF 'SYNCH' CHARACTER
      7        ; (OFF OF TSO BIT)
      151      ; EXPECTED BIT SEQUENCE (0010110)
      BCC      .+8.          ;BR IF NO ERROR
      ERROR    ;REPORT STACKED ERROR
      ESCAPE   TST           ;AND EXIT TEST                    TRAP      CSERROR
      .WORD    L10050-.
      JSR      R5,WRITEI    ;LOAD 1ST DATA CHARACTER (000)
      TDSRL
```


CVDMCA.P11 10-DEC-80 09:14

TEST 17 -- BCP TX UNDERRUN W/RESET TERMINATION

```

6127 025710 000000          000
6128
6129 025712 004537 010010    JSR    R5,TXCTRL    ;CLEAR TSOM
6130 025716 000000          000
6131 025720 000000          0
6132
6133 025722 004537 007256    JSR    R5,SERIAL    ;READ 2ND SYNCH CHARACTER VIA TSO
6134 025726 000010          8.      ; 8 BIT CHAR/CLOCK TICKS
6135 025730 000151          151     ; EXPECTED BIT SEQUENCE (010010110)
6136 025732 103003          BCC    .+8.        ;BR IF NO ERROR
6137 025734          ERROR    ;REPORT STACKED ERROR
6138 025734 104460          ESCAPE TST          ;AND EXIT TEST          TRAP    C$ERROR
6139 025736          .WORD
6140 025736 104410          TRAP    C$ESCAPE
6141 025740 000256          .WORD    L10050-.
6142
-----
6143          ; VERIFY THAT TXU AND TERR BITS ARE ZERO AT THIS POINT
-----
6144
6145 025742 004537 005432    JSR    R5,CKUSTS    ;CHECK USYRT STATUS
6146 025746 000114          TXACT!TBM!TSO      ; FOR TRANSMITTER ACTIVE (NO TXU YET!)
6147 025750 103003          BCC    .+8.        ;BR IF NO ERROR
6148 025752          ERROR    ;REPORT STACKED ERROR
6149 025752 104460          ESCAPE TST          ;SKIP TO END OF TEST          TRAP    C$ERROR
6150 025754          .WORD
6151 025754 104410          TRAP    C$ESCAPE
6152 025754 000240          .WORD    L10050-.
6153
6154 025760 004537 003610    JSR    R5,READI     ;GET TX ERROR BIT (TO SEE IF SET)
6155 025764 120403          TDSRH              ; TERR BIT IS IN TDSRH
6156 025766 000000          000               ; TDSRH STORED HERE
6157 025770 133727 025766 000200 10$: BITB    10$,#BIT7 ;CHECK TERR BIT
6158 025776 001406          BEQ    15$         ; BR IF TERR NOT SET
6159 026000          GEDF    EM98,ERR12 ;REPORT ERROR: 'TERR NOT CLEAR'
6160          ;          'DEVICE FATAL' ERROR # 52
6161 026000 104455          .WORD
6162 026002 000064          .WORD    C$ERDF
6163 026004 014357          .WORD    52
6164 026006 017722          .WORD    EM98
6165 026010          .WORD    ERR12
6166 026010 104410          ESCAPE TST          ;AND EXIT TEST          TRAP    C$ESCAPE
6167 026012 000204          .WORD    L10050-.
6168
-----
6169          ; NOW READ 1ST DATA CHARACTER (AND CHECK TERR=1,TXU=1)
-----
6170
6171 026014 004537 007256 15$: JSR    R5,SERIAL    ;READ DATA CHARACTER (000) VIA TSO
6172 026020 000010          8.      ; 8 BIT CHAR/CLOCK TICKS
6173 026022 000000          000     ; EXPECTED BIT SEQUENCE (00000000)
6174 026024 103003          BCC    .+8.        ;BR IF NO ERROR
6175 026026          ERROR    ;REPORT STACKED ERROR
6176 026026 104460          ESCAPE TST          ;AND EXIT TEST          TRAP    C$ERROR
6177 026030          .WORD
6178 026030 104410          TRAP    C$ESCAPE
6179 026032 000164          .WORD    L10050-.
6180
6181 026034 004537 011614    JSR    R5,STEPLU    ;GENERATE 1 TICK TO UNDERRUN TX
6182 026040 000001          1
    
```

CVDPCA.P11

10-DEC-80 09:14

TEST 17 -- BCP TX UNDERRUN W/RESET TERMINATION

```

6183
6184 026042 004537 005432      JSR      R5,CKUSTS      ;CHECK USYRT STATUS
6185 026046 000116      TBMT!TSO!TXACT!TXU    ; FOR TBMT AND TX UNDERRUN
6186 026050 103003      BCC      .+8.          ;BR IF NO ERROR
6187 026052      ERROR              ;REPORT STACKED ERROR
6188 026052 104460      ESCAPE  TST          ;SKIP TO END OF TEST          TRAP  C$ERROR
6189 026054      .WORD              ;
6190 026054 104410      .WORD              ;
6191 026056 000140      .WORD              ;
6192
6193 026060 004537 003610      JSR      R5,READI      ;GET TX ERROR BIT (TO SEE IF SET)
6194 026064 120403      TDSRH          ; TERR BIT IS IN TDSRH
6195 026066 000000      000          ; TDSRH STORED HERE
6196 026070 133727 026066 000200 20$:  BITB      20$,#BIT7    ;CHECK TERR BIT
6197 026076 001006      BNE      25$          ; BR IF TERR IS SET
6198 026100      GEDF      EM99,ERR12 ;REPORT ERROR: 'TERR NOT SET'
6199      ; 'DEVICE FATAL' ERROR # 53
6200 026100 104455      .WORD              ;
6201 026102 000065      .WORD              ;
6202 026104 014404      .WORD              ;
6203 026106 017722      .WORD              ;
6204 026110      ESCAPE  TST          ;AND EXIT TEST          TRAP  C$ERDF
6205 026110 104410      .WORD              ;
6206 026112 000104      .WORD              ;
6207
6208 -----
6209 25$:  JSR      R5,WRITEI    ;CLEAR IDLE BIT
6210 026114 004537 003734      PCSARH          ;DDCMP FORMAT, NO ERROR CHECKING
6211 026120 120405      BIT6!BIT2!BIT1!BIT0
6212 026122 000107
6213
6214 026124 004537 007116      JSR      R5,CHKTSO    ;CHECK 1ST BIT OF EXPECTED 'MARK'
6215 026130 000001      1              ; CHARACTER (BIT SHOULD=1)
6216 026132 103003      BCC      .+8.          ;BR IF NO ERROR
6217 026134      ERROR              ;REPORT STACKED ERROR
6218 026134 104460      ESCAPE  TST          ;AND EXIT TEST          TRAP  C$ERROR
6219 026136      .WORD              ;
6220 026136 104410      .WORD              ;
6221 026140 000056      .WORD              ;
6222
6223 026142 004537 007256      JSR      R5,SERIAL    ;READ MARK CHARACTER VIA TSO
6224 026146 000007      7              ; 7 BIT CHAR/CLOCK TICKS
6225 026150 000177      177            ; EXPECTED BIT SEQUENCE (1111111)
6226 026152 103003      BCC      .+8.          ;BR IF NO ERROR
6227 026154      ERROR              ;REPORT STACKED ERROR
6228 026154 104460      ESCAPE  TST          ;AND EXIT TEST          TRAP  C$ERROR
6229 026156      .WORD              ;
6230 026156 104410      .WORD              ;
6231 026160 000036      .WORD              ;
6232
6233 -----
6234 25$:  JSR      R5,SERIAL    ;READ SYNCH CHARACTER VIA TSO
6235      8.              ; 8 BIT CHAR/CLOCK TICKS
6236 026162 004537 007256      151            ; EXPECTED BIT SEQUENCE (00000000)
6237 026166 000010
6238 026170 000151

```

CVDHCA.P11 10-DEC-80 09:14

TEST 17 -- BCP TX UNDERRUN W/RESET TERMINATION

6239	026172	103003	BCC	+.8.	:BR IF NO ERROR		
6240	026174		ERROR		:REPORT STACKED ERROR		
6241	026174	104460				TRAP	C\$ERROR
6242	026176		ESCAPE	TST	:AND EXIT TEST		
6243	026176	104410				TRAP	C\$ESCAPE
6244	026200	000016				.WORD	L10050-
6245							
6246	026202	004537	JSR	R5,RSTCHK	:RESET USYRT/VERIFY SAME		
6247	026206	103003	BCC	+.8.	:BR IF NO ERROR		
6248	026210		ERROR		:REPORT STACKED ERROR		
6249	026210	104460				TRAP	C\$ERROR
6250	026212		ESCAPE	TST	:SKIP TO END OF TEST		
6251	026212	104410				TRAP	C\$ESCAPE
6252	026214	000002				.WORD	L10050-
6253	026216						
6254	026216		ENDTST				
6255	026216	104401				L10050:	TRAP C\$ETST

CVDMCA.P11 10-DEC-80 09:14

TEST 18 -- BCP TX DISABLE TEST

.SBTTL TEST 18 -- BCP TX DISABLE TEST

6256
6257
6258
6259
6260
6261
6262
6263
6264
6265
6266
6267
6268
6269
6270
6271
6272
6273
6274
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
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311

026220
026220 004737 005420
026224 004537 007400
026230 043626
026232 000000
026234 103003
026236 104460
026240 104410
026242 000162
026244 004537 010010
026250 000001
026252 000007
026254 004537 010010
026260 000000
026262 000000
026264 004537 007676
026270 000125
026272 000010
026274 103003
026276 104460
026300 104410
026302 000122
026304 004537 007676
026310 000252
026312 000000
026314 103003
026316 104460

```

:*****
:*
:* TEST 18 -- BCP TX DISABLE TEST
:*
:* THE USYRT IS INITIALIZED FOR BCP AND A MESSAGE IS STARTED. ONCE THE
:* SECOND DATA CHARACTER IS LOADED INTO TXDB TXE IS DROPPED. TXSO IS
:* WATCHED TO ASSURE THAT THE CHARACTER BEING TRANSMITTED IS COMPLETED.
:* WHEN IT IS, THE USYRT SHOULD DROP TXA AND STOP TRANSMITTING -- THE
:* LAST CHARACTER LOADED INTO TXDB SHOULD BE LOST.
:*
:* CHARACTERS LOADED: 125 252
:* CHARACTERS TRANSMITTED: 125
:*****

```

```

: BGNTST
:
: T18::
:
: JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP
:
: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
: DDCMP!NOCHK!SYNCH ;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 CSERROR
:
: JSR R5,TXCTRL ;OUTPUT 1ST SYNC CHARACTER
: TSOM ;AND KNOCK DOWN TBMT
: 7.
:
: JSR R5,TXCTRL ;CLEAR TSOM (GET READY TO SEND DATA)
: 000
: 0
:
: JSR R5,TXCHAR ;LOAD 125, TX 2ND SYNCH
: 125
: 8.
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
:
: ESCAPE TST ;SKIP TO END OF TEST TRAP CSERROR
:
: JSR R5,TXCHAR ;LOAD 252
: 252
: 0
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
:
: TRAP CSERROR

```

CVDPCA.P11 10-DEC-80 09:14

TEST 18 -- BCP TX DISABLE TEST

6312	026320			ESCAPE TST	;SKIP TO END OF TEST		
6313	026320	104410				TRAP	C\$ESCAPE
6314	026322	000102				.WORD	L10051-
6315							
6316	026324	004537	003734	JSR R5,WRITEI	;DROP TRANSMIT ENABLE (TXE)		
6317	026330	120000		VIAORB			
6318	026332	000102		RXEN!TTLOOP			
6319							
6320	026334	004537	011614	JSR R5,STEPLU	;CLOCK IN 125		
6321	026340	000007		7.	;***		
6322							
6323	026342	004537	011364	JSR R5,RCV1ST	;CLOCK AND RCV 125		
6324	026346	000000		0			
6325	026350	103003		BCC .+8.	;BR IF NO ERROR		
6326	026352			ERROR	;REPORT STACKED ERROR		
6327	026352	104460				TRAP	C\$ERROR
6328	026354			ESCAPE TST	;SKIP TO END OF TEST		
6329	026354	104410				TRAP	C\$ESCAPE
6330	026356	000046				.WORD	L10051-
6331							
6332	026360	004537	010110	JSR R5,RXCHAR	;READ AND CHECK FOR 125		
6333	026364	000125		125			
6334	026366	000000		0			
6335	026370	000010		8.	;AND CLOCK IN NEXT CHAR.		
6336	026372	103003		BCC .+8.	;BR IF NO ERROR		
6337	026374			ERROR	;REPORT STACKED ERROR		
6338	026374	104460				TRAP	C\$ERROR
6339	026376			ESCAPE TST	;SKIP TO END OF TEST		
6340	026376	104410				TRAP	C\$ESCAPE
6341	026400	000024				.WORD	L10051-
6342							
6343	026402	004537	010110	JSR R5,RXCHAR	;READ AND CHECK FOR 377		
6344	026406	000377		377	;** IF 252 IS READ, THEN DROPPING		
6345	026410	000000		0	;** TXEN DIDN'T WORK !!!		
6346	026412	000010		8.			
6347	026414	103003		BCC .+8.	;BR IF NO ERROR ****		
6348	026416			ERROR	;REPORT STACKED ERROR ****		
6349	026416	104460				TRAP	C\$ERROR
6350	026420			ESCAPE TST	;SKIP TO END OF TEST ****		
6351	026420	104410				TRAP	C\$ESCAPE
6352	026422	000002				.WORD	L10051-
6353	026424						
6354	026424						
6355	026424	104401				L10051:	TRAP C\$ETST

ENDTST

CVDMCA.P11 10-DEC-80 09:14

TEST 19 -- FIFO STACKING CHARACTERS TEST

.SBTTL TEST 19 -- FIFO STACKING CHARACTERS TEST

```

:*****
:*
:* TEST 19 -- FIFO STACKING CHARACTERS TEST
:* THE USYRT IS SETUP FOR BCP MODE WITH NO ERROR DETECTION.
:* THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS
:* TO FILL THE 8 CHARACTER RECEIVER FIFO WITH THE CHARACTERS:
:* 1/2(SYNCH),000,377,125,252,347,030,303,1/2(074).
:* THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. NOTE
:* THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE CHARACTERS SINCE THEY
:* ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
:* RDA = 0).
:* ALSO NOTE THAT DUE TO FIFO TIMING, TWO 'HALF CHARACTERS' ARE LOADED
:* INTO THE FIFO (THE 1ST AND LAST CHARACTERS).

```

```

6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377 026426
6378 026426 004737 005420
6379
6380 026432 004537 007400
6381 026436 043626
6382 026440 000000
6383 026442 103003
6384 026444
6385 026444 104460
6386 026446
6387 026446 104410
6388 026450 000216
6389
6390 026452 004537 010010
6391 026456 000001
6392 026460 000007
6393 026462 004537 010010
6394 026466 000000
6395 026470 000000
6396
6397
6398
6399 026472 012702 026670
6400 026476 112237 026506
6401
6402 026502 004537 007676
6403 026506 000000
6404 026510 100010
6405 026512 103003
6406 026514
6407 026514 104460
6408 026516
6409 026516 104410
6410 026520 000146
6411

```

```

:*****
:
: BGNTST
:
: JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP T19::
:
: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
: DDCMP!NOCHK!SYNCH ;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 CSERROR
:
: JSR R5,TXCTRL ;OUTPUT 1ST SYNC CHARACTERS
: TSOM ;
: 7.
: JSR R5,TXCTRL ;CLEAR TSOM
: 000
: 0
:
:-----
: FILL THE FIFO WITH CHARACTERS (DATA1 - DATA8)
:-----
5$: MOV #TXTBL3,R2 ;SET UP TABLE POINTER
: MOV (R2)+,10$ ;SETUP TRANSMIT CHARACTER
:
10$: JSR R5,TXCHAR ;TRANSMIT A CHARACTER
: 000 ;** HOLE FOR NEXT TX CHARACTER
: NCTBMT*256.!8. ;NO CHECK OF INITIAL TBMT=0
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
:
: ESCAPE TST ;SKIP TO END OF TEST TRAP CSERROR
:
: TRAP CSERROR
: .WORD L10052-.

```

CVDMA.P11 10-DEC-80 09:14

TEST 19 -- FIFO STACKING CHARACTERS TEST

6412	026522	022702	026702					
6413	026526	001363		CMP	#TXEND3,R2			
6414				BNE	5\$			
6415	026530	004537	011614	JSR	R5,STEPLU		:ADD A FEW TX TICKS TO COMPLETELY	
6416	026534	000005		5			: FILL UP FIFO.	
6417				-----				
6418				: THE FIFO IS NOW FULL OF CHARACTERS.				
6419				: NOW READ/VERIFY THEM.				
6420				-----				
6421	026536	012702	026670	MOV	#TXTBL3,R2		:SET UP TABLE POINTER	
6422	026542	112237	026552	15\$:	MOVB	(R2)+,20\$:SETUP EXPECTED CHARACTER	
6423								
6424	026546	004537	010110	JSR	R5,RXCHAR		:READ & CHK CHARACTER	
6425	026552	000000		20\$:	000		:** HOLE FOR EXPECTED RECEIVE CHAR.	
6426	026554	000000		0				
6427	026556	100000		NOCRDA			:NO INITIAL CHECK OF RDA=0	
6428	026560	103003		BCC	+.8.		:BR IF NO ERROR	
6429	026562			ERROR			:REPORT STACKED ERROR	
6430	026562	104460						TRAP CSERROR
6431	026564			ESCAPE	TST		:SKIP TO END OF TEST	
6432	026564	104410						TRAP CSERROR
6433	026566	000100						.WORD L10052-
6434								
6435	026570	022702	026700	CMP	#TXEND3-2,R2			
6436	026574	001362		BNE	15\$			
6437								
6438	026576	004537	011614	JSR	R5,STEPLU		:CLOCK IN LAST FEW CHARACTERS OFF	
6439	026602	000015		15			: OF FIFO.	
6440								
6441	026604	004537	010110	JSR	R5,RXCHAR		:READ & CHK CHARACTER	
6442	026610	000303		303				
6443	026612	000000		0				
6444	026614	100000		NOCRDA			:NO INITIAL CHECK OF RDA=0	
6445	026616	103003		BCC	+.8.		:BR IF NO ERROR	
6446	026620			ERROR			:REPORT STACKED ERROR	
6447	026620	104460						TRAP CSERROR
6448	026622			ESCAPE	TST		:SKIP TO END OF TEST	
6449	026622	104410						TRAP CSERROR
6450	026624	000042						.WORD L10052-
6451								
6452	026626	004537	010110	JSR	R5,RXCHAR		:READ & CHK CHARACTER	
6453	026632	000074		074				
6454	026634	000000		0				
6455	026636	100000		NOCRDA			:NO INITIAL CHECK OF RDA=0	
6456	026640	103003		BCC	+.8.		:BR IF NO ERROR	
6457	026642			ERROR			:REPORT STACKED ERROR	
6458	026642	104460						TRAP CSERROR
6459	026644			ESCAPE	TST		:SKIP TO END OF TEST	
6460	026644	104410						TRAP CSERROR
6461	026646	000020						.WORD L10052-
6462								
6463	026650	004537	011532	JSR	R5,ENDTRN		:SHUT DOWN TRANSMITTER, RECEIVER	
6464	026654	000010		8.				
6465	026656	103003		BCC	+.8.		:BR IF NO ERROR	
6466	026660			ERROR			:REPORT STACKED ERROR	
6467	026660	104460						TRAP CSERROR

CVDMCA.P11 10-DEC-80 09:14

TEST 19 -- FIFO STACKING CHARACTERS TEST

ESCAPE TST ;SKIP TO END OF TEST

TRAP C\$ESCAPE
.WORD L10052-

L10052: TRAP C\$ETST

6468	026662	
6469	026662	104410
6470	026664	000002
6471	026666	
6472	026666	
6473	026666	104401
6474		
6475	026670	226
6476	026671	226
6477	026672	000
6478	026673	377
6479	026674	125
6480	026675	252
6481	026676	347
6482	026677	030
6483	026700	303
6484	026701	074
6485	026702	000
6486		026704
6487		

ENDTST

```

-----
TXBL3: .BYTE 226 ;SYNCH
        .BYTE 226 ;SYNCH
        .BYTE 000
        .BYTE 377
        .BYTE 125
        .BYTE 252
        .BYTE 347
        .BYTE 030
        .BYTE 303
        .BYTE 074
TXEND3: .BYTE 000
        .EVEN
-----

```


CVDMCA.P11 10-DEC-80 09:14

TEST 20 -- BCP CHARACTER LENGTH TEST

.SBTTL TEST 20 -- BCP CHARACTER LENGTH TEST

```

+*****
*
* TEST 20 -- BCP CHARACTER LENGTH TEST
* THE USYRT IS INITIALIZED FOR BCP WITH NO ERROR CHECKING. TXC IS MANUALLY
* CONTROLLED UNTIL TWO SYNC CHARACTERS HAVE BEEN TRANSMITTED. THEN 3
* SUBTESTS FOLLOW, EACH ONE USING A DIFFERENT TRANSMIT CHARACTER LENGTH
* STARTING AT FIVE (5) AND ENDING WITH SEVEN (7).
*
* TEST PATTERN: 111 222 333 044 155 266 377
*
-*****

```

```

6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504 026704
6505 026704 004737 005420
6506
6507
6508
6509 026710
6510 026710
6511 026710 104402
6512 026712 004537 007400
6513 026716 043626
6514 026720 000245
6515 026722 103003
6516 026724
6517 026724 104460
6518 026726
6519 026726 104410
6520 026730 000126
6521
6522 026732 004537 010010
6523 026736 000001
6524 026740 000004
6525 026742 004537 010010
6526 026746 000000
6527 026750 000000
6528
6529 026752 012703 027402
6530 026756 112337 026766
6531
6532 026762 004537 007676
6533 026766 000000
6534 026770 000005
6535 026772 103003
6536 026774
6537 026774 104460
6538 026776
6539 026776 104410
6540 027000 000056
6541
6542 027002 022703 027413
6543 027006 001363

```

```

: BGNTST
: JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP T20::
-----
: SUBROUTINE # 1: 5 BIT CHARACTERS
-----
: BGNSUB
: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE TRAP CSBSUB
DDCMP!NOCHK!SYNCH ;SET CHAR MODE,NO ERROR CHECKING,S/AR=226
BIT7!BIT5!BIT2!BIT0 ;TXCL=RXCL=5 BITS
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE SUB ;SKIP TO END OF TEST TRAP CSERROR
; TRAP C$ESCAPE
; .WORD L10054-.
: JSR R5,TXCTRL ;LOAD 2ND SYNCH,TX 1ST SYNCH
TSOM
4.
: JSR R5,TXCTRL ;CLEAR TSOM
000
0
-----
10$: MOV #T24TBL,R3 ;SET UP DATA TABLE POINTER
MOVB (R3)+,1$ ;INSTALL NEXT TX CHARACTER
1$: JSR R5,TXCHAR ;TRANSMIT CHARACTER ( ==> RX/FIFO )
000 ;** HOLE FOR NEXT CHARACTER **
5.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE SUB ;SKIP TO END OF TEST TRAP CSERROR
; TRAP C$ESCAPE
; .WORD L10054-.
CMP #T24TBL+9.,R3 ;ALL CHARACTERS TRANSMITTED ?
BNE 10$ ; IF NOT, TX ANOTHER ONE

```

CVDMCA.P11 10-DEC-80 09:14

TEST 20 -- BCP CHARACTER LENGTH TEST

```

=====
6544      ;
6545 027010 012703 027402      :
6546 027014 112337 027032      40$: MOV #T24TBL,R3      ;SET UP DATA TABLE POINTER
6547 027020 142737 000340 027032  MOVB (R3)+,4$      ;INSTALL NEXT EXPECTED RX CHARACTER
6548      BICB #340,4$      ;MASK OUT UNTRANSMITTED BITS
6549 027026 004537 010110      JSR R5,RXCHAR      ;READ/CHECK NEXT CHARACTER
6550 027032 000000      4$: 000      ;** HOLE FOR NEXT EXPECTED CHARACTER
6551 027034 000000      0
6552 027036 100000      NOCRDA      ;NO INITIAL CHECK OF RDA=0
6553 027040 103003      BCC .+8.      ;BR IF NO ERROR
6554 027042      ERROR      ;REPORT STACKED ERROR
6555 027042 104460      ESCAPE SUB      ;SKIP TO END OF TEST          TRAP C$ERROR
6556 027044      .WORD      L10054-
6557 027044 104410      TRAP C$ESCAPE
6558 027046 000010      .WORD L10054-
6559
6560 027050 022703 027411      CMP #T24TBL+7,R3      ;ALL CHARACTERS CHECKED ?
6561 027054 001357      BNE 40$      ; IF NOT, CHECK ANOTHER ONE
6562
6563      ENDSUB
6564 027056
6565 027056 104403      L10054: TRAP C$ESUB
6566
6567      ;-----
6568      ; SUBROUTINE # 2: 6 BIT CHARACTERS
6569      ;-----
6570      ; BGNSUB
6571 027060 104402      T20.2: TRAP C$BSUB
6572 027062 004537 007400      JSR R5,INITRN      ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
6573 027066 043626      DDCMP!NOCHK!SYNCH      ;SET CHAR MODE,NO ERROR CHECKING,S/AR=226
6574 027070 000306      BIT7!BIT6!BIT2!BIT1      ;TXCL=RXCL=6 BITS
6575 027072 103003      BCC .+8.      ;BR IF NO ERROR
6576 027074      ERROR      ;REPORT STACKED ERROR
6577 027074 104460      ESCAPE SUB      ;SKIP TO END OF TEST          TRAP C$ERROR
6578 027076      .WORD      L10055-
6579 027076 104410      TRAP C$ESCAPE
6580 027100 000126      .WORD L10055-
6581
6582 027102 004537 010010      JSR R5,TXCTRL      ;LOAD 2ND SYNCH,TX 1ST SYNCH
6583 027106 000001      TSOM
6584 027110 000005      5.
6585 027112 004537 010010      JSR R5,TXCTRL      ;CLEAR TSOM
6586 027116 000000      000
6587 027120 000000      0
6588
6589      ;-----
6590 027122 012703 027402      :
6591 027126 112337 027136      20$: MOV #T24TBL,R3      ;SET UP DATA TABLE POINTER
6592      MOVB (R3)+,2$      ;INSTALL NEXT TX CHARACTER
6593 027132 004537 007676      JSR R5,TXCHAR      ;TRANSMIT CHARACTER ( ==> RX/FIFO )
6594 027136 000000      2$: 000      ;** HOLE FOR NEXT CHARACTER **
6595 027140 000006      6.
6596 027142 103003      BCC .+8.      ;BR IF NO ERROR
6597 027144      ERROR      ;REPORT STACKED ERROR
6598 027144 104460      ESCAPE SUB      ;SKIP TO END OF TEST          TRAP C$ERROR
6599 027146 104410      .WORD      L10055-

```


CVDMCA.P11 10-DEC-80 09:14

TEST 20 -- BCP CHARACTER LENGTH TEST

```

6656 027314          ERROR          ;REPORT STACKED ERROR
6657 027314 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  C$ERROR
6658 027316          ;
6659 027316 104410          ;
6660 027320 000056          ;
6661          ;
6662 027322 022703 027413    CMP      #T24TBL+9.,R3    ;ALL CHARACTERS TRANSMITTED ?
6663 027326 001363          BNE      30$             ; IF NOT, TX ANOTHER ONE
6664          ;-----;
6665 027330 012703 027402    MOV      #T24TBL,R3      ;SET UP DATA TABLE POINTER
6666 027334 112337 027352    60$:    MOVB   (R3)+,6$       ;INSTALL NEXT EXPECTED RX CHARACTER
6667 027340 142737 000200 027352    BICB   #200,6$          ;MASK OUT UNTRANSMITTED BITS
6668          ;
6669 027346 004537 010110    6$:    JSR    R5,RXCHAR      ;READ/CHECK NEXT CHARACTER
6670 027352 000000          000          ;** HOLE FOR NEXT EXPECTED CHARACTER
6671 027354 000000          0
6672 027356 100000          NOCRDA          ;NO INITIAL CHECK OF RDA=0
6673 027360 103003          BCC     .+8.        ;BR IF NO ERROR
6674 027362          ERROR          ;REPORT STACKED ERROR
6675 027362 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  C$ERROR
6676 027364          ;
6677 027364 104410          ;
6678 027366 000010          ;
6679          ;
6680 027370 022703 027411    CMP      #T24TBL+7,R3    ;ALL CHARACTERS CHECKED ?
6681 027374 001357          BNE      60$         ; IF NOT, CHECK ANOTHER ONE
6682          ;
6683          ENDSUB
6684 027376          ;
6685 027376 104403          ;
6686 027400          ;
6687 027400          ;
6688 027400 104401          ;
6689          ;
6690 027402          111          ;
6691 027403          222          ;
6692 027404          333          ;
6693 027405          044          ;
6694 027406          155          ;
6695 027407          266          ;
6696 027410          377          ;
6697 027411          000          ;FILLER 1
6698 027412          000          ;FILLER 2
6699 027413          000          ;FILLER 3
6700          ;
6701          ;

```

CVDMCA.P11 10-DEC-80 09:14

TEST 21 -- BOP TX TABORT/(IDLE = 0) TEST

.SBTTL TEST 21 -- BOP TX TABORT/(IDLE = 0) TEST

```

:++*****
:
:   TEST 21 -- BOP TX TABORT/(IDLE = 0) TEST
:
:   THE USYRT IS INITIALIZED FOR 'BIT-ORIENTED PROTOCOL' (BOP) WITH IDLE
:   SET TO ZERO. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
:   OBSERVING TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
:   NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
:   LOADED INTO TXDB DRIVING TXBE LOW. THE TRANSMITTER IS CLOCKED THROUGH
:   ONE CHARACTER. WHEN TXBE GOES HIGH AGAIN, TABORT IS ASSERTED CAUSING
:   ABORT TO BE TRANSMITTED. ALL CHARACTERS ARE CHECKED AT TXSO.
:   THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY TWO
:   FLAGS, ONE ZERO CHARACTER, AND ONE ABORT CHARACTER IS SENT (INTO THE BIT
:   BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
:   THE SEQUENCE.

```

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
6740
6741
6742
6743
6744
6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757

```

027414
027414 004737 005420
027420 004537 007400
027424 003400
027426 040000
027430 103003
027432
027432 104460
027434
027434 104410
027436 000150
027440 004537 007116
027444 000000
027446 103003
027450
027450 104460
027452
027452 104410
027454 000132
027456 004537 007256
027462 000007
027464 000176
027466 103003
027470
027470 104460
027472
027472 104410
027474 000112
027476 004537 010010
027502 000000
027504 000000

```

```

BGNTST
T21::
JSR PC,INIDMV ;INIT VIA
JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
NOCHK ;BOP MODE, NO ERROR CHECK
NORXEN ;NO RECEIVER ENABLE,USE 8 BIT CHARS
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP CSERROR
;WORD CSESCAPE
L10057-.

JSR R5,CHKTSO ;CHECK 1ST BIT OF EXPECTED 'FLAG'
0 ; CHARACTER (SHOULD BE 0)
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;AND EXIT TEST TRAP CSERROR
;WORD CSESCAPE
L10057-.

JSR R5,SERIAL ;READ REMAINING 7 BITS OF 'FLAG' CHARACTER
7. ; (OFF OF TSO BIT)
176 ; EXPECTED BIT SEQUENCE (1111110)
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;AND EXIT TEST TRAP CSERROR
;WORD CSESCAPE
L10057-.

