

DMC11

DDCMP LINE UNIT TESTS
MD-11-DZDME-A

EP-DZDME-A-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZDME-A-D
PRODUCT NAME:	DDCMP MODE LINE UNIT TESTS
DATE:	APRIL 1976
MAINTAINER:	DIAGNOSTICS
AUTHOR:	FAY BASHAW

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.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976 BY DIGITAL EQUIPMENT CORPORATION

.REM
 DZDME
 MACY11
 27(732)
 22-SEP-76
 13:19
 PAGE 2
 DZDME
 A.P11

4. STARTING PROCEDURE

- A. SET SWITCH REGISTER TO 000200
- B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
- C. SET SWR TO ZERO FOR 'AUTO SIZING' OR SWR BIT0=1 FOR MANUAL INPUT (QUESTIONS) OR SWR BIT7=1 TO USE EXISTING PARAMETERS SET UP BY A PREVIOUS START OR A PREVIOUSLY RUN DMC11 DIAGNOSTIC.
- D. DEPRESS 'START KEY' AND RELEASE. THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO THE FOLLOWING:

MAP OF DMC11 STATUS

PC	CSR	STAT1	STAT2	STAT3
---	---	-----	-----	-----
001500	160010	145310	177777	000000
001510	160020	145320	177777	000000

THE PROGRAM WILL TYPE 'R' AND PROCEED TO RUN THE DIAGNOSTIC. THE ABOVE IS ONLY AN EXAMPLE. THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD. 1500 IN THE PROGRAM. IN THIS EXAMPLE THE TABLE CONTAINS THE INFORMATION AND STATUS OF TWO DMC11'S. THE STATUS TABLE MUST BE VERIFIED BY THE USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS TABLE SEE SECTION 8.4 FOR HELP.

IF THE DIAGNOSTIC WAS STARTED WITH SW00=1 INDICATING MANUAL PARAMETER INPUT THEN THE FOLLOWING SHOWS AN EXAMPLE OF THE QUESTIONS ASKED AND SOME EXAMPLE ANSWERS:

HOW MANY DMC11'S TO BE TESTED?1
 01
 CSR ADDRESS?160010
 VECTOR ADDRESS?310
 BR PRIORITY LEVEL? (4,5,6,7)?5
 DOES MICRO-PROCESSOR HAVE CRAM? (Y OR N)N
 WHICH LINE UNIT? IF NONE TYPE "N". IF M8201 TYPE "1". IF M8202 TYPE "2"?1
 IS THE LOOP BACK CONNECTOR ON?Y
 SWITCH PAC#1 (DDCMP LINE#)?377
 SWITCH PAC#2 (BMS73 BOOT ADD)?377

FOLLOWING THE QUESTIONS THE STATUS MAP IS PRINTED OUT AS DESCRIBED ABOVE. THE INFORMATION IN THE MAP REFLECTS THE ANSWERS TO THE QUESTIONS. IF THE DIAGNOSTIC WAS STARTED WITH SW00=0 AND SW07=0 (AUTO-SIZING) THEN NO QUESTIONS ARE ASKED AND ONLY THE STATUS-MAP IS PRINTED OUT. IF AUTO-SIZING IS USED THE STATUS INFORMATION MUST BE VERIFIED TO BE CORRECT (MATCH THE HARDWARE). IF IT DOES NOT MATCH THE HARDWARE THE

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

F01

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 6
DZDME.A.F11

191
192

DIAGNOSTIC MUST BE RESTARTED WITH SWOO=1 AND THE QUESTIONS
ANSWERED.

193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221

4.1 CONTROL SWITCH SETTINGS

- SW 15 SET: HALT ON ERROR
- SW 14 SET: LOOP ON CURRENT TEST
- SW 13 SET: INHIBIT ERROR PRINT OUT
- SW 12 SET: INHIBIT TYPE OUT/ABELL ON ERROR.
- SW 11 SET: INHIBIT ITERATIONS. (QUICK PASS)
- SW 10 SET: ESCAPE TO NEXT TEST ON ERROR
- SW 09 SET: LOOP WITH CURRENT DATA
- SW 08 SET: CATCH ERROR AND LOOP ON IT
- SW 07 SET: USE PREVIOUS STATUS TABLE.
- SW 06 SET: HALT IN ROMCLK ROUTINE BEFORE CLOCKING
MICRO-PROCESSOR
- SW 05 SET: RESERVED
- SW 04 SET: RESERVED
- SW 03 SET: RESELECT DMC11'S DESIRED ACTIVE
- SW 02 SET: LOCK ON SELECTED TEST
- SW 01 SET: RESTART PROGRAM AT SELECTED TEST
- SW 00 SET: BUILD NEW STATUS TABLE FROM QUESTIONS. (IF SW07=0
AND SW00=0 A NEW STATUS TABLE IS BUILT BY
AUTO-SIZING)

SWITCH 06 AND 08-15 ARE DYNAMIC AND CAN BE CHANGED AS NEEDED WHILE THE DIAGNOSTIC IS RUNNING. SWITCHES 00-03 AND SWITCH 07 ARE STATIC, AND ARE USED ONLY ON STARTING OR RESTARTING THE DIAGNOSTIC.

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

7.2 OPERATING RESTRICTIONS

THE FIRST TIME A DMC11 DIAGNOSTIC IS LOADED INTO CORE AND RUN THE STATUS TABLE MUST BE SET UP. THIS IS DONE BY MANUAL INPUT (SW00=1) OR BY AUTOSIZING (SW00=0 AND SW07=0). THEREAFTER HOWEVER THE STATUS TABLE NEED NOT BE SETUP BY SUBSEQUENT RESTARTS OR EVEN LOADING THE NEXT DMC DIAGNOSTIC BECAUSE THE STATUS TABLE IS OVERLAYED. THE CURRENT PARAMETERS IN THE STATUS TABLE ARE USED WHEN SW07=1 ON START UP.

7.3 HARDWARE CONFIGURATION RESTRICTIONS

DMC11(M8200)- JUMPER W1 MUST BE IN, AND SWITCH 7 OF E76 MUST BE IN THE OFF POSITION.

IOP(M8204)- JUMPER W1 MUST BE IN.

LINE UNIT(M8201)- JUMPERS W1, W2, AND W4 MUST BE IN. JUMPERS W3, AND W5 MUST BE OUT. SW8 OF E26 MUST BE IN THE ON POSITION.

LINE UNIT (M8202)- JUMPER W1 MUST BE IN. SW8 OF E26 MUST BE IN THE OFF POSITION.

8. MISCELLANEOUS

8.1 EXECUTION TIME

ALL DMC11 DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS' MESSAGE (PROVIDING NO ERRORS AND SW12=0) WITHIN 4 MINS. THIS IS ASSUMING SW11=1 (DELETE ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION. THE ACTUAL EXECUTION TIME DEPENDS GREATLY ON THE PDP11 CPU CONFIGURATION AND THE AMOUNT OF MEMORY IN THE SYSTEM.

8.2 PASS COMPLETE

NOTE: EVERY TIME THE PROGRAM IS STARTED; THE TESTS WILL RUN AS IF SW11 (DELETE ITERATIONS) WAS UP (=1). THIS IS TO 'VERIFY NO HARD ERRORS' AS SOON AS POSSIBLE. THEREFORE THE FIRST PASS -EACH TIME PROGRAM IS STARTED- WILL BE A 'QUICK PASS' UNTIL ALL DMC11'S IN SYSTEM ARE TESTED. WHEN THE DIAGNOSTIC HAS COMPLETED A PASS THE FOLLOWING IS AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.

END PASS DZDMC CSR: 175000 VEC: 0300 PASSES: 000001
ERRORS: 000000

NOTE: THE PASS COUNT AND ERROR COUNTS ARE CUMMULITIVE FOR EACH DMC11 THAT IS RUNNING, AND ARE SET TO ZERO ONLY WHEN THE DIAGNOSTIC IS STARTED. THEREFORE AFTER AN OVERNIGHT RUN FOR EXAMPLE, THE TOTAL PASSES AND ERRORS FOR EACH DMC11 SINCE THE DIAGNOSTIC WAS STARTED ARE

L01

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 12
DZDMEA.P11

420

REFLECTED IN PASSES: AND ERRORS:.

2 3

421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476

8.4 KEY LOCATIONS

RETURN (1214) CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS ASSERTED.

NEXT (1216) CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PERFORMED.

TSTNO (1226) CONTAINS THE NUMBER OF THE TEST NOW BEING PERFORMED.

RUN (1316) THE BIT IN 'RUN' ALWAYS POINTS TO THE DMC11 CURRENTLY BEING TESTED. EXAMPLE: (RUN) 1302/0000000001000000 MEANS THAT DMC11 NO.06 IS THE DMC11 NOW RUNNING.

DMCROO-DMCR17
DMSTOO-DMST17
(1500)-(1640)

THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 16 (DECIMAL) DMC11S SEQUENTIALY. THEY CONTAIN THE CSR VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DMC11.

DMACTV (1306) EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DMC11 WILL BE TESTED IN TURN. EXAMPLE: (DMACTV) 1276/0000000000011111 MEANS THAT DMC11 NO. 00,01,02,03,04 WILL BE TESTED. EXAMPLE: (DMACTV) 1276/0000000000010001 MEANS THAT DMC11 NO. 00,04 WILL BE TESTED.

DMCSR (1402) CONTAINS THE CSR OF THE CURRENT DMC11 UNDER TEST.

8.4A 'STATUS TABLE' (1500-1640)

THE TABLE IS FILLED BY AUTO SIZING OR BY THE MANUAL PARAMETER INPUT (QUESTIONS) AS DESCRIBED PREVIOUSLY. ALSO IF DESIRED BY USER; THE LOCATIONS MAY BE ALTERED BY HAND (TOGGLED IN) TO SUIT THE SPECIFIC CONFIGURATION.

THE EXAMPLE STATUS MAP SHOWN BELOW CONTAINS INFORMATION FOR TWO DMC11'S. THE TABLE CAN CONTAIN UP TO 16 DMC11'S. FOLLOWING THE MAP IS A DESCRIPTION OF THE BITS FOR EACH MAP ENTRY

MAP OF DMC11 STATUS

PC	CSR	STAT1	STAT2	STAT3
001500	160010	145310	177777	000000

NO1.

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 14
DZDMEA.F11

477

001510 160020 016320 000000 000000

002

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 17
DZDMEA.F11

562

THE DIAGNOSTIC. THE FOLLOWING MESSAGE WILL BE DISPLAYED:

000000
 000001
 000002
 000003
 000004
 000005
 000006
 000007

 177776
 001200

 005746
 005726
 010046
 012600
 024646
 022626

 100000
 040000
 020000
 010000
 004000
 002000
 001000
 000400
 000200
 000100
 000040
 000020
 000010
 000004
 000002
 000001

:REGISTER DEFINITIONS

```

-----
R0=%0      :GENERAL REGISTER
R1=%1      :GENERAL REGISTER
R2=%2      :GENERAL REGISTER
R3=%3      :GENERAL REGISTER
R4=%4      :GENERAL REGISTER
R5=%5      :GENERAL REGISTER
SP=%6      :PROCESSOR STACK POINTER
PC=%7      :PROGRAM COUNTER
  
```

:LOCATION EQUIVALENCIES

```

-----
PS=177776  :PROCESSOR STATUS WORD
STACK=1200 :START OF PROCESSOR STACK
  
```

:INSTRUCTION DEFINITIONS

```

-----
PUSH1SP=5746 :DECREMENT PROCESSOR STACK 1 WORD
POP1SP=5726  :INCREMENT PROCESSOR STACK 1 WORD
PUSHR0=10046 :SAVE R0 ON STACK
POPR0=12600  :RESTORE R0 FROM STACK
PUSH2SP=24646 :DECREMENT STACK TWICE
POP2SP=22626 :INCREMENT STACK TWICE
.EQUIV EMT,HLT :BASIC DEFINITION OF ERROR CALL
  
```

:BIT DEFINITIONS

```

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

696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
(2)
(2)
735
736
737
738
739
740
741

000000

000024
000024 005240
000026 000340
000030 004656
000032 000340
000034 004624
000036 000340
000040
000040 000000
000042 000000
000044 000000
000046 003432
000052
000052 000000
000174
000174 000000
000176 000000
000200 000200
000200 000137 002002
001000
001000 005377 040515 047111
001025 104 041515 030461
001200
001200 177570
001202 177570

```
*****
-----
:TRAPCATCHER FOR ILLEGAL INTERRUPTS
:THE STANDARD "TRAP CATCHER" IS PLACED
:BETWEEN ADDRESS 0 TO ADDRESS 776.
:IT LOOKS LIKE "PC+2 HALT".
-----
*****

.=0
:STANDARD INTERRUPT VECTORS
-----

.=24
      .FFAIL           :POWER FAIL HANDLER
      340              :SERVICE AT LEVEL 7
      .HLT             :ERROR HANDLER
      340              :SERVICE AT LEVEL 7
      .TRPSRV         :GENERAL HANDLER DISPATCH SERVICE
      340              :SERVICE AT LEVEL 7

.=40
      0                :SAVE FOR ACT-11 OR XXDP
      0                :RETURN ADDRESS IF UNDER ACT-11 OR XXDP
      0                :SAVE FOR ACT-11 OR XXDP
      $ENDAD          :FOR USE WITH ACT-11 OR XXDP

.=52
      0                :ACT-11 PROGRAM CHARACTERISTICS

.=174
DISPREG: 0           :SOFTWARE DISPLAY REGISTER
SWREG: 0            :SOFTWARE SWITCH REGISTER

.=200
      JMP      .START  :GO TO START OF PROGRAM

.=1000
MTITLE: .ASCII <377><12>/MAINDEC-11-DZDME-A/<377>
        .ASCIZ  /DMC11 DDCMP LINE UNIT TESTS/<377>

.=1200
:INDIRECT POINTERS TO SWITCH REGISTER AND LIGHT DISPLAY
-----

DISPLAY:177570
SWR:    177570
```

```

742
743
744      ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
745      -----
746 001204 177560 TKCSR: 177560      ;TELETYPE KEYBOARD CONTROL REGISTER
747 001206 177562 TKDBR: 177562      ;TELETYPE KEYBOARD DATA BUFFER
748 001210 177564 TPCSR: 177564      ;TELEPRINTER CONTROL REGISTER
749 001212 177566 TPDBR: 177566      ;TELEPRINTER DATA BUFFER
750
751      ;PROGRAM CONTROL PARAMETERS
752      -----
753
754 001214 000000 RETURN: 0      ;SCOPE ADDRESS FOR LOOP ON TEST
755 001216 000000 NEXT: 0      ;ADDRESS OF NEXT TEST TO BE EXECUTED
756 001220 000000 LOCK: 0      ;ADDRESS FOR LOCK ON CURRENT DATA
757 001222 000003 ICOUNT: 3     ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE
758 001224 000000 LPCNT: 0     ;NUMBER OF ITERATIONS COMPLETED
759 001226 000000 TSTNO: 0     ;NUMBER OF TEST IN PROGRESS
760 001230 000000 PASCNT: 0    ;NUMBER OF PASSES COMPLETED
761 001232 000000 ERRCNT: 0    ;TOTAL NUMBER OF ERRORS
762 001234 000000 LSTERR: 0    ;PC OF LAST ERROR CALL
763
764      ;PROGRAM VARIABLES
765      -----
766
767 001236 000000 STRTSW: 0      ;SWITCHES AT START OF PROGRAM
768 001240 000000 STAT: 0      ;DM STATUS WORD STORAGE
769 001242 000000 CLKX: 0
770 001244 000000 MASKX: 0
771 001246 000000 TEMP1: 0      ;TEMPORARY STORAGE
772 001250 000000 TEMP2: 0      ;TEMPORARY STORAGE
773 001252 000000 TEMP3: 0      ;TEMPORARY STORAGE
774 001254 000000 TEMP4: 0      ;TEMPORARY STORAGE
775 001256 000000 TEMP5: 0      ;TEMPORARY STORAGE
776 001260 000000 SAVR0: 0      ;R0 STORAGE
777 001262 000000 SAVR1: 0      ;R1 STORAGE
778 001264 000000 SAVR2: 0      ;R2 STORAGE
779 001266 000000 SAVR3: 0      ;R3 STORAGE
780 001270 000000 SAVR4: 0      ;R4 STORAGE
781 001272 000000 SAVR5: 0      ;R5 STORAGE
782 001274 000000 SAVSP: 0      ;STACK POINTER STORAGE
783 001276 000000 SAVPC: 0      ;PROGRAM COUNTER STORAGE
784 001300 000000 ZERO: 0
785 001302 000001 ONE: 1
786 001304 000000 MEM LIM: 0      ;HIGHEST LOCATION FOR NPR'S
787 001306 000001 DMACTV: .BLKW 1 ;DMC11'S SELECTED ACTIVE.
788 001310 000001 DMNUM: .BLKW 1  ;OCTAL NUMBER OF DMC11'S.
789 001312 000001 SAVACT: .BLKW 1 ;ORIGINAL ACTV DEVICES
790 001314 000001 SAVNUM: .BLKW 1 ;WORKABLE NUMBER
791 001316 000000 RUN: 0      ;POINTER TO RUNNING DEVICE.
792      .EVEN
793 001320 001472 CREAM: DM.MAP-6 ;TABLE POINTER.
794 001322 001676 MILK: CNT.MAP-4 ;TABLE POINTER
  
```

```

795
796
797
798
799 001324    000
800 001325    000
801 001326    000
802 001327    000
803
804
805
806
807
808
809
810
811
812 001330    104400
813      003506
814 001330    104401
815      003644
816 001332    104402
817      003674
818 001334    104403
819      003756
820 001336    104404
821      004062
822 001340    104405
823      004102
824 001342    104406
825      004302
826 001344    104407
827      004342
828 001346    104410
829      004374
830 001350    104411
831      004400
832 001352    104412
833      005370
834 001354    104413
835      005340
836 001356    104414
837      005406
838 001360    104415
839      005454
840 001362    104416
841      005520
842 001364
843
844
845

```

```

;PROGRAM CONTROL FLAGS
-----
INIFLG: .BYTE   0      ;PROGRAM INITIALIZATION FLAG
ERRFLG: .BYTE   0      ;ERROR OCCURED FLAG
LOKFLG: .BYTE   0      ;LOCK ON CURRENT TEST FLAG
QV.FLG: .BYTE   0      ;QUICK VERIFY FLAG.
                               ;ON FIRST PASS OF EACH DMC11 ITERATIONS WILL BE
.EVEN

;DEFINITIONS FOR TRAP SUBROUTINE CALLS
;POINTERS TO SUBROUTINES CAN BE FOUND
;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS

:*****
-----
:TRPTAG:
SCOPE=TRAP+0      ;CALL TO SCOPE LOOP AND ITERATION HANDLER
      .SCOPE
SCOPI=TRAP+1      ;CALL TO LOOP ON CURRENT DATA HANDLER
      .SCOPI
TYPE=TRAP+2       ;CALL TO TELETYPE OUTPUT ROUTINE
      .TYPE
INSTR=TRAP+3      ;CALL TO ASCII STRING INPUT ROUTINE
      .INSTR
INSTER=TRAP+4     ;CALL TO INPUT ERROR HANDLER
      .INSTER
PARAM=TRAP+5      ;CALL TO NUMERICAL DATA INPUT ROUTINE
      .PARAM
SAVOS=TRAP+6      ;CALL TO REGISTER SAVE ROUTINE
      .SAVOS
RESOS=TRAP+7      ;CALL TO REGISTER RESTORE ROUTINE
      .RESOS
CONVRT=TRAP+10    ;CALL TO DATA OUTPUT ROUTINE
      .CONVRT
CNVRT=TRAP+11     ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
      .CNVRT
MSTCLR=TRAP+12    ;CALL TO ISUE A MASTER CLEAR
      .MSTCLR
DELAY=TRAP+13     ;CALL TO DELAY
      .DELAY
ROMCLK=TRAP+14    ;CALL TO CLOCK ROM ONCE
      .ROMCLK
DATACLK=TRAP+15   ;CALL TO CLK DATA
      .DATACLK
TIMER=TRAP+16    ;CALL TO DELAY A CLOCK TICK
      .TIMER
-----
:*****

```

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 24
 DZDMEA.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

846
847      ;DMC11 CONTROL INDICATORS FOR CURRENT DMC11 UNDER TEST
848      ;-----
849
850      001366 000000      STAT1: 0
851      001370 000000      STAT2: 0
852      001372 000000      STAT3: 0
853
854      ;DMC11 VECTOR AND REGISTER INDIRECT POINTERS
855      ;-----
856
857      001374 000000      DMRVEC: 0      ;POINTER TO DMC11 RECEIVER INTERRUPT VECTOR
858      001376 000000      DMLVL: 0      ;POINTER TO DMC11 RECEIVER INTERRUPT SERVICE PS
859      001400 000000      DMTVEC: 0      ;POINTER TO DMC11 TRANSMITTER INTERRUPT VECTOR
860      001402 000000      DMTLVL: 0     ;POINTER TO DMC11 TRANSMITTER INTERRUPT SERVICE PS
861      001404 000000      DMCSR: 0      ;POINTER TO DMC11 CONTROL STATUS REGISTER
862      001406 000000      DMCSRH: 0     ;POINTER TO DMC11 CONTROL STATUS REGISTER HIGH BYTE.
863      001410 000000      DMCTL: 0      ;POINTER TO DMC11 CONTROL OUT REGISTER
864      001412 000000      DMP04: 0      ;POINTER TO DMC11 PORT REGISTER(SEL 4)
865      001414 000000      DMP06: 0      ;POINTER TO DMC11 PORT REGISTER(SEL 6)
866
867
868      ;DMC11 STATUS TABLE AND ADDRESS ASSIGNMENTS
869      ;-----
870
871      001500      .=1500
872      001500 000001      DM.MAP:
873      001502 000001      DMCRO0: .BLKW 1      ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 00
874      001504 000001      DMS100: .BLKW 1     ;VECTOR FOR DMC11 NUMBER 00
875      001506 000001      DMS200: .BLKW 1     ;DDCMP LINE# FOR DMC11 NUMBER 00
876      001508 000001      DMS300: .BLKW 1     ;3RD STATUS WORD
877
878      001510 000001      DMCRO1: .BLKW 1     ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 01
879      001512 000001      DMS101: .BLKW 1     ;VECTOR FOR DMC11 NUMBER 01
880      001514 000001      DMS201: .BLKW 1     ;DDCMP LINE# FOR DMC11 NUMBER 01
881      001516 000001      DMS301: .BLKW 1     ;3RD STATUS WORD
882
883      001520 000001      DMCRO2: .BLKW 1     ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 02
884      001522 000001      DMS102: .BLKW 1     ;VECTOR FOR DMC11 NUMBER 02
885      001524 000001      DMS202: .BLKW 1     ;DDCMP LINE# FOR DMC11 NUMBER 02
886      001526 000001      DMS302: .BLKW 1     ;3RD STATUS WORD
887
888      001530 000001      DMCRO3: .BLKW 1     ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 03
889      001532 000001      DMS103: .BLKW 1     ;VECTOR FOR DMC11 NUMBER 03
890      001534 000001      DMS203: .BLKW 1     ;DDCMP LINE# FOR DMC11 NUMBER 03
891      001536 000001      DMS303: .BLKW 1     ;3RD STATUS WORD
892
893      001540 000001      DMCRO4: .BLKW 1     ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 04
894      001542 000001      DMS104: .BLKW 1     ;VECTOR FOR DMC11 NUMBER 04
895      001544 000001      DMS204: .BLKW 1     ;DDCMP LINE# FOR DMC11 NUMBER 04
896      001546 000001      DMS304: .BLKW 1     ;3RD STATUS WORD
897
898      001550 000001      DMCRO5: .BLKW 1     ;CONTROL STATUS REGISTER FOR DMC11 NUMBER 05
899      001552 000001      DMS105: .BLKW 1     ;VECTOR FOR DMC11 NUMBER 05
900      001554 000001      DMS205: .BLKW 1     ;DDCMP LINE# FOR DMC11 NUMBER 05
901      001556 000001      DMS305: .BLKW 1     ;3RD STATUS WORD

```


DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 25
 DZDMEA.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

902					
903	001560	000001	DMCR06: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 06
904	001562	000001	DMS106: .BLKW	1	:VECTOR FOR DMC11 NUMBER 06
905	001564	000001	DMS206: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 06
906	001566	000001	DMS306: .BLKW	1	:3RD STATUS WORD
907					
908	001570	000001	DMCR07: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 07
909	001572	000001	DMS107: .BLKW	1	:VECTOR FOR DMC11 NUMBER 07
910	001574	000001	DMS207: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 07
911	001576	000001	DMS307: .BLKW	1	:3RD STATUS WORD
912					
913	001600	000001	DMCR10: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 10
914	001602	000001	DMS110: .BLKW	1	:VECTOR FOR DMC11 NUMBER 10
915	001604	000001	DMS210: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 10
916	001606	000001	DMS310: .BLKW	1	:3RD STATUS WORD
917					
918	001610	000001	DMCR11: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 11
919	001612	000001	DMS111: .BLKW	1	:VECTOR FOR DMC11 NUMBER 11
920	001614	000001	DMS211: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 11
921	001616	000001	DMS311: .BLKW	1	:3RD STATUS WORD
922					
923	001620	000001	DMCR12: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 12
924	001622	000001	DMS112: .BLKW	1	:VECTOR FOR DMC11 NUMBER 12
925	001624	000001	DMS212: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 12
926	001626	000001	DMS312: .BLKW	1	:3RD STATUS WORD
927					
928	001630	000001	DMCR13: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 13
929	001632	000001	DMS113: .BLKW	1	:VECTOR FOR DMC11 NUMBER 13
930	001634	000001	DMS213: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 13
931	001636	000001	DMS313: .BLKW	1	:3RD STATUS WORD
932					
933	001640	000001	DMCR14: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 14
934	001642	000001	DMS114: .BLKW	1	:VECTOR FOR DMC11 NUMBER 14
935	001644	000001	DMS214: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 14
936	001646	000001	DMS314: .BLKW	1	:3RD STATUS WORD
937					
938	001650	000001	DMCR15: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 15
939	001652	000001	DMS115: .BLKW	1	:VECTOR FOR DMC11 NUMBER 15
940	001654	000001	DMS215: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 15
941	001656	000001	DMS315: .BLKW	1	:3RD STATUS WORD
942					
943	001660	000001	DMCR16: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 16
944	001662	000001	DMS116: .BLKW	1	:VECTOR FOR DMC11 NUMBER 16
945	001664	000001	DMS216: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 16
946	001666	000001	DMS316: .BLKW	1	:3RD STATUS WORD
947					
948	001670	000001	DMCR17: .BLKW	1	:CONTROL STATUS REGISTER FOR DMC11 NUMBER 17
949	001672	000001	DMS117: .BLKW	1	:VECTOR FOR DMC11 NUMBER 17
950	001674	000001	DMS217: .BLKW	1	:DDCMP LINE# FOR DMC11 NUMBER 17
951	001676	000001	DMS317: .BLKW	1	:3RD STATUS WORD
952					
953	001700	000000	DM.END: 000000		

```

954
955                                     ;DMC11 PASS COUNT AND ERROR COUNT TABLE
956                                     -----
957
958 001702 CNT.MAP:
959 001702 000000 PACT00: 0 ;PASS COUNT FOR DMC11 NUMBER 00
960 001704 000000 ERCT00: 0 ;ERROR COUNT FOR DMC11 NUMBER 00
961
962 001706 000000 PACT01: 0 ;PASS COUNT FOR DMC11 NUMBER 01
963 001710 000000 ERCT01: 0 ;ERROR COUNT FOR DMC11 NUMBER 01
964
965 001712 000000 PACT02: 0 ;PASS COUNT FOR DMC11 NUMBER 02
966 001714 000000 ERCT02: 0 ;ERROR COUNT FOR DMC11 NUMBER 02
967
968 001716 000000 PACT03: 0 ;PASS COUNT FOR DMC11 NUMBER 03
969 001720 000000 ERCT03: 0 ;ERROR COUNT FOR DMC11 NUMBER 03
970
971 001722 000000 PACT04: 0 ;PASS COUNT FOR DMC11 NUMBER 04
972 001724 000000 ERCT04: 0 ;ERROR COUNT FOR DMC11 NUMBER 04
973
974 001726 000000 PACT05: 0 ;PASS COUNT FOR DMC11 NUMBER 05
975 001730 000000 ERCT05: 0 ;ERROR COUNT FOR DMC11 NUMBER 05
976
977 001732 000000 PACT06: 0 ;PASS COUNT FOR DMC11 NUMBER 06
978 001734 000000 ERCT06: 0 ;ERROR COUNT FOR DMC11 NUMBER 06
979
980 001736 000000 PACT07: 0 ;PASS COUNT FOR DMC11 NUMBER 07
981 001740 000000 ERCT07: 0 ;ERROR COUNT FOR DMC11 NUMBER 07
982
983 001742 000000 PACT10: 0 ;PASS COUNT FOR DMC11 NUMBER 10
984 001744 000000 ERCT10: 0 ;ERROR COUNT FOR DMC11 NUMBER 10
985
986 001746 000000 PACT11: 0 ;PASS COUNT FOR DMC11 NUMBER 11
987 001750 000000 ERCT11: 0 ;ERROR COUNT FOR DMC11 NUMBER 11
988
989 001752 000000 PACT12: 0 ;PASS COUNT FOR DMC11 NUMBER 12
990 001754 000000 ERCT12: 0 ;ERROR COUNT FOR DMC11 NUMBER 12
991
992 001756 000000 PACT13: 0 ;PASS COUNT FOR DMC11 NUMBER 13
993 001760 000000 ERCT13: 0 ;ERROR COUNT FOR DMC11 NUMBER 13
994
995 001762 000000 PACT14: 0 ;PASS COUNT FOR DMC11 NUMBER 14
996 001764 000000 ERCT14: 0 ;ERROR COUNT FOR DMC11 NUMBER 14
997
998 001766 000000 PACT15: 0 ;PASS COUNT FOR DMC11 NUMBER 15
999 001770 000000 ERCT15: 0 ;ERROR COUNT FOR DMC11 NUMBER 15
1000
1001 001772 000000 PACT16: 0 ;PASS COUNT FOR DMC11 NUMBER 16
1002 001774 000000 ERCT16: 0 ;ERROR COUNT FOR DMC11 NUMBER 16
1003
1004 001776 000000 PACT17: 0 ;PASS COUNT FOR DMC11 NUMBER 17
1005 002000 000000 ERCT17: 0 ;ERROR COUNT FOR DMC11 NUMBER 17
1006

```

1007
 1008
 1009
 1010
 1011
 1012

FORMAT OF STATUS TABLE

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	CSR
I	C	O	N	T	R	O	L	R	E	G	I	S	T	E	R	
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
I	*	I	*	I	*	I	*	I	*	I	*	I	*	I	*	STAT1
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
I	*	I	B	I	M	I	A	D	I	D	*	I	*	I	L	STAT2
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	*	STAT3
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	

DEFINITION OF FORMAT

- CSR: CONTAINS DMC11 CSR ADDRESS
- STAT1: BITS 00-08 IS DMC11 VECTOR ADDRESS
 BIT15=1 MICRO-PROCESSOR HAS CRAM
 BIT15=0 MICRO-PROCESSOR HAS CROM
 BIT14=1 ??? TURNAROUND CONNECTOR IS ON
 BIT14=0 NO TURNAROUND CONNECTOR
 BIT13=0 LINE UNIT IS AN M8201
 BIT13=1 LINE UNIT IS AN M8202
 BIT12=1 NO LINE UNIT
 BITS 09-11 IS DMC11 BR PRIORITY LEVEL
- STAT2: LOW BYTE IS SWITCH PAC#1 (DDCMP LINE NUMBER)
 HIGH BYTE IS SWITCH PAC#2 (BM873 BOOT ADD)
- STAT3: BIT0=1 DO FREE RUNNING TESTS ON KMC
 (MUST BE SET TO A ONE MANUALLY [PROGRAMS G AND H ONLY])

```

1006 :PROGRAM INITIALIZATION
1007 :LOCK OUT INTERRUPTS
1008 :SET UP PROCESSOR STACK
1009 :SET UP POWER FAIL VECTOR
1010 :CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1011 :TYPE TITLE MESSAGE
1012
1013 :START: MOV #340,PS :LOCK OUT INTERRUPTS
1014 MOV #STACK,SP :SET UP STACK
1015 MOV #.PFAIL,2#24 :SET UP POWER FAIL VECTOR
1016 MOV DMNUM,SAVNUM :SAVE NUMBER OF DEVICES IN SYSTEM.
1017 CLR SWFLG :CLEAR SOFT TIMEOUT FLAG
1018 CLRB ERRFLG :CLEAR ERROR FLAG
1019 CLRB QV.FLG :ZERO QUICK VERIFY FLAG
1020 MOV #DM.MAP-10,CREAM :GET MAP POINTER.
1021 MOV #CNT.MAP-4,MILK :GET PASS COUNT MAP POINTER
1022 MOV #SIT15,RUN :POINT POINTER TO FIRST DEVICE.
1023 MOV #CNT.MAP,RO :PASS COUNT POINTER TO RO
1024 23$: CLR (RO)+ :CLEAR TABLE
1025 CMP #CNT.MAP+100,RO :DONE YET?
1026 BNE 23$ :KEEP GOING
1027 CLR LSTERR :CLEAR LAST ERROR POINTER
1028 MOV #1,TSTNO :SET UP FOR TEST 1
1029 MOV #.START,RETURN :SET UP FOR POWER FAIL BEFORE
1030 :TESTING STARTS
1031 :SAVE CURRENT VECTORS
1032 MOV 2#6,-(SP)
1033 MOV 2#4,-(SP)
1034 MOV #6$,2#4 :SET UP FOR TIMEOUT
1035 MOV #177570,SWR :SET SWR TO HARD SWR ADDRESS
1036 MOV #177570,DISPLAY :SET DISPLAY TO HARD SWR ADDRESS
1037 CMP #-1,2$SWR :REFERENCE HARDWARE SWITCH REGISTER
1038 BEQ 6$+2 :IF = -1 USE SOFT SWR ANYWAY
1039 BR 7$ :IF IT EXISTS AND NOT = -1 USE HARD SWR
1040 6$: CMP (SP)+(SP)+ :ADJUST STACK
1041 MOV #SWREG,SWR :POINTER TO SOFT SWR
1042 MOV #DISPREG,DISPLAY :POINTER TO SOFT DISPLAY REG
1043 7$: MOV (SP)+,2#4 :RESTORE VECTORS
1044 MOV (SP)+,2#5
1045 TSTB INIFLG :HAS INITIALIZATION BEEN PERFORMED
1046 BNE 20$ :BR IF YES
1047 CMP #SENDAC,2#42 :IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
1048 BEQ 20$
1049 TYPE M.TITLE :TYPE TITLE MESSAGE
1050 JSR PC,CKSWR :CHECK FOR SOFT SWR
1051 MOV 2$SWR,STRTSW :STORE STARTING SWITCHES
1052 TST 2#42 :IS IT RUNNING IN AUTO MODE?
1053 BEQ .+6 :BR IF NO
1054 CLR STRTSW :IF YES, CLEAR SWITCHES
1055 BIT #SW00,STRTSW :IF SW00=1, QUESTIONS ARE ASKED.
1056 BNE 17$ :BR IF SW00=1
1057 TSTB STRTSW :BIT7=1??
1058 BPL 17$ :BR IF SW07=0
1059 TST DMACTV :ARE ANY DEVICES SELECTED?
1060 BNE 16$ :BR IF YES
1061 TYPE, NOACT :NO DEVICES SELECTED.

```

C03

```

11119 0003312 000000 HALT :STOP THE SHOW
11120 0003314 000776 BR -2 :DISQUALIFY CONTINUE SWITCH
11121 0003316 004737 010360 17$: JSR PC,AUTO.SIZE :GO DO THE AUTO SIZE
11122 0003318 105737 001324 16$: TSTB INIFLG :FIRST TIME?
11123 0003320 001410 BEQ 21$: BR IF YES
11124 0003322 105737 001236 TSTB STARTSW :IF USING SAME PARAMETERS DONT TYPE MAP
11125 0003324 100421 BMI 1$:
11126 0003326 032737 000006 001236 BIT #BIT1!BIT2,STARTSW :IS TEST NO. OR LOCK SELECTED
11127 0003328 001403 BEQ 24$: :IF NO THEN TYPE STATUS
11128 0003330 000424 BR 1$: :IF YES DO NOT TYPE STATUS
11129 0003332 005137 001324 21$: COM INIFLG :SET FLAG
11130 0003334 104402 006150 24$: TYPE XHEAD :TYPE HEADER
11131 0003336 012704 001500 MOV #DM.MAP,R4 :SET POINTER
11132 0003338 010437 001246 5$: MOV R4,TEMP1 :SET ADDRESS
11133 0003340 012437 001250 MOV (R4)+,TEMP2 :SET CSR
11134 0003342 001411 BEQ 1$: :ALL DONE IF ZERO
11135 0003344 012437 001252 MOV (R4)+,TEMP3 :SET STAT1
11136 0003346 012437 001254 MOV (R4)+,TEMP4 :SET STAT2
11137 0003348 012437 001256 MOV (R4)+,TEMP5 :SET STAT3
11138 0003350 104410 CONVRT :TYPE OUT STATUS MAP
11139 0003352 007274 XSTATQ :
11140 0003354 000762 BR 5$:
11141 0003356 012700 001500 1$: MOV #DM.MAP,R0 :R0 POINTS TO STATUS TABLE

```

```

*****
: *AUTO SIZE TEST
: *THIS TEST VERIFYS THAT THE DMC11S AND/OR KMC11S ARE AT THE CORRECT FLOATING
: *ADDRESSES FOR YOUR SYSTEM. IF THIS TEST FAILS, IT IS NOT A HARDWARE ERROR.
: *CHECK THE ADDRESSES OF ALL FLOATING DEVICES (DJ,DH,DQ,DU,DUP,L,K,DMC,DZ,KMC).
: *IF THERE ARE NO OTHER FLOATING DEVICES BEFORE THE DMC11, THE FIRST
: *DMC11 ADDRESS IS 760070, KMC11 IS 760110. NO DEVICE SHOULD EVER BE AT
: *ADDRESS 760000.
*****

```

```

11142 0003358 013746 000004 MOV 2$,-(SP) :SAVE LOC 4
11143 0003360 013746 000006 MOV 4$,-(SP) :SAVE LOC 6
11144 0003362 005037 000006 CLR 2#6 :CLEAR VEC+2
11145 0003364 005037 001252 CLR TEMP3 :CLEAR FLAG
11146 0003366 005005 CLR R5 :R5=0=DMC, R5=-1=KMC
11147 0003368 011037 001404 AUSTRT: MOV (R0),DMCSR :GET NEXT DMC CSR
11148 0003370 001530 BEQ AUDONE :BR IF DONE
11149 0003372 005705 TST R5 :DMC OR KMC?
11150 0003374 001005 BNE 1$: :BR IF KMC
11151 0003376 032760 100000 000002 BIT #BIT15,2(R0) :CHECK FOR DMC CSR
11152 0003378 001044 BNE OK :SKIP IF NOT DMC
11153 0003380 000404 BR 2$: :ITS A DMC SO CONTINUE
11154 0003382 032760 100000 000002 1$: BIT #BIT15,2(R0) :CHECK FOR KMC CSR
11155 0003384 001437 BEQ OK :SKIP IF NOT KMC
11156 0003386 012737 002606 000004 2$: MOV #NODEV,2#4 :SET UP FOR TIMEOUT
11157 0003388 005705 TST R5 :DMC OR KMC?
11158 0003390 001005 BNE 3$: :BR IF KMC
11159 0003392 012703 000006 MOV #6,R3 :R3 IS COUNT OF DEVICES BEFORE DMC
11160 0003394 000402 BR 4$: :GO ON
11161 0003396 012703 000010 3$: MOV #10,R3 :R3 IS COUNT OF DEVICES BEFORE KMC
11162 0003398 012702 002722 4$: MOV #DEVTAB,R2 :R2 IS DEVICE TABLE PONTNER
11163 0003400 012701 160010 MOV #160010,R1 :START WITH ADDRESS 160010

```

1175	002536	005711		FLOAT:	TST	(R1)	:CHECK ADDRESS IN R1
1176	002540	111204			MOVB	(R2), R4	:IF NO TIMEOUT, GET NEXT ADDRESS
1177	002544	060401			ADD	R4, R1	:IN R1
1178	002548	005201			INC	R1	
1179	002552	040401			BIC	R4, R1	
1180	002556	005703			TST	R3	:ANY MORE DEVICES TO CHECK FOR?
1181	002560	001371			BNE	FLOAT	:BR IF YES
1182	002564	012737	002612 000004		MOV	#ERR, @#4	:OK ONLY DMC'S ARE LEFT, SET UP FOR TIMEOUT
1183	002568	005711		FY:	TST	(R1)	:CHECK DMC ADDRESS
1184	002572	020137	001404		CMP	R1, DMCSR	:DOES IT MATCH
1185	002576	001403			BEQ	OK	:BR IF YES
1186	002580	062701	000010		ADD	#10, R1	:GET NEXT DMC ADDRESS
1187	002584	000771			BR	FY	:DO IT AGAIN
1188	002600	062700	000010	OK:	ADD	#10, R0	:SKIP TO NEXT DMC CSR
1189	002604	000720			BR	AUSTRT	:CONTINUE
1190	002608	122243		NODEV:	CMPB	(R2)+, -(R3)	:ON TIMEOUT, INC R2, DEC R3
1191	002612	000002			RTI		:RETURN
1192	002616	005737	001252	ERR:	TST	TEMP3	:CHECK FLAG IF = 0 TYPE HEADER
1193	002620	001014			BNE	IS	:SKIP HEADER
1194	002624	104402			TYPE		:TYPEOUT HEADER MESSAGE
1195	002628	007171			CONERR		:CONFIGURATION ERROR!!!!
1196	002632	012737	002612 001276		MOV	#ERR, SAVPC	:SAVE PC FOR TYPEOUT
1197	002636	104411			CNVRT		:TYPE OUT ERROR PC
1198	002640	002702			ERRPC		
1199	002644	104402			TYPE		:TYPE REST OF HEADER
1200	002648	007233			CNERR		
1201	002652	012737	177777 001252		MOV	#-1, TEMP3	:SET FLAG SO IT ONLY GETS TYPED ONCE
1202	002656	010137	001262	IS:	MOV	R1, SAVR1	:SAVE R1 FOR TYPEOUT
1203	002660	104410			CONVRT		
1204	002664	002710			CONTAB		:TYPE CSR VALUES
1205	002668	005709			TST	R5	:DMC OR KMC ?
1206	002672	001003			BNE	3\$:BR IF KMC
1207	002676	104402			TYPE		
1208	002680	007254			DMCM		
1209	002684	000402			BR	4\$:CONTINUE
1210	002688	104402		3\$:	TYPE		
1211	002692	007264			KMCM		
1212	002696	022626		4\$:	CMP	(SP)+, (SP)+	:ADJUST STACK
1213	002700	000737			BR	OK	:BR TO GET OUT
1214	002704	000001		ERRPC:	1		
1215	002708	006	002		.BYTE	6, 2	
1216	002712	001276			SAVPC		
1217	002716	000002		CONTAB:	2		
1218	002720	006	004		.BYTE	6, 4	
1219	002724	001262			SAVR1		
1220	002728	006	002		.BYTE	6, 2	
1221	002732	001404		DEVTAB:	DMCSR		
1222	002736	007			.BYTE	7	:DJ
1223	002740	017			.BYTE	17	:DH
1224	002744	007			.BYTE	7	:DQ
1225	002748	007			.BYTE	7	:DU
1226	002752	007			.BYTE	7	:DUP
1227	002756	007			.BYTE	7	:LK
1228	002760	007			.BYTE	7	:DMC
1229	002764	007			.BYTE	7	:DZ
1230	002768	007			.BYTE	7	:KMC

E03

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 31
 DZDMEA.P11 PROGRAM INITIALIZATION AND START UP.

12861	002734	002734				.EVEN					
12862	002736	005705				AUDONE:	TST	R5	:	DMC?	
12863	002740	001005					BNE	1\$:	BR IF KMC AND ALL DONE	
12864	002744	012705	177777				MOV	#-1,R5	:	SET R5 TO -1 (KMC)	
12865	002750	012700	001500				MOV	#DM.MAP,RO	:	RESET RO TO START OF TABLE	
12866	002752	000636					BR	AUSTR	:	GO DO KMC'S	
12867	002756	012637	000006			1\$:	MOV	(SP)+,D#6	:	RESTORE LOC 6	
12868	002762	012637	000004				MOV	(SP)+,D#4	:	RESTORE LOC 4	
12869	002770	032737	000010	001236			BIT	#SW03,STRTSW	:	SELECT SPECIFIC DEVICES??	
12870	002772	001422					BEQ	3\$:	BR IF NO.	
12871	002776	104402	006070				TYPE	MNEW	:	TYPE THE MESSAGE.	
12872	003000	005000					CLR	RO	:	ZERO DATA LIGHTS	
12873	003002	027737	176174	001312			HALT		:	WAIT FOR USER TO TELL WHAT DEVICES TO RUN	
12874	003010	101404					CMP	D\$WR,SAVACT	:	IS THE NUMBER VALID?	
12875	003012	104402	005731				BLOS	2\$:	BR IF NUMBER IS OK.	
12876	003016	000000					TYPE	,MERR3	:	TELL USER OF INVALID NUMBER.	
12877	003020	000776					HALT		:	STOP EVERY THING.	
12878	003022	017737	176154	001306		2\$:	BR	-2	:	RESTART THE PROGRAM AGAIN.	
12879	003030	013700	001306				MOV	D\$WR,DMACTV	:	GET NEW DEVICE PATTERN	
12880	003034	000000					MOV	DMACTV,RO	:	SHOW THE USER WHAT HE SELECTED.	
12881	003036	012700	000300				HALT		:	CONTINUE DYNAMIC SWITCHES.	
12882	003042	012701	000302			3\$:	MOV	#300,RO	:	PREPARE TO CLEAR THE FLOATING	
12883	003046	010120					MOV	#302,R1	:	VECTOR AREA. 300-776	
12884	003050	005021				4\$:	MOV	R1,(RO)+	:	START PUTTING "PC+2 - HALT"	
12885	003052	022021					CLR	(R1)+	:	IN VECTOR AREA.	
12886	003054	022700	001000				CMP	(RO)+,(R1)+	:	POF POINTERS	
12887	003060	001372					CMP	#1000,RO	:	ALL DONE??	
							BNE	4\$:	BR IF NO.	
							:TEST START AND RESTART				
							:-----				
12888	003062	012706	001200			.BEGIN:	MOV	#STACK,SP	:	SET UP STACK	
12889	003066	013746	000006				MOV	D#6,-(SP)	:	SAVE LOC 6	
12890	003072	013746	000004				MOV	D#4,-(SP)	:	SAVE LOC 4	
12891	003076	005000					CLR	RO	:	START AT 0	
12892	003100	012737	003144	000004			MOV	#2\$,D#4	:	SET UP FOR TIME OUT	
12893	003106	005037	000006				CLR	D#6	:	TO AUTOSIZE MEMORY	
12894	003112	005720				6\$:	TST	(RO)+	:	CHECK ADDRESS IN RO	
12895	003114	022700	157776				CMP	#157776,RO	:	IS IT AT LEAST 28K	
12896	003120	001374					BNE	6\$:	BR IF NO	
12897	003122	162700	007776				SUB	#7776,RO	:	SAVE 2K FOR MONITORS	
12898	003126	010037	001304			7\$:	MOV	RO,MEMLIM	:	STORE MEMORY LIMIT	
12899	003132	012637	000004				MOV	(SP)+,D#4	:	RESTORE LOC 4	
12900	003136	012637	000006				MOV	(SP)+,D#6	:	RESTORE LOC 6	
12901	003142	000413					BR	10\$:	CONTINUE	
12902	003144	022526				2\$:	CMP	(SP)+,(SP)+	:	ADJUST STACK	
12903	003146	162700	006000				SUB	#4,RO	:	GET LAST GOOD ADDRESS	
12904	003152	162700	007776				SUB	#7776,RO	:	SAVE 2K FOR MONITORS	
12905	003156	022700	030000				CMP	#30000,RO	:	IS IT 8K?	
12906	003162	001351					BNE	7\$:	BR IF NO	
12907	003164	012700	037400				MOV	#37400,RO	:	IF 8K DON'T SAVE 2K	
12908	003170	000756					BR	7\$:		
12909	003172	012737	000340	177776		10\$:	MOV	#340,PS	:	LOCK OUT INTERRUPTS	
12910	003200	032737	000004	001236			BIT	#BIT2,STRTSW	:	CHECK FOR LOCK ON TEST	
12911	003206	001411					BEQ	1\$:	BR IF NO LOCK DESIRED.	

F03

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 32
 DZDMEAR.P11 PROGRAM INITIALIZATION AND START UP.