JSR R5,TXCTRL ;CLEAR TSOM
000
0

```

CVDMCA.P11 10-DEC-80 09:14

TEST 21 -- BOP TX TABORT/(IDLE = 0) TEST

6758								
6759	027506	004537	003734	JSR	R5,WRITEI	;LOAD 000 CHARACTER		
6760	027512	120402		TDSRL				
6761	027514	000000		000				
6762								
6763	027516	004537	007256	JSR	R5,SERIAL	;READ FLAG CHARACTER VIA TSO		
6764	027522	000010		8.		; 8 BIT CHAR/CLOCK TICKS		
6765	027524	000176		176		; EXPECTED BIT SEQUENCE (01111110)		
6766	027526	103003		BCC	+.8.	;BR IF NO ERROR		
6767	027530			ERROR		;REPORT STACKED ERROR		
6768	027530	104460					TRAP	C\$ERROR
6769	027532			ESCAPE	TST	;AND EXIT TEST		
6770	027532	104410					TRAP	C\$ESCAPE
6771	027534	000052					.WORD	L10057-.
6772								
6773	027536	004537	010010	JSR	R5,TXCTRL	;SET TXABT BIT		
6774	027542	000004		TAB				
6775	027544	000000		0				
6776								
6777	027546	004537	007256	JSR	R5,SERIAL	;READ 000 CHARACTER VIA TSO		
6778	027552	000010		8.		; 8 BIT CHAR/CLOCK TICKS		
6779	027554	000000		000		; EXPECTED BIT SEQUENCE (00000000)		
6780	027556	103003		BCC	+.8.	;BR IF NO ERROR		
6781	027560			ERROR		;REPORT STACKED ERROR		
6782	027560	104460					TRAP	C\$ERROR
6783	027562			ESCAPE	TST	;AND EXIT TEST		
6784	027562	104410					TRAP	C\$ESCAPE
6785	027564	000022					.WORD	L10057-.
6786								
6787	027566	004537	007256	JSR	R5,SERIAL	;READ ABORT CHARACTER VIA TSO		
6788	027572	000007		7		; 7 BIT CHAR/CLOCK TICKS		
6789	027574	000177		177		; EXPECTED BIT SEQUENCE (11111111)		
6790	027576	103003		BCC	+.8.	;BR IF NO ERROR		
6791	027600			ERROR		;REPORT STACKED ERROR		
6792	027600	104460					TRAP	C\$ERROR
6793	027602			ESCAPE	TST	;AND EXIT TEST		
6794	027602	104410					TRAP	C\$ESCAPE
6795	027604	000002					.WORD	L10057-.
6796	027606							
6797	027606			ENDTST				
6798	027606	104401					L10057:	TRAP C\$ETST

CVDMCA.P11 10-DEC-80 09:14

TEST 22 -- BOP TX TABORT/(IDLE = 1) TEST

.SBTTL TEST 22 -- BOP TX TABORT/(IDLE = 1) TEST

6799
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809
6810
6811
6812
6813
6814
6815
6816
6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854

027610
027610 004737 005420
027614 004537 007400
027620 007400
027622 040000
027624 103003
027626
027626 104460
027630
027630 104410
027632 000150
027634 004537 007116
027640 000000
027642 103003
027644
027644 104460
027646
027646 104410
027650 000132
027652 004537 007256
027656 000007
027660 000176
027662 103003
027664
027664 104460
027666
027666 104410
027670 000112
027672 004537 010010
027676 000000
027700 000000

```
*****
*
* TEST 22 -- BOP TX TABORT/(IDLE = 1) TEST
*
* THE USYRT IS INITIALIZED FOR 'BIT-ORIENTED PROTOCOL' (BOP) WITH IDLE
* SET TO ONE. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
* OBSERVING TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
* NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
* LOADED INTO TXDB DRIVING TXBE LOW. THE TRANSMITTER IS CLOCKED THROUGH
* ONE CHARACTER. WHEN TXBE GOES HIGH AGAIN, TABORT IS ASSERTED CAUSING
* ABORT TO BE TRANSMITTED. ALL CHARACTERS ARE CHECKED AT TXSO.
* THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY TWO
* FLAGS, ONE ZERO CHARACTER, AND ONE FLAG CHARACTER IS SENT (INTO THE BIT
* BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
* THE SEQUENCE.
*****
```

```

:
: BGNTST
:
: JSR PC,INIDMV ;INIT VIA T22::
:
: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
: NOCHK!IDLES ;BOP MODE, NO ERROR CHECK, IDLE=1
: NORXEN ;NO RECEIVER ENABLE,USE 8 BIT CHARS
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
: ESCAPE TST ;SKIP TO END OF TEST TRAP C$ERROR
:
: JSR R5,CHKTSO ;CHECK 1ST BIT OF EXPECTED 'FLAG'
: 0 ; CHARACTER (SHOULD BE 0)
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR TRAP C$ERROR
: ESCAPE TST ;AND EXIT TEST TRAP C$ESCAPE
:
: JSR R5,SERIAL ;READ REMAINING 7 BITS OF 'FLAG' CHARACTER
: 7. ; (OFF OF TSO BIT)
: 176 ; EXPECTED BIT SEQUENCE (1111110)
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR TRAP C$ERROR
: ESCAPE TST ;AND EXIT TEST TRAP C$ESCAPE
:
: JSR R5,TXCTPL ;CLEAR TSOM
: 000
: 0
:
: .WORD L10060-.
```

CVDMCA.P11 10-DEC-80 09:14

TEST 22 -- BOP TX TABORT/(IDLE = 1) TEST

6855								
6856	027702	004537	003734	JSR	R5,WRITEI	:LOAD 000 CHARACTER		
6857	027706	120402		TDSRL				
6858	027710	000000		000				
6859								
6860	027712	004537	007256	JSR	R5,SERIAL	:READ FLAG CHARACTER VIA TSO		
6861	027716	000010		8.		: 8 BIT CHAR/CLOCK TICKS		
6862	027720	000176		176		: EXPECTED BIT SEQUENCE (01111110)		
6863	027722	103003		BCC	+.8.	:BR IF NO ERROR		
6864	027724			ERROR		:REPORT STACKED ERROR		
6865	027724	104460					TRAP	C\$ERROR
6866	027726			ESCAPE	TST	:AND EXIT TEST		
6867	027726	104410					TRAP	C\$ESCAPE
6868	027730	000052					.WORD	L10060-
6869								
6870	027732	004537	010010	JSR	R5,TXCTRL	:SET TXABT BIT		
6871	027736	000004		TAB				
6872	027740	000000		0				
6873								
6874	027742	004537	007256	JSR	R5,SERIAL	:READ 000 CHARACTER VIA TSO		
6875	027746	000010		8.		: 8 BIT CHAR/CLOCK TICKS		
6876	027750	000000		000		: EXPECTED BIT SEQUENCE (00000000)		
6877	027752	103003		BCC	+.8.	:BR IF NO ERROR		
6878	027754			ERROR		:REPORT STACKED ERROR		
6879	027754	104460					TRAP	C\$ERROR
6880	027756			ESCAPE	TST	:AND EXIT TEST		
6881	027756	104410					TRAP	C\$ESCAPE
6882	027760	000022					.WORD	L10060-
6883								
6884	027762	004537	007256	JSR	R5,SERIAL	:READ ABORT (FLAG) CHARACTER VIA TSO		
6885	027766	000010		8.		: 8 BIT CHAR/CLOCK TICKS		
6886	027770	000176		176		: EXPECTED BIT SEQUENCE (01111110)		
6887	027772	103003		BCC	+.8.	:BR IF NO ERROR		
6888	027774			ERROR		:REPORT STACKED ERROR		
6889	027774	104460					TRAP	C\$ERROR
6890	027776			ESCAPE	TST	:AND EXIT TEST		
6891	027776	104410					TRAP	C\$ESCAPE
6892	030000	000002					.WORD	L10060-
6893	030002							
6894	030002			ENDTST				
6895	030002	104401					L10060:	
							TRAP	C\$ETST

CVDMCA.P11 10-DEC-80 09:14

TEST 23 -- BOP TX TXGA (TRANSMIT GO-AHEAD) TEST

.SBTTL TEST 23 -- BOP TX TXGA (TRANSMIT GO-AHEAD) TEST

6896
6897
6898
6899
6900
6901
6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943
6944
6945
6946
6947
6948
6949
6950
6951

030004
030004 004737 005420
030010 004537 007400
030014 003400
030016 040000
030020 103003
030022 104460
030024 104410
030026 000150
030030 004537 007116
030034 000000
030036 103003
030040 104460
030042 104410
030044 000132
030046 004537 007256
030052 000007
030054 000176
030056 103003
030060 104460
030062 104410
030064 000112
030066 004537 010010
030072 000000
030074 000000

```
*****
*
* TEST 23 -- BOP TX TXGA (TRANSMIT GO-AHEAD) TEST
*
* THE USYRT IS INITIALIZED FOR BOP AND TXE IS ASSERTED. TSOM IS ASSERTED
* AND TXA OBSERVED -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
* NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
* LOADED INTO TXDB DRIVING TXBE LOW. WHEN TXBE GOES HIGH AGAIN, TXGA
* IS ASSERTED CAUSING GA TO BE TRANSMITTED. THE SEQUENCE OF EVENTS IS
* CONTINUALLY MONITORED WHILE TXC IS MANUALLY CONTROLLED AND ALL
* CHARACTERS ARE CHECKED AT TXSO. THIS TEST WILL GO NO FURTHER INTO
* THE TRANSMIT SEQUENCE SO THAT ONLY TWO FLAGS, ONE ZERO CHARACTER
* AND ONE GA CHARACTER IS SENT (INTO THE BIT BUCKET).
* ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
* THE SEQUENCE.
*****
```

```

: BGNTST
:
: JSR PC,INIDMV ;INIT VIA T23::
:
: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
: NOCHK ;BOP MODE, NO ERROR CHECK
: NORXEN ;NO RECEIVER ENABLE,USE 8 BIT CHARS
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
: TRAP CSERROR
: ESCAPE TST ;SKIP TO END OF TEST
: TRAP C$ESCAPE
: .WORD L10061-.
:
: JSR R5,CHKTSO ;CHECK 1ST BIT OF EXPECTED 'FLAG'
: 0 ; CHARACTER (SHOULD BE 0)
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
: TRAP CSERROR
: ESCAPE TST ;AND EXIT TEST
: TRAP C$ESCAPE
: .WORD L10061-.
:
: JSR R5,SERIAL ;READ REMAINING 7 BITS OF 'FLAG' CHARACTER
: 7 ; (OFF OF TSO BIT)
: 176 ; EXPECTED BIT SEQUENCE (1111110)
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
: TRAP CSERROR
: ESCAPE TST ;AND EXIT TEST
: TRAP C$ESCAPE
: .WORD L10061-.
:
: JSR R5,TXCTRL ;CLEAR TSOM
: 000
: 0
```

CVDMA.P11 10-DEC-80 09:14

TEST 23 -- BOP TX TXGA (TRANSMIT GO-AHEAD) TEST

```

6952
6953 030076 004537 003734      JSR      R5,WRITEI      ;LOAD 000 CHARACTER
6954 030102 120402              TDSRL
6955 030104 000000              000
6956
6957 030106 004537 007256      JSR      R5,SERIAL      ;READ 'FLAG' CHARACTER VIA TSO
6958 030112 000010              8.          ; 8 BIT CHAR/CLOCK TICKS
6959 030114 000176              176        ; EXPECTED BIT SEQUENCE (01111110)
6960 030116 103003              BCC      .+8.          ;BR IF NO ERROR
6961 030120              ERROR        ;REPORT STACKED ERROR
6962 030120 104460              ESCAPE    TST          ;AND EXIT TEST                TRAP      C$ERROR
6963 030122
6964 030122 104410              ESCAPE    TST          ;AND EXIT TEST                TRAP      C$ESCAPE
6965 030124 000052              .WORD     L10061-.
6966
6967 030126 004537 010010      JSR      R5,TXCTRL      ;SET TX GO-AHEAD AND TEOM
6968 030132 000012              TGA!TEOM
6969 030134 000000              0
6970
6971 030136 004537 007256      JSR      R5,SERIAL      ;READ 000 CHARACTER VIA TSO
6972 030142 000010              8.          ; 8 BIT CHAR/CLOCK TICKS
6973 030144 000000              000        ; EXPECTED BIT SEQUENCE (00000000)
6974 030146 103003              BCC      .+8.          ;BR IF NO ERROR
6975 030150              ERROR        ;REPORT STACKED ERROR
6976 030150 104460              ESCAPE    TST          ;AND EXIT TEST                TRAP      C$ERROR
6977 030152
6978 030152 104410              ESCAPE    TST          ;AND EXIT TEST                TRAP      C$ESCAPE
6979 030154 000022              .WORD     L10061-.
6980
6981 030156 004537 007256      JSR      R5,SERIAL      ;READ GO-AHEAD CHARACTER VIA TSO
6982 030162 000010              8.          ; 8 BIT CHAR/CLOCK TICKS
6983 030164 000177              177        ; EXPECTED BIT SEQUENCE (01111111)
6984 030166 103003              BCC      .+8.          ;BR IF NO ERROR
6985 030170              ERROR        ;REPORT STACKED ERROR
6986 030170 104460              ESCAPE    TST          ;AND EXIT TEST                TRAP      C$ERROR
6987 030172
6988 030172 104410              ESCAPE    TST          ;AND EXIT TEST                TRAP      C$ESCAPE
6989 030174 000002              .WORD     L10061-.
6990 030176
6991 030176
6992 030176 104401              .WORD     L10061:
                                     TRAP      C$ETST

```

ENDTST

CVDMCA.P11 10-DEC-80 09:14

TEST 24 -- BOP TX MESSAGE WITHOUT CRC

.SBTTL TEST 24 -- BOP TX MESSAGE WITHOUT CRC

6993
6994
6995
6996
6997
6998
6999
7000
7001
7002
7003
7004
7005
7006
7007
7008
7009
7010
7011
7012
7013
7014
7015
7016
7017
7018
7019
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031
7032
7033
7034
7035
7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046
7047
7048

030200
030200 004737 005420
030204 004537 007400
030210 003400
030212 000000
030214 103003
030216 104460
030220
030220 104410
030222 000332
030224 004537 010010
030230 000001
030232 000007
030234 004537 010010
030240 000000
030242 000000
030244 004537 007676
030250 000000
030252 000010
030254 103003
030256 104460
030260
030260 104410
030262 000272
030264 004537 007676
030270 000307
030272 000010
030274 103003
030276 104460
030300

```
*****
*
* TEST 24 -- BOP TX MESSAGE WITHOUT CRC
*
* THE USYRT IS INITIALIZED FOR BOP MODE WITH NO ERROR DETECTION. TXC IS THEN
* MANUALLY CONTROLLED UNTIL TWO FLAG CHARACTERS HAVE BEEN TRANSMITTED. THEN
* A 5 CHARACTER MESSAGE IS TRANSMITTED, RECEIVED, CHECKED, AND TERMINATED
* (WITH TEOM). A CHECK IS MADE TO ASCERTAIN THAT NO CRC OR VRC IS GENERATED
* -- FLAG CHARACTERS SHOULD FOLLOW THE DATA.
* (NOTE: NO BIT STUFFING OCCURS IN THIS TEST)
*
* TEST MESSAGE: FLAG FLAG 000 307 125 252 201 FLAG
*
*****
```

```

BGNTST
JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP T24::
JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
NOCHK ;SET BOP MODE,NO CHECK
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 L10062-.

JSR R5,TXCTRL ;LOAD 2ND FLAG, TX 1ST FLAG
TSOM
7.

JSR R5,TXCTRL ;CLEAR TSOM
000
0

JSR R5,TXCHAR ;LOAD 000, TX 2ND FLAG
000
8.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR TRAP C$ERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP C$ESCAPE
;WORD L10062-.

JSR R5,TXCHAR ;LOAD 307, TX 000
307
8.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR TRAP C$ERROR
ESCAPE TST ;SKIP TO END OF TEST
```

CVDMA.P11 10-DEC-80 09:14

TEST 24 -- BOP TX MESSAGE WITHOUT CRC

7049	030300	104410					TRAP	C\$ESCAPE
7050	030302	000252					.WORD	L10062-.
7051								
7052	030304	004537	007676	JSR	R5,TXCHAR	:LOAD 125; TX 307		
7053	030310	000125		125				
7054	030312	000010		8.				
7055	030314	103003		BCC	+.8.	:BR IF NO ERROR		
7056	030316			ERROR		:REPORT STACKED ERROR		
7057	030316	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
7058	030320							
7059	030320	104410					TRAP	C\$ESCAPE
7060	030322	000232					.WORD	L10062-.
7061								
7062	030324	004537	007676	JSR	R5,TXCHAR	:LOAD 252		
7063	030330	000252		252				
7064	030332	000000		0				
7065	030334	103003		BCC	+.8.	:BR IF NO ERROR		
7066	030336			ERROR		:REPORT STACKED ERROR		
7067	030336	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
7068	030340							
7069	030340	104410					TRAP	C\$ESCAPE
7070	030342	000212					.WORD	L10062-.
7071								
7072	030344	004537	011364	JSR	R5,RCV1ST	:CLOCK AND RCV 000		
7073	030350	000000		0				
7074	030352	103003		BCC	+.8.	:BR IF NO ERROR		
7075	030354			ERROR		:REPORT STACKED ERROR		
7076	030354	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
7077	030356							
7078	030356	104410					TRAP	C\$ESCAPE
7079	030360	000174					.WORD	L10062-.
7080								
7081	030362	004537	010110	JSR	R5,RXCHAR	:READ & CHK 000, RCV 307		
7082	030366	000400		RXSOM!000		: & CHECK RSOM=1		
7083	030370	000000		0				
7084	030372	000010		8.				
7085	030374	103003		BCC	+.8.	:BR IF NO ERROR		
7086	030376			ERROR		:REPORT STACKED ERROR		
7087	030376	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
7088	030400							
7089	030400	104410					TRAP	C\$ESCAPE
7090	030402	000152					.WORD	L10062-.
7091								
7092	030404	004537	007676	JSR	R5,TXCHAR	:LOAD 201		
7093	030410	000201		201				
7094	030412	000000		0				
7095	030414	103003		BCC	+.8.	:BR IF NO ERROR		
7096	030416			ERROR		:REPORT STACKED ERROR		
7097	030416	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
7098	030420							
7099	030420	104410					TRAP	C\$ESCAPE
7100	030422	000132					.WORD	L10062-.
7101								
7102	030424	004537	010110	JSR	R5,RXCHAR	:READ & CHK 307, RCV 125		
7103	030430	000307		307				
7104	030432	000000		0				

CVDMCA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

.SBTTL TEST 25 -- BOP RX CHARACTER LENGTH TEST

7158
7159
7160
7161
7162
7163
7164
7165
7166
7167
7168
7169
7170
7171
7172
7173
7174
7175
7176
7177
7178
7179
7180
7181
7182
7183
7184
7185
7186
7187
7188
7189
7190
7191
7192
7193
7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210
7211
7212
7213

030556
030556 004737 005420

030562
030562 104402
030564 004537 007400
030570 004226
030572 000002
030574 103003
030576 104460
030600
030600 104410
030602 000344

030604 004537 010010
030610 000001
030612 000007
030614 004537 010010
030620 000000
030622 000000
030624 004537 007676
030630 000123
030632 000010
030634 103003

```
*****
*
* TEST 25 -- BOP RX CHARACTER LENGTH TEST
*
* THE USYRT IS INITIALIZED FOR BOP WITH CRC-CCITT PRESET TO 1'S. TXC
* IS MANUALLY CONTROLLED UNTIL TWO FLAG CHARACTERS HAVE BEEN
* TRANSMITTED. THEN 6 SUBTESTS FOLLOW, EACH ONE USING A DIFFERENT
* TRANSMIT CHARACTER LENGTH STARTING AT TWO (2) AND ENDING WITH SEVEN
* (7). IN EACH SUBTEST, TWO 8 BIT CHARACTERS WILL BE TRANSMITTED
* BEFORE TXCL IS CHANGED TO THE CHARACTER LENGTH BEING TESTED. THIS
* CORRESPONDS TO NORMAL USAGE WHERE EITHER:
*
* 1 -- A MESSAGE OF CHARACTERS WHICH ARE LESS THEN 8 BITS IS
* SENT AS A STREAM OF 8 BIT CHARACTERS AND THE REMAINING
* BITS ARE SENT AS A CHARACTER OF LESS THEN 8 BITS OR
*
* 2 -- A HEADER OF TWO 8 BIT CHARACTERS IS SENT FOLLOWED BY A
* DATA STREAM OF DATA CHARACTERS WHICH MAY BE LESS THEN 8
* BITS IN LENGTH (I.E. 2, 3, 4, 5, 6, OR 7 BIT
* CHARACTERS).
*
* THE TEST PATTERN IS: 123 321 111 222 333 044 155 266 377
*
*****
```

```
      BGNTST
      JSR    PC,INIDMV      ;INIT DMV-11, ENTER M-LOOP
      -----
      SUBROUTINE # 1: 2 BIT CHARACTERS
      -----
      BGNSUB
      JSR    R5,INITRN      ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
      IDLES!SYNCH          ;SET BOP MODE,CRC-CCITT-1,S/AR=226
      BIT1                ;INITIALLY: TXCL=8 BITS / RXCL=2 BITS
      BCC    .+8.          ;BR IF NO ERROR
      ERROR                ;REPORT STACKED ERROR
      ESCAPE SUB          ;SKIP TO END OF TEST
      JSR    R5,TXCTRL      ;LOAD 2ND FLAG,TX 1ST FLAG
      TSOM
      7.
      JSR    R5,TXCTRL      ;CLEAR TSOM
      000
      0
      JSR    R5,TXCHAR      ;LOAD 123(HEADER1), TX 2ND FLAG
      123
      8.
      BCC    .+8.          ;BR IF NO ERROR
```

CVDMCA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7214 030636          ERROR          ;REPORT STACKED ERROR
7215 030636 104460  ESCAPE SUB      ;SKIP TO END OF TEST          TRAP  CSERROR
7216 030640          ;                                     TRAP  C$ESCAPE
7217 030640 104410  ;                                     .WORD L10064-.
7218 030642 000304
7219
7220 030644 004537 007676 JSR    R5,TXCHAR      ;LOAD 321(HEADER2), TX 123(HEADER1)
7221 030650 000321      321
7222 030652 000010      8.
7223 030654 103003  BCC    .+8.          ;BR IF NO ERROR
7224 030656          ERROR          ;REPORT STACKED ERROR          TRAP  CSERROR
7225 030656 104460  ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  C$ESCAPE
7226 030660          ;                                     .WORD L10064-.
7227 030660 104410
7228 030662 000264
7229
7230 030664 004537 003734 JSR    R5,WRITEI     ;NOW CHANGE TXCL TO 2 BITS TOO
7231 030670 120407      PCR
7232 030672 000102      BIT6!BIT1
7233 030674 004537 007676 JSR    R5,TXCHAR      ;LOAD 111(DATA1), TX 321(HEADER2)
7234 030700 000111      111
7235 030702 000010      8.
7236 030704 103003  BCC    .+8.          ;BR IF NO ERROR
7237 030706          ERROR          ;REPORT STACKED ERROR          TRAP  CSERROR
7238 030706 104460  ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  C$ESCAPE
7239 030710          ;                                     .WORD L10064-.
7240 030710 104410
7241 030712 000234
7242
;=====
7243 030714 012703 033471 10$:  MOV    #T30TBL+1,R3  ;SET UP DATA TABLE POINTER
7244 030720 112337 030730  MOVB   (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7245
7246 030724 004537 007676 1$:  JSR    R5,TXCHAR      ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7247 030730 000000      000          ;** HOLE FOR NEXT CHARACTER **
7248 030732 000002      2.
7249 030734 103003  BCC    .+8.          ;BR IF NO ERROR
7250 030736          ERROR          ;REPORT STACKED ERROR          TRAP  CSERROR
7251 030736 104460  ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  C$ESCAPE
7252 030740          ;                                     .WORD L10064-.
7253 030740 104410
7254 030742 000204
7255
7256 030744 022703 033477      CMP    #T30TBL+7,R3  ;ALL CHARACTERS TRANSMITTED ?
7257 030750 001363      BNE    10$          ; IF NOT, TX ANOTHER ONE
7258
;=====
7259 030752 004537 010010 JSR    R5,TXCTRL     ;SET TEOM
7260 030756 000002      TEOM
7261 030760 000000      0
7262 030762 004537 011614 JSR    R5,STEPLU     ; AND TX DATA7 + SOME FLAGS
7263 030766 000044      36.
7264
7265 030770 004537 010110 JSR    R5,RXCHAR      ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7266 030774 000523      RXSOM!123      ; & CHECK FOR RSOM=1
7267 030776 000000      0
7268 031000 100000      NOCRDA
7269 031002 103003  BCC    .+8.          ;NO INITIAL CHECK OF RDA=0
;BR IF NO ERROR

```

CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7270 031004          ERROR          ;REPORT STACKED ERROR
7271 031004 104460          TRAP      CSERROR
7272 031006          ESCAPE SUB      ;SKIP TO END OF TEST
7273 031006 104410          TRAP      C$ESCAPE
7274 031010 000136          .WORD   L10064-.
7275
7276 031012 004537 010110    JSR      R5,RXCHAR      ;READ/CHECK 321(HEADER2),RCV 001(DATA1)
7277 031016 000321          321
7278 031020 000000          0
7279 031022 100000          NOCRDA      ;NO INITIAL CHECK OF RDA=0
7280 031024 103003          BCC      .+8.          ;BR IF NO ERROR
7281 031026          ERROR          ;REPORT STACKED ERROR
7282 031026 104460          TRAP      CSERROR
7283 031030          ESCAPE SUB      ;SKIP TO END OF TEST
7284 031030 104410          TRAP      C$ESCAPE
7285 031032 000114          .WORD   L10064-.
7286
;-----
7287 031034 012703 033470    MOV      #T30TBL,R3      ;SET UP DATA TABLE POINTER
7288 031040 112337 031056    20$:    MOVB   (R3)+,2$        ;INSTALL NEXT EXPECTED RX CHARACTER
7289 031044 142737 000374 031056    BICB   #374,2$          ;MASK OUT UNTRANSMITTED BITS
7290
7291 031052 004537 010110    JSR      R5,RXCHAR      ;READ/CHECK NEXT CHARACTER
7292 031056 000000          2$:    000              ;** HOLE FOR NEXT EXPECTED CHARACTER
7293 031060 000000          0
7294 031062 100000          NOCRDA      ;NO INITIAL CHECK OF RDA=0
7295 031064 103003          BCC      .+8.          ;BR IF NO ERROR
7296 031066          ERROR          ;REPORT STACKED ERROR
7297 031066 104460          TRAP      CSERROR
7298 031070          ESCAPE SUB      ;SKIP TO END OF TEST
7299 031070 104410          TRAP      C$ESCAPE
7300 031072 000054          .WORD   L10064-.
7301
7302 031074 022703 033475    CMP      #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
7303 031100 001357          BNE      20$            ; IF NOT, CHECK ANOTHER ONE
7304
;-----
7305 031102 004537 010110    JSR      R5,RXCHAR      ;READ/CHK DATA6(002), RCV DATA7
7306 031106 000002          002
7307 031110 000000          0
7308 031112 120000          NCRACT!NOCRDA      ;NO INITIAL CHECK OF RDA=0
7309 031114 103003          BCC      .+8.          ;DON'T CHECK FOR FINAL RXACT=1
7310 031116          ERROR          ;BR IF NO ERROR
7311 031116 104460          ;REPORT STACKED ERROR
7312 031120          TRAP      CSERROR
7313 031120 104410          ESCAPE SUB      ;SKIP TO END OF TEST
7314 031122 000024          TRAP      C$ESCAPE
7315          .WORD   L10064-.
7316 031124 004537 010110    JSR      R5,RXCHAR      ;READ/CHK DATA7(003), RCV FIRST FLAG
7317 031130 001003          RXEOM!003          ;& CHECK RERR BIT=0 (GOOD CRC) & REOM=1
7318 031132 000001          RERCHK            ;NO INITIAL CHECK OF RDA=0
7319 031134 160000          NOCRDA!NFCRDA!NCRACT ;DON'T CHECK FOR FINAL RDA=RXACT=1
7320 031136 103003          BCC      .+8.          ;BR IF NO ERROR
7321 031140          ERROR          ;REPORT STACKED ERROR
7322 031140 104460          TRAP      CSERROR
7323 031142          ESCAPE SUB      ;SKIP TO END OF TEST
7324 031142 104410          TRAP      C$ESCAPE
7325 031144 000002          .WORD   L10064-.

```


CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7326 031146
7327 031146
7328 031146 104403
7329
7330
7331
7332 031150
7333 031150
7334 031150 104402
7335 031152 004537 007400
7336 031156 000226
7337 031160 000003
7338 031162 103003
7339 031164
7340 031164 104460
7341 031166
7342 031166 104410
7343 031170 000344
7344
7345 031172 004537 010010
7346 031176 000001
7347 031200 000007
7348 031202 004537 010010
7349 031206 000000
7350 031210 000000
7351 031212 004537 007676
7352 031216 000123
7353 031220 000010
7354 031222 103003
7355 031224
7356 031224 104460
7357 031226
7358 031226 104410
7359 031230 000304
7360
7361 031232 004537 007676
7362 031236 000321
7363 031240 000010
7364 031242 103003
7365 031244
7366 031244 104460
7367 031246
7368 031246 104410
7369 031250 000264
7370
7371 031252 004537 003734
7372 031256 120407
7373 031260 000143
7374 031262 004537 007676
7375 031266 000111
7376 031270 000010
7377 031272 103003
7378 031274
7379 031274 104460
7380 031276
7381 031276 104410

```

ENDSUB

L10064: TRAP C\$ESUB

SUBROUTINE # 2: 3 BIT CHARACTERS

BGNSUB

T25.2: TRAP C\$BSUB

```

JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
SYNCH ;SET BOP MODE,CRC-CCITT-1,S/AR=226
BIT1!BIT0 ;INITIALLY: TXCL=8 BITS / RXCL=3 BITS
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR

```

TRAP C\$ERROR

```

ESCAPE SUB ;SKIP TO END OF TEST

```

TRAP C\$ESCAPE
.WORD L10065-

```

JSR R5,TXCTRL ;LOAD 2ND FLAG,TX 1ST FLAG
TSOM
7.
JSR R5,TXCTRL ;CLEAR TSOM
000
0
JSR R5,TXCHAR ;LOAD 123(HEADER1), TX 2ND FLAG
123
8.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR

```

TRAP C\$ERROR

```

ESCAPE SUB ;SKIP TO END OF TEST

```

TRAP C\$ESCAPE
.WORD L10065-

```

JSR R5,TXCHAR ;LOAD 321(HEADER2), TX 123(HEADER1)
321
8.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR

```

TRAP C\$ERROR

```

ESCAPE SUB ;SKIP TO END OF TEST

```

TRAP C\$ESCAPE
.WORD L10065-

```

JSR R5,WRITEI ;NOW CHANGE TXCL TO 3 BITS TOO
PCR
BIT6!BIT5!BIT1!BIT0
JSR R5,TXCHAR ;LOAD 111(DATA1), TX 321(HEADER2)
111
8.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR

```

TRAP C\$ERROR

```

ESCAPE SUB ;SKIP TO END OF TEST

```

TRAP C\$ESCAPE

CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7382 031300 000234                                     .WORD L10065-.
7383
7384 031302 012703 033471
7385 031306 112337 031316
7386
7387 031312 004537 007676
7388 031316 000000
7389 031320 000003
7390 031322 103003
7391 031324
7392 031324 104460
7393 031326
7394 031326 104410
7395 031330 000204
7396
7397 031332 022703 033477
7398 031336 001363
7399
7400 031340 004537 010010
7401 031344 000002
7402 031346 000000
7403 031350 004537 011614
7404 031354 000044
7405
7406 031356 004537 010110
7407 031362 000523
7408 031364 000000
7409 031366 100000
7410 031370 103003
7411 031372
7412 031372 104460
7413 031374
7414 031374 104410
7415 031376 000136
7416
7417 031400 004537 010110
7418 031404 000321
7419 031406 000000
7420 031410 100000
7421 031412 103003
7422 031414
7423 031414 104460
7424 031416
7425 031416 104410
7426 031420 000114
7427
7428 031422 012703 033470
7429 031426 112337 031444
7430 031432 142737 000370 031444
7431
7432 031440 004537 010110
7433 031444 000000
7434 031446 000000
7435 031450 100000
7436 031452 103003
7437 031454

```

```

=====
;
10$:  MOV     #T30TBL+1,R3      ;SET UP DATA TABLE POINTER
      MOVSB  (R3)+,1$         ;INSTALL NEXT TX CHARACTER

1$:   JSR     R5,TXCHAR        ;TRANSMIT CHARACTER ( ==> RX/FIFO )
      000                      ;** HOLE FOR NEXT CHARACTER **
      3.
      BCC    .+8.             ;BR IF NO ERROR
      ERROR                          ;REPORT STACKED ERROR
      ESCAPE SUB                ;SKIP TO END OF TEST
                                     TRAP   CSERROR
                                     .WORD  L10065-.

      CMP    #T30TBL+7,R3     ;ALL CHARACTERS TRANSMITTED ?
      BNE   10$              ; IF NOT, TX ANOTHER ONE
=====
;
      JSR    R5,TXCTRL        ;SET TEOM
      TEOM
      0
      JSR    R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
      36.

      JSR    R5,RXCHAR        ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
      RXSOM!123              ; & CHECK FOR RSOM=1
      0
      NOCRDA                    ;NO INITIAL CHECK OF RDA=0
      BCC    .+8.             ;BR IF NO ERROR
      ERROR                          ;REPORT STACKED ERROR
      ESCAPE SUB                ;SKIP TO END OF TEST
                                     TRAP   CSERROR
                                     .WORD  L10065-.

      JSR    R5,RXCHAR        ;READ/CHECK 321(HEADER2),RCV 001(DATA1)
      321
      0
      NOCRDA                    ;NO INITIAL CHECK OF RDA=0
      BCC    .+8.             ;BR IF NO ERROR
      ERROR                          ;REPORT STACKED ERROR
      ESCAPE SUB                ;SKIP TO END OF TEST
                                     TRAP   CSERROR
                                     .WORD  L10065-.

=====
;
40$:  MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
      MOVSB  (R3)+,4$         ;INSTALL NEXT EXPECTED RX CHARACTER
      BICB   #370,4$         ;MASK OUT UNTRANSMITTED BITS

4$:   JSR    R5,RXCHAR        ;READ/CHECK NEXT CHARACTER
      000                      ;** HOLE FOR NEXT EXPECTED CHARACTER
      0
      NOCRDA                    ;NO INITIAL CHECK OF RDA=0
      BCC    .+8.             ;BR IF NO ERROR
      ERROR                          ;REPORT STACKED ERROR

```

CVDMCA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7438 031454 104460
7439 031456
7440 031456 104410
7441 031460 000054
7442
7443 031462 022703 033475
7444 031466 001357
7445
7446 031470 004537 010110
7447 031474 000006
7448 031476 000000
7449 031500 020000
7450 031502 103003
7451 031504
7452 031504 104460
7453 031506
7454 031506 104410
7455 031510 001756
7456
7457 031512 004537 010110
7458 031516 001007
7459 031520 000001
7460 031522 060000
7461 031524 103003
7462 031526
7463 031526 104460
7464 031530
7465 031530 104410
7466 031532 001734
7467 031534
7468 031534
7469 031534 104403
7470
7471
7472
7473 031536
7474 031536
7475 031536 104402
7476 031540 004537 007400
7477 031544 000226
7478 031546 000004
7479 031550 103003
7480 031552
7481 031552 104460
7482 031554
7483 031554 104410
7484 031556 000344
7485
7486 031560 004537 010010
7487 031564 000001
7488 031566 000007
7489 031570 004537 010010
7490 031574 000000
7491 031576 000000
7492 031600 004537 007676
7493 031604 000123

      ESCAPE SUB                ;SKIP TO END OF TEST                TRAP CSERROR
                                     .WORD CS_ESCAPE L10065-.

      CMP #T30TBL+5,R3           ;ALL CHARACTERS CHECKED ?
      BNE 408                     ; IF NOT, CHECK ANOTHER ONE
;-----
      JSR R5,RXCHAR               ;READ/CHK DATA6(006), RCV DATA7
      006
      0
      NCRACT                      ;DON'T CHECK FOR FINAL RXACT=1
      BCC .+8.                   ;BR IF NO ERROR
      ERROR                       ;REPORT STACKED ERROR                TRAP CSERROR
                                     ESCAPE TST                ;SKIP TO END OF TEST                TRAP CS_ESCAPE
                                     .WORD L10063-.

      JSR R5,RXCHAR               ;READ/CHK DATA7(007), RCV FIRST FLAG
      RXEOM!007                  ;AND CHECK FOR REOM=1
      RERCHK                      ; & CHECK RERR BIT=0 (GOOD CRC)
      NRCRDA!NCRACT              ;DON'T CHECK FOR FINAL RDA=RXACT=1
      BCC .+8.                   ;BR IF NO ERROR
      ERROR                       ;REPORT STACKED ERROR                TRAP CSERROR
                                     ESCAPE TST                ;SKIP TO END OF TEST                TRAP CS_ESCAPE
                                     .WORD L10063-.

      ENDSUB

                                     L10065:
                                     TRAP CS_ESUB

;-----
      SUBROUTINE # 3: 4 BIT CHARACTERS
;-----
      BGNSUB
                                     T25.3:
                                     TRAP CSBSUB

      JSR R5,INITRN              ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
      SYNCH                       ;SET BOP MODE,CRC-CCITT-1,S/AR=226
      BIT2                         ;INITIALLY: TXCL=8 BITS / RXCL=4 BITS
      BCC .+8.                   ;BR IF NO ERROR
      ERROR                       ;REPORT STACKED ERROR                TRAP CSERROR
                                     ESCAPE SUB                ;SKIP TO END OF TEST                TRAP CS_ESCAPE
                                     .WORD L10066-.

      JSR R5,TXCTRL               ;LOAD 2ND FLAG,TX 1ST FLAG
      TSOM
      7.
      JSR R5,TXCTRL               ;CLEAR TSOM
      000
      0
      JSR R5,TXCHAR               ;LOAD 123(HEADER1), TX 2ND FLAG
      123
    
```

CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7494 031606 000010
7495 031610 103003
7496 031612
7497 031612 104460
7498 031614
7499 031614 104410
7500 031616 000304
7501
7502 031620 004537 007676
7503 031624 000321
7504 031626 000010
7505 031630 103003
7506 031632
7507 031632 104460
7508 031634
7509 031634 104410
7510 031636 000264
7511
7512 031640 004537 003734
7513 031644 120407
7514 031646 000204
7515 031650 004537 007676
7516 031654 000111
7517 031656 000010
7518 031660 103003
7519 031662
7520 031662 104460
7521 031664
7522 031664 104410
7523 031666 000234
7524
7525 031670 012703 033471
7526 031674 112337 031704
7527
7528 031700 004537 007676
7529 031704 000000
7530 031706 000004
7531 031710 103003
7532 031712
7533 031712 104460
7534 031714
7535 031714 104410
7536 031716 000204
7537
7538 031720 022703 033477
7539 031724 001363
7540
7541 031726 004537 010010
7542 031732 000002
7543 031734 000000
7544 031736 004537 011614
7545 031742 000044
7546
7547 031744 004537 010110
7548 031750 000523
7549 031752 000000

```