1297	003210	104402	005767			TYPE	,MLOCK	:TYPE LOCK SELECTED.
1298	003214	012737	000240	003522		MOV	#NOP,TTST	:ADJUST SCOPE ROUTINE.
1299	003222	012737	000240	003524		MOV	#NOP,TTST+2	:SET UP TO LOCK
1300	003230	000406				BR	3\$:CONTINUE ALONG.
1301	003232	013737	003640	003522	1\$:	MOV	BRW,TTST	:PREPARE NORMAL SCOPE ROUTINE
1302	003240	013737	003642	003524		MOV	BRX,TTST+2	:LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1303	003246	012737	007726	001214	3\$:	MOV	#CYCLE,RETURN	:START AT "CYCLE" FIND WHICH DEVICE TO TEST
1304	003254	032737	000002	001236	4\$:	BIT	#SW01,STRTSW	:IS TEST NO. SELECTED?
1305	003262	001002				BNE	5\$:BR IF YES
1306	003264	104402	005657			TYPE	,MR	:TYPE R
1307	003270	000177	175720		5\$:	JMP	,RETURN	:START TESTING

G03

1298					:END OF PASS	
1299					:TYPE NAME OF TEST	
1300					:UPDATE PASS COUNT	
1301					:CHECK FOR EXIT TO ACT-11	
1302					:RESTART TEST	
1303						
1304	003274	000005			.EOP: RESET	:MAKE THE WORLD CLEAN AGAIN.
1305	003276	005037	001234		CLR LSTERR	:CLEAR LAST ERROR PC
1306	003302	105037	001325		CLRB ERRFLG	:CLEAR ERROR FLAG
1307	003306	005237	001230		INC PASCNT	:UPDATE PASS COUNT
1308	003312	013777	001230	175660	MOV PASCNT, @DISPLAY	:DISPLAY PASS COUNT
1309	003320	104402	005635		TYPE ,MEPASS	:TYPE END PASS
1310	003324	104402	006016		TYPE ,MCSR	:TYPE CSR
1311	003330	104411	003456		CNVRT ,XCSR	:SHOW IT
1312	003334	104402	006024		TYPE ,MVECX	:TYPE VECTOR
1313	003340	104411	003464		CNVRT ,XVEC	:SHOW IT
1314	003344	104402	006032		TYPE ,MPASSX	:TYPE PASSES
1315	003350	104411	003472		CNVRT ,XPASS	:SHOW IT
1316	003354	104402	006043		TYPE ,MERRX	:TYPE ERRORS
1317	003360	104411	003500		CNVRT ,XERR	:SHOW IT
1318	003364	013700	001322		MOV MILK, RO	:GET POINTER TO PASS COUNT
1319	003370	013720	001230		MOV PASCNT, (RO)+	:STORE PASS COUNT FOR THIS DMC11
1320	003374	013720	001232		MOV ERRCNT, (RO)+	:STORE ERROR COUNT FOR THIS DMC11
1321	003400	005337	001314		DEC SAVNUM	:ARE ALL DEVICES TESTED?
1322	003404	001017			BNE RESTR	:BR IF NO.
1323	003406	112737	000377	001327	MOVB #377, QV, FLG	:SET THE QUICK VERIFY FLAG.
1324	003414	013737	001310	001314	MOV DMNUM, SAVNUM	:RESTORE THE COUNT
1325	003422	013701	000042		MOV @#42, R1	:CHECK FOR ACT-11 OR DDP
1326	003426	001406			BEQ RESTR	:IF NOT, CONTINUE TESTING
1327	003430	000005			RESET	:STOP THE SHOW--CLEAR THE WORLD
1328	003432				SENDAD:	
1329	003432	004711			JSR PC, (R1)	
1330	003434	000240			NOP	
1331	003436	000240			NOP	
1332	003440	000240			NOP	
1333	003442	000240			NOP	
1334	003444	012737	007726	001214	RESTR: MOV #CYCLE, RETURN	
1335	003452	000137	007726		JMP CYCLE	
1336	003456	000001			XCSR: 1	
1337	003460	006	002		.BYTE 5,2	
1338	003462	001404			DMCSR	
1339	003464	000001			XVEC: 1	
1340	003466	004	002		.BYTE 4,2	
1341	003470	001374			DMRVEC	
1342	003472	000001			XPASS: 1	
1343	003474	006	002		.BYTE 6,2	
1344	003476	001230			PASCNT	
1345	003500	000001			XERR: 1	
1346	003502	006	002		.BYTE 6,2	
1347	003504	001232			ERRCNT	
1348						
1349					:SCOPE LOOP AND INTERATION HANDLER	
1350					-----	
1351						
1352	003506	004737	007470		.SCOPE: JSR PC, CKSWR	:CHECK FOR SOFT SWR
1353	003512	010016			MOV RO, (SP)	:SAVE RO ON THE STACK

H03

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 34
 DZDMEA.P11 GENERAL UTILITIES (TYPEOUT, ERROR, SCOPE, ETC)

1354	003514	032777	040000	175460		BIT	#BIT14, QSWR	:"LOOP ON THIS TEST"?
1355	003522	001407			TTST:	BEQ	1\$:BR IF NO. (IF LOCK SW01=1; THIS LOC =240)
1356	003524	000437				BR	3\$:GOTO 3\$ (IF LOCK SW01=1; THIS LOC =240)
1357	003526	105777	175452			TSTB	QTKCSR	:KEYBOARD DONE?
1358	003532	100034				BPL	3\$:BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
1359	003534	017700	175446			MOV	QTKDBR, R0	:CLEAR DONE BIT
1360	003540	000415				BR	2\$:CONTINUE
1361	003542	032777	004000	175432	1\$:	BIT	#SW11, QSWR	:DELETE ITERATION? (QUICK PASS)
1362	003550	001011				BNE	2\$:BR IF YES
1363	003552	105737	001327			TSTB	QV.FLG	:HAVE PASSES BEECOMPLETED?
1364	003556	001406				BEQ	2\$:BR IF QUICK PASS.
1365	003560	005237	001224			INC	LPCNT	:UPDATE ITERATION COUNTER
1366	003564	023737	001224	001222		CMP	LPCNT, ICOUNT	:ARE ALL ITERATIONS DONE??
1367	003572	101414				BLOS	3\$:BR IF NOT YET
1368	003574	105037	001325		2\$:	CLRB	ERRFLG	:PREPARE FOR NEW TEST
1369	003600	005037	001224			CLR	LPCNT	:START ICOUNTER AT 0
1370	003604	005037	001220			CLR	LOCK	
1371	003610	012737	000020	001222		MOV	#20, ICOUNT	:RESET ITERATIONS
1372	003616	013737	001216	001214		MOV	NEXT, RETURN	:GET NEXT TEST
1373	003624	011600			3\$:	MOV	(SP), R0	:POP R0 OFF OF THE STACK
1374	003626	022626				POP2SP		:FAKE AN "RTI"
1375	003630	013701	001404			MOV	DMCSR, R1	:R1 CONTAINS BASE DMC ADDRESS
1376	003634	000177	175354			JMP	QRETURN	:GO DO THE TEST
1377	003640	001407			BRW:	1407		
1378	003642	000437			BRX:	437		
1379								
1380								:CHECK FOR FREEZE ON CURRENT DATA
1381								-----
1382								
1383	003644	004737	007470		.SCOPI:	JSR	PC, CKSWR	:CHECK FOR SOFT SWR
1384	003650	032777	001000	175324		BIT	#SW09, QSWR	:IS SW09=1(SET)?
1385	003656	001405				BEQ	1\$:BR IF NOT SET.
1386	003660	005737	001220			TST	LOCK	
1387	003664	001402				BEQ	1\$	
1388	003666	013716	001220		1\$:	MOV	LOCK, (SP)	:GOTO THE ADDRESS IN LOCK.
1389	003672	000002				RTI		:GO BACK.
1390								
1391								:TELETYPE OUTPUT ROUTINE
1392								-----
1393								
1394	003674	010546			.TYPE:	MOV	R5, -(SP)	:SAVE R5 ON THE STACK.
1395	003676	017605	000002			MOV	Q2(SP), R5	:GET ADDRESS OF MESSAGE.
1396	003702	062766	000002	000002		ADD	#2, 2(SP)	:POP OVER ADDRESS.
1397	003710	005737	007664		4\$:	TST	SWFLG	:SOFT SWR MESSAGE?
1398	003714	001004				BNE	1\$:IF YES TYPE IT OUT REGARDLESS OF SW12
1399	003716	032777	010000	175256		BIT	#SW12, QSWR	:INHIBIT ALL PRINT OUT??
1400	003724	001012				BNE	3\$:BR IF NO PRINT OUT WANTED (SW12=1)
1401	003726	105715			1\$:	TSTB	(R5)	:IS NUMBER MINUS? (MSB=1(BIT7))
1402	003730	100002				BPL	2\$:BR IF NUMBER IS PLUS
1403	003732	104402	005574			TYPE	MCRLF	:TYPE A CR/LF!
1404	003736	105777	175246		2\$:	TSTB	QTPCSR	:TTY READY?
1405	003742	100375				BPL	2\$:BR IF NO.
1406	003744	112577	175242			MOVB	(R5)+, QTPDBR	:PRINT CURRENT CHAR.
1407	003750	001357				BNE	4\$:IF NOT ZERO KEEP PRINTING!
1408	003752	012605			3\$:	MOV	(SP)+, R5	:END OF OUTPUT. RESTORE R5
1409	003754	000002				RTI		:GO HOME

```

1410
1411
1412 003756 010346 .INSTR: MOV R3,-(SP) ;SAVE R3 ON STACK
1413 003760 010446 MOV R4,-(SP) ;SAVE R4 ON STACK
1414 003762 017637 000004 004000 MOV #4(SP),.MSG
1415 003770 062766 000002 000004 ADD #2,4(SP)
1416 003776 104402 .INST1: TYPE
1417 004000 000000 .MSG: 0
1418 004002 012704 007322 MOV #INBUF,R4
1419 004006 012703 000007 MOV #7,R3
1420 004012 105777 175166 1$: TSTB @TKCSR
1421 004016 100375 BPL 1$
1422 004020 117714 175162 MOVB @TKDBR,(R4)
1423 004024 142714 000200 BICB #200,(R4)
1424 004030 122427 000015 CMPB (R4)+,#15
1425 004034 001417 BEQ INSTR2
1426 004036 105777 175146 2$: TSTB @TPCSR
1427 004042 100375 BPL 2$
1428 004044 017777 175136 175140 MOV @TKDBR,@TPDBR
1429 004052 005303 DEC R3
1430 004054 001356 BNE 1$
1431 004056 012604 MOV (SP)+,R4
1432 004060 012603 MOV (SP)+,R3
1433 004062 104402 005570 .INSTE: TYPE MQM
1434 004066 010346 MOV R3,-(SP)
1435 004070 010446 MOV R4,-(SP)
1436 004072 000741 BR .INST1
1437 004074 012604 INSTR2: MOV (SP)+,R4 ;RESTORE R4
1438 004076 012603 MOV (SP)+,R3 ;RESTORE R3
1439 004100 000002 RTI
1440
1441 ;CONVERT ASCII STRING TO OCTAL
1442
1443
1444 004102 010546 .PARAM: MOV R5,-(SP)
1445 004104 010446 MOV R4,-(SP)
1446 004106 016605 000004 MOV #4(SP),R5
1447 004112 012537 004272 MOV (R5)+,LOLIM
1448 004116 012537 004274 MOV (R5)+,HILIM
1449 004122 012537 004276 MOV (R5)+,DEVADR
1450 004126 112537 004300 MOVB (R5)+,LOBITS
1451 004132 112537 004301 MOVB (R5)+,ADRCNT
1452 004136 010566 000004 MOV R5,4(SP)
1453 004142 005005 PARAM1: CLR R5
1454 004144 012704 007322 MOV #INBUF,R4
1455 004150 122714 000015 CMPB #15,(R4)
1456 004154 001420 BEQ PARERR
1457 004156 121427 000060 1$: CMPB (R4),#60
1458 004162 002415 BLT PARERR
1459 004164 121427 000067 CMPB (R4),#67
1460 004170 003012 BGT PARERR
1461 004172 142714 000060 BICB #60,(R4)
1462 004176 152405 BISB (R4)+,R5
1463 004200 122714 000015 CMPB #15,(R4)
1464 004204 001406 BEQ LIMITS
1465 004206 006305 ASL R5

```

1466	004210	006305			ASL	R5	
1467	004212	006305			ASL	R5	
1468	004214	000760			BR	1\$	
1469	004216	104404			PARERR: INSTER		
1470	004220	000750			BR	PARAM1	
1471							
1472							
1473							
1474							
1475	004222	020537	004274		LIMITS: CMP	R5, HILIM	
1476	004226	101373			BHI	PARERR	
1477	004230	020537	004272		CMP	R5, LOLIM	
1478	004234	103770			BLO	PARERR	
1479	004236	133705	004300		BITB	LOBITS, R5	
1480	004242	001365			BNE	PARERR	
1481							
1482							
1483							
1484	004244	013704	004276				
1485	004250	010524			1\$: MOV	DEVADR, R4	
1486	004252	062705	000002		MOV	R5, (R4)+	
1487	004256	105337	004301		ADD	#2, R5	
1488	004262	001372			DECB	ADRCNT	
1489	004264	012604			BNE	1\$	
1490	004266	012605			MOV	(SP)+, R4	
1491	004270	000002			MOV	(SP)+, R5	
1492	004272	000000			RTI		
1493	004274	000000			LOLIM: 0		
1494	004276	000000			HILIM: 0		
1495	004300	000000			DEVADR: 0		
1496		004301			LOBITS: 0		
1497					ADRCNT=LOBITS+1		
1498							
1499							
1500							
1501	004302	016637	000004	001276	.SAV05: MOV	4(SP), SAVPC	;SAVE R7 (PC)
1502							
1503							
1504							
1505	004310	010537	001272		SV05: MOV	R5, SAVR5	;SAVE R5
1506	004314	010437	001270		MOV	R4, SAVR4	;SAVE R4
1507	004320	010337	001266		MOV	R3, SAVR3	;SAVE R3
1508	004324	010237	001264		MOV	R2, SAVR2	;SAVE R2
1509	004330	010137	001262		MOV	R1, SAVR1	;SAVE R1
1510	004334	010037	001260		MOV	R0, SAVR0	;SAVE R0
1511	004340	000002			RTI		;LEAVE.
1512							
1513							
1514							
1515	004342	013700	001260		.RES05: MOV	SAVR0, R0	;RESTORE R0
1516	004346	013701	001262		MOV	SAVR1, R1	;RESTORE R1
1517	004352	013702	001264		MOV	SAVR2, R2	;RESTORE R2
1518	004356	013703	001266		MOV	SAVR3, R3	;RESTORE R3
1519	004362	013704	001270		MOV	SAVR4, R4	;RESTORE R4
1520	004366	013705	001272		MOV	SAVR5, R5	;RESTORE R5
1521	004372	000002			RTI		;LEAVE.

K03

; CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

1522									
1523									
1524									
1525									
1526	004274	104402	005574		.CONVR: TYPE	MCRLF			
1527	004470	010046			.CNVRT: MOV	R0, -(SP)			
1528	004402	010146			MOV	R1, -(SP)			
1529	004404	010346			MOV	R3, -(SP)			
1530	004406	010446			MOV	R4, -(SP)			
1531	004410	010546			MOV	R5, -(SP)			
1532	004412	017601	000012		MOV	2(12(SP), R1			
1533	004416	062766	000002	000012	ADD	#2, 12(SP)			
1534	004424	012137	004616		MOV	(R1)+, WRDCNT			
1535	004430	112137	004620		1\$: MOV	(R1)+, CHRCNT			
1536	004434	112137	004621		MOV	(R1)+, SPACNT			
1537	004440	013137	004622		MOV	2(R1)+, BINWRD			
1538	004444	122737	000003	004620	CMPB	#3, CHRCNT			
1539	004452	001003			BNE	2\$			
1540	004454	042737	177400	004622	BIC	#177400, BINWRD			
1541	004462	013704	004622		2\$: MOV	BINWRD, R4			
1542	004466	113705	004620		MOV	CHRCNT, R5			
1543	004472	012700	007364		MOV	#TEMP, R0			
1544	004476	010403			3\$: MOV	R4, R3			
1545	004500	042703	177770		BIC	#177770, R3			
1546	004504	062703	000060		ADD	#060, R3			
1547	004510	110320			MOV	R3, (R0)+			
1548	004512	000241			CLC				
1549	004514	006004			ROR	R4			
1550	004516	000241			CLC				
1551	004520	006004			ROR	R4			
1552	004522	000241			CLC				
1553	004524	006004			ROR	R4			
1554	004526	005305			DEC	R5			
1555	004530	001362			BNE	3\$			
1556	004532	012703	007426		MOV	#MDATA, R3			
1557	004536	114023			4\$: MOV	-(R0), (R3)+			
1558	004540	105337	004620		DECB	CHRCNT			
1559	004544	001374			BNE	4\$			
1560	004546	105737	004621		TSTB	SPACNT			
1561	004552	001405			BEQ	5\$			
1562	004554	112723	000040		5\$: MOV	#040, (R3)+			
1563	004560	105337	004621		DECB	SPACNT			
1564	004564	001373			BNE	5\$			
1565	004566	105013			6\$: CLRB	(R3)			
1566	004570	104402	007426		TYPE	, MDATA			
1567	004574	005337	004616		DEC	WRDCNT			
1568	004600	001313			BNE	1\$			
1569	004602	012605			MOV	(SP)+, R5			
1570	004604	012604			MOV	(SP)+, R4			
1571	004606	012603			MOV	(SP)+, R3			
1572	004610	012601			MOV	(SP)+, R1			
1573	004612	012600			MOV	(SP)+, R0			
1574	004614	000002			RTI				
1575	004616	000000			WRDCNT: 0				
1576	004620	000000			CHRCNT: 0				
1577		004621			SPACNT=CHRCNT+1				

L03

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 38
 DZDMEA.P11 GENERAL UTILITIES (TYPEOUT, ERROR, SCOPE, ETC)

```

1578 004622 000000          BINWRD: 0
1579
1580
1581          ;TRAP DISPATCH SERVICE
1582          ;ARGUMENT OF TRAP IS EXTRACTED
1583          ;AND USED AS OFFSET TO OBTAIN POINTER
1584          ;TO SELECTED SUBROUTINE
1585
1586 004624 011645          .TRPSR: MOV      (SP), -(SP)          ;GET PC OF RETURN
1587 004626 162716 000002          SUB      #2, (SP)          ;=PC OF TRAP
1588 004632 017616 000000          MOV      @ (SP), (SP)      ;GET TRP
1589 004636 006316          TRPOK: ASL      (SP)          ;MULTIPLY TRAP ARG BY 2
1590 004640 042716 177001          BIC      #177001, (SP)     ;CLEAR UNWANTED BITS
1591 004644 062716 001330          ADD      #.TRPTAB, (SP)   ;POINTER TO SUBROUTINE ADDRESS
1592 004650 017616 000000          MOV      @ (SP), (SP)     ;SUBROUTINE ADDRESS
1593 004654 000136          JMP      @ (SP)+         ;GO TO SUBROUTINE
1594
1595          ;ERROR HANDLER
1596          ;-----
1597
1598 004656 004737 007470          .HLT:  JSR      PC, CKSWR      ;CHECK FOR SOFT SWR
1599 004662 032777 010000 174312          BIT      #SW12, @SWR      ;BELL ON ERROR?
1600 004670 001406          BEQ      XBX              ;BR IF NO BELL
1601 004672 105777 174312          TSTB     @TPCSR           ;TTY READY.
1602 004676 100003          BPL      XBX              ;DON'T WAIT IF TTY NOT READY.
1603 004700 112777 000207 174304          MOVB     #207, @TPDBR     ;PUSH A BELL AT THE TTY.
1604 004706 032777 020000 174266          XBX:    BIT      #SW13, @SWR ;DELETE ERROR PRINT OUT?
1605 004714 001105          BNE      HALTS           ;BR IF NO PRINT OUT WANTED.
1606 004716 021637 001234          CMP      (SP), LSTERR     ;WAS THIS ERROR FOUND LAST TIME?
1607 004722 001404          BEQ      1$              ;BR IF YES
1608 004724 011637 001234          MOV      (SP), LSTERR     ;RECORD BEING HERE
1609 004730 105037 001325          CLRB     ERRFLG          ;PREPARE HEADER
1610 004734 104406          1$:     SAVO5             ;SAVE ALL PROC REGISTERS
1611 004736 011605          MOV      (SP), R5         ;GET THE PC OF ERROR
1612 004740 162705 000002          SUB      #2, R5           ;GET ADDRESS OF TRAP CALL
1613 004744 011504          MOV      (R5), R4         ;GET HLT INSTRUCTION
1614 004746 006304          ASL      R4               ;MULT BY TWO
1615 004750 061504          ADD      (R5), R4         ;DOUBLE IT
1616 004752 006304          ASL      R4               ;MULT AGAIN
1617 004754 042704 177001          BIC      #177001, R4       ;CLEAR JUNK
1618 004760 062704 031452          ADD      #.ERRTAB, R4     ;GET POINTER
1619 004764 012437 005100          MOV      (R4)+, ERMSG     ;GET ERROR MESSAGE
1620 004770 012437 005112          MOV      (R4)+, DATAHD   ;GET DATA HEADRER
1621 004774 011437 005124          MOV      (R4), DATABP    ;GET DATA TABLE
1622 005000 105737 001325          TSTB     ERRFLG          ;TYPE HEADREER
1623 005004 001403          BEQ      TYPMSG          ;BR IF YES
1624 005006 005737 005124          TST      DATABP          ;DOES DATA TABLE EXIST?
1625 005012 001040          BNE      TYPDAT          ;BR IF YES.
1626 005014 104402 005574          TYPMSG: TYPE      , MCRLF
1627 005020 104402 005574          TYPE      , MCRLF
1628 005024 005737 001220          TST      LOCK
1629 005030 001402          BEQ      1$
1630 005032 104402 006066          TYPE      , MASTEK
1631 005036 104402 006054          1$:     TYPE      , MTSTN
1632 005042 104411 005232          CNVRT     , XTSTN
1633 005046 104402 006143          TYPE      , MERRPC          ;SHOW IT
                                ;TYPE PC.

```

M03

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 39
 DZDMEA.P11 GENERAL UTILITIES (TYPEOUT, ERROR, SCOPE, ETC)

1634	005052	104411	005224			CNVRT	,ERTABO	;SHOW IT
1635	005056	104402	005574			TYPE	,MCRLF	;GIVE A CR/LF
1636	005062	112737	177777	001325		MOVB	#-1,ERRFLG	;NO MORE HEADER UNLESS NO DATA TABLE.
1637	005070	005737	005100			TST	ERRMSG	;IS THERE AN ERROR MESSAGE?
1638	005074	001402				BEQ	WRKO.FM	;BR IF NO.
1639	005076	104402				TYPE		;TYPE
1640	005100	000000				ERRMSG: 0		ERROR MESSAGE
1641	005102					WRKO.FM:		
1642	005102	005737	005112			TST	DATAHD	;DATA HEADER?
1643	005106	001402				BEQ	TYPDAT	;BR IF NO
1644	005110	104402				TYPE		;TYPE
1645	005112	000000				DATAHD: 0		DATA HEADER
1646	005114	005737	005124			TYPDAT: TST	DATABP	;DATA TABLE?
1647	005120	001402				BEQ	RESREG	;BR IF NO.
1648	005122	104410				CONVRT		;SHOW
1649	005124	000000				DATABP: 0		DATA TABLE
1650	005126	104407				RESREG: RESOS		;RESTORE PROC REGISTERS
1651	005130	022737	003432	000042		HALTS: CMP	,\$ENDAD,2#42	;IF ACT-11 AUTOMATIC MODE, HALT!!
1652	005136	001403				BEQ	1\$	
1653	005140	005777	174036			TST	2\$SWR	;HALT ON ERROR?
1654	005144	100005				BPL	EXITER	;BR IF NO HALT ON ERROR
1655	005146	010046				1\$: PUSHRO		;SAVE RO
1656	005150	016600	000002			MOV	2(SP),RO	;SHOW ERROR PC IN DATA LIGHTS
1657	005154	000000				HALT		;HALT
1658	005156	012600				POPPO		;GET RO
1659	005160	005237	001232			EXITER: INC	ERRCNT	;UPDATE ERROR COUNT
1660	005164	032777	000400	174010		BIT	,\$SW08,2\$SWR	;GOTO TOP OF TEST?
1661	005172	001007				BNE	1\$;BR IF YES
1662	005174	032777	002000	174000		BIT	,\$SW10,2\$SWR	;GOTO NEXT TEST?
1663	005202	001407				BEQ	2\$;BR IF NO
1664	005204	013737	001216	001214		MOV	NEXT,RETURN	;SET FOR NEXT TEST
1665	005212	012706	001200			1\$: MOV	,\$STACK,SP	;RESET SP
1666	005216	000177	173772			JMP	2\$RETURN	;GOTO SPECIFIED TEST
1667	005222	000002				2\$: RTI		;RETURN
1668	005224	000001				ERTABO: 1		
1669	005226	006	002			.BYTE	6,2	
1670	005230	001276				SAVPC		
1671	005232	000001				XTSTN: 1		
1672	005234	003	002			.BYTE	3,2	
1673	005236	001226				TSTNO		
1674						;ENTER HERE ON POWER FAILURE		
1675						-----		
1676								
1677								
1678	005240					.PFAIL:		
1679	005240	012737	005252	000024		MOV	,\$RESTART,24	;SET UP FOR POWER UP TRAP
1680	005246	000000				HALT		;HALT ON POWER DOWN NORMAL
1681	005250	000777				BR		
1682								
1683								;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED
1684								
1685	005252					RESTAR:		
1686	005252	012737	005240	000024		MOV	,\$PFAIL,24	;SET UP FOR POWER FAILURE
1687	005260	012706	001200			MOV	,\$STACK,SP	;RESET THE STACK POINTER
1688	005264	013701	001404			MOV	DMCSR,R1	;RESTORE R1
1689	005270	005037	007364			CLR	TEMP	;READY FOR TIMMER

N03

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 40
 DZDMEA.P11 GENERAL UTILITIES (TYPEOUT, ERROR, SCOPE, ETC)

```

1690 005274 005237 007364          INC      TEMP          ;PLUS ONE TO THE TIMER!
1691 005300 001375          BNE      .-4          ;BR IF MORE TO GO
1692 005302 104402 005577          TYPE    .MPFAIL      ;TYPE THE MESSAGE
1693 005306 104411 005332          CNVRT   .PFTAB       ;TELL WHAT TEST TO RETURN TO.
1694 005312 105037 001325          CLR     .ERRFLG      ;START CLEAN
1695 005316 005037 001234          CLR     .LSTEPR      ;.....
1696 005322 005011          CLR     .(R1)        ;CLEAR MAINT BITS
1697 005324 104412          MSTCLR          ;START CLEAN UP OF DEVICE
1698 005326 000177 173662          JMP     @RETURN      ;START DOING THAT TEST AGAIN.
1699 005332 000001          PFTAB: 1
1700 005334 003      002      .BYTE   3,2
1701 005336 001226          TSTNO
1702
1703
1704 005340          .DELAY:
1705 005340 012777 000020 174044          MOV     #20,@DMP04
1706 005346 104414          ROMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1707 005350 121111          121111          ;POKE CLOCK DELAY BIT
1708 005352          1$:
1709 005352 104414          ROMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
1710 005354 121224          121224          ;PORT4+IBUS*11
1711 005356 032777 000020 174026          BIT     #BIT4,@DMP04 ;IS CLOCK BIT SET?
1712 005364 001772          BEQ     1$        ;BR IF NO
1713 005366 000002          RTI
1714 005370          .MSTCLR:
1715 005370 152777 000100 174010          BISB   #BIT6,@DMCSRH ;SET MASTER CLEAR
1716 005376 142777 000300 174002          BICB   #BIT6!BIT7,@DMCSRH ;CLEAR MASTER CLEAR AND RUN
1717 005404 000002          RTI          ;RETURN
1718
1719 005406          .ROMCLK:
1720 005406 152777 000002 173772          BISB   #BIT1,@DMCSRH ;SET ROMI
1721 005414 013677 173774          MOV     @ (SP)+,@DMP06 ;LOAD INSTRUCTION IN SEL6
1722 005420 062746 000002          ADD     #2,-(SP)      ;ADJUST STACK
1723 005424 032777 000100 173550          BIT     #SW06,@SWR    ;HALT IF SW06 =1
1724 005432 001401          BEQ     1$        ;BR IF SW06 =0
1725 005434 000000          HALT          ;HALT BEFORE CLOCKING INSTRUCTION
1726 005436 152777 000003 173742          1$:  BISB   #BIT1!BIT0,@DMCSRH ;CLOCK INSTRUCTION
1727 005444 142777 000007 173734          BICB   #BIT2!BIT1!BIT0,@DMCSRH ;CLEAR ROMO, ROMI, STEP
1728 005452 000002          RTI
1729
1730 005454          .DATACLK:
1731 005454 013637 007364          MOV     @ (SP)+,TEMP  ;PUT TICK COUNT IN TEMP
1732 005460 062746 000002          ADD     #2,-(SP)      ;ADJUST STACK
1733 005464 152777 000020 173714          1$:  BISB   #BIT4,@DMCSRH ;SET STEP LU
1734 005472 027777 173706 173704          CMP     @DMCSR,@DMCSR ;WASTE TIME
1735 005500 142777 000020 173700          BICB   #BIT4,@DMCSRH ;CLEAR STEP LU
1736 005506 005337 007364          DEC     TEMP          ;DEC TICK COUNT
1737 005512 001364          BNE     1$        ;BR IF NOT DONE
1738 005514 000002          RTI          ;RETURN
1739 005516 000001          3$:  .BLKW 1
1740
1741 005520          .TIMER:
1742 005520 013637 007364          MOV     @ (SP)+,TEMP  ;MOVE COUNT TO TEMP
1743 005524 062746 000002          ADD     #2,-(SP)      ;ADJUST STACK
1744 005530          1$:
1745 005530 104414          ROMCLK          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

```


000000 021364
000000 022777 000002 173650
000000 001772
000000 104414
000000 021364
000000 022777 000002 173634
000000 001372
000000 005337 007364
000000 001366
000000 000002
000570 020040 000077
000570 005015 000
000570 377 053520 020122
000570 377 047105 020104
000570 377 000123
000570 050377 047522 051107
000570 377 047111 052523
000570 377 042524 052123
000570 377 047514 045503
000601 051503 035122 000040
000601 049503 035103 000040
000601 049503 051503 051503
000601 051103 051103 051103
000601 049503 052123 047040
000606 000002
000606 051777 052105 051440
000614 120 035103 000040
000618 02012 020040 020040
000620 377 020040 020040
000624 020212 050040 020103
000626 777 026455 026455
000637 044377 053517 046440
000643 041777 051123 040440
000647 053377 041505 047524
000647 377 051102 050040
000653 044777 020105 046504
000656 053777 044510 044103
000674 051777 044527 041524
000700 051777 044527 041524
000704 044777 020123 044124
000710 050377 047522 051107
000715 377 051413 051127
000715 116 052105 020077
000717 377 042377 041515
000723 377 054105 044503
000724 024040 046504 024503
000726 024040 046512 024503
000727 000005
000727 006 003
000730 001246
000730 006 003
000730 001250
000730 006 003

021364
BIT #2, DMP04
BFO 15
25:
ROMCLK
021364
BIT #2, DMP04
ONE 25
DEC TEMP
ONE 15
RTI

:PORT4-IBUS* REG11
:IS PGM CLOCK BIT CLEAR?
:BR IF YES
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:PORT4-IBUS* REG11
:IS PGM CLOCK BIT SET?
:BR IF YES
:DEC COUNT
:BR IF NOT DONE
:RETURN

MCM: .ASCIZ / ?/
MCRLF: .ASCIZ (15)<(12)
MFAIL: .ASCIZ (377)/PWR FAILED. RESTART AT TEST /
MPASS: .ASCIZ (377)/END PASS DZDME /
MR: .ASCIZ (377)/R/
MERR2: .ASCIZ (377)/PROGRAM INDICATES NO DEVICES PRESENT./
MERR3: .ASCIZ (377)/INSUFFICIENT DATA!/
MTSTPC: .ASCIZ (377)/TEST PC-/
MLOCK: .ASCIZ (377)/LOCK ON SELECTED TEST/
MCSR: .ASCIZ /CSR: /
MVEC: .ASCIZ /VEC: /
MPASSX: .ASCIZ /PASSES: /
MERRX: .ASCIZ /ERRORS: /
MTSTN: .ASCIZ /TEST NO: /
MASTEX: .ASCIZ /*/
MNEW: .ASCIZ (377)/SET SWITCH REG TO DMC11'S DESIRED ACTIVE./
MERRPC: .ASCIZ /PC: /

MAP OF DMC11 STATUS/
-----/
(212) PC CSR STAT1 STAT2 STAT3/

(377)-----

NUM: .ASCIZ (377)/HOW MANY DMC11'S TO BE TESTED?/
CSR: .ASCIZ (377)/CSR ADDRESS?/
VEC: .ASCIZ (377)/VECTOR ADDRESS?/
PRIO: .ASCIZ (377)/BR PRIORITY LEVEL? (4,5,6,7)?/
CRAM: .ASCIZ (377)/IF DMC HAS CRAM (M8204) TYPE "Y", IF CROM (M8200) TYPE "N"
MODU: .ASCIZ (377)/WHICH LINE UNIT? IF NONE TYPE "N", IF M8201 TYPE "1", IF M...
LINE: .ASCIZ (377)/SWITCH PAC#1 (DDCMP LINE #)?/
BM: .ASCIZ (377)/SWITCH PAC#2 (BM873 BOOT ADD)?/
CONN: .ASCIZ (377)/IS THE LOOP BACK CONNECTOR ON?/
NOACT: .ASCIZ (377)/PROGRAM INDICATES NO DEVICES ARE SELECTED/
SWMES: .ASCIZ (377)<12>/SWR= /
SWMES1: .ASCIZ /NEW? /
CONFERR: .ASCIZ (377)<377>/DMC11 CONFIGURATION ERROR PC: /
CNERR: .ASCIZ (377)/EXPECTED FOUND/
DMCM: .ASCIZ / (DMC) /
KMCM: .ASCIZ / (KMC) /
.EVEN
XSTATQ: 5

.BYTE 6,3
TEMP1
.BYTE 6,3
TEMP2
.BYTE 6,3

C04

```

17777 007310 001252          TEMP3
17777 007312          .BYTE 6,3
17777 007314 001254          TEMP4
17777 007316          .BYTE 6,2
17777 007320          TEMPS
                                .EVEN
                                ;BUFFERS FOR INPUT-OUTPUT
17777 007322 000000          INBUF: 0
17777 007364 000000          =.+40
17777 007426 000000          TEMP: 0
17777 007470 000000          =.+40
17777 007470 000000          MDATA: 0
17777 007470 000000          =.+40
                                ;ROUTINE USED TO CHANGE SOFTWARE SWITCH
                                ;REGISTER USING THE CONSOLE TERMINAL
                                -----
17777 007470 022737 000176 001202          CKSWR: CMP      #SWREG,SWR          ;IS THE SOFT SWR BEING USED?
17777 007476 001071          BNE     CKSWR5           ;BR IF NO
17777 007500 022777 000007 171500          CMP     #7,ATKDBR       ;WAS CTRL G TYPED? (7 BIT ASCII)
17777 007506 001404          BEQ     1$             ;BR IF YES
17777 007510 022777 000207 171470          CMP     #207,ATKDBR     ;WAS CTRL G TYPED? (8 BIT ASCII)
17777 007516 001061          BNE     CKSWR5           ;BR IF NO
17777 007520 010246          1$:   MOV     R2,-(SP)        ;STORE R2
17777 007522 010346          MOV     R3,-(SP)        ;STORE R3
17777 007524 010446          MOV     R4,-(SP)        ;STORE R4
17777 007526 012737 177777 007664          MOV     #-1,SWFLG      ;SET SOFT TYPE OUT FLAG
17777 007534 005002          CKSWR1: CLR     R2         ;CLEAR NEW SWR CONTENTS
17777 007536 012704 177777          MOV     #-1,R4         ;SET FLAG TO ALL ONES
17777 007542 104402 007153          TYPE   ,SWMES          ;TYPE "SWR="
17777 007548 104411          CKSWR2: CNVRT          ;TYPE OUT PRESENT CONTENTS
17777 007550 007720          SOFTSW          ;OF SOFT SWITCH REGISTER
17777 007552 104402 007163          CKSWR3: TYPE   ,SWMES1  ;TYPE "NEW="
17777 007556 004737 007666          CKSWR4: JSR     PC INCHAR ;GET RESPONSE
17777 007560 022703 000015          CMP     #15,R3         ;WAS IT A CR?
17777 007566 001424          BEQ     5$             ;BR IF YES
17777 007570 022703 000012          CMP     #12,R3         ;WAS IT A LF?
17777 007574 001416          BEQ     4$             ;BR IF YES
17777 007576 022703 000025          CMP     #25,R3         ;WAS IT CTRL U?
17777 007602 001764          BEQ     CKSWR1         ;BR IF YES (START OVER)
17777 007604 022703 000007          CMP     #7,R3          ;IF CNTL G GET NEXT CHAR
17777 007610 001762          BEQ     CKSWR4
17777 007612 005004          CLR     R4             ;IT MUST BE A DIGIT SO CLR FLAG
17777 007614 042703 177770          BIC     #177770,R3     ;ONLY 0-7 ARE LEGAL SO MASK OFF BITS
17777 007620 006302          ASL     R2             ;SHIFT R2 3 TIMES
17777 007622 006302          ASL     R2
17777 007624 006302          ASL     R2
17777 007626 050302          BIS     R3,R2          ;ADD LAST DIGIT
17777 007630 000752          BR      CKSWR4         ;GET NEXT CHARACTER
17777 007632 012766 002002 000006          4$:   MOV     #,START,6(SP) ;LF WAS TYPED SO GO TO START
17777 007640 005704          5$:   TST     R4            ;IS FLAG CLEAR?
17777 007642 001002          BNE     6$             ;IF NOT DON'T CHANGE SOFT SWR
  
```

```

00000000 007644 010277 171332
00000000 007644 005037 007664
00000000 007644 012604
00000000 007644 012603
00000000 007644 012602
00000000 007644 000207
00000000 007664 000000
00000000 007666 105777 171312
00000000 007672 100375
00000000 007674 017703 171306
00000000 007700 105777 171304
00000000 007704 100375
00000000 007706 010377 171300
00000000 007712 042703 000200
00000000 007716 000207
00000000 007720 000001
00000000 007722 006 002
00000000 007724 000176

```

```

68:      MOV      R2, @SWR      ; IF YES THEN WRITE NEW CONTENTS TO SOFT SWR
          CLR      SWFLG      ; CLEAR TYPEOUT FLAG
          MOV      (SP)+, R4   ; RESTORE R4
          MOV      (SP)+, R3   ; RESTORE R3
          MOV      (SP)+, R2   ; RESTORE R2
          RTS      PC         ; RETURN

CKSWR5:  RTS      PC

SWFLG:   0

INCHAR:  TSTB     @TKCSR
          BPL     -4
          MOV     @TKDBR, R3
          TSTB   @TPCSR
          BPL     -4
          MOV     R3, @TPDBR
          BIC    #BIT7, R3
          RTS    PC

SOFTSW:  1
          .BYTE  6, 2
          SWREG

```


F04

```

1997 010226 001045          BNE      7$          ;BR IF YES
1998 010230 104402 005574    TYPE     ,MCRLF
1999 010234 104403          INSTR
2000 010236 006054          MTSTN
2001 010240 104405          PARAM
2002 010242 000001          1
2003 010244 001000          1000
2004 010246 001226          TSTNO
2005 010250 000          .BYTE 0
2006 010251 001          .BYTE 1
2007 010252 012700 012074    MOV      #TST1,R0
2008 010256 022710 5$:      CMP      (PC)+,(R0)    ;CMP FIRST WORD TO 12737
2009 010260 012737          MOV      (PC)+,3(PC)+
2010 010262 001020          BNE     6$          ;BR IF NOT SAME
2011 010264 023760 001226 000002  CMP      TSTNO,2(R0)  ;DOES TSTNO MATCH?
2012 010272 001014          BNE     6$          ;BR IF NO
2013 010274 022760 001226 000004  CMP      #TSTNO,4(R0) ;IS LAST WORD OK?
2014 010302 001010          BNE     6$          ;BR IF NO
2015 010304 010037 001214    MOV      R0,RETURN   ;IT IS A LEGAL TEST SO DO IT
2016 010310 104402 005657    TYPE     MR
2017 010314 042737 000002 001236  BIC     #SW01,STRTSW
2018 010322 000412          BR      8$
2019 010324 005720 5$:      TST     (R0)+        ;POP R0
2020 010326 020027 025604    CMP      R0,#TLAST+10 ;AT END YET?
2021 010332 001351          BNE     5$          ;BR IF NO
2022 010334 104402 005570    TYPE     ,MQM        ;YES ILLEGAL TEST NO.
2023 010340 000730          BR      4$          ;TRY AGAIN
2024
2025 010342 012737 012074 001214 7$:      MOV      #TST1,RETURN ;PREPARE RETURN ADDRESS
2026 010350 013701 001404 8$:      MOV      DMCSR,R1    ;R1 = BASE DMC11 ADDRESS
2027 010354 000177 170634    JMP     3RETURN      ;GO START TESTING.
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
010360          AUTO.SIZE:
010360 000005          CSRMAP: RESET
010362 012702 001500    1$:      MOV      #DM.MAP,R2 ;INSURE A BUS INIT.
010366 005022          CLR     (R2)+        ;LOAD MAP POINTER.
010370 022702 001700    CMP      #DM.END,R2 ;ZERO ENTIRE MAP
010374 001374          BNE     1$          ;ALL DONE?
010376 005037 001310    CLR     DMNUM        ;BR IF NO
010402 012702 001500    MOV      #DM.MAP,R2 ;SET OCTAL NUMBER OF DMC11'S TO 0
010406 005037 001306    CLR     DMACTV       ;R2 POINTS TO DMC MAP
010412 032737 000001 001236  BIT     #SW00,STRTSW ;CLEAR ACTIVE
010420 001002          BNE     +6          ;QUESTIONS?
010422 000137 011052 001256  JMP     7$          ;BR IF YES
010426 012737 000001          MOV     #1,TEMPS    ;IF NO SKIP QUESTIONS
010434 104403          INSTR ;START WITH 1
010436 006374          NUM

```

```

:ROUTINE USED TO "AUTO SIZE" THE DMC11:
:CSR AND VECTOR.
:NOTE: THE CSR MAY BE ANY WHERE IN THE FLOATING
:ADDRESS RANGE (160000:164000)
:AND THE VECTOR MAY BE ANY WHERE IN THE
:FLOATING VECTOR RANGE (300:770)
:
```

G04

1953	010440	104405			PARAM		
1954	010442	000001			1		
1955	010444	000020			16.		
1956	010446	001252			TEMP3		
1957	010450	000			.BYTE	0	
1958	010451	001			.BYTE	1	
1959	010452	013737	001252	001310	MOV	TEMP3,DMNUM	;DMNUM = HOW MANY
1960	010460	104402	005574		TYPE	,MCRLF	
1961	010464	104410		12\$:	CONVRT		;TYPE WHICH DMC IS BEING DONE
1962	010466	011556			WHICH		;TEMPS IS WHICH DMC
1963	010470	005237	001256		INC	TEMPS	
1964	010474	104403			INSTR		
1965	010476	006434			CSR		
1966	010500	104405			PARAM		
1967	010502	160000			160000		
1968	010504	164000			164000		
1969	010506	001254			TEMP4		
1970	010510	000			.BYTE	0	
1971	010511	001			.BYTE	1	
1972	010512	013722	001254		MOV	TEMP4,(R2)+	;STORE CSR IN MAP
1973	010516	104403			INSTR		
1974	010520	006452			VEC		
1975	010522	104405			PARAM		
1976	010524	000000			0		
1977	010526	000776			776		
1978	010530	001254			TEMP4		
1979	010532	000			.BYTE	0	
1980	010533	001			.BYTE	1	
1981	010534	013712	001254		MOV	TEMP4,(R2)	;STORE VECTOR IN MAP
1982	010540	104402		10\$:	TYPE		
1983	010542	006473			PRIO.		;ASK WHAT BR LEVEL
1984	010544	004737	012042		JSR	PC,INTTY	;GET RESPONSE
1985	010550	022703	000024		CMP	#24,R3	
1986	010554	101014			BHI	50\$;BR IF LESS THAN 4
1987	010556	022703	000027		CMP	#27,R3	
1988	010562	103411			BLO	50\$;BR IF GREATER THAN 7
1989	010564	012704	000011		MOV	#11,R4	;R4 = NUMBER OF SHIFTS
1990	010570	006303			ASL	R3	;SHIFT R3 LEFT
1991	010572	005304			DEC	R4	;DEC SHIFT COUNT
1992	010574	001375			BNE	.-4	;BR IF NOT DONE
1993	010576	042703	170777		BIC	#170777,R3	;BIC UNWANTED BITS
1994	010602	050312			BIS	R3,(R2)	;PUT BR LEVEL IN STATUS MAP
1995	010604	000403			BR	8\$;CONTINUE
1996	010606	104402		50\$:	TYPE		
1997	010610	005570			MQM		;RESPONSE IS OUT OF LIMITS
1998	010612	000752			BR	10\$;TRY AGAIN
1999	010614	104402		8\$:	TYPE		
2000	010616	006532			CRAM		;DOES DMC HAVE CRAM?
2001	010620	004737	012042		JSR	PC,INTTY	;GET REPLY
2002	010624	022703	000131		CMP	#131,R3	
2003	010630	001406			BEQ	9\$;YES
2004	010632	022703	000116		CMP	#116,R3	;NO
2005	010636	001405			BEQ	16\$;NOT A Y OR N
2006	010640	104402			TYPE		
2007	010642	005570			MQM		;TYPE "?"
2008	010644	000763			BR	8\$;ASK AGAIN

H04

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 47
 DZDME.P11 GENERAL UTILITIES (TYPEOUT, ERROR, SCOPE, ETC)

2009	010646	052712	100000	9\$:	BIS	#BIT15,(R2)	;SET BIT 15 IF CRAM
2010	010652	104402		16\$:	TYPE		
2011	010654	006630			MODU		;ASK WHICH LINE UNIT
2012	010656	004737	012042		JSR	PC,INTTY	;GET REPLY
2013	010662	022703	000021		CMP	#21,R3	;"1"
2014	010666	001417			BEQ	30\$	
2015	010670	022703	000022		CMP	#22,R3	;"2"
2016	010674	001412			BEQ	31\$	
2017	010676	022703	000116		CMP	#116,R3	;"N"
2018	010702	001403			BEQ	32\$	
2019	010704	104402			TYPE		
2020	010706	005570			MQM		;IF NOT A 1,2 OR N TYPE "?"
2021	010710	000760			BR	16\$;TRY AGAIN
2022	010712	052722	010000	32\$:	BIS	#BIT12,(R2)+	;SET BIT 12 IN STAT2 IF NO LU
2023	010716	022222			CMP	(R2)+,(R2)+	;POP OVER STAT2 AND STAT3
2024	010720	000447			BR	33\$	
2025	010722	052712	020000	31\$:	BIS	#BIT13,(R2)	;SET BIT 13 IN STAT2 IF M8202
2026	010726	104402		30\$:	TYPE		
2027	010730	007040			CONN		;ASK IF LOOP-BACK IS ON
2028	010732	004737	012042		JSR	PC,INTTY	;GET REPLY
2029	010736	022703	000131		CMP	#131,R3	;Y
2030	010742	001406			BEQ	17\$	
2031	010744	022703	000116		CMP	#116,R3	;N
2032	010750	001406			BEQ	18\$	
2033	010752	104402			TYPE		
2034	010754	005570			MQM		;IF NOT Y OR N TYPE "?"
2035	010756	000763			BR	30\$;TRY AGAIN
2036	010760	052722	040000	17\$:	BIS	#BIT14,(R2)+	;TURNAROUND IS CONNECTED
2037	010764	000402			BR	19\$	
2038	010766	042722	040000	18\$:	BIC	#BIT14,(R2)+	;NO TURNAROUND
2039	010772			19\$:			
2040	010772	104403			INSTR		
2041	010774	006742			LINE		
2042	010776	104405			PARAM		
2043	011000	000000			0		
2044	011002	000377			377		
2045	011004	001254			TEMP4		
2046	011006	000			.BYTE	0	
2047	011007	001			.BYTE	1	
2048	011010	113722	001254		MOVB	TEMP4,(R2)+	;STORE SWITCH PAC IN MAP
2049	011014	104403			INSTR		
2050	011016	007000			BM		
2051	011020	104405			PARAM		
2052	011022	000000			0		
2053	011024	000377			377		
2054	011026	001254			TEMP4		
2055	011030	000			.BYTE	0	
2056	011031	001			.BYTE	1	
2057	011032	113722	001254		MOVB	TEMP4,(R2)+	;STORE SWITCH PAC IN MAP
2058	011036	005722			TST	(R2)+	;POP OVER STAT3
2059	011040	005337	001252	33\$:	DEC	TEMP3	;DEC DMC COUNT
2060	011044	001205			BNE	12\$;BR IF MORE TO DO
2061	011046	000137	011456		JMP	13\$;CONTINUE
2062	011052	012701	160000	7\$:	MOV	#160000,R1	;SET FOR FIRST ADDRESS TO BE TESTED
2063	011056	012737	011550 000004		MOV	#6\$,2#4	;SET FOR NON-EXISTANT DEVICE TIME OUT
2064	011064	005011		2\$:	CLR	(R1)	;CLEAR SELO