```

      8.
      BCC      .+8.      ;BR IF NO ERROR
      ERROR    ;REPORT STACKED ERROR
                                TRAP      C$ERROR
      ESCAPE   SUB      ;SKIP TO END OF TEST
                                TRAP      C$ESCAPE
                                .WORD    L10066-
      JSR      R5,TXCHAR ;LOAD 321(HEADER2), TX 123(HEADER1)
      321
      8.
      BCC      .+8.      ;BR IF NO ERROR
      ERROR    ;REPORT STACKED ERROR
                                TRAP      C$ERROR
      ESCAPE   SUB      ;SKIP TO END OF TEST
                                TRAP      C$ESCAPE
                                .WORD    L10066-
      JSR      R5,WRITEI ;NOW CHANGE TXCL TO 4 BITS TOO
      PCR
      BIT7!BIT2
      JSR      R5,TXCHAR ;LOAD 111(DATA1), TX 321(HEADER2)
      111
      8.
      BCC      .+8.      ;BR IF NO ERROR
      ERROR    ;REPORT STACKED ERROR
                                TRAP      C$ERROR
      ESCAPE   SUB      ;SKIP TO END OF TEST
                                TRAP      C$ESCAPE
                                .WORD    L10066-
;=====
10$:  MOV      #T30TBL+1,R3 ;SET UP DATA TABLE POINTER
      MOVB    (R3)+,1$     ;INSTALL NEXT TX CHARACTER
1$:  JSR      R5,TXCHAR    ;TRANSMIT CHARACTER ( ==> RX/FIFO )
      000                ;** HOLE FOR NEXT CHARACTER **
      4.
      BCC      .+8.      ;BR IF NO ERROR
      ERROR    ;REPORT STACKED ERROR
                                TRAP      C$ERROR
      ESCAPE   SUB      ;SKIP TO END OF TEST
                                TRAP      C$ESCAPE
                                .WORD    L10066-
      CMP      #T30TBL+7,R3 ;ALL CHARACTERS TRANSMITTED ?
      BNE     10$         ; IF NOT, TX ANOTHER ONE
;=====
      JSR      R5,TXCTRL  ;SET TEOM
      TEOM
      0
      JSR      R5,STEPLU  ; AND TX DATA7 + SOME FLAGS
      36.
      JSR      R5,RXCHAR  ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
      RXSOM!123         ; & CHECK FOR RSOM=1
      0

```

CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7550 031754 100000      NOCRDA      ;NO INITIAL CHECK OF RDA=0
7551 031756 103003      BCC      .+8.      ;BR IF NO ERROR
7552 031760      ERROR      ;REPORT STACKED ERROR
7553 031760 104460      ESCAPE SUB      ;SKIP TO END OF TEST      TRAP  C$ERROR
7554 031762      ;
7555 031762 104410      ;
7556 031764 000136      ;
7557      ;
7558 031766 004537 010110  JSR      R5,RXCHAR      ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7559 031772 000321      321
7560 031774 000000      0
7561 031776 100000      NOCRDA      ;NO INITIAL CHECK OF RDA=0
7562 032000 103003      BCC      .+8.      ;BR IF NO ERROR
7563 032002      ERROR      ;REPORT STACKED ERROR
7564 032002 104460      ESCAPE SUB      ;SKIP TO END OF TEST      TRAP  C$ERROR
7565 032004      ;
7566 032004 104410      ;
7567 032006 000114      ;
7568      ;
7569 032010 012703 033470  ;=====
7570 032014 112337 032032 40$:  MOV      #T30TBL,R3      ;SET UP DATA TABLE POINTER
7571 032020 142737 000360 032032  MOVB     (R3)+,4$      ;INSTALL NEXT EXPECTED RX CHARACTER
7572      BICB     #360,4$      ;MASK OUT UNTRANSMITTED BITS
7573 032026 004537 010110  JSR      R5,RXCHAR      ;READ/CHECK NEXT CHARACTER
7574 032032 000000      000      ;** HOLE FOR NEXT EXPECTED CHARACTER
7575 032034 000000      0
7576 032036 100000      NOCRDA      ;NO INITIAL CHECK OF RDA=0
7577 032040 103003      BCC      .+8.      ;BR IF NO ERROR
7578 032042      ERROR      ;REPORT STACKED ERROR
7579 032042 104460      ESCAPE SUB      ;SKIP TO END OF TEST      TRAP  C$ERROR
7580 032044      ;
7581 032044 104410      ;
7582 032046 000054      ;
7583      ;
7584 032050 022703 033475  CMP      #T30TBL+5,R3      ;ALL CHARACTERS CHECKED ?
7585 032054 001357      BNE     40$      ; IF NOT, CHECK ANOTHER ONE
7586      ;=====
7587 032056 004537 010110  JSR      R5,RXCHAR      ;READ/CHK DATA6(006), RCV DATA7
7588 032062 000006      006
7589 032064 000000      0
7590 032066 020000      NCRACT      ;DON'T CHECK FOR FINAL RXACT=1
7591 032070 103003      BCC      .+8.      ;BR IF NO ERROR
7592 032072      ERROR      ;REPORT STACKED ERROR
7593 032072 104460      ESCAPE TST      ;SKIP TO END OF TEST      TRAP  C$ERROR
7594 032074      ;
7595 032074 104410      ;
7596 032076 001370      ;
7597      ;
7598 032100 004537 010110  JSR      R5,RXCHAR      ;READ/CHK DATA7(017), RCV FIRST FLAG
7599 032104 001017      RXEQM!017      ; AND CHECK FOR REQM=1
7600 032106 000001      RERCHK      ; & CHECK RERR BIT=0 (GOOD CRC)
7601 032110 060000      NCRDA!NCRACT      ;DON'T CHECK FOR FINAL RDA=RXACT=1
7602 032112 103003      BCC      .+8.      ;BR IF NO ERROR
7603 032114      ERROR      ;REPORT STACKED ERROR
7604 032114 104460      ESCAPE TST      ;SKIP TO END OF TEST      TRAP  C$ERROR
7605 032116      ;

```

CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7606 032116 104410                                TRAP  C$ESCAPE
7607 032120 001346                                .WORD L10063-.
7608 032122                                ENDSUB
7609 032122                                L10066:
7610 032122 104403                                TRAP  C$ESUB
7611
7612
7613
7614 032124                                BGNSUB
7615 032124                                T25.4:
7616 032124 104402                                TRAP  C$BSUB
7617 032126 004537 007400                        JSR    R5,INITRN      ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
7618 032132 000226                                SYNCH                ;SET BOP MODE,CRC-CCITT-1,S/AR=226
7619 032134 000005                                BIT2!BIT0            ;INITIALLY: TXCL=8 BITS / RXCL=5 BITS
7620 032136 103003                                BCC    .+8.          ;BR IF NO ERROR
7621 032140                                ERROR                ;REPORT STACKED ERROR
7622 032140 104460                                ESCAPE SUB           ;SKIP TO END OF TEST                                TRAP  C$ERROR
7623 032142
7624 032142 104410                                TRAP  C$ESCAPE
7625 032144 000344                                .WORD L10067-.
7626
7627 032146 004537 010010                        JSR    R5,TXCTRL     ;LOAD 2ND FLAG,TX 1ST FLAG
7628 032152 000001                                TSOM
7629 032154 000007                                7.
7630 032156 004537 010010                        JSR    R5,TXCTRL     ;CLEAR TSOM
7631 032162 000000                                000
7632 032164 000000                                0
7633 032166 004537 007676                        JSR    R5,TXCHAR     ;LOAD 123(HEADER1), TX 2ND FLAG
7634 032172 000123                                123
7635 032174 000010                                8.
7636 032176 103003                                BCC    .+8.          ;BR IF NO ERROR
7637 032200                                ERROR                ;REPORT STACKED ERROR
7638 032200 104460                                ESCAPE SUB           ;SKIP TO END OF TEST                                TRAP  C$ERROR
7639 032202
7640 032202 104410                                TRAP  C$ESCAPE
7641 032204 000304                                .WORD L10067-.
7642
7643 032206 004537 007676                        JSR    R5,TXCHAR     ;LOAD 321(HEADER2), TX 123(HEADER1)
7644 032212 000321                                321
7645 032214 000010                                8.
7646 032216 103003                                BCC    .+8.          ;BR IF NO ERROR
7647 032220                                ERROR                ;REPORT STACKED ERROR
7648 032220 104460                                ESCAPE SUB           ;SKIP TO END OF TEST                                TRAP  C$ERROR
7649 032222
7650 032222 104410                                TRAP  C$ESCAPE
7651 032224 000264                                .WORD L10067-.
7652
7653 032226 004537 003734                        JSR    R5,WRITEI     ;NOW CHANGE TXCL TO 5 BITS TOO
7654 032232 120407                                PCR
7655 032234 000245                                BIT7!BIT5!BIT2!BIT0
7656 032236 004537 007676                        JSR    R5,TXCHAR     ;LOAD 111(DATA1), TX 321(HEADER2)
7657 032242 000111                                111
7658 032244 000010                                8.
7659 032246 103003                                BCC    .+8.          ;BR IF NO ERROR
7660 032250                                ERROR                ;REPORT STACKED ERROR
7661 032250 104460                                TRAP  C$ERROR

```

CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7662 032252          ESCAPE SUB          ;SKIP TO END OF TEST
7663 032252 104410          TRAP        C$ESCAPE
7664 032254 000234          .WORD        L10067-.
7665          ;=====
7666 032256 012703 033471    MOV      #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7667 032262 112337 032272    MOV      (R3)+,1$      ;INSTALL NEXT TX CHARACTER
7668          10$:
7669 032266 004537 007676    JSR      R5,TXCHAR     ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7670 032272 000000          000                    ;** HOLE FOR NEXT CHARACTER **
7671 032274 000005          5.
7672 032276 103003          BCC      .+8.          ;BR IF NO ERROR
7673 032300          ERROR                ;REPORT STACKED ERROR
7674 032300 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP        C$ERROR
7675 032302          TRAP        C$ESCAPE
7676 032302 104410          .WORD        L10067-.
7677 032304 000204
7678          ;=====
7679 032306 022703 033477    CMP      #T30TBL+7,R3  ;ALL CHARACTERS TRANSMITTED ?
7680 032312 001363          BNE      10$          ; IF NOT, TX ANOTHER ONE
7681          ;=====
7682 032314 004537 010010    JSR      R5,TXCTRL     ;SET TEOM
7683 032320 000002          TEOM
7684 032322 000000          0
7685 032324 004537 011614    JSR      R5,STEPLU     ; AND TX DATA7 + SOME FLAGS
7686 032330 000044
7687          ;=====
7688 032332 004537 010110    JSR      R5,RXCHAR     ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7689 032336 000523          RXSOM!123            ; & CHECK FOR RSOM=1
7690 032340 000000          0
7691 032342 100000          NOCRDA                ;NO INITIAL CHECK OF RDA=0
7692 032344 103003          BCC      .+8.          ;BR IF NO ERROR
7693 032346          ERROR                ;REPORT STACKED ERROR
7694 032346 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP        C$ERROR
7695 032350          TRAP        C$ESCAPE
7696 032350 104410          .WORD        L10067-.
7697 032352 000136
7698          ;=====
7699 032354 004537 010110    JSR      R5,RXCHAR     ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7700 032360 000321          321
7701 032362 000000          0
7702 032364 100000          NOCRDA                ;NO INITIAL CHECK OF RDA=0
7703 032366 103003          BCC      .+8.          ;BR IF NO ERROR
7704 032370          ERROR                ;REPORT STACKED ERROR
7705 032370 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP        C$ERROR
7706 032372          TRAP        C$ESCAPE
7707 032374 104410          .WORD        L10067-.
7708 032374 000114
7709          ;=====
7710 032376 012703 033470    MOV      #T30TBL,R3    ;SET UP DATA TABLE POINTER
7711 032402 112337 032420    MOV      (R3)+,4$      ;INSTALL NEXT EXPECTED RX CHARACTER
7712 032406 142737 000340 032420 BICB     #340,4$      ;MASK OUT UNTRANSMITTED BITS
7713          40$:
7714 032414 004537 010110    JSR      R5,RXCHAR     ;READ/CHECK NEXT CHARACTER
7715 032420 000000          000                    ;** HOLE FOR NEXT EXPECTED CHARACTER
7716 032422 000000          0
7717 032424 100000          NOCRDA                ;NO INITIAL CHECK OF RDA=0

```

CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7718 032426 103003          BCC      .+8.          ;BR IF NO ERROR
7719 032430                ERROR          ;REPORT STACKED ERROR
7720 032430 104460                TRAP      CSERROR
7721 032432                ESCAPE  SUB          ;SKIP TO END OF TEST
7722 032432 104410                TRAP      C$ESCAPE
7723 032434 000054                .WORD    L10067-.
7724
7725 032436 022703 033475        CMP      #T30TBL+5,R3  ;ALL CHARACTERS CHECKED ?
7726 032442 001357                BNE      40$           ; IF NOT, CHECK ANOTHER ONE
7727
:-----:
7728 032444 004537 010110        JSR      R5,RXCHAR     ;READ/CHK DATA6(026), RCV DATA7
7729 032450 000026                026
7730 032452 000001                RERCHK
7731 032454 020000                NCRACT
7732 032456 103003                BCC      .+8.          ; & CHECK RERR BIT=0 (GOOD CRC)
7733 032460                ERROR          ;DON'T CHECK FOR INITIAL RDA=0
7734 032460 104460                TRAP      CSERROR
7735 032462                ESCAPE  TST          ;BR IF NO ERROR
7736 032462 104410                TRAP      C$ESCAPE
7737 032464 001002                .WORD    L10063-.
7738
7739 032466 004537 010110        JSR      R5,RXCHAR     ;READ/CHK DATA7(037), RCV FIRST FLAG
7740 032472 001037                RXEOM!037
7741 032474 000000                0
7742 032476 060000                NRCRDA!NCRACT
7743 032500 103003                BCC      .+8.          ;DON'T CHECK FOR FINAL RDA=RXACT=1
7744 032502                ERROR          ;BR IF NO ERROR
7745 032502 104460                TRAP      CSERROR
7746 032504                ESCAPE  TST          ;REPORT STACKED ERROR
7747 032504 104410                TRAP      C$ESCAPE
7748 032506 000760                .WORD    L10063-.
7749 032510
7750 032510
7751 032510 104403                L10067: TRAP      C$ESUB
7752
:-----:
7753
: SUBROUTINE # 5: 6 BIT CHARACTERS
:-----:
7754
7755 032512                BGNSUB
7756 032512
7757 032512 104402                T25.5: TRAP      C$BSUB
7758 032514 004537 007400        JSR      R5,INITRN    ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
7759 032520 000226                SYNCH
7760 032522 000006                BIT2!BIT1
7761 032524 103003                BCC      .+8.          ;SET BOP MODE,CRC-CCITT-1,S/AR=226
7762 032526                ERROR          ;INITIALLY: TXCL=8 BITS / RXCL=6 BITS
7763 032526 104460                TRAP      CSERROR
7764 032530                ESCAPE  SUB          ;BR IF NO ERROR
7765 032530 104410                TRAP      C$ESCAPE
7766 032532 000344                .WORD    L10070-.
7767
7768 032534 004537 010010        JSR      R5,TXCTRL    ;LOAD 2ND FLAG,TX 1ST FLAG
7769 032540 000001                TSOM
7770 032542 000007                7.
7771 032544 004537 010010        JSR      R5,TXCTRL    ;CLEAR TSOM
7772 032550 000000                000
7773 032552 000000                0

```


CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7774 032554 004537 007676      JSR      R5,TXCHAR      ;LOAD 123(HEADER1), TX 2ND FLAG
7775 032560 000123              123
7776 032562 000010              8.
7777 032564 103003              BCC      .+8.          ;BR IF NO ERROR
7778 032566 104460              ERROR      ;REPORT STACKED ERROR
7779 032566 104460              ESCAPE   SUB          ;SKIP TO END OF TEST      TRAP      CSERROR
7780 032570 104410              ;WORD      C$ESCAPE
7781 032570 000304              L10070-.
7782 032572 000304
7783
7784 032574 004537 007676      JSR      R5,TXCHAR      ;LOAD 321(HEADER2), TX 123(HEADER1)
7785 032600 000321              321
7786 032602 000010              8.
7787 032604 103003              BCC      .+8.          ;BR IF NO ERROR
7788 032606 104460              ERROR      ;REPORT STACKED ERROR
7789 032606 104460              ESCAPE   SUB          ;SKIP TO END OF TEST      TRAP      CSERROR
7790 032610 104410              ;WORD      C$ESCAPE
7791 032610 000264              L10070-.
7792 032612 000264
7793
7794 032614 004537 003734      JSR      R5,WRITEI      ;NOW CHANGE TXCL TO 6 BITS TOO
7795 032620 120407              PCR
7796 032622 000306              BIT7!BIT6!BIT2!BIT1
7797 032624 004537 007676      JSR      R5,TXCHAR      ;LOAD 111(DATA1), TX 321(HEADER2)
7798 032630 000111              111
7799 032632 000010              8.
7800 032634 103003              BCC      .+8.          ;BR IF NO ERROR
7801 032636 104460              ERROR      ;REPORT STACKED ERROR
7802 032636 104460              ESCAPE   SUB          ;SKIP TO END OF TEST      TRAP      CSERROR
7803 032640 104410              ;WORD      C$ESCAPE
7804 032640 000234              L10070-.
7805 032642 000234
7806
7807 032644 012703 033471      ;=====
7808 032650 112337 032660      10$:   MOV      #T30TBL+1,R3      ;SET UP DATA TABLE POINTER
7809                                MOVB     (R3)+,1$              ;INSTALL NEXT TX CHARACTER
7810 032654 004537 007676      1$:   JSR      R5,TXCHAR      ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7811 032660 000000              000      ;** HOLE FOR NEXT CHARACTER **
7812 032662 000006              6.
7813 032664 103003              BCC      .+8.          ;BR IF NO ERROR
7814 032666 104460              ERROR      ;REPORT STACKED ERROR
7815 032666 104460              ESCAPE   SUB          ;SKIP TO END OF TEST      TRAP      CSERROR
7816 032670 104410              ;WORD      C$ESCAPE
7817 032670 000204              L10070-.
7818 032672 000204
7819
7820 032674 022703 033477      CMP      #T30TBL+7,R3      ;ALL CHARACTERS TRANSMITTED ?
7821 032700 001363              BNE      10$            ; IF NOT, TX ANOTHER ONE
7822                                ;=====
7823 032702 004537 010010      JSR      R5,TXCTRL      ;SET TEOM
7824 032706 000002              TEOM
7825 032710 000000              0
7826 032712 004537 011614      JSR      R5,STEPLU      ; AND TX DATA7 + SOME FLAGS
7827 032716 000044              36.
7828
7829 032720 004537 010110      JSR      R5,RXCHAR      ;READ & CHK 123(HEADER1), RCV 321(HEADER2)

```

CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7830 032724 000523          RXSOM!123          ; & CHECK FOR RSOM=1
7831 032726 000000          0
7832 032730 100000          NOCRDA          ;NO INITIAL CHECK OF RDA=0
7833 032732 103003          BCC      .+8.   ;BR IF NO ERROR
7834 032734          ERROR          ;REPORT STACKED ERROR
7835 032734 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP      C$ERROR
7836 032736          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP      C$ESCAPE
7837 032736 104410          ESCAPE SUB          ;SKIP TO END OF TEST          .WORD    L10070-.
7838 032740 000136          ESCAPE SUB          ;SKIP TO END OF TEST          .WORD    L10070-.
7839
7840 032742 004537 010110    JSR      R5,RXCHAR   ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7841 032746 000321          321
7842 032750 000000          0
7843 032752 100000          NOCRDA          ;NO INITIAL CHECK OF RDA=0
7844 032754 103003          BCC      .+8.   ;BR IF NO ERROR
7845 032756          ERROR          ;REPORT STACKED ERROR
7846 032756 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP      C$ERROR
7847 032760          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP      C$ESCAPE
7848 032760 104410          ESCAPE SUB          ;SKIP TO END OF TEST          .WORD    L10070-.
7849 032762 000114          ESCAPE SUB          ;SKIP TO END OF TEST          .WORD    L10070-.
7850
=====
7851 032764 012703 033470    40$:    MOV      #T30TBL,R3   ;SET UP DATA TABLE POINTER
7852 032770 112337 033006    MOVB    (R3)+,4$       ;INSTALL NEXT EXPECTED RX CHARACTER
7853 032774 142737 000300 033006    BICB    #300,4$       ;MASK OUT UNTRANSMITTED BITS
7854
7855 033002 004537 010110    4$:    JSR      R5,RXCHAR   ;READ/CHECK NEXT CHARACTER
7856 033006 000000          000          ;** HOLE FOR NEXT EXPECTED CHARACTER
7857 033010 000000          0
7858 033012 100000          NOCRDA          ;NO INITIAL CHECK OF RDA=0
7859 033014 103003          BCC      .+8.   ;BR IF NO ERROR
7860 033016          ERROR          ;REPORT STACKED ERROR
7861 033016 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP      C$ERROR
7862 033020          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP      C$ESCAPE
7863 033020 104410          ESCAPE SUB          ;SKIP TO END OF TEST          .WORD    L10070-.
7864 033022 000054          ESCAPE SUB          ;SKIP TO END OF TEST          .WORD    L10070-.
7865
7866 033024 022703 033475    CMP     #T30TBL+5,R3   ;ALL CHARACTERS CHECKED ?
7867 033030 001357          BNE     40$          ; IF NOT, CHECK ANOTHER ONE
7868
=====
7869 033032 004537 010110    JSR      R5,RXCHAR   ;READ/CHK DATA6(066), RCV DATA7
7870 033036 000066          066
7871 033040 000001          RERCHK          ; & CHECK RERR BIT=0 (GOOD CRC)
7872 033042 020000          NCRACT          ;DON'T CHECK FOR INITIAL RDA=0
7873 033044 103003          BCC      .+8.   ;BR IF NO ERROR
7874 033046          ERROR          ;REPORT STACKED ERROR
7875 033046 104460          ESCAPE TST          ;SKIP TO END OF TEST          TRAP      C$ERROR
7876 033050          ESCAPE TST          ;SKIP TO END OF TEST          TRAP      C$ESCAPE
7877 033050 104410          ESCAPE TST          ;SKIP TO END OF TEST          .WORD    L10063-.
7878 033052 000414          ESCAPE TST          ;SKIP TO END OF TEST          .WORD    L10063-.
7879
7880 033054 004537 010110    JSR      R5,RXCHAR   ;READ/CHK DATA7(077), RCV FIRST FLAG
7881 033060 001077          RXEOM!077      ; & CHECK FOR REOM=1
7882 033062 000000          0
7883 033064 060000          NRCRDA!NCRACT   ;DON'T CHECK FOR FINAL RDA=RXACT=1
7884 033066 103003          BCC      .+8.   ;BR IF NO ERROR
7885 033070          ERROR          ;REPORT STACKED ERROR

```

CVDMA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7886 033070 104460
7887 033072
7888 033072 104410
7889 033074 000372
7890 033076
7891 033076
7892 033076 104403
7893
7894
7895
7896 033100
7897 033100
7898 033100 104402
7899 033102 004537 007400
7900 033106 000226
7901 033110 000007
7902 033112 103003
7903 033114
7904 033114 104460
7905 033116
7906 033116 104410
7907 033120 000344
7908
7909 033122 004537 010010
7910 033126 000001
7911 033130 000007
7912 033132 004537 010010
7913 033136 000000
7914 033140 000000
7915 033142 004537 007676
7916 033146 000123
7917 033150 000010
7918 033152 103003
7919 033154
7920 033154 104460
7921 033156
7922 033156 104410
7923 033160 000304
7924
7925 033162 004537 007676
7926 033166 000321
7927 033170 000010
7928 033172 103003
7929 033174
7930 033174 104460
7931 033176
7932 033176 104410
7933 033200 000264
7934
7935 033202 004537 003734
7936 033206 120407
7937 033210 000347
7938 033212 004537 007676
7939 033216 000111
7940 033220 000010
7941 033222 103003

```

```

          ESCAPE TST          ;SKIP TO END OF TEST          TRAP C$ERROR
          ENDSUB
          L10070:
          TRAP C$ESUB
-----
SUBROUTINE # 6: 7 BIT CHARACTERS
-----
          BGNSUB
          T25.6:
          TRAP C$BSUB
          JSR R5,INITRN      ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
          SYNCH              ;SET BOP MODE,CRC-CCITT-1,S/AR=226
          RXDL              ;INITIALLY: TXCL=8 BITS / RXCL=7 BITS
          BCC .+8.          ;BR IF NO ERROR
          ERROR             ;REPORT STACKED ERROR
          ESCAPE SUB        ;SKIP TO END OF TEST          TRAP C$ERROR
          TRAP C$ESCAPE
          .WORD L10071-.
          JSR R5,TXCTRL     ;LOAD 2ND FLAG, TX 1ST FLAG
          TSOM
          7.
          JSR R5,TXCTRL     ;CLEAR TSOM
          000
          0
          JSR R5,TXCHAR     ;LOAD 123(HEADER1), TX 2ND FLAG
          123
          8.
          BCC .+8.          ;BR IF NO ERROR
          ERROR             ;REPORT STACKED ERROR
          TRAP C$ERROR
          ESCAPE SUB        ;SKIP TO END OF TEST          TRAP C$ESCAPE
          .WORD L10071-.
          JSR R5,TXCHAR     ;LOAD 321(HEADER2), TX 123(HEADER1)
          321
          8.
          BCC .+8.          ;BR IF NO ERROR
          ERROR             ;REPORT STACKED ERROR
          TRAP C$ERROR
          ESCAPE SUB        ;SKIP TO END OF TEST          TRAP C$ESCAPE
          .WORD L10071-.
          JSR R5,WRITEI     ;NOW CHANGE TXCL TO 6 BITS TOO
          PCR
          TXDL!RXDL
          JSR R5,TXCHAR     ;LOAD 111(DATA1), TX 321(HEADER2)
          111
          8.
          BCC .+8.          ;BR IF NO ERROR

```

CVDMCA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7942 033224          ERROR          ;REPORT STACKED ERROR
7943 033224 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7944 033226          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7945 033226 104410          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7946 033230 000234          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7947          ;=====
7948 033232 012703 033471 10$: MOV  #T30TBL+1,R3 ;SET UP DATA TABLE POINTER
7949 033236 112337 033246 10$: MOVB (R3)+,1$ ;INSTALL NEXT TX CHARACTER
7950
7951 033242 004537 007676 1$: JSR  R5,TXCHAR ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7952 033246 000000 1$: 000 ;** HOLE FOR NEXT CHARACTER **
7953 033250 000007 1$: 7.
7954 033252 103003 1$: BCC  .+8. ;BR IF NO ERROR
7955 033254          ERROR          ;REPORT STACKED ERROR          TRAP  CSERROR
7956 033254 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7957 033256          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7958 033256 104410          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7959 033260 000204          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7960
7961 033262 022703 033477 10$: CMP  #T30TBL+7,R3 ;ALL CHARACTERS TRANSMITTED ?
7962 033266 001363 10$: BNE  10$ ; IF NOT, TX ANOTHER ONE
7963          ;=====
7964 033270 004537 010010 1$: JSR  R5,TXCTRL ;SET TEOM
7965 033274 000002 1$: TEOM
7966 033276 000000 1$: 0
7967 033300 004537 011614 1$: JSR  R5,STEPLU ; AND TX DATA7 + SOME FLAGS
7968 033304 000044 1$: 36.
7969
7970 033306 004537 010110 1$: JSR  R5,RXCHAR ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7971 033312 000523 1$: RXSOM:123 ; & CHECK FOR RSOM=1
7972 033314 000000 1$: 0
7973 033316 100000 1$: NOCRDA ;NO INITIAL CHECK OF RDA=0
7974 033320 103003 1$: BCC  .+8. ;BR IF NO ERROR
7975 033322          ERROR          ;REPORT STACKED ERROR          TRAP  CSERROR
7976 033322 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7977 033324          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7978 033324 104410          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7979 033326 000136          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7980
7981 033330 004537 010110 1$: JSR  R5,RXCHAR ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7982 033334 000321 1$: 321
7983 033336 000000 1$: 0
7984 033340 100000 1$: NOCRDA ;NO INITIAL CHECK OF RDA=0
7985 033342 103003 1$: BCC  .+8. ;BR IF NO ERROR
7986 033344          ERROR          ;REPORT STACKED ERROR          TRAP  CSERROR
7987 033344 104460          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7988 033346          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7989 033346 104410          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7990 033350 000114          ESCAPE SUB          ;SKIP TO END OF TEST          TRAP  CSERROR
7991          ;=====
7992 033352 012703 033470 40$: MOV  #T30TBL,R3 ;SET UP DATA TABLE POINTER
7993 033356 112337 033374 40$: MOVB (R3)+,4$ ;INSTALL NEXT EXPECTED RX CHARACTER
7994 033362 142737 000200 033374 40$: BICB #200,4$ ;MASK OUT UNTRANSMITTED BITS
7995
7996 033370 004537 010110 4$: JSR  R5,RXCHAR ;READ/CHECK NEXT CHARACTER
7997 033374 000000 4$: 000 ;** HOLE FOR NEXT EXPECTED CHARACTER
    
```

CVDMCA.P11 10-DEC-80 09:14

TEST 25 -- BOP RX CHARACTER LENGTH TEST

```

7998 033376 000000      0
7999 033400 100000      NOCRDA                ;NO INITIAL CHECK OF RDA=0
8000 033402 103003      BCC      .+8.         ;BR IF NO ERROR
8001 033404                ERROR                ;REPORT STACKED ERROR
8002 033404 104460                ESCAPE SUB                ;SKIP TO END OF TEST                TRAP      C$ERROR
8003 033406                ESCAPE SUB                ;SKIP TO END OF TEST                TRAP      C$ESCAPE
8004 033406 104410                ESCAPE SUB                ;SKIP TO END OF TEST                .WORD    L10071-.
8005 033410 000054                ESCAPE SUB                ;SKIP TO END OF TEST                TRAP      C$ERROR
8006 033410 000054                ESCAPE SUB                ;SKIP TO END OF TEST                TRAP      C$ESCAPE
8007 033412 022703 033475      CMP      #T30TBL+5,R3 ;ALL CHARACTERS CHECKED ?
8008 033416 001357      BNE      40$          ; IF NOT, CHECK ANOTHER ONE
8009 -----
8010 033420 004537 010110      JSR      R5,RXCHAR    ;READ/CHK DATA6(066), RCV DATA7
8011 033424 000066      066
8012 033426 000001      RERCHK                ; & CHECK RERR BIT=0 (GOOD CRC)
8013 033430 020000      NCRACT                ;DON'T CHECK FOR INITIAL RDA=0
8014 033432 103003      BCC      .+8.         ;BR IF NO ERROR
8015 033434                ERROR                ;REPORT STACKED ERROR
8016 033434 104460                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$ERROR
8017 033436                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$ESCAPE
8018 033436 104410                ESCAPE TST                ;SKIP TO END OF TEST                .WORD    L10063-.
8019 033440 000026                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$ERROR
8020 033440 000026                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$ESCAPE
8021 033442 004537 010110      JSR      R5,RXCHAR    ;READ/CHK DATA7(177), RCV FIRST FLAG
8022 033446 001177      RXEOM!177            ; & CHECK FOR REOM=1
8023 033450 000000      0
8024 033452 060000      NRCRDA!NCRACT        ;DON'T CHECK FOR FINAL RDA=RXACT=1
8025 033454 103003      BCC      .+8.         ;BR IF NO ERROR
8026 033456                ERROR                ;REPORT STACKED ERROR
8027 033456 104460                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$ERROR
8028 033460                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$ESCAPE
8029 033460 104410                ESCAPE TST                ;SKIP TO END OF TEST                .WORD    L10063-.
8030 033462 000004                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$ERROR
8031 033464                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$ESCAPE
8032 033464 104403                ESCAPE TST                ;SKIP TO END OF TEST                .WORD    L10071:
8033 033464 104403                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$ESUB
8034 033466                ESCAPE TST                ;SKIP TO END OF TEST                .WORD    L10063:
8035 033466 104401                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$SETST
8036 033466 104401                ESCAPE TST                ;SKIP TO END OF TEST                TRAP      C$SETST
8037 -----
8038 033470      111      T30TBL: .BYTE 111      ;D1
8039 033471      222      .BYTE 222      ;D2
8040 033472      333      .BYTE 333      ;D3
8041 033473      044      .BYTE 044      ;D4
8042 033474      155      .BYTE 155      ;D5
8043 033475      266      .BYTE 266      ;D6
8044 033476      377      .BYTE 377      ;D7
8045 033500      .EVEN
8046 -----

```

CVDMCA.P11 10-DEC-80 09:14

TEST 26 -- TX 'SPACING SEQUENCE'

8047
8048
8049
8050
8051
8052
8053
8054
8055
8056
8057
8058
8059
8060
8061 033500
8062 033500 004737 005420
8063
8064
8065
8066 033504 004537 003734
8067 033510 120013
8068 033512 000200
8069 033514 004537 003734
8070 033520 120006
8071 033522 000300
8072 033524 004537 003734
8073 033530 120007
8074 033532 000000
8075
8076 033534 004537 003734
8077 033540 120000
8078 033542 000031
8079 033544 004537 003734
8080 033550 120000
8081 033552 000030
8082
8083 033554 004537 003734
8084 033560 120404
8085 033562 000000
8086 033564 004537 003734
8087 033570 120405
8088 033572 000007
8089
8090 033574 004537 003734
8091 033600 120000
8092 033602 000172
8093
8094 033604 004537 006036
8095 033610 000001
8096 033612 103003
8097 033614
8098 033614 104460
8099 033616
8100 033616 104410
8101 033620 000234
8102

.SBTTL TEST 26 -- TX 'SPACING SEQUENCE'

```

:*****
:*
:* TEST 26 -- TX 'SPACING SEQUENCE'
:*
:* THE TRANSMITTER IS INITIALIZED AND THE 'SPACING SEQUENCE' IS FORCED
:* BY ASSERTING BOTH TSOM & TEOM AT THE SAME TIME -- CHECK THE BIT
:* STREAM FOR ACCURACY (SPACES) AND COMPLETNESS (16 OF THEM). WHEN TXBE
:* GOES HIGH (= 1) A SMALL MESSAGE IS SENT.
:*
:-----*****

```

```

: BGNTST
: JSR PC,INIDMV ;INIT VIA T26::
:-----
: SET UP USYRT AND VIA REGISTERS
:-----
: 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,WRITEI ;RESET THE USYRT
: VIAORB
: RTSND!DTR!PRESET
: JSR R5,WRITEI ;CLEAR USYRT RESET BIT
: VIAORB
: RTSND!DTR
: JSR R5,WRITEI ;LOAD USYRT PCSARL
: PCSARL
: 000
: JSR R5,WRITEI ;LOAD USYRT PCSARH
: PCSARH
: XYZ ;(NO ERROR CHECKING)
: JSR R5,WRITEI ;SET UP USYRT
: VIAORB
: RTSND!TXEN!RXEN!DTR!TTLOOP
: JSR R5,CKTBMT ;CHK FOR TBMT = 1
: 1
: BCC .+8. ;IF NO ERROR, PROCEED
: ERROR ;ELSE, REPORT IT AND
: ESCAPE TST ; EXIT THIS TEST TRAP CSError
: TRAP C$ESCAPE
: .WORD L10072-
:-----

```

CVDMA.P11 10-DEC-80 09:14

TEST 26 -- TX 'SPACING SEQUENCE'