2065	011066	005711			TST	(R1)	: IF DMC11 DMCSR S/B 0
2066	011070	001162			BNE	3\$: IF NO DEV ; TRAP TO 4. IF NO BIT 8 THEN NO DMC1
2067	011072	005061	000006		CLR	6(R1)	: CLEAR SEL6
2068	011076	005761	000006		TST	6(R1)	: IF DMC11 THEN DMCRIC S/B =0!
2069	011102	001155			BNE	3\$: BR IF NOT DMC11
2070	011104	012711	002000		MOV	#BIT10,(R1)	: SET ROMO
2071	011110	005061	000004		CLR	4(R1)	: CLEAR SEL4
2072	011114	012761	125252	000006	MOV	#125252,6(R1)	: WRITE THIS TO SEL6
2073	011122	052711	020000		BIS	#BIT13,(R1)	: WRITE IT!
2074	011126	022761	125252	000004	CMP	#125252,4(R1)	: WAS IT WRITTEN?
2075	011134	001004			BNE	21\$: IF NO IT IS NOT CROM
2076	011136	052762	100000	000002	BIS	#BIT15,2(R2)	: SET BIT15 IF CROM
2077	011144	000421			BR	22\$	
2078	011146	012711	001000		MOV	#BIT9,(R1)	: SET ROMI
2079	011152	012761	100400	000006	MOV	#100400,6(R1)	: PUT INSTRUCTION IN SEL6
2080	011160	012711	001400		MOV	#BIT9!BIT8,(R1)	: CLOCK INSTRUCTION (MICRO PROC PC TO 0)
2081	011164	012711	002000		MOV	#BIT10,(R1)	: SET ROMO
2082	011170	022761	063220	000006	CMP	#63220,6(R1)	: IS IT CROM
2083	011176	001404			BEQ	22\$: BR IF YES
2084	011200	022761	177777	000006	CMP	#-1,6(R1)	: IF =-1 IT HAS NO CROM
2085	011206	001113			BNE	3\$: BR IF NOT DMC11
2086					: AT THIS POINT IT IS ASSUMED THAT R1 HOLDS A DMC11 CSR ADDRESS.		
2087	011210	010122			MOV	R1,(R2)+	: STORE CSR IN CORE TABLE.
2088	011212	012711	001000		MOV	#BIT9,(R1)	: CLEAR LINE UNIT LOOP
2089	011216	005061	000004		CLR	4(R1)	: CLEAR PORT4
2090	011222	012761	122113	000006	MOV	#122113,6(R1)	: LOAD INSTRUCTION (CLR DTR)
2091	011230	052711	000400		BIS	#BIT8,(R1)	: CLOCK INSTRUCTION
2092	011234	012761	021264	000006	MOV	#021264,6(R1)	: LOAD INSTRUCTION
2093	011242	052711	000400		BIS	#BIT8,(R1)	: CLOCK INSTRUCTION
2094	011246	122761	000377	000004	CMPB	#377,4(R1)	: IS IT ALL ONES?
2095	011254	001003			BNE	+.10	: BR IF NO
2096	011256	052712	010000		BIS	#BIT12,(R2)	: IF YES, NO LINE UNIT, SET STATUS BIT
2097	011262	000436			BR	20\$	
2098	011264	032761	000002	000004	BIT	#BIT1,4(R1)	: IS SWITCH A ONE?
2099	011272	001403			BEQ	+.10	: BR IF M9201
2100	011274	052712	060000		BIS	#BIT13!BIT14,(R2)	: M9202 ASSUME CONNECTOR
2101	011300	000427			BR	20\$: CONNECTOR ON)
2102	011302	032761	000010	000004	BIT	#BIT3,4(R1)	: IS MRDY SET
2103	011310	001023			BNE	20\$: BR IF M9201 NO CONNECTOR (ON LINE)
2104	011312	012761	000100	000004	MOV	#BIT6,4(R1)	: LOAD PORT4
2105	011320	012761	122113	000006	MOV	#122113,6(R1)	: LOAD INSTRUCTION
2106	011326	052711	000400		BIS	#BIT8,(R1)	: CLOCK INSTRUCTION (SET DTR)
2107	011332	012761	021264	000006	MOV	#021264,6(R1)	: LOAD INSTRUCTION
2108	011340	052711	000400		BIS	#BIT8,(R1)	: CLOCK INSTRUCTION (READ MODEM REG)
2109	011344	032761	000010	000004	BIT	#BIT3,4(R1)	: IS MRDY SET NOW?
2110	011352	001402			BEQ	20\$: BR IF NO CONNECTOR
2111	011354	052712	040000		BIS	#BIT14,(R2)	: SET STATUS BIT FOR CONNECTOR
2112	011360	005722			TST	(R2)+	: POP POINTER
2113	011362	012761	021324	000006	MOV	#021324,6(R1)	: PUT INSTRUCTION IN PORT6
2114	011370	012711	001400		MOV	#BIT9!BIT8,(R1)	: PORT4+LU 15
2115	011374	156122	000004		BISB	4(R1),(R2)+	: STORE DDCMP LINE # IN TABLE
2116	011400	012761	021344	000006	MOV	#021344,6(R1)	: PORT6+INSTRUCTION
2117	011406	012711	001400		MOV	#BIT8!BIT9,(R1)	: CLOCK INSTR.
2118	011412	156122	000004		BISB	4(R1),(R2)+	: STORE BM873 ADD IN TABLE
2119	011416	005722			TST	(R2)+	: POP OVER STAT3
2120	011420	005011			CLR	(R1)	: CLEAR ROMI

2121	011422	005237	001310			INC	DMNUM	;UPDATE DEVICE COUNTER
2122	011426	022737	000020	001310		CMP	#20,DMNUM	;ARE MAX. NO. OF DEV FOUND?
2123	011434	001410				BEQ	13\$;YES DON'T LOOK FOR ANY MORE.
2124	011436	005011			3\$:	CLR	(R1)	;CLEAR BIT 10
2125	011440	005061	000006			CLR	6(R1)	;CLEAR SEL 6
2126	011444	062701	000010		14\$:	ADD	#10,R1	;UPDATE CSR POINTER ADDRESS
2127	011450	022701	164000			CMP	#164000,R1	
2128	011454	001203				BNE	2\$;BR IF MORE ADDRESS TO CHECK.
2129	011456	005037	001306		13\$:	CLR	DMACTV	
2130	011462	005737	001310			TST	DMNUM	;WERE ANY DMC11'S FOUND AT ALL?
2131	011466	001423				BEQ	5\$;ERROR AUTO SIZER FOUND NO DMC11'S IN THIS SYS.
2132	011470	013701	001310			MOV	DMNUM,R1	
2133	011474	010137	001314			MOV	R1,SAVNUM	;SAVE NUMBER OF DEVICES
2134	011500	000241			4\$:	CLC		
2135	011502	006137	001306			ROL	DMACTV	;GENERATE ACTIVE REGISTER OF DEVICES.
2136	011506	005237	001306			INC	DMACTV	;SET THE BIT
2137	011512	005301				DEC	R1	
2138	011514	001371				BNE	4\$;BR IF MORE TO GENERATE
2139	011516	012737	000006	000004		MOV	#6,2#4	;RESTORE TRAP VECTOR
2140	011524	013737	001306	001312		MOV	DMACTV,SAVACT	;SAVE ACTIVE REGISTER
2141	011532	000137	011564			JMP	VECMAP	;GO FIND THE VECTOR NOW.
2142	011536	104402	005662		5\$:	TYPE	.MERR2	;NOTIFY OPR THAT NO DMC11'S FOUND.
2143	011542	005000				CLR	RO	;MAKE DATA LIGHTS ZERO
2144	011544	000000				HALT		;STOP THE SHOW
2145	011546	000776				BR	.-2	;DISABLE CONT. SW.
2146	011550	012716	011444		6\$:	MOV	#14\$, (SP)	;ENTERED BY NON-EXISTANT TIME-OUT.
2147	011554	000002				RTI		;RETURN TO MAINSTREAM
2148								
2149	011556	000001			WHICH:	1		
2150	011560	002	002			.BYTE	2,2	
2151	011562	001256				TEMPS		
2152								
2153	011564	032737	000001	001236		VECMAP:	BIT	#SW00,STRTSW
2154	011572	001114				BNE	5\$	
2155	011574	012737	000340	000022		MOV	#340,2#22	;SET IOT TRAP PRIO TO 7
2156	011602	012737	011756	000020		MOV	#4\$,2#20	;SET IOT TRAP VECTOR
2157	011610	012702	001500			MOV	#DM.MAP,R2	;SET SOFTWARE POINTER
2158	011614	012700	000300			MOV	#300,RO	;FLOATING VECTORS START HERE.
2159	011620	012701	000302			MOV	#302,R1	;PC OF IOT INSTR.
2160	011624	010120			1\$:	MOV	R1,(RO)+	;START FILLING VECTOR AREA
2161	011626	012721	000004			MOV	#4,(R1)+	;WITH .+2; IOT
2162	011632	022021				CMP	(RO)+,(R1)+	;ADD 2 TO RO +R1
2163	011634	020127	001000			CMP	R1,#1000	
2164	011640	101771				BLOS	1\$;BR IF MORE TO FILL
2165	011642	013737	001306	001246		MOV	DMACTV,TEMP1	;STORE TEMPORALLY
2166	011650	006037	001246		2\$:	ROR	TEMP1	;BRING OUT A BIT
2167	011654	103063				BCC	5\$;BR IF ALL DONE
2168	011656	012704	000012			MOV	#12,R4	;R4 IS INDEX REGISTER
2169	011662	016437	012026	177776		MOV	BRLVL(R4),PS	;SET PS TO 7
2170	011670	011201				MOV	(R2),R1	
2171	011672	012761	000200	000004		MOV	#200,4(R1)	
2172	011700	012711	001000			MOV	#BIT9,(R1)	;SET ROMI
2173	011704	012761	121111	000006		MOV	#121111,6(R1)	;PUT INSTRUCTION IN PORT6
2174	011712	012711	001400			MOV	#BIT9:BIT8,(R1)	;FORCE AN INTERRUPT
2175	011716	105200			7\$:	INCB	RO	;STALL
2176	011720	001376				BNE	.-2	;FOR TIME TO INTERUPT

K04

```

2177 011722 162704 000002          SUB      #2,R4          ;GET NEXT LOWEST PS LEVEL
2178 011726 001404          BEQ      6$          ;BR IF R4 = 0
2179 011730 016437 012026 177776  MOV     BRLVL(R4),PS ;MOVE NEXT LOWER LEVEL IN PS
2180 011736 000767          BR       7$          ;BR TO DELAY
2181 011740 052762 005300 000002  6$:    BIS     #5300,2(R2) ;NO INTERRUPT ASSUME 300 AT LEVEL 5 AND FIX DMC11
2182 011746 005011          3$:    CLR     (R1)      ;CLEAR ROMI
2183 011750 062702 000010          ADD     #10,R2      ;POP SOFTWARE POINTER
2184 011754 000735          BR       2$          ;KEEP GOING
2185 011756 051662 000002          4$:    BIS     (SP),2(R2) ;GET VECTOR ADDRESS
2186 011762 042762 000007 000002  BIC     #7,2(R2)    ;CLEAR JUNK
2187 011770 016405 012030          MOV     BRLVL+2(R4),R5 ;GET BR LEVEL OF DMC11
2188 011774 006305          ASL     R5          ;SHIFT LEVEL 4 PLACES
2189 011776 006305          ASL     R5          ;TO THE LEFT FOR THE
2190 012000 006305          ASL     R5          ;STATUS TABLE
2191 012002 006305          ASL     R5
2192 012004 042705 170777          BIC     #170777,R5 ;CLEAR UNWANTED BITS
2193 012010 050562 000002          BIS     R5,2(R2)   ;PUT BR LEVEL IN STATUS TABLE
2194 012014 022626          CMP     (SP)+,(SP)+ ;POP IOT JUNK OFF STACK
2195 012016 012716 011746          MOV     #3$,(SP)   ;SET FOR RETURN
2196 012022 000002          RTI
2197 012024 000207          5$:    RTS     PC          ;ALL DONE WITH "AUTO SIZING"
2198
2199 012026 000000          BRLVL: 0           ;LEVEL 0
2200 012030 000000          0       ;LEVEL 0
2201 012032 000200          200     ;LEVEL 4
2202 012034 000240          240     ;LEVEL 5
2203 012036 000300          300     ;LEVEL 6
2204 012040 000340          340     ;LEVEL 7
2205
2206
2207 012042 105777 167136          INTTY: TSTB   @TKCSR   ;WAIT FOR DONE
2208 012046 100375          BPL     -4
2209 012050 017703 167132          MOV     @TKDBR,R3   ;PUT CHAR IN R3
2210 012054 105777 167130          TSTB   @TPCSR      ;WAIT UNTIL PRINTER IS READY
2211 012060 100375          BPL     -4
2212 012062 010377 167124          MOV     R3,@TPDBR   ;ECHO CHAR
2213 012066 042703 000240          BIC     #BIT7!BITS,R3 ;MASK OFF LOWER CASE
2214 012072 000207          RTS     PC          ;RETURN
2215
2216          02100
2217
2218
2219          ;***** TEST 1 *****
2220          ;*OUT CONTROL REGISTER READ/ONLY TEST
2221          ;*DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
2222          ;*BITS ARE IN THE CORRECT STATE
2223          ;*****
2224
2225          ; TEST 1
2226          -----
2227 012074 012737 000001 001226          TST1: MOV     #1,TSTNO
2228 012102 012737 012150 001216          MOV     #TST2,NEXT
2229
2230          ;R1 CONTAINS BASE DMC11 ADDRESS
2231 012110 005077 167270          CLR     @DMCSR      ;CLEAR SEL0
2232 012114 012702 000011          MOV     #11,R2     ;SAVE R2 FOR TYPEOUT
2233 012120 104414          ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
  
```

L04

```

2233 012122 021224          021004! <20*11>          ;PORT4+LINE UNIT REG 11
2234 012124 016104 000004  MOV          4(R1),R4          ;PUT "FOUND" IN R4
2235 012130 042704 000054  BIC          #54,R4          ;CLEAR UNKNOWN BITS
2236 012134 012705 000020  MOV          #20,R5          ;PUT "EXPECTED" IN R5
2237 012140 120504          CMPB         R5,R4          ;IS OUT READY SET?
2238 012142 001401          BEQ          1$            ;BR IF YES
2239 012144 104002          HLT          2              ;ERROR IN LU 11
2240 012146 104400          1$: SCOPE                ;SCOPE THIS TEST
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251 012150 012737 000002 001226  TST2: MOV          #2,TSTNO
2252 012156 012737 012216 001216  MOV          #TST3,NEXT
2253
2254 012164 012702 000012          MOV          #12,R2          ;R1 CONTAINS BASE DMC11 ADDRESS
2255 012170 104414          ROMCLK       ;SAVE R2 FOR TYPEOUT
2256 012172 021244          021004! <20*12>          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2257 012174 016104 000004  MOV          4(R1),R4          ;PORT4+LINE UNIT REG 12
2258 012200 042704 000017  BIC          #17,R4          ;PUT "FOUND" IN R4
2259 012204 005005          CLR          R5              ;CLEAR UNKNOWN BITS
2260 012206 120504          CMPB         R5,R4          ;PUT "EXPECTED" IN R5
2261 012210 001401          BEQ          1$            ;ARE ALL BITS CLEARED?
2262 012212 104002          HLT          2              ;BR IF YES
2263 012214 104400          1$: SCOPE                ;ERROR IN LU 12
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274 012216 012737 000003 001226  TST3: MOV          #3,TSTNO
2275 012224 012737 012270 001216  MOV          #TST4,NEXT
2276
2277 012232 104412          MSTCLR       ;R1 CONTAINS BASE DMC11 ADDRESS
2278 012234 012702 000013          MOV          #13,R2          ;MASTER CLEAR DMC11
2279 012240 104414          ROMCLK       ;SAVE R2 FOR TYPEOUT
2280 012242 021264          021004! <20*13>          ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2281 012244 016104 000004  MOV          4(R1),R4          ;PORT4+LINE UNIT REG 13
2282 012250 042704 000213  BIC          #213,R4          ;PUT "FOUND" IN R4
2283 012254 012705 000100  MOV          #100,R5          ;CLEAR UNKNOWN BITS
2284 012260 120504          CMPB         R5,R4          ;PUT "EXPECTED" IN R5
2285 012262 001401          BEQ          1$            ;ARE RING, DTR, AND MODEM READY SET?
2286 012264 104002          HLT          2              ;BR IF YES
2287 012266 104400          1$: SCOPE                ;ERROR IN LU 13
2288

```

M04

```

***** TEST 4 *****
*MAINTENANCE REGISTER READ/ONLY TEST
*DO A MASTER CLEAR, VERIFY THAT ALL READ/ONLY
*BITS ARE IN THE CORRECT STATE
*****
  
```

TEST 4

```

-----
TST4:  MOV    #4,TSTNO
      MOV    #TST5,NEXT
      MSTCLR
      MOV    #17,R2
      ROMCLK
      021004! <20*17>
      MOV    4(R1),R4
      BIC    #206,R4
      MOV    #51,R5
      BIT    #BIT13,STAT1
      BEQ    .+12
      BIC    #40,R4
      BIC    #BIT5,R5
      CMPB   R5,R4
      BEQ    1$
      HLT    2
1$:    SCOPE
      ;R1 CONTAINS BASE DMC11 ADDRESS
      ;MASTER CLEAR DMC11
      ;SAVE R2 FOR TYPEOUT
      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      ;PORT4+LINE UNIT REG 17
      ;PUT "FOUND" IN R4
      ;CLEAR UNKNOWN BITS
      ;PUT "EXPECTED" IN R5
      ;IS LU AN M8202 OR M8201?
      ;BR IF M8201
      ;MASK OFF SI BIT IF M8202
      ;SI BIT IS UNKNOWN ON AN M8202
      ;ARE SI AND ICIR SET?
      ;BR IF YES
      ;ERROR IN LU 17
      ;SCOPE THIS TEST
  
```

```

***** TEST 5 *****
*LINE UNIT REGISTER WRITE/READ TEST
*SET BITS IN LU REGISTER 12, VERIFY IT IS SET
*CLEAR BITS IN LU REGISTER 12, VERIFY IT IS CLEAR
*****
  
```

TEST 5

```

-----
TST5:  MOV    #5,TSTNO
      MOV    #TST6,NEXT
      MOV    #1$,LOCK
      MSTCLR
      MOV    #12,R2
      MOV    #40,4(R1)
      ROMCLK
      122112
      ROMCLK
      021245
      MOV    #40,R5
      MOVB   5(R1),R4
      BIC    #337,R4
      CMPB   R5,R4
      BEQ    2$
      HLT    3
2$:    SCOP1
      MOV    #3$,LOCK
      ;R1 CONTAINS BASE DMC11 ADDRESS
      ;MASTER CLEAR DMC11
      ;SAVE REGISTER ADDRESS FOR TYPEOUT
      ;LOAD PORT4
      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      ;SET BITS IN LU-12
      ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
      ;READ LU-12
      ;PUT "EXPECTED" IN R5
      ;PUT "FOUND" IN R4
      ;CLEAR UNWANTED BITS
      ;IS BITS SET?
      ;BR IF YES
      ;ERROR, BIT 5 IS NOT SET
      ;SCOPE SUBTEST (SW09=1)
      ;NEW SCOP1
  
```

2289					
2290					
2291					
2292					
2293					
2294					
2295					
2296					
2297					
2298	012270	012737	000004	001226	
2299	012276	012737	012362	001216	
2300					
2301	012304	104412			
2302	012306	012702	000017		
2303	012312	104414			
2304	012314	021364			
2305	012316	016104	000004		
2306	012322	042704	000206		
2307	012326	012705	000051		
2308	012332	032737	020000	001366	
2309	012340	001404			
2310	012342	042704	000040		
2311	012346	042705	000040		
2312	012352	120504			
2313	012354	001401			
2314	012356	104002			
2315	012360	104400			
2316					
2317					
2318					
2319					
2320					
2321					
2322					
2323					
2324					
2325					
2326	012362	012737	000005	001226	
2327	012370	012737	012522	001216	
2328	012376	012737	012412	001220	
2329					
2330	012404	104412			
2331	012406	012702	000012		
2332	012412	012761	000040	000004	
2333	012420	104414			
2334	012422	122112			
2335	012424	104414			
2336	012426	021245			
2337	012430	012705	000040		
2338	012434	116104	000005		
2339	012440	042704	000337		
2340	012444	120504			
2341	012446	001401			
2342	012450	104003			
2343	012452	104401			
2344	012454	012737	012462	001220	

013070 104400
 013072 012737 000010 001226
 013100 012737 013246 001216
 013106 012737 013126 001220
 013114 104412
 013116 012702 000014
 013130 012700 000001
 013132 010061 000004
 013134 104414
 013136 122114
 013138 104414
 013140 021305
 013142 010005
 013144 116104 000005
 013150 120504
 013152 001401
 013154 104003
 013156 104401
 013160 000241
 013162 106100
 013164 001360
 013166 012737 013202 001220
 013174 012700 000001
 013200 005100
 013202 010061 000004
 013204 104414
 013210 122114
 013212 104414
 013214 021305
 013216 010005
 013220 116104 000005
 013224 120504
 013226 001401
 013230 104003
 013232 104401
 013234 005100
 013236 000241
 013240 106100
 013242 001360
 013244 104400

SCOPE

;SCOPE THIS TEST

***** TEST 10 *****
 :LINE UNIT REGISTER WRITE/READ TEST
 :FLOAT A 1 THROUGH LINE UNIT REGISTER 14
 :FLOAT A 0 THROUGH LINE UNIT REGISTER 14

: TEST 10

```

TST10:  MOV    #10,TSTNO
        MOV    #TST11,NEXT
        MOV    #64$,LOCK

MSTCLR
        MOV    #14,R2
        MOV    #1,R0
64$:   MOV    R0,4(R1)
        ROMCLK 122100!14
        ROMCLK 21005!<14*20>
        MOV    R0,R5
        MOVB   5(R1),R4
        CMPB   R5,R4
        BEQ    65$
        HLT    3
65$:   SCOPI
        CLC
        ROLB   R0
        BNE   64$
        MOV    #67$,LOCK
        MOV    #1,R0
66$:   COM    R0
67$:   MOV    R0,4(R1)
        ROMCLK 122100!14
        ROMCLK 21005!<14*20>
        MOV    R0,R5
        MOVB   5(R1),R4
        CMPB   R5,R4
        BEQ    68$
        HLT    3
68$:   SCOPI
        COM    R0
        CLC
        ROLB   R0
        BNE   69$
        SCOPE

:R1 CONTAINS BASE DMC11 ADDRESS
:MASTER CLEAR DMC11
:SAVE REGISTER ADDRESS FOR TYPEOUT
:START WITH BIT 0

:PUT PATTERN INTO PORT4
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:MOV DATA TO IBUS REGISTER 14
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:READ FROM IBUS REGISTER 14
:PUT EXPECTED IN R5
:PUT "FOUND" INTO R4
:DATA CORRECT?
:BR IF YES
:ERROR
:SW09=1?
:CLEAR CARRY
:SHIFT BIT IN R0
:IF R0=0 THEN DONE
:NEW SCOPE!
:START WITH BIT 0
:CHANGE TO FLOATING ZERO

:PUT PATTERN INTO PORT4
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:MOV DATA TO IBUS REGISTER 14
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:READ FROM IBUS REGISTER 14
:PUT EXPECTED IN R5
:PUT "FOUND" INTO R4
:DATA CORRECT?
:BR IF YES
:ERROR
:SW09=1?
:CHANGE TO FLOATING 1
:CLEAR CARRY
:SHIFT BIT IN R0
:IF R0=0 THEN DONE
:SCOPE THIS TEST
  
```

***** TEST 11 *****

D05

```
:*SWITCH PAC TEST
:*THIS TEST READS SWITCH PAC#1
:*THIS SWITCH PAC CONTAINS THE DDCMP LINE #
:*****
```

: TEST 11

013246 012737 000011 001226
013254 012737 013310 001216

013262 104412
013270 104414
013278 021324
013286 016104 000004
013294 113705 001370
013300 120504
013306 001401
013312 104031
013318 104400

TST11:

1S:

```
MOV #11,TSTNO
MOV #TST12,NEXT
MSTCLR
ROMCLK
021324
MOV 4(R1),R4
MOV8 STAT2,R5
CMPB R5,R4
BEQ 1S
HLT 31
SCOPE
:R1 CONTAINS BASE DMC11 ADDRESS
:MASTER CLEAR DMC11
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:PORT4+LU15
:PUT "FOUND" IN R4
:PUT "EXPECTED" IN R5
:SW OK?
:BR IF YES
:ERROR, SWITCH PAC READ ERROR
:SCOPE THIS TEST
```

```
:***** TEST 12 *****
:*SWITCH PAC TEST
:*THIS TEST READS SWITCH PAC#2
:*THIS SWITCH PAC CONTAINS THE BM873 BOOT ADD
:*****
```

: TEST 12

013310 012737 000012 001226
013316 012737 013352 001216

013324 104412
013330 104414
013336 021344
013342 016104 000004
013348 113705 001371
013354 120504
013360 001401
013366 104031
013372 104400

TST12:

1S:

```
MOV #12,TSTNO
MOV #TST13,NEXT
MSTCLR
ROMCLK
021344
MOV 4(R1),R4
MOV8 STAT2+1,R5
CMPB R5,R4
BEQ 1S
HLT 31
SCOPE
:R1 CONTAINS BASE DMC11 ADDRESS
:MASTER CLEAR DMC11
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:PORT4+LU16
:PUT "FOUND" IN R4
:PUT "EXPECTED" IN R5
:SW OK?
:BR IF YES
:ERROR, SWITCH PAC READ ERROR
:SCOPE THIS TEST
```

```
:***** TEST 13 *****
:*LINE UNIT CLOCK TEST
:*THIS TEST VERIFYS THAT THE LU INTERNAL CLOCK
:*(BIT 1 IN LU-17) IS WORKING
:*****
```

: TEST 13

013352 012737 000013 001226
013360 012737 013452 001216

013366 104412
013370 005037 007364

TST13:

1S:

```
MOV #13,TSTNO
MOV #TST14,NEXT
MSTCLR
CLR TEMP
:R1 CONTAINS BASE DMC11 ADDRESS
:MASTER CLEAR DMC11
:PREPARE FOR DELAY
```