```

8103 ; INIT SPACING SEQUENCE BY SETTING TSOM AND TEOM
8104 -----
8105 033622 004537 010010 6$: JSR R5,TXCTRL ;SET TSOM AND TEOM
8106 033626 000003 TSOM!TEOM ; ('SPACING SEQUENCE')
8107 033630 000000 0
8108
8109 033632 004537 011614 JSR R5,STEPLU ;1 TICK TO START SPACE SEQUENCE
8110 033636 000001 1
8111
8112 033640 004537 007256 JSR R5,SERIAL ;READ 000 CHARACTER VIA TSO
8113 033644 000010 8. ; 8 BIT CHAR/CLOCK TICKS
8114 033646 000000 000 ; EXPECTED BIT SEQUENCE (0000000)
8115 033650 103003 BCC .+8. ;BR IF NO ERROR
8116 033652 ERROR ;REPORT STACKED ERROR
8117 033652 104460 ESCAPE TST ;AND EXIT TEST TRAP CSERROR
8118 033654 104410 ; TRAP C$ESCAPE
8119 033654 104410 .WORD L10072-.
8120 033656 000176
8121
8122 033660 004537 006036 JSR R5,CKTBMT ;CHK FOR TBMT = 1
8123 033664 000001 1
8124 033666 103003 BCC .+8. ;IF NO ERROR, PROCEED
8125 033670 ERROR ;ELSE, REPORT IT AND TRAP CSERROR
8126 033670 104460 ESCAPE TST ; EXIT THIS TEST TRAP C$ESCAPE
8127 033672 104410 ; TRAP L10072-.
8128 033672 104410 .WORD
8129 033674 000160
8130 -----
8131 033676 004537 010010 JSR R5,TXCTRL ;CLEAR TEOM
8132 033702 000001 TSOM
8133 033704 000000 0
8134
8135 033706 004537 007256 JSR R5,SERIAL ;READ 000 CHARACTER VIA TSO
8136 033712 000010 8. ; 8 BIT CHAR/CLOCK TICKS
8137 033714 000000 000 ; EXPECTED BIT SEQUENCE (00000000)
8138 033716 103003 BCC .+8. ;BR IF NO ERROR
8139 033720 ERROR ;REPORT STACKED ERROR TRAP CSERROR
8140 033720 104460 ESCAPE TST ;AND EXIT TEST TRAP C$ESCAPE
8141 033722 104410 ; TRAP L10072-.
8142 033722 104410 .WORD
8143 033724 000130
8144
8145 033726 004537 006036 JSR R5,CKTBMT ;CHK FOR TBMT = 1
8146 033732 000001 1
8147 033734 103003 BCC .+8. ;IF NO ERROR, PROCEED
8148 033736 ERROR ;ELSE, REPORT IT AND TRAP CSERROR
8149 033736 104460 ESCAPE TST ; EXIT THIS TEST TRAP C$ESCAPE
8150 033740 104410 ; TRAP L10072-.
8151 033740 104410 .WORD
8152 033742 000112
8153 -----
8154 SEND 'SHORT MESSAGE' (125,252)
8155 -----
8156 033744 004537 003734 JSR R5,WRITEI ;LOAD 125 CHARACTER
8157 033750 120402 TDSRL
8158 033752 000125 125

```

CVDMCA.P11 10-DEC-80 09:14

TEST 26 -- TX 'SPACING SEQUENCE'

8159	033754	004537	010010	JSR	R5,TXCTRL	:CLEAR TSOM		
8160	033760	000000		000				
8161	033762	000000		0				
8162								
8163	033764	004537	007256	JSR	R5,SERIAL	:READ FLAG CHARACTER (176) VIA TSO		
8164	033770	000010		8.		: 8 BIT CHAR/CLOCK TICKS		
8165	033772	000176		176		: EXPECTED BIT SEQUENCE (01111110)		
8166	033774	103003		BCC	+.8.	:BR IF NO ERROR		
8167	033776			ERROR		:REPORT STACKED ERROR		
8168	033776	104460					TRAP	C\$ERROR
8169	034000			ESCAPE	TST	:AND EXIT TEST		
8170	034000	104410					TRAP	C\$ESCAPE
8171	034002	000052					.WORD	L10072-.
8172								
8173	034004	004537	003734	JSR	R5,WRITEI	:LOAD 252 CHARACTER		
8174	034010	120402		TDSRL				
8175	034012	000252		252				
8176								
8177	034014	004537	007256	JSR	R5,SERIAL	:READ 1ST DATA CHARACTER (125) VIA TSO		
8178	034020	000010		8.		: 8 BIT CHAR/CLOCK TICKS		
8179	034022	000252		252		: EXPECTED BIT SEQUENCE (01010110)		
8180	034024	103003		BCC	+.8.	:BR IF NO ERROR		
8181	034026			ERROR		:REPORT STACKED ERROR		
8182	034026	104460					TRAP	C\$ERROR
8183	034030			ESCAPE	TST	:AND EXIT TEST		
8184	034030	104410					TRAP	C\$ESCAPE
8185	034032	000022					.WORD	L10072-.
8186								
8187	034034	004537	007256	JSR	R5,SERIAL	:READ 2ND DATA CHARACTER (252) VIA TSO		
8188	034040	000010		8.		: 8 BIT CHAR/CLOCK TICKS		
8189	034042	000125		125		: EXPECTED BIT SEQUENCE (10101010)		
8190	034044	103003		BCC	+.8.	:BR IF NO ERROR		
8191	034046			ERROR		:REPORT STACKED ERROR		
8192	034046	104460					TRAP	C\$ERROR
8193	034050			ESCAPE	TST	:AND EXIT TEST		
8194	034050	104410					TRAP	C\$ESCAPE
8195	034052	000002					.WORD	L10072-.
8196	034054							
8197	034054							
8198	034054	104401					TRAP	C\$ETST

ENDTST

L10072:

CVDMCA.P11 10-DEC-80 09:14

TEST 27 -- FIFO OVERRUN INTEGRITY TEST

.SBTTL TEST 27 -- FIFO OVERRUN INTEGRITY TEST

8199
8200
8201
8202
8203
8204
8205
8206
8207
8208
8209
8210
8211
8212
8213
8214
8215
8216
8217
8218
8219
8220 034056
8221 034056 004737 005420
8222
8223 034062 004537 007400
8224 034066 043626
8225 034070 000000
8226 034072 103003
8227 034074
8228 034074 104460
8229 034076
8230 034076 104410
8231 034100 000324
8232 034102 004537 010010
8233 034106 000001
8234 034110 000007
8235
8236 034112 004537 010010
8237 034116 000000
8238 034120 000000
8239
8240
8241
8242 034122 012702 034426
8243 034126 112237 034136
8244
8245 034132 004537 007676
8246 034136 000000
8247 034140 100010
8248 034142 103003
8249 034144
8250 034144 104460
8251 034146
8252 034146 104410
8253 034150 000254
8254

```

:*****
:*
:* TEST 27 -- FIFO OVERRUN INTEGRITY TEST
:*
:* THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS TO FILL
:* THE 8 CHARACTER RECEIVER FIFO UNTIL RXOR WITH THE CHARACTERS:
:* (SYNCH),000,377,125,252,347,030,303,074,125.
:* THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. OF IMPORTANCE
:* IS THE INTEGRITY OF THE LAST OVERRUN-CAUSING FIFO CHARACTER (IT SHOULD
:* REMAIN INTACT).
:* NOTE THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE CHARACTERS SINCE THEY
:* ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
:* RDA = 0).
:*****

```

```

:
: BGNTST
:
: JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP T27::
:
: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
: DDCMP!NOCHK!SYNCH ;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 CSERROR
:
: JSR R5,TXCTRL ;OUTPUT 1ST SYNC CHARACTERS TRAP C$ESCAPE
: TSOM ;.WORD L10073-.
: 7.
:
: JSR R5,TXCTRL ;CLEAR TSOM
: 000
: 0
:-----
: FILL THE FIFO WITH CHARACTERS AND FORCE OVERRUN
:-----
:
: 5$: MOV #TXTBL4,R2 ;SET UP TABLE POINTER
: MOV (R2)+,10$ ;SETUP TRANSMIT CHARACTER
:
: 10$: JSR R5,TXCHAR ;TRANSMIT A CHARACTER
: 000 ;** HOLE FOR NEXT TX CHARACTER
: NCTBMT*256,!8. ;NO CHECK OF INITIAL TBMT=0
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
:
: ESCAPE TST ;SKIP TO END OF TEST TRAP CSERROR
:
: TRAP C$ESCAPE
: .WORD L10073-.

```

CVDMA.P11 10-DEC-80 09:14

TEST 27 -- FIFO OVERRUN INTEGRITY TEST

```

8255 034152 022702 034441      CMP      #TXEND4,R2      ;CONTINUE TRANSMITTING UNTIL
8256 034156 001363              BNE      5$              ;OVERRUN.
8257
8258 034160 004537 006476      JSR      R5,CKROR        ;VERIFY RECEIVER OVERRUN OCCURED
8259 034164 000001              1
8260 034166 103003              BCC      .+8.            ;BR IF OK
8261 034170                      ERROR                    ;REPORT STACKED ERROR
8262 034170 104460                      ESCAPE   TST              ;SKIP TO END OF TEST              TRAP   C$ERROR
8263 034172                      .WORD
8264 034172 104410                      .WORD   C$ESCAPE
8265 034174 000230                      .WORD   L10073-.
8266
8267
8268
8269
-----
: THE RECEIVER SHOULD HAVE OVERRUN. VERIFY THIS AND READ/VERIFY
: THE REMAINING FIFO CHARACTERS (ESPECIALLY THE LAST ONE).
-----
8270 034176 004537 010110      JSR      R5,RXCHAR        ;READ & CHK CHARACTER
8271 034202 000226              SYNCH                    ; (THIS CHARACTER ISN'T AFFECTED)
8272 034204 000000              0
8273 034206 100000              NOCRDA                    ;NO INITIAL CHECK OF RDA=0
8274 034210 103003              BCC      .+8.            ;BR IF NO ERROR
8275 034212                      ERROR                    ;REPORT STACKED ERROR
8276 034212 104460                      ESCAPE   TST              ;SKIP TO END OF TEST              TRAP   C$ERROR
8277 034214                      .WORD
8278 034214 104410                      .WORD   C$ESCAPE
8279 034216 000206                      .WORD   L10073-.
8280
8281 034220 004537 010110      JSR      R5,RXCHAR        ;READ & CHK CHARACTER
8282 034224 000000              000                      ; (THIS IS THE OVERRITTEN CHARACTER)
8283 034226 000000              0                          ; ... WAS A 'SYNCH' BEFORE OVERRITTEN
8284 034230 100000              NOCRDA                    ;NO INITIAL CHECK OF RDA=0
8285 034232 103003              BCC      .+8.            ;BR IF NO ERROR
8286 034234                      ERROR                    ;REPORT STACKED ERROR
8287 034234 104460                      ESCAPE   TST              ;SKIP TO END OF TEST              TRAP   C$ERROR
8288 034236                      .WORD
8289 034236 104410                      .WORD   C$ESCAPE
8290 034240 000164                      .WORD   L10073-.
8291
8292 034242 012702 034431      MOV      #TXTBL4+3,R2    ;SET UP TABLE POINTER
8293 034246 112237 034256      15$:    MOVB     (R2)+,20$      ;SETUP EXPECTED CHARACTER
8294
8295 034252 004537 010110      JSR      R5,RXCHAR        ;READ & CHK CHARACTER
8296 034256 000000              000                      ;** HOLE FOR EXPECTED RECEIVE CHAR.
8297 034260 000000              0
8298 034262 100000              NOCRDA                    ;NO INITIAL CHECK OF RDA=0
8299 034264 103003              BCC      .+8.            ;BR IF NO ERROR
8300 034266                      ERROR                    ;REPORT STACKED ERROR
8301 034266 104460                      ESCAPE   TST              ;SKIP TO END OF TEST              TRAP   C$ERROR
8302 034270                      .WORD
8303 034270 104410                      .WORD   C$ESCAPE
8304 034272 000132                      .WORD   L10073-.
8305 034274 022702 034436      CMP      #TXEND4-3,R2
8306 034300 001362              BNE      15$
8307
8308 034302 012702 000004      MOV      #4,R2           ;TRANSMIT 4 EXTRA CHARACTERS TO AVOID
8309 034306 004537 007676      30$:    JSR      R5,TXCHAR        ;TRANSMITTER UNDERRUN.
8310 034312 000333              333                      ;FILLER CHARACTER

```

CVDMCA.P11 10-DEC-80 09:14

TEST 27 -- FIFO OVERRUN INTEGRITY TEST

```

8311 034314 100010      NCTBMT*256.!8.      ;NO CHECK OF INITIAL TBMT=0
8312 034316 077205      SOB      R2,30$
8313
8314 034320 004537 010110 JSR      R5,RXCHAR      ;READ & CHK CHARACTER
8315 034324 000303      303
8316 034326 000000      0
8317 034330 100000      NOCRDA      ;NO INITIAL CHECK OF RDA=0
8318 034332 103003      BCC      .+8.      ;BR IF NO ERROR
8319 034334      ERROR      ;REPORT STACKED ERROR
8320 034334 104460      TRAP      C$ERROR
8321 034336      ESCAPE TST      ;SKIP TO END OF TEST
8322 034336 104410      TRAP      C$ESCAPE
8323 034340 000064      .WORD      L10073-.
8324
8325 034342 004537 010110 JSR      R5,RXCHAR      ;READ & CHK CHARACTER
8326 034346 000074      074
8327 034350 000000      0
8328 034352 100000      NOCRDA      ;NO INITIAL CHECK OF RDA=0
8329 034354 103003      BCC      .+8.      ;BR IF NO ERROR
8330 034356      ERROR      ;REPORT STACKED ERROR
8331 034356 104460      TRAP      C$ERROR
8332 034360      ESCAPE TST      ;SKIP TO END OF TEST
8333 034360 104410      TRAP      C$ESCAPE
8334 034362 000042      .WORD      L10073-.
8335
8336 034364 004537 010110 JSR      R5,RXCHAR      ;* READ & CHK FINAL CHARACTER
8337 034370 000125      125
8338 034372 000000      0
8339 034374 100000      NOCRDA      ;* NO INITIAL CHECK OF RDA=0
8340 034376 103003      BCC      .+8.      ;* BR IF NO ERROR
8341 034400      ERROR      ;* REPORT STACKED ERROR
8342 034400 104460      TRAP      C$ERROR
8343 034402      ESCAPE TST      ;* SKIP TO END OF TEST
8344 034402 104410      TRAP      C$ESCAPE
8345 034404 000020      .WORD      L10073-.
8346
8347 034406 004537 011532 JSR      R5,ENDTRN      ;SHUT DOWN TRANSMITTER, RECEIVER
8348 034412 000011      9.
8349 034414 103003      BCC      .+8.      ;BR IF NO ERROR
8350 034416      ERROR      ;REPORT STACKED ERROR
8351 034416 104460      TRAP      C$ERROR
8352 034420      ESCAPE TST      ;SKIP TO END OF TEST
8353 034420 104410      TRAP      C$ESCAPE
8354 034422 000002      .WORD      L10073-.
8355 034424      ENDTST
8356 034424      L10073:
8357 034424 104401      TRAP      C$ETST
8358
8359 034426      226
8360 034427      226
8361 034430      000
8362 034431      377
8363 034432      125
8364 034433      252
8365 034434      347
8366 034435      030

```

```

-----
TXTBL4: .BYTE 226
        .BYTE 226
        .BYTE 000
        .BYTE 377
        .BYTE 125
        .BYTE 252
        .BYTE 347
        .BYTE 030

```

CVDMCA.P11 10-DEC-80 09:14

TEST 27 -- FIFO OVERRUN INTEGRITY TEST

8367	034436	303
8368	034437	074
8369	034440	125
8370	034441	000
8371		
8372		

	.BYTE	303
	.BYTE	074
	.BYTE	125
TXEND4:	.BYTE	000
	.EVEN	

;-----

CVDMCA.P11 10-DEC-80 09:14

TEST 28 -- BCP RX OVERRUN SET AND CLEAR TEST

.SBTTL TEST 28 -- BCP RX OVERRUN SET AND CLEAR TEST

TEST 28 -- BCP RX OVERRUN SET AND CLEAR TEST

THE USYRT IS INITIALIZED AND THREE SUBTESTS ARE PERFORMED.

- 1 -- AN OVERRUN CONDITION IS FORCED, RECEIVER STATUS REGISTER IS READ TWICE: ONCE TO VERIFY ROR BIT = 1, AND AGAIN TO VERIFY THAT THE FIRST READ CLEARED ROR .
- 2 -- AN OVERRUN CONDITION IS FORCED (BY THE SAME TECHNIQUE USED IN (2), THE USYRT IS RESET AND THE PROPER STATE OF ALL REGISTERS IS VERIFIED.
- 3 -- AN OVERRUN CONDITION IS FORCED (AS ABOVE). RXE IS THEN DROPPED AND A DELAY IS PROVIDED TO ALLOW TIME FOR THE FIFO TO FLUSH (CAUSED BY RDA GOING LOW). RXE IS THEN RE-INITIALIZED AND ROR IS CHECKED = 0. THE RECEIVER IS THEN RE-SYNCHED AND THE TEST IS TERMINATED.

BGNTST

T28::

***** SUBTEST #1 *****

BGNSUB

T28.1:

TRAP C\$BSUB

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

JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!NOCHK!SYNCH ;SET DDCMP, NO ERR CHECKING, SYNCH=226
0 ;USE 8 BIT CHARS

BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR

TRAP C\$ERROR

ESCAPE SUB ;SKIP TO END OF TEST

TRAP C\$ESCAPE
.WORD L10075-

JSR R5,TXCTRL ;OUTPUT 1ST SYNC CHARACTERS

TSOM
7.

JSR R5,TXCTRL ;OUTPUT 2ND SYNC CHARACTER

TSOM
8.

JSR R5,TXCTRL ;CLEAR TSOM

000
0

JSR R5,CKROR ;CHECK FOR RXOR = 0

0
BCC .+8.

;BR IF NO ERROR

8373
8374
8375
8376
8377
8378
8379
8380
8381
8382
8383
8384
8385
8386
8387
8388
8389
8390
8391
8392
8393
8394
8395
8396
8397
8398
8399 034442
8400
8401 034442
8402 034442
8403 034442 104402
8404 034444 004737 005420
8405
8406 034450 004537 007400
8407 034454 043626
8408 034456 000000
8409 034460 103003
8410 034462
8411 034462 104460
8412 034464
8413 034464 104410
8414 034466 000112
8415
8416 034470 004537 010010
8417 034474 000001
8418 034476 000007
8419 034500 004537 010010
8420 034504 000001
8421 034506 000010
8422 034510 004537 010010
8423 034514 000000
8424 034516 000000
8425
8426 034520 004537 006476
8427 034524 000000
8428 034526 103003

CVDPCA.P11 10-DEC-80 09:14

TEST 28 -- BCP RX OVERRUN SET AND CLEAR TEST

```

8429 034530          ERROR          ;REPORT STACKED ERROR
8430 034530 104460          ESCAPE SUB      ;SKIP TO END OF TEST          TRAP  C$ERROR
8431 034532          ESCAPE SUB      ;SKIP TO END OF TEST          TRAP  C$ESCAPE
8432 034532 104410          ESCAPE SUB      ;SKIP TO END OF TEST          .WORD L10075-.
8433 034534 000044          ESCAPE SUB      ;SKIP TO END OF TEST
8434 034534 000044          ESCAPE SUB      ;SKIP TO END OF TEST
8435 034536 004537 011614  JSR      R5,STEPLU      ;FORCE RECEIVER OVERRUN
8436 034542 000116          JSR      R5,STEPLU      ;FORCE RECEIVER OVERRUN
8437 034542 000116          JSR      R5,STEPLU      ;FORCE RECEIVER OVERRUN
8438 034544 004537 006476  JSR      R5,CKROR      ;CHECK FOR RXOR = 1
8439 034550 000001          JSR      R5,CKROR      ;CHECK FOR RXOR = 1
8440 034552 103003          BCC     .+8.          ;BR IF NO ERROR
8441 034554          ERROR          ;REPORT STACKED ERROR
8442 034554 104460          ESCAPE SUB      ;SKIP TO END OF TEST          TRAP  C$ERROR
8443 034556          ESCAPE SUB      ;SKIP TO END OF TEST          TRAP  C$ESCAPE
8444 034556 104410          ESCAPE SUB      ;SKIP TO END OF TEST          .WORD L10074-.
8445 034560 000262          ESCAPE SUB      ;SKIP TO END OF TEST
8446 034560 000262          ESCAPE SUB      ;SKIP TO END OF TEST
8447 034562 004537 006476  JSR      R5,CKROR      ;CHECK FOR RXOR = 0
8448 034566 000000          JSR      R5,CKROR      ;CHECK FOR RXOR = 0
8449 034570 103003          BCC     .+8.          ;BR IF NO ERROR
8450 034572          ERROR          ;REPORT STACKED ERROR
8451 034572 104460          ESCAPE SUB      ;SKIP TO END OF TEST          TRAP  C$ERROR
8452 034574          ESCAPE SUB      ;SKIP TO END OF TEST          TRAP  C$ESCAPE
8453 034574 104410          ESCAPE SUB      ;SKIP TO END OF TEST          .WORD L10075-.
8454 034576 000002          ESCAPE SUB      ;SKIP TO END OF TEST
8455 034600          ENDSUB
8456 034600          ENDSUB
8457 034600 104403          ENDSUB
8458 034600 104403          ENDSUB
8459 034602          ;***** SUBTEST #2 *****
8460 034602          BGNSUB
8461 034602 104402          BGNSUB
8462 034604 004737 005420  JSR      PC,INIDMV      ;INIT DMV-11, ENTER M-LOOP
8463 034604 004737 005420  JSR      PC,INIDMV      ;INIT DMV-11, ENTER M-LOOP
8464 034610 004537 007400  JSR      R5,INITRN      ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
8465 034614 043626          DDCMP!NOCHK!SYNCH ;SET DDCMP, NO ERR CHECKING, SYNCH=226
8466 034616 000000          0 ;USE 8 BIT CHARS
8467 034620 103003          BCC     .+8.          ;BR IF NO ERROR
8468 034622          ERROR          ;REPORT STACKED ERROR
8469 034622 104460          ESCAPE SUB      ;SKIP TO END OF TEST          TRAP  C$ERROR
8470 034624          ESCAPE SUB      ;SKIP TO END OF TEST          TRAP  C$ESCAPE
8471 034624 104410          ESCAPE SUB      ;SKIP TO END OF TEST          .WORD L10076-.
8472 034626 000054          ESCAPE SUB      ;SKIP TO END OF TEST
8473 034626 000054          ESCAPE SUB      ;SKIP TO END OF TEST
8474 034630 004537 010010  JSR      R5,TXCTRL      ;OUTPUT 1ST SYNC CHARACTERS
8475 034634 000001          TSOM
8476 034636 000007          7.
8477 034640 004537 010010  JSR      R5,TXCTRL      ;OUTPUT 2ND SYNC CHARACTER
8478 034644 000001          TSOM
8479 034646 000010          8.
8480 034650 004537 010010  JSR      R5,TXCTRL      ;CLEAR TSOM
8481 034654 000000          000
8482 034656 000000          0
8483 034660 004537 011614  JSR      R5,STEPLU      ;FORCE RECEIVER OVERRUN
8484 034664 000116          JSR      R5,STEPLU      ;FORCE RECEIVER OVERRUN

```

CVDMA.P11 10-DEC-80 09:14

TEST 28 -- BCP RX OVERRUN SET AND CLEAR TEST

```

8485
8486 034666 004537 005126      JSR      R5,RSTCHK      ;RESET USYRT/VERIFY SAME
8487 034672 103003              BCC      .+8.           ;BR IF NO ERROR
8488 034674                      ERROR                    ;REPORT STACKED ERROR
8489 034674 104460              ESCAPE TST              ;SKIP TO END OF TEST
8490 034676                      TRAP      C$ERROR
8491 034676 104410              .WORD    C$ESCAPE
8492 034700 000142              L10074-
8493 034702                      ENDSUB
8494 034702                      L10076:
8495 034702 104403              TRAP      C$ESUB
8496
8497 034704                      ;***** SUBTEST #3 *****
8498 034704                      BGNSUB
8499 034704 104402              T28.3:
8500 034706 004737 005420      JSR      PC,INIDMV      ;INIT DMV-11, ENTER M-LOOP
8501
8502 034712 004537 007400      JSR      R5,INITRN      ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
8503 034716 043626              DDCMP!NOCHK!SYNCH ;SET DDCMP, NO ERR CHECKING, SYNCH=226
8504 034720 000000              0                      ;USE 8 BIT CHARS
8505 034722 103003              BCC      .+8.           ;BR IF NO ERROR
8506 034724                      ERROR                    ;REPORT STACKED ERROR
8507 034724 104460              ESCAPE SUB              ;SKIP TO END OF TEST
8508 034726                      TRAP      C$ERROR
8509 034726 104410              .WORD    C$ESCAPE
8510 034730 000110              L10077-
8511
8512 034732 004537 010010      JSR      R5,TXCTRL      ;OUTPUT 1ST SYNC CHARACTERS
8513 034736 000001              TSOM                    ;
8514 034740 000007              7.
8515 034742 004537 010010      JSR      R5,TXCTRL      ;OUTPUT 2ND SYNC CHARACTER
8516 034746 000001              TSOM                    ;
8517 034750 000010              8.
8518 034752 004537 010010      JSR      R5,TXCTRL      ;CLEAR TSOM
8519 034756 000000              000
8520 034760 000000              0
8521 034762 004537 011614      JSR      R5,STEPLU      ;FORCE RECEIVER OVERRUN
8522 034766 000116              78.
8523
8524 034770 004537 003734      JSR      R5,WRITEI      ;DROP RECEIVER ENABLE (RXEN)
8525 034774 120000              VIAORB                  ; (RDA SHOULD ALSO DROP, WHICH WILL
8526 034776 000042              TXEN!TTLOOP            ; CAUSE FIFO TO FLUSH ITSELF).
8527
8528 035000 012701 000050      MOV      #50,R1         ;DELAY FOR NNN SEC. TO ALLOW FIFO TIME
8529 035004 004737 005312      JSR      PC,WAIT50      ;TO FLUSH ITSELF.
8530 035010 077103              SOB
8531
8532 035012 004537 003734      JSR      R5,WRITEI      ;TURN ON RECEIVER ENABLE (RXEN)
8533 035016 120000              VIAORB
8534 035020 000142              TXEN!RXEN!TTLOOP
8535
8536 035022 004537 006476      JSR      R5,CKROR       ;VERIFY CLEARING OF RECEIVER OVERRUN
8537 035026 000000              0
8538 035030 103003              BCC      .+8.           ;BR IF NO ERROR
8539 035032                      ERROR                    ;REPORT STACKED ERROR
8540 035032 104460              TRAP      C$ERROR

```

CVDMCA.P11 10-DEC-80 09:14

TEST 28 -- BCP RX OVERRUN SET AND CLEAR TEST

8541	035034	
8542	035034	104410
8543	035036	000002
8544	035040	
8545	035040	
8546	035040	104403
8547	035042	
8548	035042	
8549	035042	104401

ESCAPE SUB ;SKIP TO END OF TEST

TRAP C\$ESCAPE
.WORD L10077-

ENDSUB

L10077:

TRAP C\$ESUB

ENDTST

L10074:

TRAP C\$ETST

CVDMCA.P11 10-DEC-80 09:14

TEST 29 -- BCP RX SYNC CHARACTER RECOGNITION

.SBTTL TEST 29 -- BCP RX SYNC CHARACTER RECOGNITION

```

:*****
:*
:* TEST 29 -- BCP RX SYNC CHARACTER RECOGNITION
:* THE FOLLOWING MESSAGE IS INITIATED WITHOUT ASSERTING RXE AND ONCE
:* THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):
:*
:* SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA
:* SYNC SYNC DATA DATA DATA SYNC SYNC
:*
:* THE RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE
:* THE NEXT TWO SYNC CHARACTERS FOR SYNCHRONIZATION, THEN PASS THE REST
:* OF THE MESSAGE (7 SYNC AND 6 DATA CHARACTERS) THROUGH RXDB REGISTER.
:*****

```

```

8550
8551
8552
8553
8554
8555
8556
8557
8558
8559
8560
8561
8562
8563
8564
8565
8566
8567
8568
8569 035044
8570
8571 035044 004737 005420
8572
8573 035050 004537 007400
8574 035054 043626
8575 035056 040000
8576 035060 103003
8577 035062
8578 035062 104460
8579 035064
8580 035064 104410
8581 035066 000414
8582
8583 035070 004537 010010
8584 035074 000001
8585 035076 000007
8586 035100 004537 010010
8587 035104 000000
8588 035106 000000
8589 035110 004537 003734
8590 035114 120000
8591 035116 000142
8592
8593
8594
8595 035120 004537 007676
8596 035124 000000
8597 035126 000010
8598 035130 103003
8599 035132
8600 035132 104460
8601 035134
8602 035134 104410
8603 035136 000344
8604
8605 035140 004537 007676

```

```

:
: BGNTST
:
: T29::
:
: JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP
:
: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
: DDCMP!NOCHK!SYNCH ;SET DDCMP,NO CHECK,SYNCH=226
: NORXEN ;USE 8 BIT CHARS; LEAVE RXEN=0
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
:
: ESCAPE TST ;SKIP TO END OF TEST TRAP CSERROR
:
: JSR R5,TXCTRL ;OUTPUT 1ST SYNC CHARACTER
: TSOM ;(IGNORED BY RECEIVER)
: 7.
: JSR R5,TXCTRL ;CLEAR TSOM
: 000
: 0
: JSR R5,WRITEI ;ENABLE RECEIVER (RXEN => 1)
: VIAORB
: RXEN!TXEN!TTLOOP
:
:-----
: THE RECEIVER SHOULD IGNORE THE NEXT STRING OF CHARACTERS
:-----
:
: JSR R5,TXCHAR ;LOAD 000, TX 2ND SYNCH
: 000
: 8.
: BCC .+8. ;BR IF NO ERROR
: ERROR ;REPORT STACKED ERROR
:
: ESCAPE TST ;SKIP TO END OF TEST TRAP CSERROR
:
: JSR R5,TXCHAR ;LOAD 377, TX 000
:
: TRAP C$ESCAPE
: .WORD L10100-.

```

CVDMA.P11 10-DEC-80 09:14

TEST 29 -- BCP RX SYNC CHARACTER RECOGNITION

8606	035144	000377		377				
8607	035146	000010		8.				
8608	035150	103003		BCC	.+8.	:BR IF NO ERROR		
8609	035152			ERROR		:REPORT STACKED ERROR		
8610	035152	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
8611	035154							
8612	035154	104410					TRAP	C\$ESCAPE
8613	035156	000324					.WORD	L10100-.
8614								
8615	035160	004537	007676	JSR	R5,TXCHAR	:LOAD 125, TX 377		
8616	035164	000125		125				
8617	035166	000010		8.				
8618	035170	103003		BCC	.+8.	:BR IF NO ERROR		
8619	035172			ERROR		:REPORT STACKED ERROR		
8620	035172	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
8621	035174							
8622	035174	104410					TRAP	C\$ESCAPE
8623	035176	000304					.WORD	L10100-.
8624								
8625	035200	004537	007676	JSR	R5,TXCHAR	:TX 125, LOAD SYNCH		
8626	035204	000226		SYNCH				
8627	035206	000010		8.				
8628	035210	103003		BCC	.+8.	:BR IF NO ERROR		
8629	035212			ERROR		:REPORT STACKED ERROR		
8630	035212	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
8631	035214							
8632	035214	104410					TRAP	C\$ESCAPE
8633	035216	000264					.WORD	L10100-.
8634								
8635	035220	004537	006176	JSR	R5,CKRDA	:CHECK RECEIVE DATA AVAILABLE		
8636	035224	000000		0		: (NO DATA EXPECTED)		
8637	035226	103003		BCC	.+8.	:BR IF NO ERROR		
8638	035230			ERROR		:REPORT STACKED ERROR		
8639	035230	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
8640	035232							
8641	035232	104410					TRAP	C\$ESCAPE
8642	035234	000246					.WORD	L10100-.
8643								
8644								
8645								
8646								
8647	035236	004537	007676	JSR	R5,TXCHAR	:LOAD 2ND SYNCH, TX 1ST SYNCH		
8648	035242	000226		SYNCH				
8649	035244	000010		8.				
8650	035246	103003		BCC	.+8.	:BR IF NO ERROR		
8651	035250			ERROR		:REPORT STACKED ERROR		
8652	035250	104460		ESCAPE	TST	:SKIP TO END OF TEST	TRAP	C\$ERROR
8653	035252							
8654	035252	104410					TRAP	C\$ESCAPE
8655	035254	000226					.WORD	L10100-.
8656								
8657	035256	004537	007676	JSR	R5,TXCHAR	:LOAD 3ND SYNCH, TX 2ND SYNCH		
8658	035262	000226		SYNCH				
8659	035264	000010		8.				
8660	035266	103003		BCC	.+8.	:BR IF NO ERROR		
8661	035270			ERROR		:REPORT STACKED ERROR		

THE RECEIVER SHOULD SYNCHRONIZE ON THE NEXT TWO SYNC CHARACTERS
AND THEN READ THE REMAINING ONES.

CVDMA.P11 10-DEC-80 09:14

TEST 29 -- BCP RX SYNC CHARACTER RECOGNITION

```

8662 035270 104460
8663 035272          ESCAPE TST          ;SKIP TO END OF TEST          TRAP   C$ERROR
8664 035272 104410
8665 035274 000206          .WORD   C$ESCAPE
8666                                     L10100-.
8667 035276 004537 007676      JSR    R5,TXCHAR          ;LOAD 4TH SYNCH, TX 3RD SYNCH
8668 035302 000226      SYNCH
8669 035304 000010      8.
8670 035306 103003      BCC    .+8.              ;BR IF NO ERROR
8671 035310          ERROR          ;REPORT STACKED ERROR
8672 035310 104460          TRAP   C$ERROR
8673 035312          ESCAPE TST          ;SKIP TO END OF TEST          TRAP   C$ESCAPE
8674 035312 104410          .WORD   L10100-.
8675 035314 000166
8676
8677 035316 004537 007676      JSR    R5,TXCHAR          ;LOAD 5TH SYNCH
8678 035322 000226      SYNCH
8679 035324 000000      0
8680 035326 103003      BCC    .+8.              ;BR IF NO ERROR
8681 035330          ERROR          ;REPORT STACKED ERROR
8682 035330 104460          TRAP   C$ERROR
8683 035332          ESCAPE TST          ;SKIP TO END OF TEST          TRAP   C$ESCAPE
8684 035332 104410          .WORD   L10100-.
8685 035334 000146
8686
8687 035336 004537 011364      JSR    R5,RCV1ST         ;CLOCK AND RECEIVE 3RD SYNCH
8688 035342 000000      0
8689 035344 103003      BCC    .+8.              ;BR IF NO ERROR
8690 035346          ERROR          ;REPORT STACKED ERROR
8691 035346 104460          TRAP   C$ERROR
8692 035350          ESCAPE TST          ;SKIP TO END OF TEST          TRAP   C$ESCAPE
8693 035350 104410          .WORD   L10100-.
8694 035352 000130
8695
8696 035354 004537 010110      JSR    R5,RXCHAR          ;RECEIVE/CHECK SYNCH CHARACTER
8697 035360 000226      SYNCH
8698 035362 000000      0
8699 035364 000010      8.
8700 035366 103003      BCC    .+8.              ;BR IF NO ERROR
8701 035370          ERROR          ;REPORT STACKED ERROR
8702 035370 104460          TRAP   C$ERROR
8703 035372          ESCAPE TST          ;SKIP TO END OF TEST          TRAP   C$ESCAPE
8704 035372 104410          .WORD   L10100-.
8705 035374 000106
8706
8707 035376 012702 035504      ;-----
8708 035402 112237 035440      5$:  MOV    #TXTBL1,R2          ;SET UP TABLE POINTER
8709 035406 116237 000001 035420      MOVB   (R2)+,20$          ;SETUP EXPECTED CHARACTER
8710                                     MOVB   1(R2),10$          ;SETUP TRANSMIT CHARACTER
8711 035414 004537 007676      JSR    R5,TXCHAR          ;LOAD A CHARACTER
8712 035420 000000      10$:  000                ;** HOLE FOR NEXT TX CHARACTER
8713 035422 000000      0
8714 035424 103003      BCC    .+8.              ;BR IF NO ERROR
8715 035426          ERROR          ;REPORT STACKED ERROR
8716 035426 104460          TRAP   C$ERROR
8717 035430          ESCAPE TST          ;SKIP TO END OF TEST

```

CVDMCA.P11 10-DEC-80 09:14

TEST 29 -- BCP RX SYNC CHARACTER RECOGNITION

8718	035430	104410							
8719	035432	000050						TRAP	C\$ESCAPE
8720								.WORD	L10100-
8721	035434	004537	010110	20\$:	JSR	R5,RXCHAR			
8722	035440	000000			000				:CLOCK/RECEIVE/CHECK PREVIOUS CHARACTER
8723	035442	000000			0				:** HOLE FOR EXPECTED CHARACTER
8724	035444	000010			8.				
8725	035446	103003			BCC	+.8.			:BR IF NO ERROR
8726	035450				ERROR				:REPORT STACKED ERROR
8727	035450	104460			ESCAPE	TST		TRAP	C\$ERROR
8728	035452								:SKIP TO END OF TEST
8729	035452	104410						TRAP	C\$ESCAPE
8730	035454	000026						.WORD	L10100-
8731									
8732	035456	022702	035521		CMP	#TXEND1,R2			
8733	035462	001347			BNE	5\$			
8734					-----				
8735	035464	004537	011532		JSR	R5,ENDTRN			:SHUT DOWN TRANSMITTER, RECEIVER
8736	035470	000010			8.				
8737	035472	103003			BCC	+.8.			:BR IF NO ERROR
8738	035474				ERROR				:REPORT STACKED ERROR
8739	035474	104460			ESCAPE	TST		TRAP	C\$ERROR
8740	035476								:SKIP TO END OF TEST
8741	035476	104410						TRAP	C\$ESCAPE
8742	035500	000002						.WORD	L10100-
8743									
8744	035502			ENDTST					
8745	035502								
8746	035502	104401						L10100:	TRAP
8747									C\$ETST
8748	035504	226			-----				
8749	035505	226		TXTBL1:	.BYTE	226			:SYNCH
8750	035506	000			.BYTE	226			:SYNCH
8751	035507	377			.BYTE	000			
8752	035510	125			.BYTE	377			
8753	035511	226			.BYTE	125			:SYNCH
8754	035512	226			.BYTE	226			:SYNCH
8755	035513	252			.BYTE	226			
8756	035514	101			.BYTE	252			
8757	035515	202			.BYTE	101			
8758	035516	226			.BYTE	202			:SYNCH
8759	035517	226			.BYTE	226			:SYNCH
8760	035520	000			.BYTE	226			
8761	035521	000		TXEND1:	.BYTE	000			
8762					.EVEN	000			
8763					-----				

CVDMCA.P11 10-DEC-80 09:14

TEST 30 -- BCP RX STRIP-SYNC TEST

.SBTTL TEST 30 -- BCP RX STRIP-SYNC TEST

TEST 30 -- BCP RX STRIP-SYNC TEST

THE USYRT IS INITIALIZED WITH THE STRIP-SYNC CONTROL BIT ASSERTED. THE FOLLOWING MESSAGE IS THEN INITIATED WITHOUT ASSERTING RXE AND ONCE THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):

SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA SYNC SYNC DATA DATA DATA SYNC SYNC

THE RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE THE NEXT TWO SYNC CHARACTERS FOR SYNCHRONIZATION, IGNORE THE NEXT THREE SYNC CHARACTERS, AND PASS THE REST OF THE MESSAGE (4 SYNC AND 6 DATA CHARACTERS) THROUGH RXDB REGISTER.

8764
8765
8766
8767
8768
8769
8770
8771
8772
8773
8774
8775
8776
8777
8778
8779
8780
8781
8782
8783
8784
8785 035522
8786 035522 004737 005420
8787
8788 035526 004537 007400
8789 035532 063626
8790 035534 040000
8791 035536 103003
8792 035540
8793 035540 104460
8794 035542
8795 035542 104410
8796 035544 000474
8797
8798 035546 004537 010010
8799 035552 000001
8800 035554 000007
8801 035556 004537 010010
8802 035562 000000
8803 035564 000000
8804 035566 004537 003734
8805 035572 120000
8806 035574 000142
8807
8808
8809
8810 035576 004537 007676
8811 035602 000000
8812 035604 000010
8813 035606 103003
8814 035610
8815 035610 104460
8816 035612
8817 035612 104410
8818 035614 000424
8819

BGNTST

JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP T30::
JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!STRIPS!NOCHK!SYNCH ;SET DDCMP,NO CHECK,SYNCH=226
NORXEN ;USE 8 BIT CHARS; LEAVE RXEN=0
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP CSERROR
TRAP CSESCAPE
.WORD L10101-.
JSR R5,TXCTRL ;OUTPUT 1ST SYNC CHARACTER
TSOM ; (IGNORED BY RECEIVER)
7
JSR R5,TXCTRL ;CLEAR TSOM
000
0
JSR R5,WRITEI ;ENABLE RECEIVER (RXEN => 1)
VIAORB
RXEN!TXEN!TTLOOP

THE RECEIVER SHOULD IGNORE THE NEXT STRING OF CHARACTERS

JSR R5,TXCHAR ;LOAD 000, TX 2ND SYNCH
000
8
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR TRAP CSERROR
ESCAPE TST ;SKIP TO END OF TEST TRAP CSESCAPE
.WORD L10101-.

CVDMA.P11 10-DEC-80 09:14

TEST 30 -- BCP RX STRIP-SYNC TEST

```

8820 035616 004537 007676      JSR      R5,TXCHAR      ;LOAD 377, TX 000
8821 035622 000377              377
8822 035624 000010              8.
8823 035626 103003              BCC      .+8.          ;BR IF NO ERROR
8824 035630                      ERROR          ;REPORT STACKED ERROR
8825 035630 104460                      TRAP     CSERROR
8826 035632                      ESCAPE   TST          ;SKIP TO END OF TEST
8827 035632 104410                      TRAP     C$ESCAPE
8828 035634 000404                      .WORD   L10101-.
8829
8830 035636 004537 007676      JSR      R5,TXCHAR      ;LOAD 125, TX 377
8831 035642 000125              125
8832 035644 000010              8.
8833 035646 103003              BCC      .+8.          ;BR IF NO ERROR
8834 035650                      ERROR          ;REPORT STACKED ERROR
8835 035650 104460                      TRAP     CSERROR
8836 035652                      ESCAPE   TST          ;SKIP TO END OF TEST
8837 035652 104410                      TRAP     C$ESCAPE
8838 035654 000364                      .WORD   L10101-.
8839
8840 035656 004537 007676      JSR      R5,TXCHAR      ;LOAD SYNCH, TX 125
8841 035662 000226              SYNCH
8842 035664 000010              8.
8843 035666 103003              BCC      .+8.          ;BR IF NO ERROR
8844 035670                      ERROR          ;REPORT STACKED ERROR
8845 035670 104460                      TRAP     CSERROR
8846 035672                      ESCAPE   TST          ;SKIP TO END OF TEST
8847 035672 104410                      TRAP     C$ESCAPE
8848 035674 000344                      .WORD   L10101-.
8849
8850 035676 004537 006176      JSR      R5,CKRDA       ;CHECK RECEIVE DATA AVAILABLE
8851 035702 000000              0                ; (NO DATA EXPECTED)
8852 035704 103003              BCC      .+8.          ;BR IF NO ERROR
8853 035706                      ERROR          ;REPORT STACKED ERROR
8854 035706 104460                      TRAP     CSERROR
8855 035710                      ESCAPE   TST          ;SKIP TO END OF TEST
8856 035710 104410                      TRAP     C$ESCAPE
8857 035712 000326                      .WORD   L10101-.
8858
8859
8860
8861
8862 035714 004537 007676      JSR      R5,TXCHAR      ;LOAD 2ND SYNCH, TX 1ST SYNCH
8863 035720 000226              SYNCH
8864 035722 000010              8.
8865 035724 103003              BCC      .+8.          ;BR IF NO ERROR
8866 035726                      ERROR          ;REPORT STACKED ERROR
8867 035726 104460                      TRAP     CSERROR
8868 035730                      ESCAPE   TST          ;SKIP TO END OF TEST
8869 035730 104410                      TRAP     C$ESCAPE
8870 035732 000306                      .WORD   L10101-.
8871
8872 035734 004537 007676      JSR      R5,TXCHAR      ;LOAD 3ND SYNCH, TX 2ND SYNCH
8873 035740 000226              SYNCH
8874 035742 000010              8.
8875 035744 103003              BCC      .+8.          ;BR IF NO ERROR

```

```

: THE RECEIVER SHOULD SYNCHRONIZE ON THE NEXT TWO SYNC CHARACTERS,
: STRIP THE NEXT THREE, AND THEN READ THE REMAINING ONES.

```


CVDMA.P11 10-DEC-80 09:14

TEST 30 -- BCP RX STRIP~SYNC TEST

```

8932 036074 004537 011364      JSR    R5,RCV1ST      ;CLOCK AND RECEIVE DATA1
8933 036100 000000              0
8934 036102 103003      BCC    .+8.           ;BR IF NO ERROR
8935 036104              ERROR                ;REPORT STACKED ERROR
8936 036104 104460              ESCAPE TST            ;SKIP TO END OF TEST
8937 036106              TRAP    C$ERROR
8938 036106 104410              TRAP    C$ESCAPE
8939 036110 000130              .WORD  L10101-.
8940
8941 036112 004537 010110      JSR    R5,RXCHAR      ;RECEIVE/CHECK DATA1 (252)
8942 036116 000252              252
8943 036120 000000              0
8944 036122 000010              8.
8945 036124 103003      BCC    .+8.           ;BR IF NO ERROR
8946 036126              ERROR                ;REPORT STACKED ERROR
8947 036126 104460              ESCAPE TST            ;SKIP TO END OF TEST
8948 036130              TRAP    C$ERROR
8949 036130 104410              TRAP    C$ESCAPE
8950 036132 000106              .WORD  L10101-.
8951
-----
8952 036134 012702 036242      ;
8953 036140 112237 036176      5$:  MOV    #TXTBL2,R2    ;SET UP TABLE POINTER
8954 036144 116237 000001 036156  MOVB   (R2)+,20$        ;SETUP EXPECTED CHARACTER
8955                                MOVB   1(R2),10$        ;SETUP TRANSMIT CHARACTER
8956 036152 004537 007676      10$: JSR    R5,TXCHAR      ;LOAD A CHARACTER
8957 036156 000000              000                    ;** HOLE FOR NEXT TX CHARACTER
8958 036160 000000              0
8959 036162 103003      BCC    .+8.           ;BR IF NO ERROR
8960 036164              ERROR                ;REPORT STACKED ERROR
8961 036164 104460              ESCAPE TST            ;SKIP TO END OF TEST
8962 036166              TRAP    C$ERROR
8963 036166 104410              TRAP    C$ESCAPE
8964 036170 000050              .WORD  L10101-.
8965
8966 036172 004537 010110      20$: JSR    R5,RXCHAR      ;CLOCK/RECEIVE/CHECK PREVIOUS CHARACTER
8967 036176 000000              000                    ;** HOLE FOR EXPECTED CHARACTER
8968 036200 000000              0
8969 036202 000010              8.
8970 036204 103003      BCC    .+8.           ;BR IF NO ERROR
8971 036206              ERROR                ;REPORT STACKED ERROR
8972 036206 104460              ESCAPE TST            ;SKIP TO END OF TEST
8973 036210              TRAP    C$ERROR
8974 036210 104410              TRAP    C$ESCAPE
8975 036212 000026              .WORD  L10101-.
8976
8977 036214 022702 036254      CMP    #TXEND2,R2
8978 036220 001347      BNE    5$
8979
-----
8980 036222 004537 011532      JSR    R5,ENDTRN      ;SHUT DOWN TRANSMITTER, RECEIVER
8981 036226 000010              8.
8982 036230 103003      BCC    .+8.           ;BR IF NO ERROR
8983 036232              ERROR                ;REPORT STACKED ERROR
8984 036232 104460              ESCAPE TST            ;SKIP TO END OF TEST
8985 036234              TRAP    C$ERROR
8986 036234 104410              TRAP    C$ESCAPE
8987 036236 000002              .WORD  L10101-.

```


CVDMCA.P11 10-DEC-80 09:14

TEST 30 -- BCP RX STRIP-SYNC TEST

ENDTST

L10101: TRAP CSETST

8988	036240	
8989	036240	
8990	036240	104401
8991		
8992	036242	347
8993	036243	030
8994	036244	226
8995	036245	226
8996	036246	252
8997	036247	101
8998	036250	202
8999	036251	226
9000	036252	226
9001	036253	000
9002	036254	000
9003		036256
9004		

```

-----
TXBL2: .BYTE 347
        .BYTE 030
        .BYTE 226      :SYNCH
        .BYTE 226      :SYNCH
        .BYTE 252
        .BYTE 101
        .BYTE 202
        .BYTE 226      :SYNCH
        .BYTE 226      :SYNCH
        .BYTE 000
TXEND2: .BYTE 000
        .EVEN
-----

```

CVDMCA.P11 10-DEC-80 09:14

TEST 31 -- BCP RX LOST RXE TEST

.SBTTL TEST 31 -- BCP RX LOST RXE TEST

```

:*****
:*
:* TEST 31 -- BCP RX LOST RXE TEST
:* THE USYRT IS INITIALIZED (CRC16,STRIPS,BCP MODE) AND A MESSAGE IS STARTED.
:* WHILE IN THE MIDDLE OF TEXT, RXE IS DROPPED AND THE REACTION OF THE
:* RECEIVER IS MONITORED.
:*****

```

```

9005
9006
9007
9008
9009
9010
9011
9012
9013
9014
9015
9016
9017
9018
9019 036256
9020 036256 004737 005420
9021
9022 036262 004537 007400
9023 036266 065626
9024 036270 000000
9025 036272 103003
9026 036274
9027 036274 104460
9028 036276
9029 036276 104410
9030 036300 000334
9031
9032 036302 004537 007676
9033 036306 000226
9034 036310 000007
9035 036312 103003
9036 036314
9037 036314 104460
9038 036316
9039 036316 104410
9040 036320 000314
9041
9042 036322 004537 010010
9043 036326 000000
9044 036330 000000
9045 036332 004537 007676
9046 036336 000000
9047 036340 000010
9048 036342 103003
9049 036344
9050 036344 104460
9051 036346
9052 036346 104410
9053 036350 000264
9054
9055 036352 004537 007676
9056 036356 000125
9057 036360 000010
9058 036362 103003
9059 036364
9060 036364 104460

```

```

: BGNTST
:
: JSR PC,INIDMV ;INIT DMV-11, ENTER M-LOOP T31::
:
: JSR R5,INITRN ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP, IDLE, CRC-16, SYNCH=226
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 L10102-.
:
: JSR R5, TXCHAR ;LOAD 2ND SYNCH, TX 1ST SYNCH
SYNCH
7.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
: TRAP C$ERROR
ESCAPE TST ;SKIP TO END OF TEST
: TRAP C$ESCAPE
: .WORD L10102-.
:
: JSR R5, TXCTRL ;CLEAR TSOM
000
0
: JSR R5, TXCHAR ;LOAD 000, TX 2ND SYNCH
000
8.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
: TRAP C$ERROR
ESCAPE TST ;SKIP TO END OF TEST
: TRAP C$ESCAPE
: .WORD L10102-.
:
: JSR R5, TXCHAR ;LOAD 125, TX 000
125
8.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
: TRAP C$ERROR

```

CVDMA.P11 10-DEC-80 09:14

TEST 31 -- BCP RX LOST RXE TEST

9061	036366			ESCAPE TST	;SKIP TO END OF TEST		
9062	036366	104410				TRAP	C\$ESCAPE
9063	036370	000244				.WORD	L10102-.
9064							
9065	036372	004537	007676	JSR R5,TXCHAR	;LOAD 252, TX 125		
9066	036376	000252		252			
9067	036400	000010		8.			
9068	036402	103003		BCC	.+8.	;BR IF NO ERROR	
9069	036404			ERROR		;REPORT STACKED ERROR	
9070	036404	104460				TRAP	C\$ERROR
9071	036406			ESCAPE TST	;SKIP TO END OF TEST		
9072	036406	104410				TRAP	C\$ESCAPE
9073	036410	000224				.WORD	L10102-.
9074							
9075	036412	004537	007676	JSR R5,TXCHAR	;LOAD 377, TX 252		
9076	036416	000377		377			
9077	036420	000010		8.			
9078	036422	103003		BCC	.+8.	;BR IF NO ERROR	
9079	036424			ERROR		;REPORT STACKED ERROR	
9080	036424	104460				TRAP	C\$ERROR
9081	036426			ESCAPE TST	;SKIP TO END OF TEST		
9082	036426	104410				TRAP	C\$ESCAPE
9083	036430	000204				.WORD	L10102-.
9084							
9085	036432	004537	007676	JSR R5,TXCHAR	;LOAD 000		
9086	036436	000000		000			
9087	036440	000000		0			
9088	036442	103003		BCC	.+8.	;BR IF NO ERROR	
9089	036444			ERROR		;REPORT STACKED ERROR	
9090	036444	104460				TRAP	C\$ERROR
9091	036446			ESCAPE TST	;SKIP TO END OF TEST		
9092	036446	104410				TRAP	C\$ESCAPE
9093	036450	000164				.WORD	L10102-.
9094							
9095	036452	004537	011364	JSR R5,RCV1ST	;CLOCK AND RCV 000		
9096	036456	000000		0			
9097	036460	103003		BCC	.+8.	;BR IF NO ERROR	
9098	036462			ERROR		;REPORT STACKED ERROR	
9099	036462	104460				TRAP	C\$ERROR
9100	036464			ESCAPE TST	;SKIP TO END OF TEST		
9101	036464	104410				TRAP	C\$ESCAPE
9102	036466	000146				.WORD	L10102-.
9103							
9104	036470	004537	010110	JSR R5,RXCHAR	;READ & CHK 000, RCV 125		
9105	036474	000000		000			
9106	036476	000000		0			
9107	036500	000010		8.			
9108	036502	103003		BCC	.+8.	;BR IF NO ERROR	
9109	036504			ERROR		;REPORT STACKED ERROR	
9110	036504	104460				TRAP	C\$ERROR
9111	036506			ESCAPE TST	;SKIP TO END OF TEST		
9112	036506	104410				TRAP	C\$ESCAPE
9113	036510	000124				.WORD	L10102-.
9114							
9115							
9116							

```

: TX AND RX NOW SYNC'D. ONE CHARACTER HAS BEEN READ/VERIFIED....
: NOW CLEAR RXEN (RECEIVER ENABLE).

```

CVDMCA.P11 10-DEC-80 09:14

TEST 31 -- BCP RX LOST RXE TEST

```

9117
9118 036512 004537 003476
9119 036516 120000
9120 036520 036546
9121 036522 103003
9122 036524
9123 036524 104460
9124 036526
9125 036526 104410
9126 036530 000104
9127
9128 036532 042737 000100 036546
9129
9130 036540 004537 003734
9131 036544 120000
9132 036546 000000
9133 036550 103003
9134 036552
9135 036552 104460
9136 036554
9137 036554 104410
9138 036556 000056
9139
9140
9141
9142
9143 036560 004537 005432
9144 036564 000104
9145 036566 103003
9146 036570
9147 036570 104460
9148 036572
9149 036572 104410
9150 036574 000040
9151
9152 036576 004537 007676
9153 036602 000303
9154 036604 100010
9155 036606 103003
9156 036610
9157 036610 104460
9158 036612
9159 036612 104410
9160 036614 000020
9161
9162 036616 004537 005432
9163 036622 000114
9164 036624 103003
9165 036626
9166 036626 104460
9167 036630
9168 036630 104410
9169 036632 000002
9170 036634
9171 036634
9172 036634 104401

```

```

-----
JSR R5,READ ;READ VALUE OF VIAORB REGISTER
VIAORB ; AND PUT IT IN LOCATION PROVIDED
CGORB ;BELOW (CGORB).
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT ERROR
TRAP C$ERROR
ESCAPE TST TRAP C$ESCAPE
WORD L10102-.
BIC #RXEN,CGORB ;CLEAR RXEN BIT OF ORB STATUS WORD
JSR R5,WRITEI ;WRITE NEW STATUS WORD (W/RXEN=0)
VIAORB ;BACK INTO VIAORB
CGORB: .WORD 0 ;ACTUAL DATA WRITTEN TO VIAORB
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT ERROR
TRAP C$ERROR
ESCAPE TST TRAP C$ESCAPE
WORD L10102-.
-----
; RXEN NOW CLEARED. CHECK USYRT STATUS, TX ANOTHER CHARACTER, AND
; CHECK USYRT STATUS AGAIN.
-----
JSR R5,CKUSTS ;CHK USYRT STATUS FOR PROPER STATE
TXACT!TBMT ;TXACT = 1, TBMT = 1
BCC .+8. ;BR IF NO ERROR
ERROR
TRAP C$ERROR
ESCAPE TST TRAP C$ESCAPE
WORD L10102-.
JSR R5,TXCHAR ;LOAD/TRANSMIT 303
303
NCTBMT*256.!8.
BCC .+8. ;BR IF NO ERROR
ERROR ;REPORT STACKED ERROR
TRAP C$ERROR
ESCAPE TST ;SKIP TO END OF TEST
TRAP C$ESCAPE
WORD L10102-.
JSR R5,CKUSTS ;CHK USYRT STATUS FOR PROPER STATE
TXACT!TSO!TBMT ;TXACT = 1
BCC .+8. ;BR IF NO ERROR
ERROR
TRAP C$ERROR
ESCAPE TST TRAP C$ESCAPE
WORD L10102-.
ENDTST
L10102: TRAP C$SETST

```

CVDMCA.P11 10-DEC-80 09:14

HARDWARE PARAMETER CODING SECTION

.SBTTL HARDWARE PARAMETER CODING SECTION

```

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

```

```

9173
9174
9175
9176
9177
9178
9179
9180
9181
9182
9183
9184
9185
9186 036636
9187 036636 000041
9188 036640
9189
9190 036640
9191 036640 000031
9192 036642 036742
9193 036644 160020
9194 036646 177776
9195 036650
9196 036650 001031
9197 036652 036770
9198 036654 000000
9199 036656 000674
9200 036660
9201 036660 002032
9202 036662 037021
9203 036664 007000
9204 036666 000004
9205 036670 000007
9206 036672
9207 036672 003032
9208 036674 037052
9209 036676 000377
9210 036700 000000
9211 036702 000377
9212 036704
9213 036704 004032
9214 036706 037113
9215 036710 000377
9216 036712 000000
9217 036714 000377
9218 036716
9219 036716 005032
9220 036720 037155
9221 036722 000007
9222 036724 000000
9223 036726 000002
9224 036730
9225 036730 007032
9226 036732 037240
9227 036734 000017
9228 036736 000000

```

```

          BGNHRD
          LSHARD:: .WORD L10103-LSHARD/2
          GPRMA  ADDRES,0,0,160020,177776,YES
          GPRMA  VECTOR,2,0,0,674,YES
          GPRMD  PRIRTY,4,0,7000,4,7,YES
          GPRMD  SW1.M,6,0,377,0,377,YES
          GPRMD  SW2.M,10,0,377,0,377,YES
          GPRMD  BDTY.M,12,0,7,0,2,YES
          GPRMD  BR.M,16,0,17,0,1,YES
          .WORD  TSCODE
          .WORD  ADDRES
          .WORD  TSLOLIM
          .WORD  TSHILIM
          .WORD  TSCODE
          .WORD  VECTOR
          .WORD  TSLOLIM
          .WORD  TSHILIM
          .WORD  TSCODE
          .WORD  PRIRTY
          .WORD  7000
          .WORD  TSLOLIM
          .WORD  TSHILIM
          .WORD  TSCODE
          .WORD  SW1.M
          .WORD  377
          .WORD  TSLOLIM
          .WORD  TSHILIM
          .WORD  TSCODE
          .WORD  SW2.M
          .WORD  377
          .WORD  TSLOLIM
          .WORD  TSHILIM
          .WORD  TSCODE
          .WORD  BDTY.M
          .WORD  7
          .WORD  TSLOLIM
          .WORD  TSHILIM
          .WORD  TSCODE
          .WORD  BR.M
          .WORD  17
          .WORD  TSLOLIM

```

CVDMCA.P11 10-DEC-80 09:14

HARDWARE PARAMETER CODING SECTION

9229 036740 000001

.WORD TSHILIM

9230

9231 036742

ENDHRD

9232

9233 036742

L10103: .EVEN

9234

9235

036742	042504	044526	042503
036770	042504	044526	042503
037021	104	053105	041511
037052	053523	052111	044103
037113	123	044527	041524
037155	102	040517	042122
037240	040502	042125	051040