E05

```

013374
013374 104414
013376 021364
013400 032761 000002 000004
013406 001004
013410 005237 007364
013414 001367
013416 104004
013420 005037 007364
013424 104414
013426 021364
013430 032761 000002 000004
013436 001404
013440 005237 007364
013444 001367
013446 104004
013450 104400
  
```

```

18: ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021364 :PORT4+LU-17
BIT #2,4(R1) :IS CLOCK BIT SET?
BNE 23 :BR IF YES
INC TEMP :DELAY
BNE 19 :DELAY FINISHED?
HLT 4 :ERROR BIT IS STUCK CLEAR
CLR TEMP :PREPARE FOR DELAY

39: ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021364 :PORT4+LU-17
BIT #2,4(R1) :IS CLOCK BIT CLEAR?
BEQ 49 :BR IF YES
INC TEMP :DELAY
BNE 39 :BR IF DELAY NOT DONE
HLT 4 :ERROR BIT IS STUCK SET

45: SCOPE
  
```

```

:***** TEST 14 *****
:OUT DATA SILO TEST
:SET SOM AND LOAD OUT DATA SILO
:VERIFY THAT OCOR SET, INDICATING THAT THE
:CHARACTER IS AT THE BOTTOM OF THE OUT SILO
:*****
  
```

: TEST 14

```

013452 012737 000014 001226
013460 012737 013552 001216

013466 104412
013470 012711 004000
013474 012761 000001 000004
013502 104414
013504 122111
013506 104414
013510 122110
013512 104416 000002
013516 012702 000017
013522 104414
013524 021364
013526 016104 000004
013532 042704 000357
013536 012705 000020
013542 120504
013544 001401
013546 104005
013550
013550 104400
  
```

```

TST14: MOV #14,TSTNO
MOV #TST15,NEXT

MSTCLR :R1 CONTAINS BASE DMC11 ADDRESS
MOV #BIT11 (R1) :MASTER CLEAR DMC11
MOV #1,4(R1) :SET LINE UNIT LOOP
ROMCLK :LOAD PORT4 WITH BIT0
122111 :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
ROMCLK :SET SOM
122110 :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
TIMER, 2 :LOAD OUT DATA SILO
MOV #17,R2 :WAIT FOR OCOR
ROMCLK :SAVE ADDRESS FOR TYPEOUT
021364 :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
MOV 4(R1),R4 :PORT4+LU 17
BIC #357,R4 :PUT "FOUND" IN R4
MOV #20,R5 :CLEAR UNWANTED BITS
CMPB R5,R4 :PUT "EXPECTED" IN R5
BEQ 18 :IS OCOR SET?
HLT 5 :BR IF YES

19: SCOPE :SCOPE THIS TEST
  
```

```

:***** TEST 15 *****
:DDCMP TEST OF RTS AND OUT ACTIVE
:SET SOM AND LOAD OUT DATA SILO
  
```

F05

```

: *SINGLE STEP 2 DATA CLOCKS, VERIFY
: *THAT RTS AND ACTIVE ARE SET
: *****

```

TEST 15

```

013552 012737 000015 001226
013560 012737 013710 001216

013566 104412
013570 012711 004000
013574 012761 000001 000004
013602 104414
013604 122111
013606 104414
013610 122110
013612 004737 026144
013616 104415 000002
013622 012702 000011
013626 104414
013630 021224
013632 016104 000004
013636 042704 000257
013642 012705 000120
013646 120504
013650 001401
013652 104005
013654
013654 012702 000013
013660 104414
013662 021264
013664 016104 000004
013670 042704 000337
013674 012705 000040
013700 120504
013702 001401
013704 104005
013706
013706 104400

```

```

TST15: MOV #15,TSTNO
MOV #TST16,NEXT

MSTCLR ;R1 CONTAINS BASE DMC11 ADDRESS
MOV #BIT11,(R1) ;MASTER CLEAR DMC11
MOV #1,4(R1) ;SET LINE UNIT LOOP
ROMCLK ;LOAD PORT4 WITH BIT0
122111 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
ROMCLK ;SET SOM
122110 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR PC,OCOR ;LOAD OUT DATA SILO
DATACLK, ;WAIT FOR OCOR
MOV #11,R2 ;CLOCK DATA FOUR TIMES
ROMCLK ;SAVE ADDRESS FOR TYPEOUT
021224 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
MOV 4(R1),R4 ;PORT4+LU 11
BIC #257,R4 ;PUT "FOUND" IN R4
MOV #120,R5 ;CLEAR UNWANTED BITS
CMPB R5,R4 ;PUT "EXPECTED" IN R5
BEQ 1$ ;IS ACTIVE SET?
HLT 5 ;BR IF YES

15: MOV #13,R2 ;SAVE ADDRESS FOR TYPEOUT
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021264 ;PORT4+LU 13
MOV 4(R1),R4 ;PUT EXPECTED IN R4
BIC #337,R4 ;CLEAR UNWANTED BITS
MOV #BITS,R5 ;PUT "EXPECTED" IN R5, RTS SHOULD BE SET
CMPB R5,R4 ;IS RTS OK?
BEQ 2$ ;BR IF YES
HLT 5 ;RTS ERROR

23: SCOPE ;SCOPE THIS TEST

```

```

: ***** TEST 16 *****
: *TEST OF OUT CLEAR
: *SET SOM AND LOAD OUT DATA SILO
: *SINGLE STEP DATA CLOCK, SET OUT CLEAR
: *VERIFY THAT OCOR, RTS, AND ACTIVE ARE CLEARED
: *****

```

TEST 16

```

013710 012737 000016 001226
013716 012737 014106 001216

013724 104412
013726 012711 004000
013732 012761 000001 000004

```

```

TST16: MOV #16,TSTNO
MOV #TST17,NEXT

MSTCLR ;R1 CONTAINS BASE DMC11 ADDRESS
MOV #BIT11,(R1) ;MASTER CLEAR DMC11
MOV #1,4(R1) ;SET LINE UNIT LOOP
;LOAD PORT4 WITH BIT0

```

G05

2681	013740	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304	
2682	013742	122111			122111		:SET SOM	
2683	013744	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304	
2684	013746	122110			122110		:LOAD OUT DATA SILO	
2685	013750	004737	026144		JSR	PC,OCOR	:WAIT FOR OCOR	
2686	013754	104415	000002		DATACLK,	2	:CLOCK DATA FOUR TIMES	
2687	013760	012761	000200	000004	MOV	#BIT7,4(R1)	:SET BIT7 IN PORT4	
2688	013766	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304	
2689	013770	122111			122111		:SET OUT CLEAR	
2690	013772	104415	000001		DATACLK,	1	:GIVE A TICK TO CLEAR RTS	
2691	013776	012702	000017		MOV	#17,R2	:SAVE ADDRESS FOR TYPEOUT	
2692	014002	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304	
2693	014004	021364			021364		:PORT4+LU 17	
2694	014006	016104	000004		MOV	4(R1),R4	:PUT "FOUND" IN R4	
2695	014012	042704	000357		BIC	#357,R4	:CLEAR UNWANTED BITS	
2696	014016	005005			CLR	R5	:PUT "EXPECTED" IN R5	
2697	014020	120504			CMPB	R5,R4	:IS OCOR CLEARED?	
2698	014022	001401			BEQ	1\$:BR IF YES	
2699	014024	104005			HLT	5		
2700	014026							
2701	014026	012702	000013		1\$:	MOV	#13,R2	:SAVE ADDRESS FOR TYPEOUT
2702	014032	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304	
2703	014034	021264			021264		:PORT4+LU 13	
2704	014036	016104	000004		MOV	4(R1),R4	:PUT EXPECTED IN R4	
2705	014042	042704	000337		BIC	#337,R4	:CLEAR UNWANTED BITS	
2706	014046	005005			CLR	R5	:PUT "EXPECTED" IN R5, RTS SHOULD BE CLEARED	
2707	014050	120504			CMPB	R5,R4	:IS RTS OK?	
2708	014052	001401			BEQ	2\$:BR IF YES	
2709	014054	104005			HLT	5	:RTS ERROR	
2710	014056							
2711	014056	012702	000011		2\$:	MOV	#11,R2	:SAVE ADDRESS FOR TYPEOUT
2712	014062	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304	
2713	014064	021224			021224		:PORT4+LU 11	
2714	014066	016104	000004		MOV	4(R1),R4	:PUT "FOUND" IN R4	
2715	014072	012705	000020		MOV	#BIT4,R5	:ONLY OUT READY SHOULD BE SET	
2716	014076	120504			CMPB	R5,R4	:IS ACTIVE CLEAR?	
2717	014100	001401			BEQ	3\$:BR IF YES	
2718	014102	104005			HLT	5	:ERROR ACTIVE NOT CLEARED	
2719	014104							
2720	014104	104400			3\$:	SCOPE	:SCOPE THIS TEST	
2721								
2722								
2723								
2724								
2725								
2726								
2727								
2728								
2729								
2730								
2731								
2732								
2733	014106	012737	000017	001226	TST17:	MOV	#17,TSTNO	
2734	014114	012737	014270	001216		MOV	#TST20,NEXT	
2735								
2736	014122	104412				MSTCLR	:R1 CONTAINS BASE DMC11 ADDRESS :MASTER CLEAR DMC11	

```

:***** TEST 17 *****
:DDCMP TRANSMITTER TEST
:SINGLE CLOCK THE CHARACTER 0
:VERIFY EACH BIT POSITION AS IT
:PASSES THE BIT WINDOW (SI BIT)
:ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
:*****

```

: TEST 17

H05

2737 014124 012711 004000
2738 014130 004737 026276
2739 014134 012761 000001 000004
2740 014142 104414
2741 014144 122111
2742 014146 012705 000000
2743 014152 004737 026276
2744 014156 010561 000004
2745 014162 104414
2746 014164 122110
2747 014166 004737 026144
2748 014172 005003
2749 014174 010502
2750 014176 104415 000002
2751 014202 104415 000001
2752 014206 106002
2753 014210 103005
2754 014212 004737 026112
2755 014216 103406
2756 014220 104006
2757 014222 000404
2758 014224 004737 026112
2759 014230 103001
2760 014232 104006
2761 014234
2762 014234 005203
2763 014236 022703 000010
2764 014242 001357
2765 014244 104415 000014
2766 014250 104414
2767 014252 021264
2768 014254 032761 000040 000004
2769 014262 001401
2770 014264 104034
2771 014266 104400
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784 014270 012737 000020 001226
2785 014276 012737 014452 001216
2786
2787 014304 104412
2788 014306 012711 004000
2789 014312 004737 026276
2790 014316 012761 000001 000004
2791 014324 104414
2792 014326 122111

```
MOV #BIT11,(R1) ;SET LINE UNIT LOOP
JSR PC,OUTRDY ;WAIT FOR OUT-READY
MOV #1,4(R1) ;SET BIT0 IN PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122111 ;SET SOM!
MOV #0,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
JSR PC,OUTRDY ;WAIT FOR OUT-READY
MOV R5,4(R1) ;LOAD PORT4 WITH CHARACTER
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122110 ;LOAD OUT DATA
JSR PC,OCOR ;WAIT FOR OCOR TO SET
CLR R3 ;CLEAR BIT COUNTER
MOV R5,R2 ;LOAD CHARACTER IN R2
DATACLK, 2 ;2 TICKS TO SET UP TRANSMITTER
DATACLK, 1 ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
RORB R2 ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
BCC 2$ ;BR IF CARRY CLEAR
JSR PC,GETSI ;GET THE WINDOW
BCS 3$ ;BR IF BIT IS A MARK
HLT 6 ;ERROR BIT WAS A SPACE
BR 3$ ;CONTINUE WITH TEST
2$: JSR PC,GETSI ;GET THE WINDOW
BCC 3$ ;BR IF BIT IS A SPACE
HLT 6 ;ERROR BIT WAS A MARK
3$: INC R3 ;NEXT BIT
CMP #10,R3 ;DONE YET?
BNE 1$ ;BR IF NO
DATACLK, 14 ;CLOCK TRANSMITTER 14 MORE TICKS
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021264 ;PORT4-LU-13
BIT #BITS,4(R1) ;RTS SHOULD BE CLEAR NOW
BEQ 4$ ;BR IF YES
HLT 34 ;ERROR, RTS NOT CLEAR
4$: SCOPE ;SCOPE THIS TEST
```

***** TEST 20 *****
*DDCMP TRANSMITTER TEST
*SINGLE CLOCK THE CHARACTER 125
*VERIFY EACH BIT POSITION AS IT
*PASSES THE BIT WINDOW (SI BIT)
*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE

: TEST 20

```
TST20: MOV #20,TSTNO
MOV #TST21,NEXT
MSTCLR ;R1 CONTAINS BASE DMC11 ADDRESS
MOV #BIT11,(R1) ;MASTER CLEAR DMC1
JSR PC,OUTRDY ;SET LINE UNIT LOOP
MOV #1,4(R1) ;WAIT FOR OUT-READY
ROMCLK ;SET BIT0 IN PORT4
122111 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;SET SOM!
```


J05

```

2859 014532 004737 026144 JSR PC,OCOR ;WAIT FOR OCOR TO SET
2860 014536 005003 CLR R3 ;CLEAR BIT COUNTER
2861 014540 010502 MOV R5,R2 ;LOAD CHARACTER IN R2
2862 014542 104415 000002 DATACLK, 2 ;2 TICKS TO SET UP TRANSMITTER
2863 014546 104415 000001 1$: DATACLK, 1 ;SHIFT NEXT BIT IN THE WINDOW (SI BIT)
2864 014552 106002 RORB R2 ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
2865 014554 103005 BCC 2$ ;BR IF CARRY CLEAR
2866 014556 004737 026112 JSR PC,GETSI ;GET THE WINDOW
2867 014562 103406 BCS 3$ ;BR IF BIT IS A MARK
2868 014564 104006 HLT 6 ;ERROR BIT WAS A SPACE
2869 014566 000404 BR 3$ ;CONTINUE WITH TEST
2870 014570 004737 026112 2$: JSR PC,GETSI ;GET THE WINDOW
2871 014574 103001 BCC 3$ ;BR IF BIT IS A SPACE
2872 014576 104006 HLT 6 ;ERROR BIT WAS A MARK
2873 014600 3$: INC R3 ;NEXT BIT
2874 014600 005203 CMP #10,R3 ;DONE YET?
2875 014602 022703 000010 BNE 1$ ;BR IF NO
2876 014606 001357 DATACLK, 14 ;CLOCK TRANSMITTER 14 MORE TICKS
2877 014610 104415 000014 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2878 014614 104414 021264 ;PORT4+LU-13
2879 014616 021264 BIT #BITS,4(R1) ;RTS SHOULD BE CLEAR NOW
2880 014620 032761 000040 000004 BEQ 4$ ;BR IF YES
2881 014626 001401 HLT 34 ;ERROR, RTS NOT CLEAR
2882 014630 104034 4$: SCOPE ;SCOPE THIS TEST
2883 014632 104400

```

```

***** TEST 22 *****
*DDCMP TRANSMITTER TEST
*SINGLE CLOCK THE CHARACTER 377
*VERIFY EACH BIT POSITION AS IT
*PASSES THE BIT WINDOW (SI BIT)
*ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
*****

```

: TEST 22

```

2886 014634 012737 000022 001226 TST22: MOV #22,TSTNO
2887 014642 012737 015016 001216 MOV #TST23,NEXT
2888
2889 014650 104412 MSTCLR ;R1 CONTAINS BASE DMC11 ADDRESS
2890 014652 012711 004000 MOV #BIT11,(R1) ;MASTER CLEAR DMC11
2891 014656 004737 026276 JSR PC,OUTRDY ;SET LINE UNIT LOOP
2892 014662 012761 000001 000004 MOV #1,4(R1) ;WAIT FOR OUT-READY
2893 014670 104414 ROMCLK ;SET BIT0 IN PORT4
2894 014672 122111 122111 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2895 014674 012705 000377 MOV #377,R5 ;LOAD CHARACTER IN R5 FOR TYPEOUT
2896 014700 004737 026276 JSR PC,OUTRDY ;SET SOM!
2897 014704 010561 000004 MOV R5,4(R1) ;LOAD CHARACTER IN R5 FOR TYPEOUT
2898 014710 104414 ROMCLK ;WAIT FOR OUT-READY
2899 014712 122110 122110 ;LOAD PORT4 WITH CHARACTER
2900 014714 004737 026144 JSR PC,OCOR ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2901 014720 005003 CLR R3 ;LOAD OUT DATA
2902 014722 010502 MOV R5,R2 ;WAIT FOR OCOR TO SET
2903 014724 104415 000002 DATACLK, 2 ;CLEAR BIT COUNTER
2904 014730 104415 000001 1$: DATACLK, 1 ;LOAD CHARACTER IN R2
;2 TICKS TO SET UP TRANSMITTER
;SHIFT NEXT BIT IN THE WINDOW (SI BIT)

```

K05

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 63
DZDMEA.P11 BASIC TRANSMITTER TESTS

```

2905 014734 106002 RORB R2 ;SHIFT NEXT SOFTWARE BIT IN TO CARRY
2906 014736 103005 BCC 2$ ;BR IF CARRY CLEAR
2907 014740 004737 026112 JSR PC,GETSI ;GET THE WINDOW
2908 014744 103406 BCS 3$ ;BR IF BIT IS A MARK
2909 014746 104006 HLT 6 ;ERROR BIT WAS A SPACE
2910 014750 000404 BR 3$ ;CONTINE WITH TEST
2911 014752 004737 026112 2$: JSR PC,GETSI ;GET THE WINDOW
2912 014756 103001 BCC 3$ ;BR IF BIT IS A SPACE
2913 014760 104006 HLT 6 ;ERROR BIT WAS A MARK
2914 014762 3$:
2915 014762 005203 INC R3 ;NEXT BIT
2916 014764 022703 000010 CMP #10,R3 ;DONE YET?
2917 014770 001357 BNE 1$ ;BR IF NO
2918 014772 104415 000014 DATACLK, 14 ;CLOCK TRANSMITTER 14 MORE TICKS
2919 014776 104414 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2920 015000 021264 021264 ;PORT4+LU-13
2921 015002 032761 000040 000004 BIT #BITS,4(R1) ;RTS SHOULD BE CLEAR NOW
2922 015010 001401 BEQ 4$ ;BR IF YES
2923 015012 104034 HLT 34 ;ERROR, RTS NOT CLEAR
2924 015014 104400 4$: SCOPE ;SCOPE THIS TEST

```

```

;***** TEST 23 *****
;DDCMP TRANSMITTER TEST
;SINGLE CLOCK A BINARY COUNT PATTERN
;VERIFY EACH BIT POSITION AS IT
;PASSES THE BIT WINDOW (SI BIT)
;ON AN ERROR, R3 CONTAINS BIT POSITION OF FAILURE
;AND R5 CONTAINS THE CHARACTER THAT FAILED
;*****

```

TEST 23

```

2937 ;-----
2938 015016 012737 000023 001226 TST23: MOV #23,TSTNO
2939 015024 012737 015224 001216 MOV #TST24,NEXT
2940 ;R1 CONTAINS BASE DMC11 ADDRESS
2941 015032 104412 MSTCLR ;MASTER CLEAR DMC11
2942 015034 012711 004000 MOV #BIT11,(R1) ;SET LINE UNIT LOOP
2943 015040 005003 CLR R3 ;R3 CONTAINS BIT COUNT
2944 015042 005004 CLR R4 ;R4 CONTAINS CHAR TO BE LOADED IN SILO
2945 015044 005005 CLR R5 ;R5 CONTAINS CHARACTER CURRENTLY BEING SHIFTED 0
2946 015046 004737 026276 JSR PC,OUTRDY ;WAIT FOR OUT-READY
2947 015052 012761 000001 000004 MOV #1,4(R1) ;SET BIT0 IN PORT4
2948 015060 104414 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2949 015062 122111 122111 ;SET SOM!
2950 015064 004737 026276 JSR PC,OUTRDY ;WAIT FOR OUT-READY
2951 015070 010461 000004 MOV R4,4(R1) ;LOAD PORT4 WITH CHARACTER
2952 015074 104414 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2953 015076 122110 122110 ;LOAD OUT DATA
2954 015100 005204 INC R4 ;INCREMENT TO NEXT CHARACTER
2955 015102 004737 026276 JSR PC,OUTRDY ;WAIT FOR OUT-READY
2956 015106 010461 000004 MOV R4,4(R1) ;LOAD PORT4 WITH CHARACTER
2957 015112 104414 ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2958 015114 122110 122110 ;LOAD OUT DATA
2959 015116 004737 026144 JSR PC,OCOR ;WAIT FOR OCOR TO SET
2960 015122 104415 000002 DATACLK, 2 ;2 TICKS TO SET UP TRANSMITTER

```

L05

2961	015126	005003			4\$:	CLR R3		:CLEAR BIT COUNTER
2962	015130	010502				MOV R5,R2		:LOAD CHARACTER IN R2
2963	015132	104415	000001		1\$:	DATACLK, 1		:SHIFT NEXT BIT IN THE WINDOW (SI BIT)
2964	015136	106002				RORB R2		:SHIFT NEXT SOFTWARE BIT IN TO CARRY
2965	015140	103005				BCC 2\$:BR IF CARRY CLEAR
2966	015142	004737	026112			JSR PC,GETSI		:GET THE WINDOW
2967	015146	103406				BCS 3\$:BR IF BIT IS A MARK
2968	015150	104006				HLT 6		:ERROR BIT WAS A SPACE
2969	015152	000404				BR 3\$:CONTINUE WITH TEST
2970	015154	004737	026112		2\$:	JSR PC,GETSI		:GET THE WINDOW
2971	015160	103001				BCC 3\$:BR IF BIT IS A SPACE
2972	015162	104006				HLT 6		:ERROR BIT WAS A MARK
2973	015164				3\$:			
2974	015164	005203				INC R3		:NEXT BIT
2975	015166	022703	000010			CMP #10,R3		:DONE YET?
2976	015172	001357				BNE 1\$:BR IF NO
2977	015174	005204				INC R4		:NEXT CHARACTER
2978	015176	004737	026276			JSR PC,OUTRDY		:WAIT FOR OUT-READY
2979	015202	010461	000004			MOV R4,4(R1)		:LOAD PORT4 WITH CHARACTER
2980	015206	104414				ROMCLK 122110		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
2981	015210	122110						:LOAD OUT DATA
2982	015212	005205				INC R5		:NEXT CHARACTER
2983	015214	022705	000400			CMP #400,R5		:DONE YET?
2984	015220	001342				BNE 4\$:BR IF NO
2985	015222	104400			5\$:	SCOPE		:SCOPE THIS TEST
2986								
2987								
2988								
2989								
2990								
2991								
2992								
2993								
2994								
2995								
2996	015224	012737	000024	001226	TST24:	MOV #24,TSTNO		
2997	015232	012737	015312	001216		MOV #TST25,NEXT		
2998								:R1 CONTAINS BASE DMC11 ADDRESS
2999	015240	104412				MSTCLR		:MASTER CLEAR DMC11
3000	015242	012711	004000			MOV #BIT11,(R1)		:SET LU LOOP
3001	015246	012702	000012			MOV #12,R2		:SAVE LU REG FOR TYPEOUT
3002	015252	004737	026162			JSR PC,SYNC		:SINGLE CLOCK 5 SYNC CHARACTERS
3003	015256	000005				5		
3004	015260	104415	000054			DATACLK, 54		
3005	015264	104414				ROMCLK 021244		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3006	015266	021244						:PORT4+LU12
3007	015270	016104	000004			MOV 4(R1),R4		:PUT "FOUND" IN R4
3008	015274	042704	000277			BIC #277,R4		:CLEAR UNWANTED BITS
3009	015300	005005				CLR R5		:PUT "EXPECTED" IN R5
3010	015302	120504				CMPB R5,R4		:IS ACTIVE CLEAR?
3011	015304	001401				BEQ 1\$:BR IF YES
3012	015306	104040				HLT 40		:ERROR ACTIVE IS NOT CLEAR
3013	015310	104400			1\$:	SCOPE		:SCOPE THIS TEST
3014								
3015								
3016								