```

.NLIST BEX
ADDRESS: .ASCIZ /DEVICE CSR ADDRESS : /
VECTOR: .ASCIZ /DEVICE VECTOR ADDRESS : /
PRIRTY: .ASCIZ /DEVICE PRIORITY LEVEL : /
SW1.M: .ASCIZ /SWITCH PACK # 1 (BOOT ADDRESS): /
SW2.M: .ASCIZ /SWITCH PACK # 2 (DDCMP ADDRESS): /
BDTY.M: .ASCIZ /BOARD TYPE (0=M8064, 1=M8053-V.35, 2=M8053-EIA) : /
BR.M: .ASCIZ /BAUD RATE (0=LOW (19.2K), 1=HIGH (56K)): /
.LIST BEX
.EVEN

```

9236

CVDMCA.P11

10-DEC-80 09:14

SOFTWARE PARAMETER CODING SECTION

.SBTTL SOFTWARE PARAMETER CODING SECTION

9237
9238
9239
9240
9241
9242
9243
9244
9245
9246
9247
9248
9249
9250
9251
9252
9253
9254
9255

037312
037312 000000
037314
037314
037314

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

BGNSFT

ENDSFT

.WORD L10104-LSSOFT/2
LSSOFT::
L10104: .EVEN

CVDMCA.P11 10-DEC-80 09:14

***** PATCH AREA FOR DEBUG *****

.SBTTL ***** PATCH AREA FOR DEBUG *****

PATCH:

.=.+100
NOP
NOP
NOP

.SBTTL 'ENDMOD' STATEMENT

ENDMOD

.SBTTL 'LASTAD' STATEMENT & END OF PROGRAM
LASTAD

.EVEN
.WORD 0
.WORD 0

LSLAST::

.END

9256		
9257		
9258	037314	
9259		037414
9260	037414	000240
9261	037416	000240
9262	037420	000240
9263		
9264		
9265		
9266		
9267	037422	
9268		
9269		
9270	037422	
9271		
9272	037422	000000
9273	037424	000000
9274	037426	
9275		
9276		000001

CVDMA.P11

10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

ERR4	017030	G	2440	2484	2531	2602	4098#	4806									
ERR5	017154	G	4137#	4825													
ERR5\$	020160		4129	4156	4370#												
ERR7A	017252	G	2701	4164#	5389												
EVL	= 000004	G	1559#														
EVRC	= 002400		1701#														
EXADD	= 000020		1713#														
EXCON	= 000010		1714#														
EXECUT	= 000005		1606#														
ESEND	= 002100		1297#														
ESLOAD	= 000035		1297#	1392													
FLGCA1	= 000002		1942#														
FLGCA2	= 000001		1943#														
FLGCB1	= 000020		1939#														
FLGCB2	= 000010		1940#														
FLGIRQ	= 000200		1931#														
FLGSR	= 000004		1941#														
FLGT1	= 000100		1937#														
FLGT2	= 000040		1938#														
FMT10	012252		4064#	4151													
FMT10A	012326		4064#														
FMT11	012347		4064#	4384	4401												
FMT15	012366		4064#	4170													
FMT15A	012420		4064#														
FMT19	012472		4064#														
FMT2	011650		4064#	4141													
FMT21	012523		4064#	4203	4248	4293	4328	4973									
FMT22	012533		4064#	4209	4254	4299											
FMT23	012555		4064#	4229	4274	4338											
FMT24	012634		4064#	4423	4500												
FMT25	012647		4064#	4433	4467	4510											
FMT26	012677		4064#	4450	4484	4527											
FMT27	012732		4064#	4219	4264	4309											
FMT29	012741		4064#	4440	4457	4474	4517										
FMT3	011705		4064#	4180													
FMT30	012746		4064#														
FMT31	013003		4064#														
FMT32	013051		4064#														
FMT4	011771		4064#	4374													
FMT4A	012027		4064#														
FMT4B	012062		4064#	4391													
FMT4C	012067		4064#														
FMT5	012122		4064#	4106													
FMT5A	012165		4064#	4122													
FRSTIM	002444		2080#	4563	4572*												
FSAU	= 000015		1297#	4750	4752												
FSAUTO	= 000020		1297#	4670	4706												
F\$BGN	= 000040		1297#	1300	4098	4137	4164	4199	4244	4289	4324	4543	4556	4670	4720		
			4734	4750	4770	4777	4786	4808	4828	4851	4855	4881	4885	4896	4913		
			4917	4921	4969	5008	5042	5048	5073	5091	5096	5119	5157	5168	5185		
			5191	5199	5210	5214	5220	5237	5243	5251	5262	5266	5272	5291	5297		
			5305	5316	5320	5326	5351	5357	5365	5391	5395	5401	5417	5423	5431		
			5442	5446	5452	5469	5475	5494	5511	5517	5536	5564	5596	5622	5649		
			5679	5690	5711	5722	5730	5734	5766	5777	5787	5797	5807	5816	5827		
			5835	5838	5865	5875	5884	5894	5912	5923	5938	5950	5962	5977	5988		
			5998	6014	6024	6039	6054	6066	6069	6093	6103	6112	6122	6140	6151		

CVDMCA.P11

10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

6166	6178	6190	6205	6220	6230	6243	6251	6254	6274	6285	6303	6313
6329	6340	6351	6354	6377	6387	6409	6432	6449	6460	6469	6472	6504
6510	6519	6539	6557	6564	6570	6579	6599	6617	6624	6630	6639	6659
6677	6684	6687	6723	6733	6742	6752	6770	6784	6794	6797	6820	6830
6839	6849	6867	6881	6891	6894	6917	6927	6936	6946	6964	6978	6988
6991	7011	7021	7039	7049	7059	7069	7078	7089	7099	7110	7120	7131
7142	7153	7156	7186	7192	7201	7217	7227	7240	7253	7273	7284	7299
7313	7324	7327	7333	7342	7358	7368	7381	7394	7414	7425	7440	7454
7465	7468	7474	7483	7499	7509	7522	7535	7555	7566	7581	7595	7606
7609	7615	7624	7640	7650	7663	7676	7696	7707	7722	7736	7747	7750
7756	7765	7781	7791	7804	7817	7837	7848	7863	7877	7888	7891	7897
7906	7922	7932	7945	7958	7978	7989	8004	8018	8029	8032	8035	8061
8100	8119	8128	8142	8151	8170	8184	8194	8197	8220	8230	8252	8264
8278	8289	8303	8322	8333	8344	8353	8356	8399	8402	8413	8432	8444
8453	8456	8460	8471	8491	8494	8498	8509	8542	8545	8548	8569	8580
8602	8612	8627	8632	8641	8654	8664	8674	8684	8693	8704	8718	8729
8741	8745	8785	8795	8817	8827	8837	8847	8856	8869	8879	8889	8899
8909	8919	8929	8938	8949	8963	8974	8986	8989	9019	9029	9039	9052
9062	9072	9082	9092	9101	9112	9125	9137	9149	9159	9168	9171	9187
9250	9268											
1297#	4720	4724										
1297#	4734	4739										
1297#	1300	4133	4160	4194	4238	4283	4318	4347	4653	4708	4726	4741
4754	4770	4777	4786	4808	4828	4830	4851	4855	4881	4883	4885	4896
4913	4915	4917	4919	4967	4982	5008	5042	5048	5050	5073	5091	5096
5098	5119	5157	5168	5170	5185	5191	5199	5210	5214	5216	5220	5222
5237	5243	5251	5262	5266	5268	5272	5274	5291	5297	5305	5316	5320
5322	5326	5328	5351	5357	5365	5391	5395	5397	5401	5403	5417	5423
5431	5442	5446	5448	5452	5454	5469	5475	5494	5496	5511	5517	5536
5538	5564	5596	5622	5649	5651	5679	5690	5711	5722	5730	5734	5736
5766	5777	5787	5797	5807	5816	5827	5835	5838	5840	5865	5875	5884
5894	5912	5923	5938	5950	5962	5977	5988	5998	6014	6024	6039	6054
6066	6069	6071	6093	6103	6112	6122	6140	6151	6166	6178	6190	6205
6220	6230	6243	6251	6254	6256	6274	6285	6303	6313	6329	6340	6351
6354	6356	6377	6387	6409	6432	6449	6460	6469	6472	6474	6504	6510
6519	6539	6557	6564	6566	6570	6579	6599	6617	6624	6626	6630	6639
6659	6677	6684	6686	6687	6689	6723	6733	6742	6752	6770	6784	6794
6797	6799	6820	6830	6839	6849	6867	6881	6891	6894	6896	6917	6927
6936	6946	6964	6978	6988	6991	6993	7011	7021	7039	7049	7059	7069
7078	7089	7099	7110	7120	7131	7142	7153	7156	7158	7186	7192	7201
7217	7227	7240	7253	7273	7284	7299	7313	7324	7327	7329	7333	7342
7358	7368	7381	7394	7414	7425	7440	7454	7465	7468	7470	7474	7483
7499	7509	7522	7535	7555	7566	7581	7595	7606	7609	7611	7615	7624
7640	7650	7663	7676	7696	7707	7722	7736	7747	7750	7752	7756	7765
7781	7791	7804	7817	7837	7848	7863	7877	7888	7891	7893	7897	7906
7922	7932	7945	7958	7978	7989	8004	8018	8029	8032	8034	8035	8037
8061	8100	8119	8128	8142	8151	8170	8184	8194	8197	8199	8220	8230
8252	8264	8278	8289	8303	8322	8333	8344	8353	8356	8358	8399	8402
8413	8432	8444	8453	8456	8458	8460	8471	8491	8494	8496	8498	8509
8542	8545	8547	8548	8550	8569	8580	8602	8612	8622	8632	8641	8654
8664	8674	8684	8693	8704	8718	8729	8741	8745	8747	8785	8795	8817
8827	8837	8847	8856	8869	8879	8889	8899	8909	8919	8929	8938	8949
8963	8974	8986	8989	8991	9019	9029	9039	9052	9062	9072	9082	9092
9101	9112	9125	9137	9149	9159	9168	9171	9173	9234	9256	9268	
1297#	9187	9232										
1297#	1465	1481										

F\$CLEA= 000007
F\$DU = 000016
F\$END = 000041

F\$HARD= 000004
F\$HW = 000013

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

L\$LAST	037426	G	1350	9274#				
L\$LOAD	002100	G	1391#					
L\$LUN	002074	G	1387#					
L\$PREV	002050	G	1367#					
L\$NAME	002000	G	1324#					
L\$PRIO	002042	G	1361#					
L\$PROT	021072	G	1402	4543#				
L\$PRT	002112	G	1401#					
L\$REPP	002062	G	1377#					
L\$REV	002010	G	1333#					
L\$SOFT	037314	G	9250	9251#				
L\$SPC	002056	G	1373#					
L\$SPCP	002020	G	1343#					
L\$SPTP	002024	G	1347#					
L\$STA	002030	G	1351#					
L\$SW	002246	G	1490	1491#				
L\$TEST	002114	G	1403#					
L\$TIML	002014	G	1339#					
L\$UNIT	002012	G	1337#					
L10000	002244		1465	1481#				
L10001	002246		1490	1495#				
L10002	017152		4131#					
L10003	017250		4158#					
L10004	017370		4192#					
L10005	017544		4236#					
L10006	017720		4281#					
L10007	020034		4316#					
L10010	020132		4345#					
L10012	021412		4651#					
L10013	021530		4706#					
L10014	021540		4724#					
L10015	021544		4739#					
L10016	021546		4752#					
L10017	021726		4778	4787	4809	4828#		
L10020	022114		4917#					
L10021	022012		4881#					
L10022	022112		4897	4913#				
L10023	022306		4965#					
L10024	022344		4980#					
L10025	023112		5043	5048#				
L10026	023176		5092	5096#				
L10027	023422		5158	5168#				
L10030	023514		5192	5211	5220#			
L10031	023510		5214#					
L10032	023606		5244	5263	5272#			
L10033	023602		5266#					
L10034	023700		5298	5317	5326#			
L10035	023674		5320#					
L10036	024030		5358	5392	5401#			
L10037	024024		5395#					
L10040	024122		5424	5443	5452#			
L10041	024116		5446#					
L10042	024176		5476	5494#				
L10043	024252		5518	5536#				
L10044	024574		5597	5623	5649#			
L10045	024726		5691	5712	5723	5731	5734#	

CVDMA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

L10046	025110	5778	5788	5798	5808	5817	5828	5836	5838#					
L10047	025616	5876	5885	5895	5913	5924	5939	5951	5963	5978	5989	5999	6015	6025
		6040	6055	6067	6069#									
L10050	026216	6104	6113	6123	6141	6152	6167	6179	6191	6206	6221	6231	6244	6252
		6254#												
L10051	026424	6286	6304	6314	6330	6341	6352	6354#						
L10052	026666	6388	6410	6433	6450	6461	6470	6472#						
L10053	027400	6687#												
L10054	027056	6520	6540	6558	6564#									
L10055	027226	6580	6600	6618	6624#									
L10056	027376	6640	6660	6678	6684#									
L10057	027606	6734	6743	6753	6771	6785	5795	6797#						
L10060	030002	6831	6840	6850	6868	6882	6892	6894#						
L10061	030176	6928	6937	6947	6965	6979	6989	6991#						
L10062	030554	7022	7040	7050	7060	7070	7079	7090	7100	7111	7121	7132	7143	7154
		7156#												
L10063	033466	7455	7466	7596	7607	7737	7748	7878	7889	8019	8030	8035#		
L10064	031146	7202	7218	7228	7241	7254	7274	7285	7300	7314	7325	7327#		
L10065	031534	7343	7359	7369	7382	7395	7415	7426	7441	7468#				
L10066	032122	7484	7500	7510	7523	7536	7556	7567	7582	7609#				
L10067	032510	7625	7641	7651	7664	7677	7697	7708	7723	7750#				
L10070	033076	7766	7782	7792	7805	7818	7838	7849	7864	7891#				
L10071	033464	7907	7923	7933	7946	7959	7979	7990	8005	8032#				
L10072	034054	8101	8120	8129	8143	8152	8171	8185	8195	8197#				
L10073	034424	8231	8253	8265	8279	8290	8304	8323	8334	8345	8354	8356#		
L10074	035042	8445	8492	8548#										
L10075	034600	8414	8433	8454	8456#									
L10076	034702	8472	8494#											
L10077	035040	8510	8543	8545#										
L10100	035502	8581	8603	8613	8623	8633	8642	8655	8665	8675	8685	8694	8705	8719
		8730	8742	8745#										
L10101	036240	8796	8818	8828	8838	8848	8857	8870	8880	8890	8900	8910	8920	8930
		8939	8950	8964	8975	8987	8989#							
L10102	036634	9030	9040	9053	9063	9073	9083	9093	9102	9113	9126	9138	9150	9160
		9169	9171#											
L10103	036742	9187	9233#											
L10104	037314	9250	9255#											
MCLR =	000100	1590#	1592	2424	2434									
MDMRDY=	000040	1760#												
MLWRI	003744	2561	2586#											
MPCSR	002472	2095#	4201	4246	4291	4326	4629	4971						
MPIVEC	002532	2119#	4637*											
MPOVEC	002534	2120#	4638*	4639*										
MPRIOR	002536	2121#	4640*											
MRDY =	000200	1597#	2431	2474	2521	2593	4796							
MREQ =	000001	1591#	1592	2424	2434									
MSTCLR	003374	2424#	3047	4772										
NCRACT=	020000	2005#	7137	7148	7308	7319	7449	7460	7590	7601	7731	7742	7872	7883
		8013	8024											
NCTBMT=	000200	2001#	3644	6404	8247	8311	9154							
NEWLIN	011645	4064#	4959											
NEWST	021244	4593	4608#											
NFCRDA=	040000	2004#	7148	7319	7460	7601	7742	7883	8024					
NOCHK =	003400	1700#	5684	5771	5869	6097	6279	6381	6513	6573	6633	6727	6824	6921
		7015	8224	8407	8465	8503	8574	8789						
NOCRDA=	100000	2003#	6427	6444	6455	6552	6612	6672	7268	7279	7294	7308	7319	7409

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

	7420	7435	7550	7561	7576	7691	7702	7717	7832	7843	7858	7973	7984
NLOOP= 001000	7999	8273	8284	8298	8317	8328	8339						
NORXEN= 040000	1999#												
NULCLK= 000200	1998#	5870	6098	6728	6825	6922	8575	8790					
OVRC = 002000	1742#	5579	5585	5605	5631	5637							
OSAPTS= 000000	1702#												
OSAU = 000001	1297#	1351											
OSBGNR= 000000	1297#	1321#	1383										
OSBGNS= 000000	1297#	1377											
OSDU = 000001	1297#	1343											
OSERRT= 000001	1297#	1321#	1385										
OSGNSW= 000000	1297#	1321#	1393										
OSPOIN= 000001	1297#	1347											
OSSETU= 000000	1297#	1321#	1409										
PALENB= 000001	1297#	1337	9272										
PATCH 037314	1907#												
PATE 002652	9258#												
PATF 002662	2175#	5011	5076										
PATG 002672	2186#	2955	5022	5121									
PATH 002721	2197#	5124	5194	5195									
PATI 002752	2223#	5195	5246	5247	5426	5427	5478	5479	5520	5521			
PATJ 003013	2251#	5247	5300	5301	5427	5479	5521						
PATK 003023	2287#	5129	5301										
PATL 003044	2298#	5360	5361										
PATQ 003065	2318#	5361	5362										
PATQB 003075	2338#												
PBLENB= 000002	2348#												
PCR = 120407	1906#												
PCSARH= 120405	1708#	3557	5310	7231	7372	7513	7654	7795	7936				
PCSARL= 120404	1681#	3551	5204	6004	6211	8087							
PNT = 001000 G	1674#	3547	5256	8084									
PRESET= 000001	1566#												
PRI = 002000 G	1749#	2949	3541	8078									
PRIOR 002416	1567#												
PRIRTY 037021	2068#												
PRI00 = 000000 G	9202	9235#											
PRI01 = 000040 G	1555#												
PRI02 = 000100 G	1554#												
PRI03 = 000140 G	1553#												
PRI04 = 000200 G	1552#												
PRI05 = 000240 G	1551#												
PRI06 = 000300 G	1550#												
PRI07 = 000340 G	1549#												
PROTO = 000100	1548#												
PSTACK 002414	1686#	6005											
RABGA = 000004	2067#	4558*											
RAMADR= 001000	1631#	3831	3833	3843									
RCVBUF 003106	1993#												
RCVDAT= 000002	2362#												
RCV1ST 011364	1764#												
RDA = 000200	3952#	5705	5810	6323	7072	8687	8932	9095					
RDSRH = 120401	1724#	3238	3249	3973									
RDSRL = 120400	1625#	3322	3369	3731	5526	5530							
READ 003476	1619#	3734	5484	5488									
READI 003610	2466#	2687	2729	2734	2759	5375	9118						
	2512#	2816	2903	2957	3062	3098	3143	3188	3233	3278	3321	3368	3435

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

	3591	3730	3733	3744	3970	4857	4860	4889	5024	5140	5487	5529	5576
	5602	5628	5926	5965	6042	6154	6193						
REDBYT 002432	2075#												
REDDAT 002572	2140#	5122	5130*	5133*	5143	5147							
REDLOC= 000001	1602#	2467	2478	2514	2525								
REDPAG= 000003	1604#												
REGNUM 002412	2066#	2963*	2964*	3068*	3097*	3142*	3187*	3232*	3277*	3320*	3367*	3434*	3749*
	3758*	3776*	3781*	4139	4165	4214	4259	4304	4819*	5029*	5030*	5145*	5196*
	5248*	5302*	5383*	5428*	5584*	5610*	5636*						
	2143#	5575*	5581*	5601*	5607*	5627*	5633*						
REG0 002602	2144#												
REG1 002604	2145#												
REG2 002606	2146#												
REG3 002610	2147#												
REG4 002612	2148#												
REG5 002614	2149#												
REG6 002616	2150#												
REG7 002620	1632#	3397	3408	3854	3856	3866							
REOM = 000002	1642#	3782	7318	7459	7600	7730	7871	8012					
RERCHK= 000001	1628#	3785	3787	3797									
RERR = 000200	2074#												
RETADR 002430	1758#												
RING = 000200	1630#	3326	3337	3808	3810	3820							
ROR = 000010	1727#	3283	3294										
RSA = 000020	1633#	3373	3384	3877	3879	3889							
RSOM = 000001	2943#	5086	5725	5830	6246	8486							
RSTCHK 005126	1746#	2949	2952	3025	3541	3544	8078	8081	8092				
RTSND = 000010	1589#	1592	2424	2434									
RUN = 000200	1638#												
RXABGA= 002000	1726#	3148	3159										
RXACT = 000040	3727#	5714	5819	6332	6343	6424	6441	6452	6549	6609	6669	7081	7102
RXCHAR 010110	7123	7134	7145	7265	7276	7291	7305	7316	7406	7417	7432	7446	7457
	7547	7558	7573	7587	7598	7688	7699	7714	7728	7739	7829	7840	7855
	7869	7880	7970	7981	7996	8010	8021	8270	8281	8295	8314	8325	8336
	8696	8721	8941	8966	9104								
	1715#	3738	6634	7901	7937								
RXDL = 000007	1743#	3577	3607	6318	8092	8534	8591	8806	9128				
RXEN = 000100	1639#	7146	7317	7458	7599	7740	7881	8022					
RXCOM = 001000	1636#												
RXERR = 100000	1637#												
RXOR = 004000	1640#	7082	7266	7407	7548	7689	7830	7971					
RXSOM = 000400	2081#	4565*	4569										
SAVE4 002446	2082#	4566*	4570										
SAVE6 002450	2085#	3554*	3555*	3738	4562*								
SAVLEN 002456	2064#												
SCRACH 002406	1688#												
SECAD = 000020	1697#												
SECADR= 010000	2094#	2649											
SELO 002472	2107#	2653											
SEL10 002512	2110#	2654											
SEL12 002516	2113#	2655											
SEL14 002522	2116#	2656											
SEL16 002526	2098#	2650	4789*										
SEL2 002476	2101#	2466*	2513*	2559*	2584*	2651							
SEL4 002502	2104#	2536	2560*	2585*	2652								
SEL6 002506	3479#	5887	5905	5943	5991	6007	6017	6059	6115	6133	6171	6223	6236

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

	6745	6763	6777	6787	6842	6860	6874	6884	6939	6957	6971	6981	8112
	8135	8163	8177	8187									
SETVIA 005326	3013#	3048											
SFPTBL 002246 G	1492#												
SFR = 000001	1731#												
SPEED = 000020	1761#												
SRMODE= 000034	1895#												
STALL 004400	2712#												
STARES 002466	2089#	4604*	4610*										
STARST 021234	4579	4586	4603#										
STEPLU 011614	3488	3589	3650	3698	3918	3967	4018	4041#	5953	6181	6320	6415	6438
	7262	7403	7544	7685	7826	7967	8109	8435	8483	8521			
STRIP = 000040	1687#												
STRIPS= 020000	1696#	8789	9023										
STRTML= 000301	1592#												
STUREG 004272	2678#	5205	5257	5311	5437								
SUBRPC 002420	2069#	4559*											
SVCGBL= 000000	1297#	1300	1307#	1324	1333	1335	1337	1339	1341	1343	1345	1347	1349
	1351	1353	1355	1357	1359	1361	1363	1365	1367	1370	1373	1375	1377
	1379	1381	1383	1385	1387	1389	1391	1393	1395	1397	1399	1401	1403
	1405	1407	1423	1466	1467	1491	1492	2019	2379	2391	4098	4137	4164
	4199	4244	4289	4324	4543	4556	4670	4720	4734	4750	4921	4969	9188
	9251	9274#	9275										
SVCINS= 000001	1297#	1304#	1325	1326	1327	1328	1329	1330	1331	1332	1334	1336	1338
	1340	1342	1344	1346	1348	1350	1352	1354	1356	1358	1360	1362	1364
	1366	1368	1369	1371	1372	1374	1376	1378	1380	1382	1384	1386	1388
	1390	1392	1394	1396	1398	1400	1402	1404	1406	1408	1422	1424	1425
	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438
	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451
	1452	1453	1454	1465	1490	2380	2383	2392	2399	2437	2438	2439	2440
	2481	2482	2483	2484	2528	2529	2530	2531	2599	2600	2601	2602	2698
	2699	2700	2701	2971	2972	2973	2974	3075	3076	3077	3078	3108	3109
	3110	3111	3119	3120	3121	3122	3153	3154	3155	3156	3164	3165	3166
	3167	3198	3199	3200	3201	3209	3210	3211	3212	3243	3244	3245	3246
	3254	3255	3256	3257	3288	3289	3290	3291	3299	3300	3301	3302	3331
	3332	3333	3334	3342	3343	3344	3345	3378	3379	3380	3381	3389	3390
	3391	3392	3402	3403	3404	3405	3413	3414	3415	3416	3445	3446	3447
	3448	3457	3458	3459	3460	3506	3507	3508	3509	3553	3754	3755	3756
	3766	3767	3768	3769	3792	3793	3794	3795	3802	3803	3804	3805	3815
	3816	3817	3818	3825	3826	3827	3828	3838	3839	3840	3841	3848	3849
	3850	3851	3861	3862	3863	3864	3871	3872	3873	3874	3884	3885	3886
	3887	3894	3895	3896	3897	4105	4106	4107	4108	4109	4110	4120	4121
	4122	4123	4124	4125	4126	4132	4139	4140	4141	4142	4143	4144	4145
	4148	4149	4150	4151	4152	4153	4154	4155	4159	4168	4169	4170	4171
	4172	4173	4174	4177	4178	4179	4180	4181	4182	4183	4184	4186	4187
	4188	4189	4190	4193	4201	4202	4203	4204	4205	4206	4207	4209	4210
	4211	4212	4213	4217	4218	4219	4220	4221	4222	4223	4226	4227	4228
	4229	4230	4231	4232	4233	4237	4246	4247	4248	4249	4250	4251	4252
	4254	4255	4256	4257	4258	4262	4263	4264	4265	4266	4267	4268	4271
	4272	4273	4274	4275	4276	4277	4278	4282	4291	4292	4293	4294	4295
	4296	4297	4299	4300	4301	4302	4303	4307	4308	4309	4310	4311	4312
	4313	4317	4326	4327	4328	4329	4330	4331	4332	4335	4336	4337	4338
	4339	4340	4341	4342	4346	4372	4373	4374	4375	4376	4377	4378	4380
	4381	4382	4383	4384	4385	4386	4387	4388	4390	4391	4392	4393	4394
	4395	4397	4398	4399	4400	4401	4402	4403	4404	4405	4407	4408	4409
	4410	4411	4421	4422	4423	4424	4425	4426	4427	4429	4430	4431	4432

CVDMA.P11

10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

4433	4434	4435	4436	4437	4439	4440	4441	4442	4443	4444	4446	4447
4448	4449	4450	4451	4452	4453	4454	4456	4457	4458	4459	4460	4461
4463	4464	4465	4466	4467	4468	4469	4470	4471	4473	4474	4475	4476
4477	4478	4480	4481	4482	4483	4484	4485	4486	4487	4488	4498	4499
4500	4501	4502	4503	4504	4506	4507	4508	4509	4510	4511	4512	4513
4514	4516	4517	4518	4519	4520	4521	4523	4524	4525	4526	4527	4528
4529	4530	4531	4576	4577	4579	4583	4584	4586	4590	4591	4593	4597
4598	4600	4618	4619	4620	4622	4648	4652	4673	4674	4675	4676	4677
4678	4695	4696	4700	4701	4707	4725	4737	4740	4753	4775	4777	4778
4784	4786	4787	4803	4804	4805	4806	4808	4809	4822	4823	4824	4825
4829	4856	4872	4873	4874	4875	4877	4882	4886	4894	4896	4897	4908
4909	4910	4911	4914	4918	4923	4924	4925	4926	4927	4929	4930	4931
4932	4933	4935	4936	4937	4938	4939	4941	4942	4943	4944	4945	4946
4947	4951	4952	4953	4954	4955	4956	4957	4959	4960	4961	4962	4963
4966	4971	4972	4973	4974	4975	4976	4977	4981	5037	5038	5039	5040
5042	5043	5049	5089	5091	5092	5097	5152	5153	5154	5155	5157	5158
5169	5189	5191	5192	5200	5208	5210	5211	5215	5221	5241	5243	5244
5252	5260	5262	5263	5267	5273	5295	5297	5298	5306	5314	5316	5317
5321	5327	5355	5357	5358	5366	5386	5387	5388	5389	5391	5392	5396
5402	5421	5423	5424	5432	5440	5442	5443	5447	5453	5473	5475	5476
5495	5515	5517	5518	5537	5591	5592	5593	5594	5596	5597	5617	5618
5619	5620	5622	5623	5643	5644	5645	5646	5650	5688	5690	5691	5709
5711	5712	5720	5722	5723	5728	5730	5731	5735	5775	5777	5778	5785
5787	5788	5795	5797	5798	5805	5807	5808	5814	5816	5817	5825	5827
5828	5833	5835	5836	5839	5873	5875	5876	5882	5884	5885	5892	5894
5895	5910	5912	5913	5921	5923	5924	5933	5934	5935	5936	5938	5939
5948	5950	5951	5960	5962	5963	5972	5973	5974	5975	5977	5978	5986
5988	5989	5996	5998	5999	6012	6014	6015	6022	6024	6025	6037	6039
6040	6049	6050	6051	6052	6054	6055	6064	6066	6067	6070	6101	6103
6104	6110	6112	6113	6120	6122	6123	6138	6140	6141	6149	6151	6152
6161	6162	6163	6164	6166	6167	6176	6178	6179	6188	6190	6191	6200
6201	6202	6203	6205	6206	6218	6220	6221	6228	6230	6231	6241	6243
6244	6249	6251	6252	6255	6283	6285	6286	6301	6303	6304	6311	6313
6314	6327	6329	6330	6338	6340	6341	6349	6351	6352	6355	6385	6387
6388	6407	6409	6410	6430	6432	6433	6447	6449	6450	6458	6460	6461
6467	6469	6470	6473	6511	6517	6519	6520	6537	6539	6540	6555	6557
6558	6565	6571	6577	6579	6580	6597	6599	6600	6615	6617	6618	6625
6631	6637	6639	6640	6657	6659	6660	6675	6677	6678	6685	6688	6731
6733	6734	6740	6742	6743	6750	6752	6753	6768	6770	6771	6782	6784
6785	6792	6794	6795	6798	6828	6830	6831	6837	6839	6840	6847	6849
6850	6865	6867	6868	6879	6881	6882	6889	6891	6892	6895	6925	6927
6928	6934	6936	6937	6944	6946	6947	6962	6964	6965	6976	6978	6979
6986	6988	6989	6992	7019	7021	7022	7037	7039	7040	7047	7049	7050
7057	7059	7060	7067	7069	7070	7076	7078	7079	7087	7089	7090	7097
7099	7100	7108	7110	7111	7118	7120	7121	7129	7131	7132	7140	7142
7143	7151	7153	7154	7157	7193	7199	7201	7202	7215	7217	7218	7225
7227	7228	7238	7240	7241	7251	7253	7254	7271	7273	7274	7282	7284
7285	7297	7299	7300	7311	7313	7314	7322	7324	7325	7328	7334	7340
7342	7343	7356	7358	7359	7366	7368	7369	7379	7381	7382	7392	7394
7395	7412	7414	7415	7423	7425	7426	7438	7440	7441	7452	7454	7455
7463	7465	7466	7469	7475	7481	7483	7484	7497	7499	7500	7507	7509
7510	7520	7522	7523	7533	7535	7536	7553	7555	7556	7564	7566	7567
7579	7581	7582	7593	7595	7596	7604	7606	7607	7610	7616	7622	7624
7625	7638	7640	7641	7648	7650	7651	7661	7663	7664	7674	7676	7677
7694	7696	7697	7705	7707	7708	7720	7722	7723	7734	7736	7737	7745
7747	7748	7751	7757	7763	7765	7766	7779	7781	7782	7789	7791	7792

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

	7802	7804	7805	7815	7817	7818	7835	7837	7838	7846	7848	7849	7861
	7863	7864	7875	7877	7878	7886	7888	7889	7892	7898	7904	7906	7907
	7920	7922	7923	7930	7932	7933	7943	7945	7946	7956	7958	7959	7976
	7978	7979	7987	7989	7990	8002	8004	8005	8016	8018	8019	8027	8029
	8030	8033	8036	8098	8100	8101	8117	8119	8120	8126	8128	8129	8140
	8142	8143	8149	8151	8152	8168	8170	8171	8182	8184	8185	8192	8194
	8195	8198	8228	8230	8231	8250	8252	8253	8262	8264	8265	8276	8278
	8279	8287	8289	8290	8301	8303	8304	8320	8322	8323	8331	8333	8334
	8342	8344	8345	8351	8353	8354	8357	8403	8411	8413	8414	8430	8432
	8433	8442	8444	8445	8451	8453	8454	8457	8461	8469	8471	8472	8489
	8491	8492	8495	8499	8507	8509	8510	8540	8542	8543	8546	8549	8578
	8580	8581	8600	8602	8603	8610	8612	8613	8620	8622	8623	8630	8632
	8633	8639	8641	8642	8652	8654	8655	8662	8664	8665	8672	8674	8675
	8682	8684	8685	8691	8693	8694	8702	8704	8705	8716	8718	8719	8727
	8729	8730	8739	8741	8742	8746	8793	8795	8796	8815	8817	8818	8825
	8827	8828	8835	8837	8838	8845	8847	8848	8854	8856	8857	8867	8869
	8870	8877	8879	8880	8887	8889	8890	8897	8899	8900	8907	8909	8910
	8917	8919	8920	8927	8929	8930	8936	8938	8939	8947	8949	8950	8961
	8963	8964	8972	8974	8975	8984	8986	8987	8990	9027	9029	9030	9037
	9039	9040	9050	9052	9053	9060	9062	9063	9070	9072	9073	9080	9082
	9083	9090	9092	9093	9099	9101	9102	9110	9112	9113	9123	9125	9126
	9135	9137	9138	9147	9149	9150	9157	9159	9160	9166	9168	9169	9172
	9187	9191	9192	9193	9194	9196	9197	9198	9199	9201	9202	9203	9204
	9205	9207	9208	9209	9210	9211	9213	9214	9215	9216	9217	9219	9220
	9221	9222	9223	9225	9226	9227	9228	9229	9232	9250	9254	9271	9272
	9273												
SVCSUB= 000001	1297#	1306#	4855	4885	5199	5251	5305	5365	5431	6510	6570	6630	7192
	7333	7474	7615	7756	7897	8402	8460	8498					
SVCTAG= 000001	1297#	1308#	1415	1419	1481	1495	4131	4158	4192	4236	4281	4316	4345
	4651	4706	4724	4739	4752	4828	4881	4913	4917	4965	4980	5048	5096
	5168	5214	5220	5266	5272	5320	5326	5395	5401	5446	5452	5494	5536
	5649	5734	5838	6069	6254	6354	6472	6564	6624	6684	6687	6797	6894
	6991	7156	7327	7468	7609	7750	7891	8032	8035	8197	8356	8456	8494
	8545	8548	8745	8989	9171	9233	9255						
SVCTST= 000001	1297#	1305#	4770	4851	5008	5073	5119	5185	5237	5291	5351	5417	5469
	5511	5564	5679	5766	5865	6093	6274	6377	6504	6723	6820	6917	7011
	7186	8061	8220	8399	8569	8785	9019						
SWPBOT= 121000	1987#	4861											
SWPDDC= 121400	1988#	4858											
SW1.M 037052	9208	9235#											
SW2.M 037113	9214	9235#											
SYNCH = 000226	1675#	3584	5771	5781	5869	6097	6279	6381	6513	6573	6633	7195	7336
	7477	7618	7759	7900	8224	8271	8407	8465	8503	8574	8626	8648	8658
	8668	8678	8697	8789	8841	8863	8873	8883	8893	9023	9033		
SSL SYM= 010000	1297#	1482#	1496#	4132#	4159#	4193#	4237#	4282#	4317#	4346#	4652#	4707#	4725#
	4740#	4753#	4829#	4882#	4914#	4918#	4966#	4981#	5049#	5097#	5169#	5215#	5221#
	5267#	5273#	5321#	5327#	5396#	5402#	5447#	5453#	5495#	5537#	5650#	5735#	5839#
	6070#	6255#	6355#	6473#	6565#	6625#	6685#	6688#	6798#	6895#	6992#	7157#	7328#
	7469#	7610#	7751#	7892#	8033#	8036#	8198#	8357#	8457#	8495#	8546#	8549#	8746#
	8990#	9172#	9234#	9256#									
TAB = 000004	1659#	6774	6871										
TBMT = 000100	1725#	3193	3204	3594	5918	5957	6146	6185	9144	9163			
TCCHEK= 100000	1992#												
TDATA 002376	2060#	2683	5202*	5254*	5308*	5368*	5373	5434*					
TDSRH = 120403	1654#	3580	3685	5372	5376	5927	5966	6043	6155	6194			
TDSRL = 120402	1648#	3583	3632	5131	5436	5898	6126	6760	6857	6954	8157	8174	

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

TXTML7	016255	4064#	4073					
TXTNP	016437	4064#	4085					
TXTNPT	016766	4086#						
TXTNP0	016444	4064#	4086					
TXTNP1	016454	4064#	4086					
TXTNP2	016464	4064#	4086					
TXTNP3	016474	4064#	4086					
TXTNP4	016511	4064#	4086					
TXTNP5	016526	4064#	4086					
TXTNP6	016543	4064#	4086					
TXTNP7	016557	4064#	4086					
TXTNP8	016573	4064#	4086					
TXTNUL	016111	4064#						