:***** TEST 25 *****

N05

```

3073                                     ;*DDCMP IN ACTIVE TEST
3074                                     ;*SET LU LOOP, SINGLE STEP 2 SYNC AND A NON-SYNC (301)
3075                                     ;*VERIFY THAT IN ACTIVE IS SET
3076                                     ;*****
3077
3078                                     ; TEST 27
3079                                     -----
3080 015470 012737 000027 001226      TST27: MOV      #27,TSTNO
3081 015476 012737 015560 001216      MOV      #TST30,NEXT
3082
3083 015504 104412                       MSTCLR
3084 015506 012711 004000               MOV      #BIT11,(R1)
3085 015512 012702 000012               MOV      #12,R2
3086 015516 004737 026162               JSR      PC,SYNC
3087 015522 000002                       2
3088 015524 104415 000034               DATACLK,      34
3089 015530 104414                       ROMCLK
3090 015532 021244                       021244
3091 015534 016104 000004               MOV      4(R1),R4
3092 015540 042704 000277               BIC      #277,R4
3093 015544 012705 000100               MOV      #BIT6,R5
3094 015550 120504                       CMPB     R5,R4
3095 015552 001401                       BEQ      1$
3096 015554 104040                       HLT      40
3097 015556 104400                       1$: SCOPE
3098
3099
3100                                     ;***** TEST 30 *****
3101                                     ;*IN CLEAR TEST
3102                                     ;*SYNC UP RECEIVER AND TRANSMIT A CHARACTER
3103                                     ;*WAIT FOR IN RDY, THEN SET IN CLEAR
3104                                     ;*VERIFY THAT IN ACTIVE AND IN RDY ARE CLEARED
3105                                     ;*****
3106
3107                                     ; TEST 30
3108                                     -----
3109 015560 012737 000030 001226      TST30: MOV      #30,TSTNO
3110 015566 012737 015732 001216      MOV      #TST31,NEXT
3111
3112 015574 104412                       MSTCLR
3113 015576 012702 000012               MOV      #12,R2
3114 015602 012711 004000               MOV      #BIT11,(R1)
3115 015606 004737 026330               JSR      PC,CHAR
3116 015612 000301                       30!
3117 015614 104415 000053               DATACLK,      53
3118 015620 104416 000002               TIMER,      2
3119 015624 104414                       ROMCLK
3120 015626 021244                       021244
3121 015630 016104 000004               MOV      4(R1),R4
3122 015634 042704 000357               BIC      #357,R4
3123 015640 012705 000020               MOV      #BIT4,R5
3124 015644 120504                       CMPB     R5,R4
3125 015646 001401                       BEQ      1$
3126 015650 104040                       HLT      40
3127 015652
3128 015652 012761 000200 000004      1$: MOV      #BIT7,4(R1)

```

```

;R1 CONTAINS BASE DMC11 ADDRESS
;MASTER CLEAR DMC11
;SET LU LOOP
;SAVE LU REG FOR TYPEOUT
;SINGLE CLOCK 2 SYNC CHARACTERS

;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;PORT4+LU12
;PUT "FOUND" IN R4
;CLEAR UNWANTED BITS
;PUT "EXPECTED" IN R5

;IS ACTIVE SET?
;BR IF YES
;ERROR ACTIVE IS NOT SET
;SCOPE THIS TEST

```

```

;R1 CONTAINS BASE DMC11 ADDRESS
;MASTER CLEAR DMC11
;SAVE REG ADDRESS IN R2 FOR TYPEOUT
;SET LINE UNIT LOOP
;LOAD SILO WITH 3 SYNC
;AND A NON-SYNC (301)
;SINGLE CLOCK THE DATA
;WAIT FOR INRDY
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;PORT4+LU 12
;PUT "FOUND" IN R4
;CLEAR UNWANTED BITS
;PUT "EXPECTED" IN R5

;IS INRDY SET?
;ERROR, INRDY IS NOT SET
;LOAD PORT4

```

015730	104414		
015732	122112		
015734	104414		
015736	104414		
015738	000004		
015740	000277		
015742	000000		
015744	000000		
015746	000000		
015748	120504		
015750	001401		
015752	104414		
015754	104414		
015756	016104	000004	
015758	042704	000357	
015760	005005		
015762	120504		
015764	001401		
015766	104414		
015768	104414		
015770	016104	000004	
015772	042704	000357	
015774	012705	000020	
015776	120504		
015778	001401		
015780	104414		
015782	104414		
015784	000004		
015786	005005		
015788	120504		
015790	001401		
015792	104414		
015794	104414		

```

ROMCLK 122112 :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
ROMCLK 021244 :SET IN CLEAR
MOV 4(R1),R4 :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
BIC #277,R4 :PORT4+LU 12
CLR R5 :PUT "FOUND" IN R4
CMPB R5,R4 :CLEAR UNWANTED BITS
BEQ 28 :PUT "EXPECTED" IN R5
HLT 40 :IS IN ACTIVE CLEAR?

28: :ERROR, IN ACTIVE IS NOT CLEAR
MOV 4(R1),R4 :PUT "FOUND" IN R4
BIC #357,R4 :CLEAR UNWANTED BITS
CLR R5 :PUT "EXPECTED" IN R5
CMPB R5,R4 :IS INRDY CLEARED?
BEQ 38 :ERROR, INRDY IS NOT CLEARED
HLT 40 :SCOPE THIS TEST
SCOPE

```

```

:***** TEST 31 *****
:DDCMP BASIC RECEICER TEST
:SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 0
:VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED
:*****

```

```

: TEST 31
:-----
TST31: MOV #31,TSTNO
      MOV #TST32,NEXT

MSTCLR :R1 CONTAINS BASE DMC11 ADDRESS
MOV #12,R2 :MASTER CLEAR DMC11
MOV #BIT11,(R1) :SAVE REG ADDRESS IN R2 FOR TYPEOUT
JSR PC,CHAR :SET LINE UNIT LOOP
0 :LOAD SILO WITH 3 SYNC
DATACLK, 53 :AND THE CHARACTER 0
TIMER, 2 :SINGLE CLOCK THE DATA
ROMCLK :WAIT FOR INRDY
021244 :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
MOV 4(R1),R4 :PORT4+LU 12
BIC #357,R4 :PUT "FOUND" IN R4
MOV #BIT4,R5 :CLEAR UNWANTED BITS
CMPB R5,R4 :PUT "EXPECTED" IN R5
BEQ 18 :IS INRDY SET?
HLT 40 :ERROR, INRDY IS NOT SET

18: :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
ROMCLK 021204 :PORT4+IN DATA
MOV 4(R1),R4 :PUT "FOUND" IN R4
CLR R5 :PUT "EXPECTED" IN R5
CMPB R5,R4 :WAS A 0 RECEIVED?
BEQ 28 :ERROR, RECEIVED DATA IS WRONG
HLT 10 :SCOPE THIS TEST
SCOPE

```



```

016330 104414
016332 021244
016334 016104 000004
016340 042704 000357
016344 012705 000020
016350 120504
016352 001401
016354 104040
016356 104414
016358 021204
016360 016104 000004
016362 012705 000252
016370 120504
016374 001401
016376 104010
016300 104400

```

15:

25:

```

ROMCLK
021244
MOV 4(R1),R4
BIC #357,R4
MOV #BIT4,R5
CMPB R5,R4
BEQ 15
HLT 40

```

```

ROMCLK
021204
MOV 4(R1),R4
MOV #252,R5
CMPB R5,R4
BEQ 25
HLT 10
SCOPE

```

```

:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:PORT4+LU 12
:PUT "FOUND" IN R4
:CLEAR UNWANTED BITS
:PUT "EXPECTED" IN R5
:IS INRDY SET?
:ERROR, INRDY IS NOT SET

```

```

:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:PORT4+IN DATA
:PUT "FOUND" IN R4
:PUT "EXPECTED" IN R5
:WAS A 252 RECEIVED?
:ERROR, RECEIVED DATA IS WRONG
:SCOPE THIS TEST

```

```

:***** TEST 34 *****
:*DDCMP BASIC RECEIVER TEST
:*SYNC UP RECEIVER AND SINGLE CLOCK THE CHARACTER 377
:*VERIFY THAT IN RDY IS SET, AND THAT THE CHARACTER WAS RECEIVED
:*****

```

: TEST 34

TST34:

```

016302 012737 000034 001226
016310 012737 016420 001216
016316 104412
016320 012702 000012
016324 012711 004000
016330 004737 026330
016334 000377
016336 104415 000053
016342 104416 000002
016346 104414
016350 021244
016352 016104 000004
016356 042704 000357
016362 012705 000020
016366 120504
016370 001401
016372 104040
016374
016374 104414
016376 021204
016400 016104 000004
016404 012705 000377
016410 120504
016412 001401
016414 104010
016416 104400

```

15:

25:

```

MOV #34,TSTNO
MOV #TST35,NEXT
MSTCLR
MOV #12,R2
MOV #BIT11,(R1)
JSR PC,CHAR
377
DATACLK, 53
TIMER, 2
ROMCLK
021244
MOV 4(R1),R4
BIC #357,R4
MOV #BIT4,R5
CMPB R5,R4
BEQ 15
HLT 40

```

```

ROMCLK
021204
MOV 4(R1),R4
MOV #377,R5
CMPB R5,R4
BEQ 25
HLT 10
SCOPE

```

```

:R1 CONTAINS BASE DMC11 ADDRESS
:MASTER CLEAR DMC11
:SAVE REG ADDRESS IN R2 FOR TYPEOUT
:SET LINE UNIT LOOP
:LOAD SILO WITH 3 SYNC
:AND THE CHARACTER 377
:SINGLE CLOCK THE DATA
:WAIT FOR INRDY
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:PORT4+LU 12
:PUT "FOUND" IN R4
:CLEAR UNWANTED BITS
:PUT "EXPECTED" IN R5
:IS INRDY SET?
:ERROR, INRDY IS NOT SET
:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
:PORT4+IN DATA
:PUT "FOUND" IN R4
:PUT "EXPECTED" IN R5
:WAS A 377 RECEIVED?
:ERROR, RECEIVED DATA IS WRONG
:SCOPE THIS TEST

```


F06

3353	016600	012703	000073		MOV	#73,R3	:R3 IS CHARACTER COUNT
3354	016604	005011			CLR	(R1)	:CLEAR LU LOOP IN MAINT REG
3355	016606	012761	000040	000004	MOV	#BITS,4(R1)	:LOAD PORT4
3356	016614	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3357	016616	122112			122112		:SET LU LOOP IN LU REG 12
3358	016620	004737	026506		JSR	PC,SIOLD	:LOAD SILO WITH COUNT PATTERN
3359	016624	104415	000043		DATACLK,	43	:SYNC RECEIVER AND GET IT ACTIVE
3360	016630	104415	000730		DATACLK,	730	:CLOCK IN 73 CHARACTERS
3361	016634	004737	026752	18:	JSR	PC,INRDY	:WAIT FOR INRDY
3362	016640	104414		48:	ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3363	016642	021204			021204		:PORT4+IN DATA
3364	016644	016104	000004		MOV	4(R1),R4	:PUT "FOUND" IN R4
3365	016650	010205			MOV	R2,R5	:PUT "EXPECTED" IN R5
3366	016652	120504			CMPB	R5,R4	:IS DATA CORRECT?
3367	016654	001401			BEQ	28	:BR IF YES
3368	016656	104010			HLT	10	:DATA ERROR
3369	016660	005202		28:	INC	R2	:NEXT CHARACTER
3370	016662	022702	000400		CMP	#400,R2	:ALL DONE?
3371	016666	001407			BEQ	38	:BR IF YES
3372	016670	005303			DEC	R3	:DECREMENT CHARACTER COUNT
3373	016672	001360			BNE	48	:BR IF SILO NOT EMPTY
3374	016674	004737	026506		JSR	PC,SIOLD	:LOAD SILO WITH MORE OF COUNT PATTERN
3375	016700	012703	000073		MOV	#73,R3	:RELOAD CHARACTER COUNT
3376	016704	000751			BR	18	:CONTINUE
3377	016706	104400		38:	SCOPE		:SCOPE THIS TEST

```

:***** TEST 37 *****
:*TRANSMITTER MARK TEST
:*SINGLE CLOCK 3 SYNC AND A 301 AND 20 EXTRA
:*CLOCK TICKS, VERIFY THAT A 301, A 377 AND A 377
:*WERE RECEIVED INDICATING THAT THE TRANSMITTER WENT
:*TO A MARK STATE FOR 16 BITS WHEN OUT SILO WAS EMPTY
:*****

```

```

: TEST 37
:-----
TST37: MOV #37,TSTNO
      MOV #TST40,NEXT

```

3380	016710	012737	000037	001226		MSTCLR		:R1 CONTAINS BASE DMC11 ADDRESS
3381	016716	012737	017050	001216		MOV	#BIT11,(R1)	:MASTER CLEAR DMC11
3382	016724	104412			MOV	#BIT11,(R1)	:SET LINE UNIT LOOP	
3383	016726	012711	004000		JSR	PC,CHAR	:LOAD SILO WITH 3 SYNC	
3384	016732	004737	026330		301		:AND A 301	
3385	016736	000301			DATACLK,	73	:CLOCK THE 301 IN AND 20 EXTRA TICKS	
3386	016740	104415	000073		JSR	PC,INRDY	:WAIT FOR INRDY	
3387	016744	004737	026752		ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304	
3388	016750	104414			021204		:PORT4+IN DATA	
3389	016752	021204			MOV	4(R1),R4	:PUT "FOUND" IN R4	
3390	016754	016104	000004		MOV	#301,R5	:PUT "EXPECTED" IN R5	
3391	016760	012705	000301		CMPB	R5,R4	:WAS A 301 RECEIVED?	
3392	016764	120504			BEQ	18		
3393	016766	001401			HLT	10	:ERROR FIRST CHARACTER INCORRECT	
3394	016770	104010			JSR	PC,INRDY	:WAIT FOR INRDY	
3395	016772	004737	026752	18:	ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304	
3396	016776	104414			021204		:PORT4+IN DATA	
3397	017000	021204						

G06

017002	016104	000004		MOV	4(R1),R4	:PUT "FOUND" IN R4
017006	012705	000377		MOV	#377,R5	:PUT "EXPECTED" IN R5
017012	120504			CMPB	R5,R4	:WAS A 377 RECEIVED?
017014	001401			BEQ	2\$	
017016	104010			HLT	10	:ERROR, 377 WAS NOT RECEIVED
017020	004737	026752	2\$:	JSR	PC,INRDY	:WAIT FOR INRDY
017024	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017026	021204			021204		:PORT4+IN DATA
017030	016104	000004		MOV	4(R1),R4	:PUT "FOUND" IN R4
017034	012705	000377		MOV	#377,R5	:PUT "EXPECTED" IN R5
017040	120504			CMPB	R5,R4	:WAS A 377 RECEIVED?
017042	001401			BEQ	3\$	
017044	104010			HLT	10	:ERROR, 177 WAS NOT RECEIVED
017046	104400		3\$:	SCOPE		:SCOPE THIS TEST

***** TEST 40 *****
 :CABLE TURNAROUND TEST
 :CLEAR LINE UNIT LOOP, SET DTR
 :VERIFY THAT RING AND MODEM READY ARE SET
 :CLEAR DTR, VERIFY THAT RING AND MRDY ARE CLEARED
 :*****

				:	TEST 40	
017050	012737	000040	001226	TST40:	MOV	#40,TSTNO
017056	012737	017246	001216		MOV	#TST41,NEXT
017064	104412				MSTCLR	:R1 CONTAINS BASE DMC11 ADDRESS
017066	032737	020000	001366		BIT	:MASTER CLEAR DMC11
017074	001004				.+12	:IS LINE UNIT M8202?
017076	032737	040000	001366		BIT	:BR IF YES (DO TEST EVEN IF NO LOOP-BACK CONN)
017104	001457				BEQ	:IS TURNAROUND CONNECTOR ON?
017106	005011				CLR	:SKIP TEST IF NO
017110	012761	000100	000004		MOV	:CLEAR LINE UNIT LOOP
017116	104414				ROMCLK	:LOAD PORT4
017120	122113				122113	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017122	104416	000002			TIMER,	:SET DTR
017126	104414				ROMCLK	:WAIT
017130	021264				021264	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017132	016104	000004			MOV	:PORT4+LU13
017136	042704	000023			BIC	:PUT "FOUND" IN R4
017142	012705	000310			MOV	:CLEAR UNWANTED BITS
017146	032737	020000	001366		BIT	:PUT "EXPECTED" IN R5
017154	001402				BEQ	:IS LINE UNIT M8202?
017156	042705	000200			.+6	:BR IF NO
017162	120504				BIC	:NO RING ON M8202
017164	001401				CMPB	:ARE RING AND MRDY SET?
017166	104011				BEQ	
017170	005061	000004		1\$:	HLT	:ERROR, RING OR MRDY NOT SET
017174	104414				CLR	:CLEAR PORT4
017176	122113				ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017200	104416	000002			122113	:CLEAR DTR
017204	104414				TIMER,	
017206	021264				ROMCLK	:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
017210	016104	000004			021264	:PORT4+LU13
					MOV	:PUT "FOUND" IN R4

H06

017214	042704	000023	
017220	005005		
017222	032737	020000	001366
017223	001402		
017226	052705	000010	
017229	120504		
017234	001401		
017234	104011		
017244	104400		

23:

```

BIC #23,R4 ;CLEAR UNWANTED BITS
CLR R5 ;PUT "EXPECTED" IN R5
BIT #BIT13,STAT1 ;IS LINE UNIT M8202?
BEQ .+6 ;BR IF NO
BIS #BIT3,R5 ;MRDY SET ON M8202
CMPB R5,R4 ;ARE RING AND MRDY CLEAR?
BEQ 23 ;
HLT 11 ;ERROR, RING OR MRDY NOT CLEAR
SCOPE ;SCOPE THIS TEST

```

```

***** TEST 41 *****
:CABLE TURNAROUND TEST
:CLEAR LINE UNIT LOOP, LOAD OUT DATA SILO
:VERIFY THAT ALL MODEM SIGNALS ARE SET
*****

```

TEST 41

017246	012737	000041	001226
017254	012737	017426	001216
017262	104412		
017264	032737	020000	001366
017272	001004		
017274	032737	040000	001366
017302	001450		
017304	012711	004000	
017310	012761	000100	000004
017316	104414		
017320	122113		
017322	104416	000002	
017326	012761	000001	000004
017334	104414		
017336	122111		
017340	004537	027412	
017344	027674		
017346	000100		
017350	012700	000050	
017354	005011		
017356	104414		
017360	021264		
017362	016104	000004	
017366	042704	000023	
017372	012705	000354	
017376	032737	020000	001366
017404	001402		
017406	042705	000200	
017412	120504		
017414	001402		
017416	005300		
017420	001366		
017422	104011		
017424	104400		

24:

13:

```

TST41: MOV #41,TSTNO
MOV #TST42,NEXT ;R1 CONTAINS BASE DMC11 ADDRESS
MSTCLR ;MASTER CLEAR DMC11
BIT #BIT13,STAT1 ;IS LINE UNIT M8202?
BNE .+12 ;BR IF YES (DO TEST EVEN IF NO LOOP-BACK CONN)
BIT #BIT14,STAT1 ;IS TURNAROUND CONNECTOR ON?
BEQ 13 ;SKIP TEST IF NO
MOV #BIT11,(R1) ;SET LINE UNIT LOOP
MOV #100,4(R1) ;LOAD PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122113 ;CLEAR ALL MODEM SIGNALS, EXCEPT DTR
TIMER, 2 ;WAIT
MOV #1,4(R1) ;LOAD PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122111 ;SET SOM
JSR R5,MESLD ;FILL OUT DATA SILO
MESDAT ;WITH 64 CHARACTERS
64.
MOV #50,R0 ;PREPARE FOR DELAY
CLR (R1) ;CLEAR LINE UNIT LOOP
24: ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021264 ;PORT4+LU13
MOV 4(R1),R4 ;PUT "FOUND" IN R4
BIC #23,R4 ;CLEAR UNWANTED BITS
MOV #354,R5 ;PUT "EXPECTED" IN R5
BIT #BIT13,STAT1 ;IS LINE UNIT M8202?
BEQ .+6 ;BR IF NO
BIC #BIT7,R5 ;NO RING ON M8202
CMPB R5,R4 ;COMPARE EXPECTED AND FOUND
BEQ 13 ;BR IF OK
DEC R0 ;DEC DELAY COUNT
BNE 23 ;BR IF NOT ZERO
HLT 11 ;ERROR, ALL SIGNALS ARE NOT SET
SCOPE ;SCOPE THIS TEST

```

017426
017434
017442
017450
017452
017456
017462
017464
017472
017500
017504
017510
017512
017516
017522
017524
017530
017532
017534
017536
017540
017544
017546
017550
017552
017556
017560
017562
017566
017574
017600
017604
017610
017616
017620
017626
017634
017640
017644
017646
017652
017656
017660

012737 000042 001226
012737 017742 001216
012737 017456 001220
104412
012711 004000
004737 027454
005000
012737 120001 027110
012737 000000 017532
005037 017534
004737 027114
000000
104415 000021
104415 000001
005200
004537 027004
000001
000000
000000
103405
004737 027226
103006
104012
000404
004737 027226
103401
104016
006037 017532
013737 027112 017534
000010
000010
017612 001220
004737 027454
005000
012737 120001 027110
012737 000000 017666
005037 017670
004737 027114
000000
104415 000032
104415 000001
005200
004537 027004

TST42:
64\$:
65\$:
66\$:
67\$:
68\$:
69\$:
71\$:
72\$:

***** TEST 42 *****
*TEST OF CRC OPERATION
*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
*0, VERIFY THE LSB OF THE BCC ON EACH SHIFT
*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC

TEST 42

MOV #42,TSTNO
MOV #TST43,NEXT
MOV #64\$,LOCK
MSTCLR
MOV #BIT11,(R1)
JSR PC,CLR10
CLR RO
MOV #CRC16,XPOLY
MOV #0,66\$
CLR 67\$
JSR PC,BCCLD
0
DATACLK, 21
DATACLK, 1
INC RO
JSR R5,SIMBCC
1
0
0
BCS 68\$
JSR PC,GETQ0
BCC 69\$
HLT 12
BR 69\$
JSR PC,GETQ0
BCS 69\$
HLT 16
ROR 56\$
MOV CALBCC,67\$
CMP #10,RO
BNE 65\$
SCOPE1
MOV #71\$,LOCK
JSR PC,CLR10
CLR RO
MOV #CRC16,XPOLY
MOV #0,73\$
CLR 74\$
JSR PC,BCCLD
0
DATACLK, 32
DATACLK, 1
INC RO
JSR R5,SIMBCC

:R1 CONTAINS BASE DMC11 ADDRESS
:MASTER CLEAR DMC11
:SET L1 LOOP
:CLEAR BCC REGISTERS
:START SHIFT COUNTER AT ZERO
:LOAD POLYNOMIAL FOR SOFTWARE BCC
:LOAD CHAR FOR SOFTWARE BCC
:CLEAR OLD SOFTWARE BCC
:LOAD OUT SILO WITH 2 SYNC
:AND THE CHARACTER 0
:GET TRANSMITTER ACTIVE
:SHIFT BCC ONCE
:BUMP SHIFT COUNT
:CALCULATE SOFTWARE BCC LSB
:ONE SHIFT
:DATA CHARACTER
:OLD BCC
:BR IF SOFT BCC LSB IS SET
:GET HARDWARE TRANSMITTER BCC LSB
:BR IF HARD BCC LSB IS CLEAR
:ERROR, BCC LSB IS SET
:CONTINUE
:GET HARDWARE TRANSMITTER BCC LSB
:BR IF HARD BCC LSB IS SET
:ERROR, HARD BCC LSB IS CLEAR
:SHIFT SOFT DATA
:LOAD OLD SOFT BCC
:DONE YET?
:BR IF NOT DONE
:SCOPE SUBTEST (SW09=1)
:NEW SCOPE1
:CLEAR BCC REGISTERS
:START SHIFT COUNTER AT ZERO
:LOAD POLYNOMIAL FOR SOFTWARE BCC
:LOAD CHAR FOR SOFTWARE BCC
:CLEAR OLD SOFTWARE BCC
:LOAD OUT SILO WITH 2 SYNC
:AND THE CHARACTER 0
:GET RECEIVER ACTIVE
:SHIFT BCC ONCE
:BUMP SHIFT COUNT
:CALCULATE SOFTWARE BCC LSB

3577	017664	000001				1		: ONE SHIFT
3578	017666	000000			73\$:	0		: DATA CHARACTER
3579	017670	000000			74\$:	0		: OLD BCC
3580	017672	103405				BCS	75\$: BR IF SOFT BCC LSB IS SET
3581	017674	004737	027240			JSR	PC,GETQI	: GET HARDWARE RECEIVER BCC LSB
3582	017700	103006				BCC	76\$: BR IF HARD BCC LSB IS CLEAR
3583	017702	104013				HLT	13	: ERROR, BCC LSB IS SET
3584	017704	000404				BR	76\$: CONTINUE
3585	017706	004737	027240		75\$:	JSR	PC,GETQI	: GET HARDWARE RECEIVER BCC LSB
3586	017712	103401				BCS	76\$: BR IF HARD BCC LSB IS SET
3587	017714	104017				HLT	17	: ERROR, BCC LSB IS CLEAR
3588	017716				76\$:			
3589	017716	006037	017666			ROR	73\$: SHIFT SOFT DATA
3590	017722	013737	027112	017670		MOV	CALBCC,74\$: LOAD OLD SOFT BCC
3591	017730	022700	000010			CMP	#10,R0	: DONE YET?
3592	017734	001346				BNE	72\$: BR IF NOT DONE
3593	017736	104401				SCOPE1		: SCOPE SUBTEST (SW09=1)
3594	017740	104400			77\$:	SCOPE		: SCOPE THIS TEST

:***** TEST 43 *****
 :*TEST OF CRC OPERATION
 :*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
 :*377: VERIFY THE LSB OF THE BCC ON EACH SHIFT
 :*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
 :*****