TXTUR	016607	4064#	4169	4218		4308		
TXTURT	017010	4089#	4168	4217		4307		
TXTURO	016622	4064#	4089					
TXTUR1	016630	4064#	4089					
TXTUR2	016636	4064#	4089					
TXTUR3	016644	4064#	4089					
TXTUR4	016652	4064#	4089					
TXTUR5	016661	4064#	4089					
TXTUR6	016670	4064#	4089					
TXTUR7	016674	4064#	4089					
TXTVR	016312	4064#	4077	4263				
TXTVRA	016410	4064#	4081					
TXTVRB	016413	4064#	4081					
TXTVRC	016417	4064#	4081					
TXTVRD	016423	4064#	4081					
TXTVRE	016427	4064#	4081					
TXTVRF	016433	4064#	4081					
TXTVRT	016724	4078#	4262					
TXTVR0	016330	4064#	4078					
TXTVR1	016334	4064#	4078					
TXTVR2	016340	4064#	4078					
TXTVR3	016345	4064#	4078					
TXTVR4	016352	4064#	4078					
TXTVR5	016357	4064#	4078					
TXTVR6	016364	4064#	4078					
TXTVR7	016371	4064#	4078					
TXTVR8	016376	4064#	4081					
TXTVR9	016403	4064#	4081					
TXT1	014541	4064#						
TXT10	015341	4064#						
TXT11	015361	4064#						
TXT11A	015433	4064#						
TXT11B	015471	4064#						
TXT12	015541	4064#	4202	4247	4292	4327	4972	
TXT13	015567	4064#	4499					
TXT14	015604	4064#	4498					
TXT15	015642	4064#	4516					
TXT16	015704	4064#	4422					
TXT17	015717	4064#	4421					
TXT18	015754	4064#	4439					
TXT19	016015	4064#	4456					
TXT2	014577	4064#						
TXT2A	014641	4064#						

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE --- USER SYMBOLS

T\$LTNO= 000037
T\$NEST= 177777

9275#													
1297#	1300#	1465#	1481#	1490#	1495#	4098#	4131#	4137#	4158#	4164#	4192#	4199#	
4236#	4244#	4281#	4289#	4316#	4324#	4345#	4543#	4548#	4556#	4651#	4670#	4706#	
4720#	4724#	4734#	4739#	4750#	4752#	4771#	4828#	4852#	4856#	4881#	4886#	4913#	
4917#	4921#	4965#	4969#	4980#	5009#	5048#	5074#	5096#	5120#	5168#	5186#	5200#	
5214#	5220#	5238#	5252#	5266#	5272#	5292#	5306#	5320#	5326#	5352#	5366#	5395#	
5401#	5418#	5432#	5446#	5452#	5470#	5494#	5512#	5536#	5565#	5649#	5680#	5734#	
5767#	5838#	5866#	6069#	6094#	6254#	6275#	6354#	6378#	6472#	6505#	6511#	6564#	
6571#	6624#	6631#	6684#	6687#	6724#	6797#	6821#	6894#	6918#	6991#	7012#	7156#	
7187#	7193#	7327#	7334#	7468#	7475#	7609#	7616#	7750#	7757#	7891#	7898#	8032#	
8035#	8062#	8197#	8221#	8356#	8400#	8403#	8456#	8461#	8494#	8499#	8545#	8548#	
8570#	8745#	8786#	8989#	9020#	9171#	9187#	9232#	9250#	9254#	9268#			

T\$NSO = 000000
T\$NS1 = 000005

1300#	9268												
1465#	1481	1490#	1495	4098#	4131	4137#	4158	4164#	4192	4199#	4236	4244#	
4281	4289#	4316	4324#	4345	4543#	4548	4556#	4651	4670#	4706	4720#	4724	
4734#	4739	4750#	4752	4771#	4828	4852#	4917	4921#	4965	4969#	4980	5009#	
5048	5074#	5096	5120#	5168	5186#	5220	5238#	5272	5292#	5326	5352#	5401	
5418#	5452	5470#	5494	5512#	5536	5565#	5649	5680#	5734	5767#	5838	5866#	
6069	6094#	6254	6275#	6354	6378#	6472	6505#	6687	6724#	6797	6821#	6894	
6918#	6991	7012#	7156	7187#	8035	8062#	8197	8221#	8356	8400#	8548	8570#	
8745	8786#	8989	9020#	9171	9187#	9232	9250#	9254					
4856#	4881	4886#	4913	5200#	5214	5252#	5266	5306#	5320	5366#	5395	5432#	
5446	6511#	6564	6571#	6624	6631#	6684	7193#	7327	7334#	7468	7475#	7609	
7616#	7750	7757#	7891	7898#	8032	8403#	8456	8461#	8494	8499#	8545		

T\$NS2 = 000002

1297#													
1297#													
1297#													
1297#	4770#	4851#	4855#	4885#	5008#	5073#	5119#	5185#	5199#	5237#	5251#	5291#	
5305#	5351#	5365#	5417#	5431#	5469#	5511#	5564#	5679#	5766#	5865#	6093#	6274#	
6377#	6504#	6510#	6570#	6630#	6723#	6820#	6917#	7011#	7186#	7192#	7333#	7474#	
7615#	7756#	7897#	8061#	8220#	8399#	8402#	8460#	8498#	8569#	8785#	9019#		

T\$PTNU= 000000
T\$SAVL= 177777
T\$SEGL= 177777
T\$SUBN= 000000

1297#													
1297#													
1297#													
1297#	4770#	4851#	4855#	4885#	5008#	5073#	5119#	5185#	5199#	5237#	5251#	5291#	
5305#	5351#	5365#	5417#	5431#	5469#	5511#	5564#	5679#	5766#	5865#	6093#	6274#	
6377#	6504#	6510#	6570#	6630#	6723#	6820#	6917#	7011#	7186#	7192#	7333#	7474#	
7615#	7756#	7897#	8061#	8220#	8399#	8402#	8460#	8498#	8569#	8785#	9019#		

T\$TAGL= 177777
T\$TAGN= 010105

1297#													
1297#	1465#	1490#	4098#	4137#	4164#	4199#	4244#	4289#	4324#	4543#	4556#	4670#	
4720#	4734#	4750#	4771#	4852#	4856#	4886#	4921#	4969#	5009#	5074#	5120#	5186#	
5200#	5238#	5252#	5292#	5306#	5352#	5366#	5418#	5432#	5470#	5512#	5565#	5680#	
5767#	5866#	6094#	6275#	6378#	6505#	6511#	6571#	6631#	6724#	6821#	6918#	7012#	
7187#	7193#	7334#	7475#	7616#	7757#	7898#	8062#	8221#	8400#	8403#	8461#	8499#	
8570#	8786#	9020#	9187#	9250#									

T\$TEMP= 000000

1415#	1419#	1424#	1425#	1426#	1427#	1428#	1429#	1430#	1431#	1432#	1433#	1434#	
1435#	1436#	1437#	1438#	1439#	1440#	1441#	1442#	1443#	1444#	1445#	1446#	1447#	
1448#	1449#	1450#	1451#	1452#	1453#	1454#	1455#	1481#	1495#	4131#	4158#	4192#	
4236#	4281#	4316#	4345#	4548#	4651#	4706#	4724#	4739#	4752#	4777#	4778	4786#	
4787	4808#	4809	4828#	4881#	4896#	4897	4913#	4917#	4965#	4980#	5042#	5043	
5048#	5091#	5092	5096#	5157#	5158	5168#	5191#	5192	5210#	5211	5214#	5220#	
5243#	5244	5262#	5263	5266#	5272#	5297#	5298	5316#	5317	5320#	5326#	5357#	
5358	5391#	5392	5395#	5401#	5423#	5424	5442#	5443	5446#	5452#	5475#	5476	
5494#	5517#	5518	5536#	5596#	5597	5622#	5623	5649#	5690#	5691	5711#	5712	
5722#	5723	5730#	5731	5734#	5777#	5778	5787#	5788	5797#	5798	5807#	5808	
5816#	5817	5827#	5828	5835#	5836	5838#	5875#	5876	5884#	5885	5894#	5895	
5912#	5913	5923#	5924	5938#	5939	5950#	5951	5962#	5963	5977#	5978	5988#	
5989	5998#	5999	6014#	6015	6024#	6025	6039#	6040	6054#	6055	6066#	6067	
6069#	6103#	6104	6112#	6113	6122#	6123	6140#	6141	6151#	6152	6166#	6167	
6178#	6179	6190#	6191	6205#	6206	6220#	6221	6230#	6231	6243#	6244	6251#	
6252	6254#	6285#	6286	6303#	6304	6313#	6314	6329#	6330	6340#	6341	6351#	
6352	6354#	6387#	6388	6409#	6410	6432#	6433	6449#	6450	6460#	6461	6469#	
6470	6472#	6519#	6520	6539#	6540	6557#	6558	6564#	6579#	6580	6599#	6600	

CVDMCA.P11

10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

6617#	6618	6624#	6639#	6640	6659#	6660	6677#	6678	6684#	6687#	6733#	6734
6742#	6743	6752#	6753	6770#	6771	6784#	6785	6794#	6795	6797#	6830#	6831
6839#	6840	6849#	6850	6867#	6868	6881#	6882	6891#	6892	6894#	6927#	6928
6936#	6937	6946#	6947	6964#	6965	6978#	6979	6988#	6989	6991#	7021#	7022
7039#	7040	7049#	7050	7059#	7060	7069#	7070	7078#	7079	7089#	7090	7099#
7100	7110#	7111	7120#	7121	7131#	7132	7142#	7143	7153#	7154	7156#	7201#
7202	7217#	7218	7227#	7228	7240#	7241	7253#	7254	7273#	7274	7284#	7285
7299#	7300	7313#	7314	7324#	7325	7327#	7342#	7343	7358#	7359	7368#	7369
7381#	7382	7394#	7395	7414#	7415	7425#	7426	7440#	7441	7454#	7455	7465#
7466	7468#	7483#	7484	7499#	7500	7509#	7510	7522#	7523	7535#	7536	7555#
7556	7566#	7567	7581#	7582	7595#	7596	7606#	7607	7609#	7624#	7625	7640#
7641	7650#	7651	7663#	7664	7676#	7677	7696#	7697	7707#	7708	7722#	7723
7736#	7737	7747#	7748	7750#	7765#	7766	7781#	7782	7791#	7792	7804#	7805
7817#	7818	7837#	7838	7848#	7849	7863#	7864	7877#	7878	7888#	7889	7891#
7906#	7907	7922#	7923	7932#	7933	7945#	7946	7958#	7959	7978#	7979	7989#
7990	8004#	8005	8018#	8019	8029#	8030	8032#	8035#	8100#	8101	8119#	8120
8128#	8129	8142#	8143	8151#	8152	8170#	8171	8184#	8185	8194#	8195	8197#
8230#	8231	8252#	8253	8264#	8265	8278#	8279	8289#	8290	8303#	8304	8322#
8323	8333#	8334	8344#	8345	8353#	8354	8356#	8413#	8414	8432#	8433	8444#
8445	8453#	8454	8456#	8471#	8472	8491#	8492	8494#	8509#	8510	8542#	8543
8545#	8548#	8580#	8581	8602#	8603	8612#	8613	8622#	8623	8632#	8633	8641#
8642	8654#	8655	8664#	8665	8674#	8675	8684#	8685	8693#	8694	8704#	8705
8718#	8719	8729#	8730	8741#	8742	8745#	8795#	8796	8817#	8818	8827#	8828
8837#	8838	8847#	8848	8856#	8857	8869#	8870	8879#	8880	8889#	8890	8899#
8900	8909#	8910	8919#	8920	8929#	8930	8938#	8939	8949#	8950	8963#	8964
8974#	8975	8986#	8987	8989#	9029#	9030	9039#	9040	9052#	9053	9062#	9063
9072#	9073	9082#	9083	9092#	9093	9101#	9102	9112#	9113	9125#	9126	9137#
9138	9149#	9150	9159#	9160	9168#	9169	9171#	9191#	9196#	9201#	9207#	9213#
9219#	9225#	9232#	9254#	9268#								
1297#	4770#	4851#	4855	4885	5008#	5073#	5119#	5185#	5199	5237#	5251	5291#
5305	5351#	5365	5417#	5431	5469#	5511#	5564#	5679#	5766#	5865#	6093#	6274#
6377#	6504#	6510	6570	6630	6723#	6820#	6917#	7011#	7186#	7192	7333	7474
7615	7756	7897	8061#	8220#	8399#	8402	8460	8498	8569#	8785#	9019#	9275
1297#	4109	4125	4132	4144	4154	4159	4173	4183	4189	4193	4206	4212
4222	4232	4237	4251	4257	4267	4277	4282	4296	4302	4312	4317	4331
4341	4346	4377	4387	4394	4404	4410	4426	4436	4443	4453	4460	4470
4477	4487	4503	4513	4520	4530	4577	4584	4591	4598	4619	4648	4652
4677	4696	4701	4707	4725	4737	4740	4753	4775	4777	4784	4786	4803
4808	4822	4829	4856	4872	4877	4882	4886	4894	4896	4908	4914	4918
4926	4932	4938	4946	4956	4962	4966	4976	4981	5037	5042	5049	5089
5091	5097	5152	5157	5169	5189	5191	5200	5208	5210	5215	5221	5241
5243	5252	5260	5262	5267	5273	5295	5297	5306	5314	5316	5321	5327
5355	5357	5366	5386	5391	5396	5402	5421	5423	5432	5440	5442	5447
5453	5473	5475	5495	5515	5517	5537	5591	5596	5617	5622	5643	5650
5688	5690	5709	5711	5720	5722	5728	5730	5735	5775	5777	5785	5787
5795	5797	5805	5807	5814	5816	5825	5827	5833	5835	5839	5873	5875
5882	5884	5892	5894	5910	5912	5921	5923	5933	5938	5948	5950	5960
5962	5972	5977	5986	5988	5996	5998	6012	6014	6022	6024	6037	6039
6049	6054	6064	6066	6070	6101	6103	6110	6112	6120	6122	6138	6140
6149	6151	6161	6166	6176	6178	6188	6190	6200	6205	6218	6220	6228
6230	6241	6243	6249	6251	6255	6283	6285	6301	6303	6311	6313	6327
6329	6338	6340	6349	6351	6355	6385	6387	6407	6409	6430	6432	6447
6449	6458	6460	6467	6469	6473	6511	6517	6519	6537	6539	6555	6557
6565	6571	6577	6579	6597	6599	6615	6617	6625	6631	6637	6639	6657
6659	6675	6677	6685	6688	6731	6733	6740	6742	6750	6752	6768	6770
6782	6784	6792	6794	6798	6828	6830	6837	6839	6847	6849	6865	6867

TSTEST= 000037

TSTSTM= 177777

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

6879	6881	6889	6891	6895	6925	6927	6934	6936	6944	6946	6962	6964	
6976	6978	6986	6988	6992	7019	7021	7037	7039	7047	7049	7057	7059	
7067	7069	7076	7078	7087	7089	7097	7099	7108	7110	7118	7120	7129	
7131	7140	7142	7151	7153	7157	7193	7199	7201	7215	7217	7225	7227	
7238	7240	7251	7253	7271	7273	7282	7284	7297	7299	7311	7313	7322	
7324	7328	7334	7340	7342	7356	7358	7366	7368	7379	7381	7392	7394	
7412	7414	7423	7425	7438	7440	7452	7454	7463	7465	7469	7475	7481	
7483	7497	7499	7507	7509	7520	7522	7533	7535	7553	7555	7564	7566	
7579	7581	7593	7595	7604	7606	7610	7616	7622	7624	7638	7640	7648	
7650	7661	7663	7674	7676	7694	7696	7705	7707	7720	7722	7734	7736	
7745	7747	7751	7757	7763	7765	7779	7781	7789	7791	7802	7804	7815	
7817	7835	7837	7846	7848	7861	7863	7875	7877	7886	7888	7892	7898	
7904	7906	7920	7922	7930	7932	7943	7945	7956	7958	7976	7978	7987	
7989	8002	8004	8016	8018	8027	8029	8033	8036	8098	8100	8117	8119	
8126	8128	8140	8142	8149	8151	8168	8170	8182	8184	8192	8194	8198	
8228	8230	8250	8252	8262	8264	8276	8278	8287	8289	8301	8303	8320	
8322	8331	8333	8342	8344	8351	8353	8357	8403	8411	8413	8430	8432	
8442	8444	8451	8453	8457	8461	8469	8471	8489	8491	8495	8499	8507	
8509	8540	8542	8546	8549	8578	8580	8600	8602	8610	8612	8620	8622	
8630	8632	8639	8641	8652	8654	8662	8664	8672	8674	8682	8684	8691	
8693	8702	8704	8716	8718	8727	8729	8739	8741	8746	8793	8795	8815	
8817	8825	8827	8835	8837	8845	8847	8854	8856	8867	8869	8877	8879	
8887	8889	8897	8899	8907	8909	8917	8919	8927	8929	8936	8938	8947	
8949	8961	8963	8972	8974	8984	8986	8990	9027	9029	9037	9039	9050	
9052	9060	9062	9070	9072	9080	9082	9090	9092	9099	9101	9110	9112	
9123	9125	9135	9137	9147	9149	9157	9159	9166	9168	9172			
TSTSTS= 000001	1297#	4771#	4852#	5009#	5074#	5120#	5186#	5238#	5292#	5352#	5418#	5470#	5512#
	5565#	5680#	5767#	5866#	6094#	6275#	6378#	6505#	6724#	6821#	6918#	7012#	7187#
	8062#	8221#	8400#	8570#	8786#	9020#							
TSSAU = 010016	4750#	4752											
TSSAUT= 010013	4670#	4706											
TSSCLE= 010014	4720#	4724											
TSSDU = 010015	4734#	4739											
TSSHAR= 010103	9187#	9233											
TSSHW = 010000	1465#	1481											
TSSINI= 010012	4556#	4651											
TSSMSG= 010024	4098#	4131	4137#	4158	4164#	4192	4199#	4236	4244#	4281	4289#	4316	4324#
	4345	4921#	4965	4969#	4980								
TSSPRO= 010011	4543#												
TSSSOF= 010104	9250#	9255											
TSSSUB= 010077	4856#	4881	4886#	4896	4913	5200#	5214	5252#	5266	5306#	5320	5366#	5395
	5432#	5446	6511#	6519	6539	6557	6564	6571#	6579	6599	6617	6624	6631#
	6639	6659	6677	6684	7193#	7201	7217	7227	7240	7253	7273	7284	7299
	7313	7324	7327	7334#	7342	7358	7368	7381	7394	7414	7425	7440	7468
	7475#	7483	7499	7509	7522	7535	7555	7566	7581	7609	7616#	7624	7640
	7650	7663	7676	7696	7707	7722	7750	7757#	7765	7781	7791	7804	7817
	7837	7848	7863	7891	7898#	7906	7922	7932	7945	7958	7978	7989	8004
	8032	8403#	8413	8432	8453	8456	8461#	8471	8494	8499#	8509	8542	8545
TSSSW = 010001	1490#	1495											
TSSTES= 010102	4771#	4777	4786	4808	4828	4852#	4917	5009#	5042	5048	5074#	5091	5096
	5120#	5157	5168	5186#	5191	5210	5220	5238#	5243	5262	5272	5292#	5297
	5316	5326	5352#	5357	5391	5401	5418#	5423	5442	5452	5470#	5475	5494
	5512#	5517	5536	5565#	5596	5622	5649	5680#	5690	5711	5722	5730	5734
	5767#	5777	5787	5797	5807	5816	5827	5835	5838	5866#	5875	5884	5894
	5912	5923	5938	5950	5962	5977	5988	5998	6014	6024	6039	6054	6066
	6069	6094#	6103	6112	6122	6140	6151	6166	6178	6190	6205	6220	6230

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- USER SYMBOLS

VIASR = 120012	1868#													
VIAT1A= 120004	1798#	2839	5567											
VIAT1B= 120005	1810#	2844	4045	5570										
VIAT1C= 120006	1823#	2830	3563	8070										
VIAT1D= 120007	1834#	2835	3566	8073										
VIAT2A= 120010	1845#	2917												
VIAT2B= 120011	1857#	2922												
VREGS 002336	2056#	2756	4429	4430	4431	4432	4446	4447	4448	4449	4463	4464	4465	
	4466	4480	4481	4482	4483									
WAIT50 005312	2990#	8529												
WRIBYT 002434	2076#													
WRILOC= 000002	1603#	2586	2596											
WRIPAG= 000004	1605#													
WRITE 003722	2559#	2681	5371											
WRITEI 003734	2583#	2812	2825	2829	2834	2838	2843	2899	2912	2916	2921	2947	2950	
	3014	3017	3020	3023	3026	3029	3032	3539	3542	3546	3550	3556	3559	
	3562	3565	3575	3579	3582	3631	3684	4015	4044	5014	5079	5134	5483	
	5525	5566	5569	5897	6003	6125	6210	6316	6759	6856	6953	7230	7371	
	7512	7653	7794	7935	8066	8069	8072	8076	8079	8083	8086	8090	8156	
	8173	8524	8532	8589	8804	9130								
WSRO 002256	2028#	2649*	4383											
WSR10 002266	2036#	2653*	4400											
WSR12 002270	2038#	2654*	4399											
WSR14 002272	2040#	2655*	4398											
WSR16 002274	2042#	2656*	4397											
WSR2 002260	2030#	2650*	4382	4812	4818									
WSR4 002262	2032#	2651*	4381											
WSR6 002264	2034#	2652*	4380											
XDATA 002404	2063#	4148	4177	4226	4271	4335	4361*	4362*						
XORGB 020134	4146	4175	4224	4269	4333	4359#								
XYZ = 000007	1690#	6005	8088											
XSALWA= 000000	1297#													
XSFALS= 000040	1297#													
XSOFFS= 000400	1297#													
XSTRUE= 000020	1297#													
SE = 000065	2007#	2436#	2480#	2527#	2598#	2697#	2970#	3074#	3107#	3118#	3152#	3163#	3197#	
	3208#	3242#	3253#	3287#	3298#	3330#	3341#	3377#	3388#	3401#	3412#	3444#	3456#	
	3505#	3752#	3765#	3791#	3801#	3814#	3824#	3837#	3847#	3860#	3870#	3883#	3893#	
	4802#	4821#	4871#	4907#	5036#	5151#	5385#	5590#	5616#	5642#	5932#	5971#	6048#	
	6160#	6199#												
\$LSTIN= 000001	1302#													
\$LSTTA= 000001	1303#													
\$T = 000037	2007#	4755#	4831#	4985#	5050#	5098#	5170#	5222#	5274#	5328#	5403#	5454#	5496#	
	5538#	5651#	5736#	5841#	6071#	6256#	6356#	6488#	6702#	6799#	6896#	6993#	7158#	
	8047#	8199#	8373#	8550#	8764#	9005#								
	1293#	2053#	2056#	2140#	2357#	2362#	2383#	2399#	3511	4773	4778	4782	4787	
	4809	4892	4897	5043	5087	5092	5158	5192	5211	5244	5263	5298	5317	
	5358	5392	5424	5443	5476	5518	5597	5623	5686	5691	5707	5712	5718	
	5723	5726	5731	5773	5778	5783	5788	5793	5798	5803	5808	5812	5817	
	5823	5828	5831	5836	5871	5876	5880	5885	5890	5895	5908	5917	5919	
	5924	5939	5946	5951	5958	5963	5978	5984	5989	5994	5999	6010	6015	
	6020	6025	6035	6040	6055	6062	6067	6099	6104	6108	6113	6118	6123	
	6136	6141	6147	6152	6167	6174	6179	6186	6191	6206	6216	6221	6226	
	6231	6239	6244	6247	6252	6281	6286	6299	6304	6309	6314	6325	6330	
	6336	6341	6347	6352	6383	6388	6405	6410	6428	6433	6445	6450	6456	
	6461	6465	6470	6486#	6515	6520	6535	6540	6553	6558	6575	6580	6595	

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

GETPRI	1#	1297#													
GETWOR	1#	1297#													
GMANIA	1#	1297#													
GMANID	1#	1297#													
GMANIL	1#	1297#													
GPHARD	1#	1297#	4617												
GPRMA	1#	1297#	9190	9195											
GPRMD	1#	1297#	9200	9206	9212	9218	9224								
GPRML	1#	1297#													
GTDF	2007#	2435	2479	2526	2597	2696	2969	3073	3106	3117	3151	3162	3196	3207	3241
	3252	3286	3297	3329	3340	3376	3387	3400	3411	3443	3455	3504	3751	3764	3790
	3800	3813	3823	3836	3846	3859	3869	3882	3892						
GTHRD	2007#														
GTSF	2007#														
GTSFT	2007#														
HEADER	1#	1297#	1323												
INLOOP	1#	1297#													
IOSETU	1#	1297#													
IOSTAR	1#	1297#													
KT11	1#	1297#													
LASTAD	1#	1297#	9270												
MANUAL	1#	1297#													
MEMORY	1#	1297#													
MSG	4755#	4761	4831#	4837	4985#	4991	5050#	5056	5098#	5104	5170#	5176	5222#	5228	5274#
	5280	5328#	5334	5403#	5409	5454#	5460	5496#	5502	5538#	5544	5651#	5657	5736#	5742
	5841#	5847	6071#	6077	6256#	6262	6356#	6362	6488#	6494	6702#	6708	6799#	6805	6896#
	6902	6993#	6999	7158#	7164	8047#	8053	8199#	8205	8373#	8379	8550#	8556	8764#	8770
	9005#	9011													
MSBYTE	1#	1297#	1324#	1330	1331	1332									
MSCHEC	1#	1297#													
MSCNTO	1#	1297#	9191#	9196#	9201#	9207#	9213#	9219#	9225#						
MSCOUN	1#	1297#	4105#	4120#	4139#	4148#	4168#	4177#	4186#	4201#	4209#	4217#	4226#	4246#	4254#
	4262#	4271#	4291#	4299#	4307#	4326#	4335#	4372#	4380#	4390#	4397#	4407#	4421#	4429#	4439#
	4446#	4456#	4463#	4473#	4480#	4498#	4506#	4516#	4523#	4923#	4929#	4935#	4941#	4951#	4959#
	4971#														
MSDATA	1#	1297#	1324#	1333	1335	1337	1339	1341	1343	1345	1347	1349	1351	1353	1355
	1357	1359	1361	1363#	1365	1367	1370	1373	1375	1377	1379	1381	1383	1385	1387
	1389	1391	1393	1395	1397	1399	1401	1403	1405	1407	2379#	2391#			
MSDECR	1#	1297#	1481#	1495#	4131#	4158#	4192#	4236#	4281#	4316#	4345#	4548#	4651#	4706#	4724#
	4739#	4752#	4828#	4881#	4913#	4917#	4965#	4980#	5048#	5096#	5168#	5214#	5220#	5266#	5272#
	5320#	5326#	5395#	5401#	5446#	5452#	5494#	5536#	5649#	5734#	5838#	6069#	6254#	6354#	6472#
	6564#	6624#	6684#	6687#	6797#	6894#	6991#	7156#	7327#	7468#	7609#	7750#	7891#	8032#	8035#
	8197#	8356#	8456#	8494#	8545#	8548#	8745#	8989#	9171#	9232#	9254#	9268#			
MSDEFA	1#	1297#	9191#	9196#	9201#	9207#	9213#	9219#	9225#						
MSENDE	1#	1297#	1481#	1495#	4131#	4158#	4192#	4236#	4281#	4316#	4345#	4651#	4706#	4724#	4739#
	4752#	4828#	4881#	4913#	4917#	4965#	4980#	5048#	5096#	5168#	5214#	5220#	5266#	5272#	5320#
	5326#	5395#	5401#	5446#	5452#	5494#	5536#	5649#	5734#	5838#	6069#	6254#	6354#	6472#	6564#
	6624#	6684#	6687#	6797#	6894#	6991#	7156#	7327#	7468#	7609#	7750#	7891#	8032#	8035#	8197#
	8356#	8456#	8494#	8545#	8548#	8745#	8989#	9171#	9232#	9254#	9268#				
MSERRI	1#	1297#	4803#	4822#	4872#	4908#	5037#	5152#	5386#	5591#	5617#	5643#	5933#	5972#	6049#
	6161#	6200#													
MSESCA	1#	1297#	4777#	4778	4786#	4787	4808#	4809	4896#	4897	5042#	5043	5091#	5092	5157#
	5158	5191#	5192	5210#	5211	5243#	5244	5262#	5263	5297#	5298	5316#	5317	5357#	5358
	5391#	5392	5423#	5424	5442#	5443	5475#	5476	5517#	5518	5596#	5597	5622#	5623	5690#
	5691	5711#	5712	5722#	5723	5730#	5731	5777#	5778	5787#	5788	5797#	5798	5807#	5808
	5816#	5817	5827#	5828	5835#	5836	5875#	5876	5884#	5885	5894#	5895	5912#	5913	5923#

CVDMCA.P11

10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

	5924	5938#	5939	5950#	5951	5962#	5963	5977#	5978	5988#	5989	5998#	5999	6014#	6015
	6024#	6025	6039#	6040	6054#	6055	6066#	6067	6103#	6104	6112#	6113	6122#	6123	6140#
	6141	6151#	6152	6166#	6167	6178#	6179	6190#	6191	6205#	6206	6220#	6221	6230#	6231
	6243#	6244	6251#	6252	6285#	6286	6303#	6304	6313#	6314	6329#	6330	6340#	6341	6351#
	6352	6387#	6388	6409#	6410	6432#	6433	6449#	6450	6460#	6461	6469#	6470	6519#	6520
	6539#	6540	6557#	6558	6579#	6580	6599#	6600	6617#	6618	6639#	6640	6659#	6660	6677#
	6678	6733#	6734	6742#	6743	6752#	6753	6770#	6771	6784#	6785	6794#	6795	6830#	6831
	6839#	6840	6849#	6850	6867#	6868	6881#	6882	6891#	6892	6927#	6928	6936#	6937	6946#
	6947	6964#	6965	6978#	6979	6988#	6989	7021#	7022	7039#	7040	7049#	7050	7059#	7060
	7069#	7070	7078#	7079	7089#	7090	7099#	7100	7110#	7111	7120#	7121	7131#	7132	7142#
	7143	7153#	7154	7201#	7202	7217#	7218	7227#	7228	7240#	7241	7253#	7254	7273#	7274
	7284#	7285	7299#	7300	7313#	7314	7324#	7325	7342#	7343	7358#	7359	7368#	7369	7381#
	7382	7394#	7395	7414#	7415	7425#	7426	7440#	7441	7454#	7455	7465#	7466	7483#	7484
	7499#	7500	7509#	7510	7522#	7523	7535#	7536	7555#	7556	7566#	7567	7581#	7582	7595#
	7596	7606#	7607	7624#	7625	7640#	7641	7650#	7651	7663#	7664	7676#	7677	7696#	7697
	7707#	7708	7722#	7723	7736#	7737	7747#	7748	7765#	7766	7781#	7782	7791#	7792	7804#
	7805	7817#	7818	7837#	7838	7848#	7849	7863#	7864	7877#	7878	7888#	7889	7906#	7907
	7922#	7923	7932#	7933	7945#	7946	7958#	7959	7978#	7979	7989#	7990	8004#	8005	8018#
	8019	8029#	8030	8100#	8101	8119#	8120	8128#	8129	8142#	8143	8151#	8152	8170#	8171
	8184#	8185	8194#	8195	8230#	8231	8252#	8253	8264#	8265	8278#	8279	8289#	8290	8303#
	8304	8322#	8323	8333#	8334	8344#	8345	8353#	8354	8413#	8414	8432#	8433	8444#	8445
	8453#	8454	8471#	8472	8491#	8492	8509#	8510	8542#	8543	8580#	8581	8602#	8603	8612#
	8613	8622#	8623	8632#	8633	8641#	8642	8654#	8655	8664#	8665	8674#	8675	8684#	8685
	8693#	8694	8704#	8705	8718#	8719	8729#	8730	8741#	8742	8795#	8796	8817#	8818	8827#
	8828	8837#	8838	8847#	8848	8856#	8857	8869#	8870	8879#	8880	8889#	8890	8899#	8900
	8909#	8910	8919#	8920	8929#	8930	8938#	8939	8949#	8950	8963#	8964	8974#	8975	8986#
	8987	9029#	9030	9039#	9040	9052#	9053	9062#	9063	9072#	9073	9082#	9083	9092#	9093
	9101#	9102	9112#	9113	9125#	9126	9137#	9138	9149#	9150	9159#	9160	9168#	9169	
MSESCS	1#	1297#	4777#	4786#	4808#	4896#	5042#	5091#	5157#	5191#	5210#	5243#	5262#	5297#	5316#
	5357#	5391#	5423#	5442#	5475#	5517#	5596#	5622#	5690#	5711#	5722#	5730#	5777#	5787#	5797#
	5807#	5816#	5827#	5835#	5875#	5884#	5894#	5912#	5923#	5938#	5950#	5962#	5977#	5988#	5998#
	6014#	6024#	6039#	6054#	6066#	6103#	6112#	6122#	6140#	6151#	6166#	6178#	6190#	6205#	6220#
	6230#	6243#	6251#	6285#	6303#	6313#	6329#	6340#	6351#	6387#	6409#	6432#	6449#	6460#	6469#
	6519#	6539#	6557#	6579#	6599#	6617#	6639#	6659#	6677#	6733#	6742#	6752#	6770#	6784#	6794#
	6830#	6839#	6849#	6867#	6881#	6891#	6927#	6936#	6946#	6964#	6978#	6988#	7021#	7039#	7049#
	7059#	7069#	7078#	7089#	7099#	7110#	7120#	7131#	7142#	7153#	7201#	7217#	7227#	7240#	7253#
	7273#	7284#	7299#	7313#	7324#	7342#	7358#	7368#	7381#	7394#	7414#	7425#	7440#	7454#	7465#
	7483#	7499#	7509#	7522#	7535#	7555#	7566#	7581#	7595#	7606#	7624#	7640#	7650#	7663#	7676#
	7696#	7707#	7722#	7736#	7747#	7765#	7781#	7791#	7804#	7817#	7837#	7848#	7863#	7877#	7888#
	7906#	7922#	7932#	7945#	7958#	7978#	7989#	8004#	8018#	8029#	8100#	8119#	8128#	8142#	8151#
	8170#	8184#	8194#	8230#	8252#	8264#	8278#	8289#	8303#	8322#	8333#	8344#	8353#	8413#	8432#
	8444#	8453#	8471#	8491#	8509#	8542#	8580#	8602#	8612#	8622#	8632#	8641#	8654#	8664#	8674#
	8684#	8693#	8704#	8718#	8729#	8741#	8795#	8817#	8827#	8837#	8847#	8856#	8869#	8879#	8889#
	8899#	8909#	8919#	8929#	8938#	8949#	8963#	8974#	8986#	9029#	9039#	9052#	9062#	9072#	9082#
	9092#	9101#	9112#	9125#	9137#	9149#	9159#	9168#							
MSEXCP	1#	1297#	9191#	9196#	9201#	9207#	9213#	9219#	9225#						
MSEXIT	1#	1297#													
MSEXSE	1#	1297#													
MSEXTJ	1#	1297#													
M\$GEN	1#	1297#	1300#	1324#	1333#	1335#	1337#	1339#	1341#	1343#	1345#	1347#	1349#	1351#	1353#
	1355#	1357#	1359#	1361#	1363#	1365#	1367#	1370#	1373#	1375#	1377#	1379#	1381#	1383#	1385#
	1387#	1389#	1391#	1393#	1395#	1397#	1399#	1401#	1403#	1405#	1407#	1423#	1466#	1467#	1481#
	1491#	1492#	1495#	2019#	2379#	2391#	4098#	4131#	4137#	4158#	4164#	4192#	4199#	4236#	4244#
	4281#	4289#	4316#	4324#	4345#	4543#	4556#	4651#	4670#	4706#	4720#	4724#	4734#	4739#	4750#
	4752#	4770#	4828#	4851#	4855#	4881#	4885#	4913#	4917#	4921#	4965#	4969#	4980#	5008#	5048#
	5073#	5096#	5119#	5168#	5185#	5199#	5214#	5220#	5237#	5251#	5266#	5272#	5291#	5305#	5320#

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

	5326#	5351#	5365#	5395#	5401#	5417#	5431#	5446#	5452#	5469#	5494#	5511#	5536#	5564#	5649#
	5679#	5734#	5766#	5838#	5865#	6069#	6093#	6254#	6274#	6354#	6377#	6472#	6504#	6510#	6564#
	6570#	6624#	6630#	6684#	6687#	6723#	6797#	6820#	6894#	6917#	6991#	7011#	7156#	7186#	7192#
	7327#	7333#	7468#	7474#	7609#	7615#	7750#	7756#	7891#	7897#	8032#	8035#	8061#	8197#	8220#
	8356#	8399#	8402#	8456#	8460#	8494#	8498#	8545#	8548#	8569#	8745#	8785#	8989#	9019#	9171#
	9188#	9233#	9251#	9255#	9274#										
MSGENB	1#	1297#													
MSGETS	1#	1297#	1481#	1495#	4131#	4158#	4192#	4236#	4281#	4316#	4345#	4548#	4651#	4706#	4724#
	4739#	4752#	4828#	4881#	4913#	4917#	4965#	4980#	5048#	5096#	5168#	5214#	5220#	5266#	5272#
	5320#	5326#	5395#	5401#	5446#	5452#	5494#	5536#	5649#	5734#	5838#	6069#	6254#	6354#	6472#
	6564#	6624#	6684#	6687#	6797#	6894#	6991#	7156#	7327#	7468#	7609#	7750#	7891#	8032#	8035#
MSGETT	1#	1297#	4777#	4786#	4808#	4896#	5042#	5091#	5157#	5191#	5210#	5243#	5262#	5297#	5316#
	5357#	5391#	5423#	5442#	5475#	5517#	5596#	5622#	5690#	5711#	5722#	5730#	5777#	5787#	5797#
	5807#	5816#	5827#	5835#	5875#	5884#	5894#	5912#	5923#	5938#	5950#	5962#	5977#	5988#	5998#
	6014#	6024#	6039#	6054#	6066#	6103#	6112#	6122#	6140#	6151#	6166#	6178#	6190#	6205#	6220#
	6230#	6243#	6251#	6285#	6303#	6313#	6329#	6340#	6351#	6387#	6409#	6432#	6449#	6460#	6469#
	6519#	6539#	6557#	6579#	6599#	6617#	6639#	6659#	6677#	6733#	6742#	6752#	6770#	6784#	6794#
	6830#	6839#	6849#	6867#	6881#	6891#	6927#	6936#	6946#	6964#	6978#	6988#	7021#	7039#	7049#
	7059#	7069#	7078#	7089#	7099#	7110#	7120#	7131#	7142#	7153#	7201#	7217#	7227#	7240#	7253#
	7273#	7284#	7299#	7313#	7324#	7342#	7358#	7368#	7381#	7394#	7414#	7425#	7440#	7454#	7465#
	7483#	7499#	7509#	7522#	7535#	7555#	7566#	7581#	7595#	7606#	7624#	7640#	7650#	7663#	7676#
	7696#	7707#	7722#	7736#	7747#	7765#	7781#	7791#	7804#	7817#	7837#	7848#	7863#	7877#	7888#
	7906#	7922#	7932#	7945#	7958#	7978#	7989#	8004#	8018#	8029#	8100#	8119#	8128#	8142#	8151#
	8170#	8184#	8194#	8230#	8252#	8264#	8278#	8289#	8303#	8322#	8333#	8344#	8353#	8413#	8432#
	8444#	8453#	8471#	8491#	8509#	8542#	8580#	8602#	8612#	8622#	8632#	8641#	8654#	8664#	8674#
	8684#	8693#	8704#	8718#	8729#	8741#	8795#	8817#	8827#	8837#	8847#	8856#	8869#	8879#	8889#
	8899#	8909#	8919#	8929#	8938#	8949#	8963#	8974#	8986#	9029#	9039#	9052#	9062#	9072#	9082#
	9092#	9101#	9112#	9125#	9137#	9149#	9159#	9168#							
MSGNGB	1#	1297#	1300#	1324#	1333#	1335#	1337#	1339#	1341#	1343#	1345#	1347#	1349#	1351#	1353#
	1355#	1357#	1359#	1361#	1363#	1365#	1367#	1370#	1373#	1375#	1377#	1379#	1381#	1383#	1385#
	1387#	1389#	1391#	1393#	1395#	1397#	1399#	1401#	1403#	1405#	1407#	1422#	1423	1465#	1466
	1467	1490#	1491	1492	2019#	2379#	2391#	4098#	4137#	4164#	4199#	4244#	4289#	4324#	4543#
	4556#	4670#	4720#	4734#	4750#	4921#	4969#	9187#	9188	9250#	9251	9271#	9274		
MSGNIN	1#	1297#	1324#	1325	1326	1327	1328	1329	1330#	1331#	1332#	1333#	1334	1335#	1336
	1337#	1338	1339#	1340	1341#	1342	1343#	1344	1345#	1346	1347#	1348	1349#	1350	1351#
	1352	1353#	1354	1355#	1356	1357#	1358	1359#	1360	1361#	1362	1363#	1364	1365#	1366
	1367#	1368	1369	1370#	1371	1372#	1373#	1374	1375#	1376	1377#	1378	1379#	1380	1381#
	1382	1383#	1384	1385#	1386	1387#	1388	1389#	1390	1391#	1392	1393#	1394	1395#	1396
	1397#	1398	1399#	1400	1401#	1402	1403#	1404	1405#	1406	1407#	1408	1422#	1424#	1425#
	1426#	1427#	1428#	1429#	1430#	1431#	1432#	1433#	1434#	1435#	1436#	1437#	1438#	1439#	1440#
	1441#	1442#	1443#	1444#	1445#	1446#	1447#	1448#	1449#	1450#	1451#	1452#	1453#	1454#	1465#
	1490#	2379#	2380	2383	2391#	2392	2399	2437	2438	2439	2440	2481	2482	2483	2484
	2528	2529	2530	2531	2599	2600	2601	2602	2698	2699	2700	2701	2971	2972	2973
	2974	3075	3076	3077	3078	3108	3109	3110	3111	3119	3120	3121	3122	3153	3154
	3155	3156	3164	3165	3166	3167	3198	3199	3200	3201	3209	3210	3211	3212	3243
	3244	3245	3246	3254	3255	3256	3257	3288	3289	3290	3291	3299	3300	3301	3302
	3331	3332	3333	3334	3342	3343	3344	3345	3378	3379	3380	3381	3389	3390	3391
	3392	3402	3403	3404	3405	3413	3414	3415	3416	3445	3446	3447	3448	3457	3458
	3459	3460	3506	3507	3508	3509	3753	3754	3755	3756	3766	3767	3768	3769	3792
	3793	3794	3795	3802	3803	3804	3805	3815	3816	3817	3818	3825	3826	3827	3828
	3838	3839	3840	3841	3848	3849	3850	3851	3861	3862	3863	3864	3871	3872	3873
	3874	3884	3885	3886	3887	3894	3895	3896	3897	4105#	4106#	4107#	4108	4109#	4110
	4120#	4121#	4122#	4123#	4124	4125#	4126	4132#	4139#	4140#	4141#	4142#	4143	4144#	4145
	4148#	4149#	4150#	4151#	4152#	4153	4154#	4155	4159#	4168#	4169#	4170#	4171#	4172	4173#
	4174	4177#	4178#	4179#	4180#	4181#	4182	4183#	4184	4186#	4187#	4188	4189#	4190	4193#

CVDMA.P11

10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

4201#	4202#	4203#	4204#	4205	4206#	4207	4209#	4210#	4211	4212#	4213	4217#	4218#	4219#
4220#	4221	4222#	4223	4226#	4227#	4228#	4229#	4230#	4231	4232#	4233	4237#	4246#	4247#
4248#	4249#	4250	4251#	4252	4254#	4255#	4256	4257#	4258	4262#	4263#	4264#	4265#	4266
4267#	4268	4271#	4272#	4273#	4274#	4275#	4276	4277#	4278	4282#	4291#	4292#	4293#	4294#
4295	4296#	4297	4299#	4300#	4301	4302#	4303	4307#	4308#	4309#	4310#	4311	4312#	4313
4317#	4326#	4327#	4328#	4329#	4330	4331#	4332	4335#	4336#	4337#	4338#	4339#	4340	4341#
4342	4346#	4372#	4373#	4374#	4375#	4376	4377#	4378	4380#	4381#	4382#	4383#	4384#	4385#
4386	4387#	4388	4390#	4391#	4392#	4393	4394#	4395	4397#	4398#	4399#	4400#	4401#	4402#
4403	4404#	4405	4407#	4408#	4409	4410#	4411	4421#	4422#	4423#	4424#	4425	4426#	4427
4429#	4430#	4431#	4432#	4433#	4434#	4435	4436#	4437	4439#	4440#	4441#	4442	4443#	4444
4446#	4447#	4448#	4449#	4450#	4451#	4452	4453#	4454	4456#	4457#	4458#	4459	4460#	4461
4463#	4464#	4465#	4466#	4467#	4468#	4469	4470#	4471	4473#	4474#	4475#	4476	4477#	4478
4480#	4481#	4482#	4483#	4484#	4485#	4486	4487#	4488	4498#	4499#	4500#	4501#	4502	4503#
4504	4506#	4507#	4508#	4509#	4510#	4511#	4512	4513#	4514	4516#	4517#	4518#	4519	4520#
4521	4523#	4524#	4525#	4526#	4527#	4528#	4529	4530#	4531	4576#	4577#	4579#	4583#	4584#
4586#	4590#	4591#	4593#	4597#	4598#	4600#	4618#	4619#	4620#	4622#	4648#	4652#	4673#	4674#
4675#	4676#	4677#	4678	4695#	4696#	4700#	4701#	4707#	4725#	4737#	4740#	4753#	4775#	4777#
4778#	4784#	4786#	4787#	4803#	4804#	4805#	4806#	4808#	4809#	4822#	4823#	4824#	4825#	4829#
4856#	4872#	4873#	4874#	4875#	4877#	4882#	4886#	4894#	4896#	4897#	4908#	4909#	4910#	4911#
4914#	4918#	4923#	4924#	4925	4926#	4927	4929#	4930#	4931	4932#	4933	4935#	4936#	4937
4938#	4939	4941#	4942#	4943#	4944#	4945	4946#	4947	4951#	4952#	4953#	4954#	4955	4956#
4957	4959#	4960#	4961	4962#	4963	4966#	4971#	4972#	4973#	4974#	4975	4976#	4977	4981#
5037#	5038#	5039#	5040#	5042#	5043#	5049#	5089#	5091#	5092#	5097#	5152#	5153#	5154#	5155#
5157#	5158#	5169#	5189#	5191#	5192#	5200#	5208#	5210#	5211#	5215#	5221#	5241#	5243#	5244#
5252#	5260#	5262#	5263#	5267#	5273#	5295#	5297#	5298#	5306#	5314#	5316#	5317#	5321#	5327#
5355#	5357#	5358#	5366#	5386#	5387#	5388#	5389#	5391#	5392#	5396#	5402#	5421#	5423#	5424#
5432#	5440#	5442#	5443#	5447#	5453#	5473#	5475#	5476#	5495#	5515#	5517#	5518#	5537#	5591#
5592#	5593#	5594#	5596#	5597#	5617#	5618#	5619#	5620#	5622#	5623#	5643#	5644#	5645#	5646#
5650#	5688#	5690#	5691#	5709#	5711#	5712#	5720#	5722#	5723#	5728#	5730#	5731#	5735#	5775#
5777#	5778#	5785#	5787#	5788#	5795#	5797#	5798#	5805#	5807#	5808#	5814#	5816#	5817#	5825#
5827#	5828#	5833#	5835#	5836#	5839#	5873#	5875#	5876#	5882#	5884#	5885#	5892#	5894#	5895#
5910#	5912#	5913#	5921#	5923#	5924#	5933#	5934#	5935#	5936#	5938#	5939#	5948#	5950#	5951#
5960#	5962#	5963#	5972#	5973#	5974#	5975#	5977#	5978#	5986#	5988#	5989#	5996#	5998#	5999#
6012#	6014#	6015#	6022#	6024#	6025#	6037#	6039#	6040#	6049#	6050#	6051#	6052#	6054#	6055#
6064#	6066#	6067#	6070#	6101#	6103#	6104#	6110#	6112#	6113#	6120#	6122#	6123#	6138#	6140#
6141#	6149#	6151#	6152#	6161#	6162#	6163#	6164#	6166#	6167#	6176#	6178#	6179#	6180#	6190#
6191#	6200#	6201#	6202#	6203#	6205#	6206#	6218#	6220#	6221#	6228#	6230#	6231#	6241#	6243#
6244#	6249#	6251#	6252#	6255#	6283#	6285#	6286#	6301#	6303#	6304#	6311#	6313#	6314#	6327#
6329#	6330#	6338#	6340#	6341#	6349#	6351#	6352#	6355#	6385#	6387#	6388#	6407#	6409#	6410#
6430#	6432#	6433#	6447#	6449#	6450#	6458#	6460#	6461#	6467#	6469#	6470#	6473#	6511#	6517#
6519#	6520#	6537#	6539#	6540#	6555#	6557#	6558#	6565#	6571#	6577#	6579#	6580#	6597#	6599#
6600#	6615#	6617#	6618#	6625#	6631#	6637#	6639#	6640#	6657#	6659#	6660#	6675#	6677#	6678#
6685#	6688#	6731#	6733#	6734#	6740#	6742#	6743#	6750#	6752#	6753#	6768#	6770#	6771#	6782#
6784#	6785#	6792#	6794#	6795#	6798#	6828#	6830#	6831#	6837#	6839#	6840#	6847#	6849#	6850#
6865#	6867#	6868#	6879#	6881#	6882#	6889#	6891#	6892#	6895#	6925#	6927#	6928#	6934#	6936#
6937#	6944#	6946#	6947#	6962#	6964#	6965#	6976#	6978#	6979#	6986#	6988#	6989#	6992#	7019#
7021#	7022#	7037#	7039#	7040#	7047#	7049#	7050#	7057#	7059#	7060#	7067#	7069#	7070#	7076#
7078#	7079#	7087#	7089#	7090#	7097#	7099#	7100#	7108#	7110#	7111#	7118#	7120#	7121#	7129#
7131#	7132#	7140#	7142#	7143#	7151#	7153#	7154#	7157#	7193#	7199#	7201#	7202#	7215#	7217#
7218#	7225#	7227#	7228#	7238#	7240#	7241#	7251#	7253#	7254#	7271#	7273#	7274#	7282#	7284#
7285#	7297#	7299#	7300#	7311#	7313#	7314#	7322#	7324#	7325#	7328#	7334#	7340#	7342#	7343#
7356#	7358#	7359#	7366#	7368#	7369#	7379#	7381#	7382#	7392#	7394#	7395#	7412#	7414#	7415#
7423#	7425#	7426#	7438#	7440#	7441#	7452#	7454#	7455#	7463#	7465#	7466#	7469#	7475#	7481#
7483#	7484#	7497#	7499#	7500#	7507#	7509#	7510#	7520#	7522#	7523#	7533#	7535#	7536#	7553#
7555#	7556#	7564#	7566#	7567#	7579#	7581#	7582#	7593#	7595#	7596#	7604#	7606#	7607#	7610#
7616#	7622#	7624#	7625#	7638#	7640#	7641#	7648#	7650#	7651#	7661#	7663#	7664#	7674#	7676#

CVDMA.P11

10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

7677#	7694#	7696#	7697#	7705#	7707#	7708#	7720#	7722#	7723#	7734#	7736#	7737#	7745#	7747#	
7748#	7751#	7757#	7763#	7765#	7766#	7779#	7781#	7782#	7789#	7791#	7792#	7802#	7804#	7805#	
7815#	7817#	7818#	7835#	7837#	7838#	7846#	7848#	7849#	7861#	7863#	7864#	7875#	7877#	7878#	
7886#	7888#	7889#	7892#	7898#	7904#	7906#	7907#	7920#	7922#	7923#	7930#	7932#	7933#	7943#	
7945#	7946#	7956#	7958#	7959#	7976#	7978#	7979#	7987#	7989#	7990#	8002#	8004#	8005#	8016#	
8018#	8019#	8027#	8029#	8030#	8033#	8036#	8098#	8100#	8101#	8117#	8119#	8120#	8126#	8128#	
8129#	8140#	8142#	8143#	8149#	8151#	8152#	8168#	8170#	8171#	8182#	8184#	8185#	8192#	8194#	
8195#	8198#	8228#	8230#	8231#	8250#	8252#	8253#	8262#	8264#	8265#	8276#	8278#	8279#	8287#	
8289#	8290#	8301#	8303#	8304#	8320#	8322#	8323#	8331#	8333#	8334#	8342#	8344#	8345#	8351#	
8353#	8354#	8357#	8403#	8411#	8413#	8414#	8430#	8432#	8433#	8442#	8444#	8445#	8451#	8453#	
8454#	8457#	8461#	8469#	8471#	8472#	8489#	8491#	8492#	8495#	8499#	8507#	8509#	8510#	8540#	
8542#	8543#	8546#	8549#	8578#	8580#	8581#	8600#	8602#	8603#	8610#	8612#	8613#	8620#	8622#	
8623#	8630#	8632#	8633#	8639#	8641#	8642#	8652#	8654#	8655#	8662#	8664#	8665#	8672#	8674#	
8675#	8682#	8684#	8685#	8691#	8693#	8694#	8702#	8704#	8705#	8716#	8718#	8719#	8727#	8729#	
8730#	8739#	8741#	8742#	8746#	8793#	8795#	8796#	8815#	8817#	8818#	8825#	8827#	8828#	8835#	
8837#	8838#	8845#	8847#	8848#	8854#	8856#	8857#	8867#	8869#	8870#	8877#	8879#	8880#	8887#	
8889#	8890#	8897#	8899#	8900#	8907#	8909#	8910#	8917#	8919#	8920#	8927#	8929#	8930#	8936#	
8938#	8939#	8947#	8949#	8950#	8961#	8963#	8964#	8972#	8974#	8975#	8984#	8986#	8987#	8990#	
9027#	9029#	9030#	9037#	9039#	9040#	9050#	9052#	9053#	9060#	9062#	9063#	9070#	9072#	9073#	
9080#	9082#	9083#	9090#	9092#	9093#	9099#	9101#	9102#	9110#	9112#	9113#	9123#	9125#	9126#	
9135#	9137#	9138#	9147#	9149#	9150#	9157#	9159#	9160#	9166#	9168#	9169#	9172#	9187#	9191#	
9192	9193	9194	9196#	9197	9198	9199	9201#	9202	9203	9204	9205	9207#	9208	9209	
9210	9211	9213#	9214	9215	9216	9217	9219#	9220	9221	9222	9223	9225#	9226	9227	
9228	9229	9232#	9250#	9254#	9271#	9272#	9273#								
MSGNLS	1#	1297#													
MSGNSU	1#	1297#	4855#	4885#	5199#	5251#	5305#	5365#	5431#	6510#	6570#	6630#	7192#	7333#	7474#
MSGNTA	1#	1297#	7897#	8402#	8460#	8498#									
	7615#	7756#	7897#	8402#	8460#	8498#									
	1#	1297#	1481#	1495#	4131#	4158#	4192#	4236#	4281#	4316#	4345#	4651#	4706#	4724#	4739#
	4752#	4828#	4881#	4913#	4917#	4965#	4980#	5048#	5096#	5168#	5214#	5220#	5266#	5272#	5320#
	5326#	5395#	5401#	5446#	5452#	5494#	5536#	5649#	5734#	5838#	6069#	6254#	6354#	6472#	6564#
	6624#	6684#	6687#	6797#	6894#	6991#	7156#	7327#	7468#	7609#	7750#	7891#	8032#	8035#	8197#
	8356#	8456#	8494#	8545#	8548#	8745#	8989#	9171#	9232#	9233	9254#	9255			
MSGNTE	1#	1297#	4770#	4851#	5008#	5073#	5119#	5185#	5237#	5291#	5351#	5417#	5469#	5511#	5564#
	5679#	5766#	5865#	6093#	6274#	6377#	6504#	6723#	6820#	6917#	7011#	7186#	8061#	8220#	8399#
	8569#	8785#	9019#												
MSHAPT	1#	1297#	1324#												
MSHNAP	1#	1297#	1324#	1363											
MSINCR	1#	1297#	1300#	1465#	1490#	4098#	4109#	4125#	4132#	4137#	4144#	4154#	4159#	4164#	4173#
	4183#	4189#	4193#	4199#	4206#	4212#	4222#	4232#	4237#	4244#	4251#	4257#	4267#	4277#	4282#
	4289#	4296#	4302#	4312#	4317#	4324#	4331#	4341#	4346#	4377#	4387#	4394#	4404#	4410#	4426#
	4436#	4443#	4453#	4460#	4470#	4477#	4487#	4503#	4513#	4520#	4530#	4543#	4556#	4577#	4584#
	4591#	4598#	4619#	4648#	4652#	4670#	4677#	4696#	4701#	4707#	4720#	4725#	4734#	4737#	4740#
	4750#	4753#	4770#	4771#	4775#	4777#	4784#	4786#	4803#	4808#	4822#	4829#	4851#	4852#	4855#
	4856#	4872#	4877#	4882#	4885#	4886#	4894#	4896#	4908#	4914#	4918#	4921#	4926#	4932#	4938#
	4946#	4956#	4962#	4966#	4969#	4976#	4981#	5008#	5009#	5037#	5042#	5049#	5073#	5074#	5089#
	5091#	5097#	5119#	5120#	5152#	5157#	5169#	5185#	5186#	5189#	5191#	5199#	5200#	5208#	5210#
	5215#	5221#	5237#	5238#	5241#	5243#	5251#	5252#	5260#	5262#	5267#	5273#	5291#	5292#	5295#
	5297#	5305#	5306#	5314#	5316#	5321#	5327#	5351#	5352#	5355#	5357#	5365#	5366#	5386#	5391#
	5396#	5402#	5417#	5418#	5421#	5423#	5431#	5432#	5440#	5442#	5447#	5453#	5469#	5470#	5473#
	5475#	5495#	5511#	5512#	5515#	5517#	5537#	5564#	5565#	5591#	5596#	5617#	5622#	5643#	5650#
	5679#	5680#	5688#	5690#	5709#	5711#	5720#	5722#	5728#	5730#	5735#	5766#	5767#	5775#	5777#
	5785#	5787#	5795#	5797#	5805#	5807#	5814#	5816#	5825#	5827#	5833#	5835#	5839#	5865#	5866#
	5873#	5875#	5882#	5884#	5892#	5894#	5910#	5912#	5921#	5923#	5933#	5938#	5948#	5950#	5960#
	5962#	5972#	5977#	5986#	5988#	5996#	5998#	6012#	6014#	6022#	6024#	6037#	6039#	6049#	6054#
	6064#	6066#	6070#	6093#	6094#	6101#	6103#	6110#	6112#	6120#	6122#	6138#	6140#	6149#	6151#
	6161#	6166#	6176#	6178#	6188#	6190#	6200#	6205#	6218#	6220#	6228#	6230#	6241#	6243#	6249#

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

	4262#	4271#	4291#	4299#	4307#	4326#	4335#	4372#	4380#	4390#	4397#	4407#	4421#	4429#	4439#
	4446#	4456#	4463#	4473#	4480#	4498#	4506#	4516#	4523#	4673#	4923#	4929#	4935#	4941#	4951#
	4959#	4971#													
MSPUT1	1#	1297#	4105#	4106	4107	4120#	4121	4122	4123	4139#	4140	4141	4142	4148#	4149
	4150	4151	4152	4168#	4169	4170	4171	4177#	4178	4179	4180	4181	4186#	4187	4201#
	4202	4203	4204	4209#	4210	4217#	4218	4219	4220	4226#	4227	4228	4229	4230	4246#
	4247	4248	4249	4254#	4255	4262#	4263	4264	4265	4271#	4272	4273	4274	4275	4291#
	4292	4293	4294	4299#	4300	4307#	4308	4309	4310	4326#	4327	4328	4329	4335#	4336
	4337	4338	4339	4372#	4373	4374	4375	4380#	4381	4382	4383	4384	4385	4390#	4391
	4392	4397#	4398	4399	4400	4401	4402	4407#	4408	4421#	4422	4423	4424	4429#	4430
	4431	4432	4433	4434	4439#	4440	4441	4446#	4447	4448	4449	4450	4451	4456#	4457
	4458	4463#	4464	4465	4466	4467	4468	4473#	4474	4475	4480#	4481	4482	4483	4484
	4485	4498#	4499	4500	4501	4506#	4507	4508	4509	4510	4511	4516#	4517	4518	4523#
	4524	4525	4526	4527	4528	4673#	4674	4675	4676	4923#	4924	4929#	4930	4935#	4936
	4941#	4942	4943	4944	4951#	4952	4953	4954	4959#	4960	4971#	4972	4973	4974	
MSRADI	1#	1297#	9191#	9196#	9201#	9207#	9213#	9219#	9225#						
MSRBRO	1#	1297#													
MSRNRO	1#	1297#	4618#	4620											
MSSETS	1#	1297#	1300#	1465#	1490#	4098#	4137#	4164#	4199#	4244#	4289#	4324#	4543#	4556#	4670#
	4720#	4734#	4750#	4771#	4852#	4856#	4886#	4921#	4969#	5009#	5074#	5120#	5186#	5200#	5238#
	5252#	5292#	5306#	5352#	5366#	5418#	5432#	5470#	5512#	5565#	5680#	5767#	5866#	6094#	6275#
	6378#	6505#	6511#	6571#	6631#	6724#	6821#	6918#	7012#	7187#	7193#	7334#	7475#	7616#	7757#
	7898#	8062#	8221#	8400#	8403#	8461#	8499#	8570#	8786#	9020#	9187#	9250#			
MSSTAR	1#	1297#													
MS SVC	1#	1297#	4105#	4109	4120#	4125	4131#	4132	4139#	4144	4148#	4154	4158#	4159	4168#
	4173	4177#	4183	4186#	4189	4192#	4193	4201#	4206	4209#	4212	4217#	4222	4226#	4232
	4236#	4237	4246#	4251	4254#	4257	4262#	4267	4271#	4277	4281#	4282	4291#	4296	4299#
	4302	4307#	4312	4316#	4317	4326#	4331	4335#	4341	4345#	4346	4372#	4377	4380#	4387
	4390#	4394	4397#	4404	4407#	4410	4421#	4426	4429#	4436	4439#	4443	4446#	4453	4456#
	4460	4463#	4470	4473#	4477	4480#	4487	4498#	4503	4506#	4513	4516#	4520	4523#	4530
	4576#	4577	4583#	4584	4590#	4591	4597#	4598	4618#	4619	4648#	4651#	4652	4673#	4677
	4695#	4696	4700#	4701	4706#	4707	4724#	4725	4737#	4739#	4740	4752#	4753	4775#	4777#
	4784#	4786#	4803	4808#	4822	4828#	4829	4855#	4856	4872	4877#	4881#	4882	4885#	4886
	4894#	4896#	4908	4913#	4914	4917#	4918	4923#	4926	4929#	4932	4935#	4938	4941#	4946
	4951#	4956	4959#	4962	4965#	4966	4971#	4976	4980#	4981	5037	5042#	5048#	5049	5089#
	5091#	5096#	5097	5152	5157#	5168#	5169	5189#	5191#	5199#	5200	5208#	5210#	5214#	5215
	5220#	5221	5241#	5243#	5251#	5252	5260#	5262#	5266#	5267	5272#	5273	5295#	5297#	5305#
	5306	5314#	5316#	5320#	5321	5326#	5327	5355#	5357#	5365#	5366	5386	5391#	5395#	5396
	5401#	5402	5421#	5423#	5431#	5432	5440#	5442#	5446#	5447	5452#	5453	5473#	5475#	5494#
	5495	5515#	5517#	5536#	5537	5591	5596#	5617	5622#	5643	5649#	5650	5688#	5690#	5709#
	5711#	5720#	5722#	5728#	5730#	5734#	5735	5775#	5777#	5785#	5787#	5795#	5797#	5805#	5807#
	5814#	5816#	5825#	5827#	5833#	5835#	5838#	5839	5873#	5875#	5882#	5884#	5892#	5894#	5910#
	5912#	5921#	5923#	5933	5938#	5948#	5950#	5960#	5962#	5972	5977#	5986#	5988#	5996#	5998#
	6012#	6014#	6022#	6024#	6037#	6039#	6049	6054#	6064#	6066#	6069#	6070	6101#	6103#	6110#
	6112#	6120#	6122#	6138#	6140#	6149#	6151#	6161	6166#	6176#	6178#	6188#	6190#	6200	6205#
	6218#	6220#	6228#	6230#	6241#	6243#	6249#	6251#	6254#	6255	6283#	6285#	6301#	6303#	6311#
	6313#	6327#	6329#	6338#	6340#	6349#	6351#	6354#	6355	6385#	6387#	6407#	6409#	6430#	6432#
	6447#	6449#	6458#	6460#	6467#	6469#	6472#	6473	6510#	6511	6517#	6519#	6537#	6539#	6555#
	6557#	6564#	6565	6570#	6571	6577#	6579#	6597#	6599#	6615#	6617#	6624#	6625	6630#	6631
	6637#	6639#	6657#	6659#	6675#	6677#	6684#	6685	6687#	6688	6731#	6733#	6740#	6742#	6750#
	6752#	6768#	6770#	6782#	6784#	6792#	6794#	6797#	6798	6828#	6830#	6837#	6839#	6847#	6849#
	6865#	6867#	6879#	6881#	6889#	6891#	6894#	6895	6925#	6927#	6934#	6936#	6944#	6946#	6962#
	6964#	6976#	6978#	6986#	6988#	6991#	6992	7019#	7021#	7037#	7039#	7047#	7049#	7057#	7059#
	7067#	7069#	7076#	7078#	7087#	7089#	7097#	7099#	7108#	7110#	7118#	7120#	7129#	7131#	7140#
	7142#	7151#	7153#	7156#	7157	7192#	7193	7199#	7201#	7215#	7217#	7225#	7227#	7238#	7240#
	7251#	7253#	7271#	7273#	7282#	7284#	7297#	7299#	7311#	7313#	7322#	7324#	7327#	7328	7333#

CVDMCA.P11

10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

7334#	7340#	7342#	7356#	7358#	7366#	7368#	7379#	7381#	7392#	7394#	7412#	7414#	7423#	7425#	
7438#	7440#	7452#	7454#	7463#	7465#	7468#	7469	7474#	7475	7481#	7483#	7497#	7499#	7507#	
7509#	7520#	7522#	7533#	7535#	7553#	7555#	7564#	7566#	7579#	7581#	7593#	7595#	7604#	7606#	
7609#	7610	7615#	7616	7622#	7624#	7638#	7640#	7648#	7650#	7661#	7663#	7674#	7676#	7694#	
7696#	7705#	7707#	7720#	7722#	7734#	7736#	7745#	7747#	7750#	7751	7756#	7757	7763#	7765#	
7779#	7781#	7789#	7791#	7802#	7804#	7815#	7817#	7835#	7837#	7846#	7848#	7861#	7863#	7875#	
7877#	7886#	7888#	7891#	7892	7897#	7898	7904#	7906#	7920#	7922#	7930#	7932#	7943#	7945#	
7956#	7958#	7976#	7978#	7987#	7989#	8002#	8004#	8016#	8018#	8027#	8029#	8032#	8033	8035#	
8036	8098#	8100#	8117#	8119#	8126#	8128#	8140#	8142#	8149#	8151#	8168#	8170#	8182#	8184#	
8192#	8194#	8197#	8198	8228#	8230#	8250#	8252#	8262#	8264#	8276#	8278#	8287#	8289#	8301#	
8303#	8320#	8322#	8331#	8333#	8342#	8344#	8351#	8353#	8356#	8357	8402#	8403	8411#	8413#	
8430#	8432#	8442#	8444#	8451#	8453#	8456#	8457	8460#	8461	8469#	8471#	8489#	8491#	8494#	
8495	8498#	8499	8507#	8509#	8540#	8542#	8545#	8546	8548#	8549	8578#	8580#	8600#	8602#	
8610#	8612#	8620#	8622#	8630#	8632#	8639#	8641#	8652#	8654#	8662#	8664#	8674#	8672#	8682#	
8684#	8691#	8693#	8702#	8704#	8716#	8718#	8727#	8729#	8739#	8741#	8745#	8746	8793#	8795#	
8815#	8817#	8825#	8827#	8835#	8837#	8845#	8847#	8854#	8856#	8867#	8869#	8877#	8879#	8887#	
8889#	8897#	8899#	8907#	8909#	8917#	8919#	8927#	8929#	8936#	8938#	8947#	8949#	8961#	8963#	
8972#	8974#	8984#	8986#	8989#	8990	9027#	9029#	9037#	9039#	9050#	9052#	9060#	9062#	9070#	
9072#	9080#	9082#	9090#	9092#	9099#	9101#	9110#	9112#	9123#	9125#	9135#	9137#	9147#	9149#	
9157#	9159#	9166#	9168#	9171#	9172										
MSTLAB	1#	1297#	4109#	4125#	4132#	4144#	4154#	4159#	4173#	4183#	4189#	4193#	4206#	4212#	4222#
4232#	4237#	4251#	4257#	4267#	4277#	4282#	4296#	4302#	4312#	4317#	4331#	4341#	4346#	4377#	
4387#	4394#	4404#	4410#	4426#	4436#	4443#	4453#	4460#	4470#	4477#	4487#	4503#	4513#	4520#	
4530#	4577#	4584#	4591#	4598#	4619#	4648#	4652#	4677#	4696#	4701#	4707#	4725#	4737#	4740#	
4753#	4775#	4777#	4784#	4786#	4803#	4808#	4822#	4829#	4856#	4872#	4877#	4882#	4886#	4894#	
4896#	4908#	4914#	4918#	4926#	4932#	4938#	4946#	4956#	4962#	4966#	4976#	4981#	5037#	5042#	
5049#	5089#	5091#	5097#	5152#	5157#	5169#	5189#	5191#	5200#	5208#	5210#	5215#	5221#	5241#	
5243#	5252#	5260#	5262#	5267#	5273#	5295#	5297#	5306#	5314#	5316#	5321#	5327#	5355#	5357#	
5366#	5386#	5391#	5396#	5402#	5421#	5423#	5432#	5440#	5442#	5447#	5453#	5473#	5475#	5495#	
5515#	5517#	5537#	5591#	5596#	5617#	5622#	5643#	5650#	5688#	5690#	5709#	5711#	5720#	5722#	
5728#	5730#	5735#	5775#	5777#	5785#	5787#	5795#	5797#	5805#	5807#	5814#	5816#	5825#	5827#	
5833#	5835#	5839#	5873#	5875#	5882#	5884#	5892#	5894#	5910#	5912#	5921#	5923#	5933#	5938#	
5948#	5950#	5960#	5962#	5972#	5977#	5986#	5988#	5996#	5998#	6012#	6014#	6022#	6024#	6037#	
6039#	6049#	6054#	6064#	6066#	6070#	6101#	6103#	6110#	6112#	6120#	6122#	6138#	6140#	6149#	
6151#	6161#	6166#	6176#	6178#	6188#	6190#	6200#	6205#	6218#	6220#	6228#	6230#	6241#	6243#	
6249#	6251#	6255#	6283#	6285#	6301#	6303#	6311#	6313#	6327#	6329#	6338#	6340#	6349#	6351#	
6355#	6385#	6387#	6407#	6409#	6430#	6432#	6447#	6449#	6458#	6460#	6467#	6469#	6473#	6511#	
6517#	6519#	6537#	6539#	6555#	6557#	6565#	6571#	6577#	6579#	6597#	6599#	6615#	6617#	6625#	
6631#	6637#	6639#	6657#	6659#	6675#	6677#	6685#	6688#	6731#	6733#	6740#	6742#	6750#	6752#	
6768#	6770#	6782#	6784#	6792#	6794#	6798#	6828#	6830#	6837#	6839#	6847#	6849#	6865#	6867#	
6879#	6881#	6889#	6891#	6895#	6925#	6927#	6934#	6936#	6944#	6946#	6962#	6964#	6976#	6978#	
6986#	6988#	6992#	7019#	7021#	7037#	7039#	7047#	7049#	7057#	7059#	7067#	7069#	7076#	7078#	
7087#	7089#	7097#	7099#	7108#	7110#	7118#	7120#	7129#	7131#	7140#	7142#	7151#	7153#	7157#	
7193#	7199#	7201#	7215#	7217#	7225#	7227#	7238#	7240#	7251#	7253#	7271#	7273#	7282#	7284#	
7297#	7299#	7311#	7313#	7322#	7324#	7328#	7334#	7340#	7342#	7356#	7358#	7366#	7368#	7379#	
7381#	7392#	7394#	7412#	7414#	7423#	7425#	7438#	7440#	7452#	7454#	7463#	7465#	7469#	7475#	
7481#	7483#	7497#	7499#	7507#	7509#	7520#	7522#	7533#	7535#	7553#	7555#	7564#	7566#	7579#	
7581#	7593#	7595#	7604#	7606#	7610#	7616#	7622#	7624#	7638#	7640#	7648#	7650#	7661#	7663#	
7674#	7676#	7694#	7696#	7705#	7707#	7720#	7722#	7734#	7736#	7745#	7747#	7751#	7756#	7763#	
7765#	7779#	7781#	7789#	7791#	7802#	7804#	7815#	7817#	7835#	7837#	7846#	7848#	7861#	7863#	
7875#	7877#	7886#	7888#	7892#	7898#	7904#	7906#	7920#	7922#	7930#	7932#	7943#	7945#	7956#	
7958#	7976#	7978#	7987#	7989#	8002#	8004#	8016#	8018#	8027#	8029#	8033#	8036#	8098#	8100#	
8117#	8119#	8126#	8128#	8140#	8142#	8149#	8151#	8168#	8170#	8182#	8184#	8192#	8194#	8198#	
8228#	8230#	8250#	8252#	8262#	8264#	8276#	8278#	8287#	8289#	8301#	8303#	8320#	8322#	8331#	
8333#	8342#	8344#	8351#	8353#	8357#	8403#	8411#	8413#	8430#	8432#	8442#	8444#	8451#	8453#	
8457#	8461#	8469#	8471#	8489#	8491#	8495#	8499#	8507#	8509#	8540#	8542#	8546#	8549#	8578#	

CVDMCA.P11

10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

MSTSTL

8580#	8600#	8602#	8610#	8612#	8620#	8622#	8630#	8632#	8639#	8641#	8652#	8654#	8662#	8664#
8672#	8674#	8682#	8684#	8691#	8693#	8702#	8704#	8716#	8718#	8727#	8729#	8739#	8741#	8746#
8793#	8795#	8815#	8817#	8825#	8827#	8835#	8837#	8845#	8847#	8854#	8856#	8867#	8869#	8877#
8879#	8887#	8889#	8897#	8899#	8907#	8909#	8917#	8919#	8927#	8929#	8936#	8938#	8947#	8949#
8961#	8963#	8972#	8974#	8984#	8986#	8990#	9027#	9029#	9037#	9039#	9050#	9052#	9060#	9062#
9070#	9072#	9080#	9082#	9090#	9092#	9099#	9101#	9110#	9112#	9123#	9125#	9135#	9137#	9147#
9149#	9157#	9159#	9166#	9168#	9172#									
1#	1297#	4109#	4125#	4132#	4144#	4154#	4159#	4173#	4183#	4189#	4193#	4206#	4212#	4222#
4232#	4237#	4251#	4257#	4267#	4277#	4282#	4296#	4302#	4312#	4317#	4331#	4341#	4346#	4377#
4387#	4394#	4404#	4410#	4426#	4436#	4443#	4453#	4460#	4470#	4477#	4487#	4503#	4513#	4520#
4530#	4577#	4584#	4591#	4598#	4619#	4648#	4652#	4677#	4696#	4701#	4707#	4725#	4737#	4740#
4753#	4775#	4777#	4784#	4786#	4803#	4808#	4822#	4829#	4856#	4872#	4877#	4882#	4886#	4894#
4896#	4908#	4914#	4918#	4926#	4932#	4938#	4946#	4956#	4962#	4966#	4976#	4981#	5037#	5042#
5049#	5089#	5091#	5097#	5152#	5157#	5169#	5189#	5191#	5200#	5208#	5210#	5215#	5221#	5241#
5243#	5252#	5260#	5262#	5267#	5273#	5295#	5297#	5306#	5314#	5316#	5321#	5327#	5355#	5357#
5366#	5386#	5391#	5396#	5402#	5421#	5423#	5432#	5440#	5442#	5447#	5453#	5473#	5475#	5495#
5515#	5517#	5537#	5591#	5596#	5617#	5622#	5643#	5650#	5688#	5690#	5709#	5711#	5720#	5722#
5728#	5730#	5735#	5775#	5777#	5785#	5787#	5795#	5797#	5805#	5807#	5814#	5816#	5825#	5827#
5833#	5835#	5839#	5873#	5875#	5882#	5884#	5892#	5894#	5910#	5912#	5921#	5923#	5933#	5938#
5948#	5950#	5960#	5962#	5972#	5977#	5986#	5988#	5996#	5998#	6012#	6014#	6022#	6024#	6037#
6039#	6049#	6054#	6064#	6066#	6070#	6101#	6103#	6110#	6112#	6120#	6122#	6138#	6140#	6149#
6151#	6161#	6166#	6176#	6178#	6188#	6190#	6200#	6205#	6218#	6220#	6228#	6230#	6241#	6243#
6249#	6251#	6255#	6283#	6285#	6301#	6303#	6311#	6313#	6327#	6329#	6338#	6340#	6349#	6351#
6355#	6385#	6387#	6407#	6409#	6430#	6432#	6447#	6449#	6458#	6460#	6467#	6469#	6473#	6511#
6517#	6519#	6537#	6539#	6555#	6557#	6565#	6571#	6577#	6579#	6597#	6599#	6615#	6617#	6625#
6631#	6637#	6639#	6657#	6659#	6675#	6677#	6685#	6688#	6731#	6733#	6740#	6742#	6750#	6752#
6768#	6770#	6782#	6784#	6792#	6794#	6798#	6828#	6830#	6837#	6839#	6847#	6849#	6865#	6867#
6879#	6881#	6889#	6891#	6895#	6925#	6927#	6934#	6936#	6944#	6946#	6962#	6964#	6976#	6978#
6986#	6988#	6992#	7019#	7021#	7037#	7039#	7047#	7049#	7057#	7059#	7067#	7069#	7076#	7078#
7087#	7089#	7097#	7099#	7108#	7110#	7118#	7120#	7129#	7131#	7140#	7142#	7151#	7153#	7157#
7193#	7199#	7201#	7215#	7217#	7225#	7227#	7238#	7240#	7251#	7253#	7271#	7273#	7282#	7284#
7297#	7299#	7311#	7313#	7322#	7324#	7328#	7334#	7340#	7342#	7356#	7358#	7366#	7368#	7379#
7381#	7392#	7394#	7412#	7414#	7423#	7425#	7438#	7440#	7452#	7454#	7463#	7465#	7469#	7475#
7481#	7483#	7497#	7499#	7507#	7509#	7520#	7522#	7533#	7535#	7553#	7555#	7564#	7566#	7579#
7581#	7593#	7595#	7604#	7606#	7610#	7616#	7622#	7624#	7638#	7640#	7648#	7650#	7661#	7663#
7674#	7676#	7694#	7696#	7705#	7707#	7720#	7722#	7731#	7736#	7745#	7747#	7751#	7757#	7763#
7765#	7779#	7781#	7789#	7791#	7802#	7804#	7815#	7817#	7835#	7837#	7846#	7848#	7861#	7863#
7875#	7877#	7886#	7888#	7892#	7898#	7904#	7906#	7920#	7922#	7930#	7932#	7943#	7945#	7956#
7958#	7976#	7978#	7987#	7989#	8002#	8004#	8016#	8018#	8027#	8029#	8033#	8036#	8098#	8100#
8117#	8119#	8126#	8128#	8140#	8142#	8149#	8151#	8168#	8170#	8182#	8184#	8192#	8194#	8198#
8228#	8230#	8250#	8252#	8262#	8264#	8276#	8278#	8287#	8289#	8301#	8303#	8320#	8322#	8331#
8333#	8342#	8344#	8351#	8353#	8357#	8403#	8411#	8413#	8430#	8432#	8442#	8444#	8451#	8453#
8457#	8461#	8469#	8471#	8489#	8491#	8495#	8499#	8507#	8509#	8540#	8542#	8546#	8549#	8578#
8580#	8600#	8602#	8610#	8612#	8620#	8622#	8630#	8632#	8639#	8641#	8652#	8654#	8662#	8664#
8672#	8674#	8682#	8684#	8691#	8693#	8702#	8704#	8716#	8718#	8727#	8729#	8739#	8741#	8746#
8793#	8795#	8815#	8817#	8825#	8827#	8835#	8837#	8845#	8847#	8854#	8856#	8867#	8869#	8877#
8879#	8887#	8889#	8897#	8899#	8907#	8909#	8917#	8919#	8927#	8929#	8936#	8938#	8947#	8949#
8961#	8963#	8972#	8974#	8984#	8986#	8990#	9027#	9029#	9037#	9039#	9050#	9052#	9060#	9062#
9070#	9072#	9080#	9082#	9090#	9092#	9099#	9101#	9110#	9112#	9123#	9125#	9135#	9137#	9147#
9149#	9157#	9159#	9166#	9168#	9172#									
1#	1297#	1363#	1372	1422#	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433
1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448
1449	1450	1451	1452	1453	1454	4803#	4804	4805	4806	4822#	4823	4824	4825	4872#
4873	4874	4875	4908#	4909	4910	4911	5037#	5038	5039	5040	5152#	5153	5154	5155
5386#	5387	5388	5389	5591#	5592	5593	5594	5617#	5618	5619	5620	5643#	5644	5645
5646	5933#	5934	5935	5936	5972#	5973	5974	5975	6049#	6050	6051	6052	6161#	6162

MSWORD

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

	6163	6164	6200#	6201	6202	6203	9191#	9196#	9201#	9207#	9213#	9219#	9225#	9272	9273
MSXFER	1#	1297#													
NEWTST	2007#	4755	4831	4985	5050	5098	5170	5222	5274	5328	5403	5454	5496	5538	5651
	5736	5841	6071	6256	6356	6488	6702	6799	6896	6993	7158	8047	8199	8373	8550
	8764	9005													
NTST	2007#	4755	4831	4985	5050	5098	5170	5222	5274	5328	5403	5454	5496	5538	5651
	5736	5841	6071	6256	6356	6488	6702	6799	6896	6993	7158	8047	8199	8373	8550
	8764	9005													
OPEN	1#	1297#													
POINTE	1#	1297#	1320												
PRINTB	1#	1297#	4138	4147	4167	4176	4185	4200	4208	4216	4225	4245	4253	4261	4270
	4290	4298	4306	4325	4334	4406	4922	4958	4970						
PRINTF	1#	1297#													
PRINTS	1#	1297#													
PRINTX	1#	1297#	4104	4119	4371	4379	4389	4396	4420	4428	4438	4445	4455	4462	4472
	4479	4497	4505	4515	4522	4928	4934	4940	4950						
READBU	1#	1297#													
READEF	1#	1297#	4575	4582	4589	4596									
RFLAGS	1#	1297#													
SETDF	2007#	2437	2481	2528	2599	2698	2971	3075	3108	3119	3153	3164	3198	3209	3243
	3254	3288	3299	3331	3342	3378	3389	3402	3413	3445	3457	3506	3753	3766	3792
	3802	3815	3825	3838	3848	3861	3871	3884	3894						
SETHRD	2007#														
SETPRI	1#	1297#													
SETSF	2007#														
SETSFT	2007#														
SETVEC	1#	1297#	4672												
SLASH	1#	1297#	1414	1418											
STARS	1#	1297#													
SVC	1#	1295#	1296												
T\$GEN	2007#	2437	2481	2528	2599	2698	2971	3075	3108	3119	3153	3164	3198	3209	3243
	3254	3288	3299	3331	3342	3378	3389	3402	3413	3445	3457	3506	3753	3766	3792
	3802	3815	3825	3838	3848	3861	3871	3884	3894						
XFER	1#	1297#													
XFERF	1#	1297#													
XFERT	1#	1297#													
\$GEDF	2007#	4802	4821	4871	4907	5036	5151	5385	5590	5616	5642	5932	5971	6048	6160
	6199														
\$GEHRD	2007#														
\$GESF	2007#														
\$GESFT	2007#														
\$GTDF	2007#	2436	2480	2527	2598	2697	2970	3074	3107	3118	3152	3163	3197	3208	3242
	3253	3287	3298	3330	3341	3377	3388	3401	3412	3444	3456	3505	3752	3765	3791
	3801	3814	3824	3837	3847	3860	3870	3883	3893						
\$GTHRD	2007#														
\$GTSF	2007#														
\$GTSFT	2007#														

. ABS. 037426 000

ERRORS DETECTED: 0

CVDMCA.BIN, CVDMCA.SEQ/CRF/SOL=SVC34R.MAC, CVDMCA.P11
RUN-TIME: 40 52 6 SECONDS

CVDMCA.P11 10-DEC-80 09:14

CROSS REFERENCE TABLE -- MACRO NAMES

RUN-TIME RATIO: 204/99=2.0
CORE USED: 24K (47 PAGES)