: TEST 43

3606	017742	012737	000043	001226	TST43:	MOV	#43,TSTNO	
3607	017750	012737	020256	001216		MOV	#TST44,NEXT	
3608	017756	012737	017772	001220		MOV	#64\$,LOCK	
3609								
3610	017764	104412				MSTCLR		: R1 CONTAINS BASE DMC11 ADDRESS
3611	017766	012711	004000			MOV	#BIT11,(R1)	: MASTER CLEAR DMC11
3612	017772	004737	027454		64\$:	JSR	PC,CLRIC	: SET LU LOOP
3613	017776	005000				CLR	R0	: CLEAR BCC REGISTERS
3614	020000	012737	120001	027110		MOV	#CRC16,XPOLY	: START SHIFT COUNTER AT ZERO
3615	020006	012737	000377	020046		MOV	#377,65\$;	: LOAD POLYNOMIAL FOR SOFTWARE BCC
3616	020014	005037	020050			CLR	67\$: LOAD CHAR FOR SOFTWARE BCC
3617	020020	004737	027114			JSR	PC,BCCLD	: CLEAR OLD SOFTWARE BCC
3618	020024	000377				377		: LOAD OUT SILO WITH 2 SYNC
3619	020026	104415	000021			DATACLK,	21	: AND THE CHARACTER 377
3620	020032	104415	000001		65\$:	DATACLK,	1	: GET TRANSMITTER ACTIVE
3621	020036	005200				INC	R0	: SHIFT BCC ONCE
3622	020040	004537	027004			JSR	R5,SIMBCC	: BUMP SHIFT COUNT
3623	020044	000001				1		: CALCULATE SOFTWARE BCC LSB
3624	020046	000000			66\$:	0		: ONE SHIFT
3625	020050	000000			67\$:	0		: DATA CHARACTER
3626	020052	103405				BCS	68\$: OLD BCC
3627	020054	004737	027226			JSR	PC,GETQ0	: BR IF SOFT BCC LSB IS SET
3628	020060	103006				BCC	69\$: GET HARDWARE TRANSMITTER BCC LSB
3629	020062	104012				HLT	12	: BR IF HARD BCC LSB IS CLEAR
3630	020064	000404				BR	69\$: ERROR, BCC LSB IS SET
3631	020066	004737	027226		68\$:	JSR	PC,GETQ0	: CONTINUE
3632	020072	103401				BCS	69\$: GET HARDWARE TRANSMITTER BCC LSB
								: BR IF HARD BCC LSB IS SET

K06

3633	020074	104016			HLT	16	;ERROR, HARD BCC LSB IS CLEAR
3634	020076			69\$:	ROR	66\$;SHIFT SOFT DATA
3635	020076	006037	020046		MOV	CALBCC,67\$;LOAD OLD SOFT BCC
3636	020102	013737	027112	020050	CMP	#10,RO	;DONE YET?
3637	020110	022700	000010		BNE	65\$;BR IF NOT DONE
3638	020114	001346			SCOPE1		;SCOPE SUBTEST (SW09=1)
3639	020116	104401			MOV	#71\$,LOCK	;NEW SCOPE1
3640	020120	012737	020126	001220	JSR	PC,CLRIO	;CLEAR BCC REGISTERS
3641	020126	004737	027454	71\$:	CLR	RO	;START SHIFT COUNTER AT ZERO
3642	020132	005000			MOV	#CRC16,XPOLY	;LOAD POLYNOMIAL FOR SOFTWARE BCC
3643	020134	012737	120001	027110	MOV	#377,73\$;	;LOAD CHAR FOR SOFTWARE BCC
3644	020142	012737	000377	020202	CLR	74\$;CLEAR OLD SOFTWARE BCC
3645	020150	005037	020204		JSR	PC,BCCLD	;LOAD OUT SILO WITH 2 SYNC
3646	020154	004737	027114		377		;AND THE CHARACTER 377
3647	020160	000377			DATACLK,	32	;GET RECEIVER ACTIVE
3648	020162	104415	000032		DATACLK,	1	;SHIFT BCC ONCE
3649	020166	104415	000001	72\$:	INC	RO	;BUMP SHIFT COUNT
3650	020172	005200			JSR	R5,SIMBCC	;CALCULATE SOFTWARE BCC LSB
3651	020174	004537	027004		1		;ONE SHIFT
3652	020200	000001			0		;DATA CHARACTER
3653	020202	000000		73\$:	0		;OLD BCC
3654	020204	000000		74\$:	BCS	75\$;BR IF SOFT BCC LSB IS SET
3655	020206	103405			JSR	PC,GETQI	;GET HARDWARE RECEIVER BCC LSB
3656	020210	004737	027240		BCC	76\$;BR IF HARD BCC LSB IS CLEAR
3657	020214	103006			HLT	13	;ERROR, BCC LSB IS SET
3658	020216	104013			BR	76\$;CONTINUE
3659	020220	000404			JSR	PC,GETQI	;GET HARDWARE RECEIVER BCC LSB
3660	020222	004737	027240	75\$:	BCS	76\$;BR IF HARD BCC LSB IS SET
3661	020226	103401			HLT	17	;ERROR, BCC LSB IS CLEAR
3662	020230	104017					
3663	020232			76\$:	ROR	73\$;SHIFT SOFT DATA
3664	020232	006037	020202		MOV	CALBCC,74\$;LOAD OLD SOFT BCC
3665	020236	013737	027112	020204	CMP	#10,RO	;DONE YET?
3666	020244	022700	000010		BNE	72\$;BR IF NOT DONE
3667	020250	001346			SCOPE1		;SCOPE SUBTEST (SW09=1)
3668	020252	104401			SCOPE		;SCOPE THIS TEST
3669	020254	104400		77\$:			

3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688

```
***** TEST 44 *****  
*TEST OF CRC OPERATION  
*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER  
*125. VERIFY THE LSB OF THE BCC ON EACH SHIFT  
*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC  
*****
```

```
TEST 44  
-----  
TST44: MOV #44,TSTNO  
MOV #TST45,NEXT  
MOV #64$,LOCK  
  
MSTCLR ;R1 CONTAINS BASE DMC11 ADDRESS  
MOV #BIT11,(R1) ;MASTER CLEAR DMC11  
JSR PC,CLRIO ;SET LU LOOP  
CLR RO ;CLEAR BCC REGISTERS  
 ;START SHIFT COUNTER AT ZERO
```

L06

3689	020314	012737	120001	027110		MOV	#CRC16,XPOLY	;LOAD POLYNOMIAL FOR SOFTWARE BCC
3690	020322	012737	000125	020362		MOV	#125,66\$;	;LOAD CHAR FOR SOFTWARE BCC
3691	020330	005037	020364			CLR	67\$;CLEAR OLD SOFTWARE BCC
3692	020334	004737	027114			JSR	PC,BCCLD	;LOAD OUT SILO WITH 2 SYNC
3693	020340	000125				125		;AND THE CHARACTER 125
3694	020342	104415	000021			DATACLK,	21	;GET TRANSMITTER ACTIVE
3695	020346	104415	000001		55\$:	DATACLK,	1	;SHIFT BCC ONCE
3696	020352	005200				INC	RO	;BUMP SHIFT COUNT
3697	020354	004537	027004			JSR	R5,SIMBCC	;CALCULATE SOFTWARE BCC LSB
3698	020360	000001				1		;ONE SHIFT
3699	020362	000000			66\$:	0		;DATA CHARACTER
3700	020364	000000			67\$:	0		;OLD BCC
3701	020366	103405				BCS	68\$;BR IF SOFT BCC LSB IS SET
3702	020370	004737	027226			JSR	PC,GETQ0	;GET HARDWARE TRANSMITTER BCC LSB
3703	020374	103006				BCC	69\$;BR IF HARD BCC LSB IS CLEAR
3704	020376	104012				HLT	12	;ERROR, BCC LSB IS SET
3705	020400	000404				BR	69\$;CONTINUE
3706	020402	004737	027226		68\$:	JSR	PC,GETQ0	;GET HARDWARE TRANSMITTER BCC LSB
3707	020406	103401				BCS	69\$;BR IF HARD BCC LSB IS SET
3708	020410	104016				HLT	16	;ERROR, HARD BCC LSB IS CLEAR
3709	020412				69\$:			
3710	020412	006037	020362			ROR	66\$;SHIFT SOFT DATA
3711	020416	013737	027112	020364		MOV	CALBCC,67\$;LOAD OLD SOFT BCC
3712	020424	022700	000010			CMP	#10,RO	;DONE YET?
3713	020430	001346				BNE	65\$;BR IF NOT DONE
3714	020432	104401				SCOPE1		;SCOPE SUBTEST (SW09=1)
3715	020434	012737	020442	001220		MOV	#71\$,LOCK	;NEW SCOPE1
3716	020442	004737	027454		71\$:	JSR	PC,CLR10	;CLEAR BCC REGISTERS
3717	020446	005000				CLR	RO	;START SHIFT COUNTER AT ZERO
3718	020450	012737	120001	027110		MOV	#CRC16,XPOLY	;LOAD POLYNOMIAL FOR SOFTWARE BCC
3719	020456	012737	000125	020516		MOV	#125,73\$;	;LOAD CHAR FOR SOFTWARE BCC
3720	020464	005037	020520			CLR	74\$;CLEAR OLD SOFTWARE BCC
3721	020470	004737	027114			JSR	PC,BCCLD	;LOAD OUT SILO WITH 2 SYNC
3722	020474	000125				125		;AND THE CHARACTER 125
3723	020476	104415	000032			DATACLK,	32	;GET RECEIVER ACTIVE
3724	020502	104415	000001		72\$:	DATACLK,	1	;SHIFT BCC ONCE
3725	020506	005200				INC	RO	;BUMP SHIFT COUNT
3726	020510	004537	027004			JSR	R5,SIMBCC	;CALCULATE SOFTWARE BCC LSB
3727	020514	000001				1		;ONE SHIFT
3728	020516	000000			73\$:	0		;DATA CHARACTER
3729	020520	000000			74\$:	0		;OLD BCC
3730	020522	103405				BCS	75\$;BR IF SOFT BCC LSB IS SET
3731	020524	004737	027240			JSR	PC,GETQI	;GET HARDWARE RECEIVER BCC LSB
3732	020530	103006				BCC	76\$;BR IF HARD BCC LSB IS CLEAR
3733	020532	104013				HLT	13	;ERROR, BCC LSB IS SET
3734	020534	000404				BR	76\$;CONTINUE
3735	020536	004737	027240		75\$:	JSR	PC,GETQI	;GET HARDWARE RECEIVER BCC LSB
3736	020542	103401				BCS	76\$;BR IF HARD BCC LSB IS SET
3737	020544	104017				HLT	17	;ERROR, BCC LSB IS CLEAR
3738	020546				76\$:			
3739	020546	006037	020516			ROR	73\$;SHIFT SOFT DATA
3740	020552	013737	027112	020520		MOV	CALBCC,74\$;LOAD OLD SOFT BCC
3741	020560	022700	000010			CMP	#10,RO	;DONE YET?
3742	020564	001346				BNE	72\$;BR IF NOT DONE
3743	020566	104401				SCOPE1		;SCOPE SUBTEST (SW09=1)
3744	020570	104400			77\$:	SCOPE		;SCOPE THIS TEST

```

3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756 020572 012737 000045 001226
3757 020600 012737 021106 001216
3758 020606 012737 020622 001220
3759
3760 020614 104412
3761 020616 012711 004000
3762 020622 004737 027454
3763 020626 005000
3764 020630 012737 120001 027110
3765 020636 012737 000252 020676
3766 020644 005037 020700
3767 020650 004737 027114
3768 020654 000252
3769 020656 104415 000021
3770 020662 104415 000001
3771 020666 005200
3772 020670 004537 027004
3773 020674 000001
3774 020676 000000
3775 020700 000000
3776 020702 103405
3777 020704 004737 027226
3778 020710 103006
3779 020712 104012
3780 020714 000404
3781 020716 004737 027226
3782 020722 103401
3783 020724 104016
3784 020726
3785 020726 006037 020676
3786 020732 013737 027112 020700
3787 020740 022700 000010
3788 020744 001346
3789 020746 104401
3790 020750 012737 020756 001220
3791 020756 004737 027454
3792 020762 005000
3793 020764 012737 120001 027110
3794 020772 012737 000252 021032
3795 021000 005037 021034
3796 021004 004737 027114
3797 021010 000252
3798 021012 104415 000032
3799 021016 104415 000001
3800 021022 005200

```

```

:***** TEST 45 *****
:*TEST OF CRC OPERATION
:*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK THE CHARACTER
:*252, VERIFY THE LSB OF THE BCC ON EACH SHIFT
:*TEST TRANSMITTER FIRST THEN THE RECEIVER BCC
:*****

```

```

: TEST 45
:-----
TST45: MOV #45,TSTNO
MOV #TST46,NEXT
MOV #64$,LOCK
MSTCLR
64$: MOV #BIT11,(R1)
JSR PC,CLRIO
CLR RO
MOV #CRC16,XPOLY
MOV #252,66$;
CLR 67$
JSR PC,BCCLD
252
DATACLK, 21
65$: DATACLK, 1
INC RO
JSR R5,SIMBCC
1
66$: 0
67$: 0
BCS 68$
JSR PC,GETQ0
BCC 69$
HLT 12
BR 69$
68$: JSR PC,GETQ0
BCS 69$
HLT 16
69$: ROR 66$
MOV CALBCC,67$
CMP #10,RO
BNE 65$
SCOPE1
71$: MOV #71$,LOCK
JSR PC,CLRIO
CLR RO
MOV #CRC16,XPOLY
MOV #252,73$;
CLR 74$
JSR PC,BCCLD
252
72$: DATACLK, 32
DATACLK, 1
INC RO

```

```

;R1 CONTAINS BASE DMC11 ADDRESS
;MASTER CLEAR DMC11
;SET LU LOOP
;CLEAR BCC REGISTERS
;START SHIFT COUNTER AT ZERO
;LOAD POLYNOMIAL FOR SOFTWARE BCC
;LOAD CHAR FOR SOFTWARE BCC
;CLEAR OLD SOFTWARE BCC
;LOAD OUT SILO WITH 2 SYNCs
;AND THE CHARACTER 252
;GET TRANSMITTER ACTIVE
;SHIFT BCC ONCE
;BUMP SHIFT COUNT
;CALCULATE SOFTWARE BCC LSB
;ONE SHIFT
;DATA CHARACTER
;OLD BCC
;BR IF SOFT BCC LSB IS SET
;GET HARDWARE TRANSMITTER BCC LSB
;BR IF HARD BCC LSB IS CLEAR
;ERROR, BCC LSB IS SET
;CONTINUE
;GET HARDWARE TRANSMITTER BCC LSB
;BR IF HARD BCC LSB IS SET
;ERROR, HARD BCC LSB IS CLEAR
;SHIFT SOFT DATA
;LOAD OLD SOFT BCC
;DONE YET?
;BR IF NOT DONE
;SCOPE SUBTEST (SW09=1)
;NEW SCOPE1
;CLEAR BCC REGISTERS
;START SHIFT COUNTER AT ZERO
;LOAD POLYNOMIAL FOR SOFTWARE BCC
;LOAD CHAR FOR SOFTWARE BCC
;CLEAR OLD SOFTWARE BCC
;LOAD OUT SILO WITH 2 SYNCs
;AND THE CHARACTER 252
;GET RECEIVER ACTIVE
;SHIFT BCC ONCE
;BUMP SHIFT COUNT

```

3801	021024	004537	027004		JSR	R5,SIMBCC	;CALCULATE SOFTWARE BCC LSB
3802	021030	000001			1		;ONE SHIFT
3803	021032	000000		73\$:	0		;DATA CHARACTER
3804	021034	000000		74\$:	0		;OLD BCC
3805	021036	103405			BCS	75\$;BR IF SOFT BCC LSB IS SET
3806	021040	004737	027240		JSR	PC,GETQI	;GET HARDWARE RECEIVER BCC LSB
3807	021044	103006			BCC	76\$;BR IF HARD BCC LSB IS CLEAR
3808	021046	104013			HLT	13	;ERROR, BCC LSB IS SET
3809	021050	000404			BR	76\$;CONTINUE
3810	021052	004737	027240		JSR	PC,GETQI	;GET HARDWARE RECEIVER BCC LSB
3811	021056	103401		75\$:	BCS	76\$;BR IF HARD BCC LSB IS SET
3812	021060	104017			HLT	17	;ERROR, BCC LSB IS CLEAR
3813	021062			76\$:			
3814	021062	006037	021032		ROR	73\$;SHIFT SOFT DATA
3815	021066	013737	027112	021034	MOV	CALBCC,74\$;LOAD OLD SOFT BCC
3816	021074	022700	000010		CMP	#10,R0	;DONE YET?
3817	021100	001346			BNE	72\$;BR IF NOT DONE
3818	021102	104401			SCOPE		;SCOPE SUBTEST (SW09=1)
3819	021104	104400		77\$:	SCOPE		;SCOPE THIS TEST
3820							
3821							
3822							
3823							
3824							
3825							
3826							
3827							
3828							
3829							
3830	021106	012737	000046	001226	TST46:	MOV	#46,TSTNO
3831	021114	012737	021344	001216		MOV	#TST47,NEXT
3832							
3833	021122	104412			MSTCLR		;R1 CONTAINS BASE DMC11 ADDRESS
3834	021124	012711	004000		MOV	#BIT11,(R1)	;MASTER CLEAR DMC11
3835	021130	005003			CLR	R3	;SET LINE UNIT LOOP
3836	021132	005004			CLR	R4	;ZERO BIT COUNT
3837	021134	005005			CLR	R5	;R4 CONTAINS CHAR TO BE LOADED IN SILO
3838	021136	005037	021240		CLR	4\$;R5 CONTAINS CHAR CURRENTLY BEING SHIFTED OUT
3839	021142	012737	120001	027110	MOV	#CRC16,XPOLY	;CLEAR SOFT BCC
3840	021150	004737	027256		JSR	PC,SYNLD	;LOAD POLYNOMINAL
3841	021154	010461	000004		MOV	R4,4(R1)	;LOAD SILO WITH 2 SYNCS, SOM SET
3842	021160	104414			ROMCLK		;PORT4+CHAR
3843	021162	122110			122110		;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3844	021164	005204			INC	R4	;LOAD OUT DATA
3845	021166	010461	000004		MOV	R4,4(R1)	;INCREMENT TO NEXT CHARACTER
3846	021172	104414			ROMCLK		;PORT4+CHAR
3847	021174	122110			122110		;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3848	021176	005204			INC	R4	;LOAD OUT DATA
3849	021200	010461	000004		MOV	R4,4(R1)	;INCREMENT TO NEXT CHARACTER
3850	021204	104414			ROMCLK		;PORT4+CHAR
3851	021206	122110			122110		;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
3852	021210	004737	026144		JSR	PC,OCOR	;LOAD OUT DATA
3853	021214	104415	000021		DATACLK,	21	;WAIT FOR OCOR
3854	021220	010537	021236		MOV	R5,3\$;CLOCK DATA
3855	021224	104415	000001		DATACLK,	1	;LOAD CHAR FOR SOFT CRC
3856	021230	004537	027004		JSR	R5,SIMBCC	;SHIFT BCC ONCE
				1\$:			;CALCULATE SOFT BCC
				2\$:			

```

;***** TEST 46 *****
;*TRANSMITTER CRC TEST
;*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK A BINARY
;*COUNT PATTERN, VERIFY THE LSB OF THE TRANSMITTER BCC ON EACH SHIFT
;*****

```

```

; TEST 46
-----

```

TST46:

```

MOV #46,TSTNO
MOV #TST47,NEXT

MSTCLR
MOV #BIT11,(R1)
CLR R3
CLR R4
CLR R5
CLR 4$
MOV #CRC16,XPOLY
JSR PC,SYNLD
MOV R4,4(R1)
ROMCLK
122110
INC R4
MOV R4,4(R1)
ROMCLK
122110
INC R4
MOV R4,4(R1)
ROMCLK
122110
JSR PC,OCOR
DATACLK,21
MOV R5,3$
DATACLK,1
JSR R5,SIMBCC

;R1 CONTAINS BASE DMC11 ADDRESS
;MASTER CLEAR DMC11
;SET LINE UNIT LOOP
;ZERO BIT COUNT
;R4 CONTAINS CHAR TO BE LOADED IN SILO
;R5 CONTAINS CHAR CURRENTLY BEING SHIFTED OUT
;CLEAR SOFT BCC
;LOAD POLYNOMINAL
;LOAD SILO WITH 2 SYNCS, SOM SET
;PORT4+CHAR
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;LOAD OUT DATA
;INCREMENT TO NEXT CHARACTER
;PORT4+CHAR
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;LOAD OUT DATA
;INCREMENT TO NEXT CHARACTER
;PORT4+CHAR
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;LOAD OUT DATA
;INCREMENT TO NEXT CHARACTER
;PORT4+CHAR
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;LOAD OUT DATA
;WAIT FOR OCOR
;CLOCK DATA
;LOAD CHAR FOR SOFT CRC
;SHIFT BCC ONCE
;CALCULATE SOFT BCC

```

```

00000000 00000000 00000001 00000000 00000000 1
00000000 00000000 00000000 00000000 00000000 38: 0
00000000 00000000 00000000 00000000 00000000 48: 0
00000000 00000000 00000000 00000000 00000000 BCS 58
00000000 00000000 00000000 00000000 00000000 JSR PC,GETQO :SOFT SHIFT COUNT
00000000 00000000 00000000 00000000 00000000 BCC 68 :SOFT CHARACTER
00000000 00000000 00000000 00000000 00000000 HLT 20 :OLD SOFT BCC
00000000 00000000 00000000 00000000 00000000 BR 68 :BR IF SOFT BCC LSB IS SET
00000000 00000000 00000000 00000000 00000000 JSR PC,GETQO :GET HARDWARE TRANSMITTER BCC LSB
00000000 00000000 00000000 00000000 00000000 BCS 68 :BR IF OK (CLEARED)
00000000 00000000 00000000 00000000 00000000 HLT 21 :ERROR, BCC LSB WAS SET
00000000 00000000 00000000 00000000 00000000 BR 68 :CONTINUE WITH TEST
00000000 00000000 00000000 00000000 00000000 JSR PC,GETQO :GET HARDWARE TRANSMITTER BCC LSB
00000000 00000000 00000000 00000000 00000000 BCS 68 :BR IF OK (SET)
00000000 00000000 00000000 00000000 00000000 HLT 21 :ERROR, BCC LSB WAS CLEAR
00000000 00000000 00000000 00000000 00000000 55:
00000000 00000000 00000000 00000000 00000000 ROR 38 :SHIFT SOFT DATA
00000000 00000000 00000000 00000000 00000000 MOV CALBCC,48 :LOAD OLD SOFT BCC
00000000 00000000 00000000 00000000 00000000 INC R3 :INCREMENT BIT COUNTER
00000000 00000000 00000000 00000000 00000000 CMP #10,R3 :DONE A FULL CHARACTER YET?
00000000 00000000 00000000 00000000 00000000 BNE 28 :BR IF NO
00000000 00000000 00000000 00000000 00000000 CLR R3 :RESTART BIT COUNTER
00000000 00000000 00000000 00000000 00000000 INC R4 :INCREMENT DATA FOR SILO
00000000 00000000 00000000 00000000 00000000 CMP #400,R4 :DONE BINARY COUNT YET?
00000000 00000000 00000000 00000000 00000000 BLE 98 :BR IF YES
00000000 00000000 00000000 00000000 00000000 MOV R4,4(R1) :PORT4+DATA
00000000 00000000 00000000 00000000 00000000 ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
00000000 00000000 00000000 00000000 00000000 122110 :LOAD OUT DATA
00000000 00000000 00000000 00000000 00000000 98: INC R5 :INCREMENT DATA
00000000 00000000 00000000 00000000 00000000 CMP #400,R5 :DONE BINARY PATTERN YET?
00000000 00000000 00000000 00000000 00000000 BNE 18 :BR IF NO
00000000 00000000 00000000 00000000 00000000 78: SCOPE :SCOPE THIS TEST

```

```

***** TEST 47 *****
*RECEIVER CRC TEST
*USING THE CRC16 POLYNOMIAL, SINGLE CLOCK A BINARY
*COUNT PATTERN, VERIFY THE LSB OF THE RECEIVER BCC ON EACH SHIFT
*****

```

TEST 47

```

00000000 00000000 00000000 00000000 00000000 TST47: MOV #47,TSTNO
00000000 00000000 00000000 00000000 00000000 MOV #TST50,NEXT
00000000 00000000 00000000 00000000 00000000 MSTCLR :R1 CONTAINS BASE DMC11 ADDRESS
00000000 00000000 00000000 00000000 00000000 MOV #BIT11,(R1) :MASTER CLEAR DMC11
00000000 00000000 00000000 00000000 00000000 CLR R3 :SET LINE UNIT LOOP
00000000 00000000 00000000 00000000 00000000 CLR R4 :ZERO BIT COUNT
00000000 00000000 00000000 00000000 00000000 CLR R5 :R4 CONTAINS CHAR TO BE LOADED IN SILO
00000000 00000000 00000000 00000000 00000000 CLR 48 :R5 CONTAINS CHAR CURRENTLY BEING SHIFTED OUT
00000000 00000000 00000000 00000000 00000000 MOV #CRC16,XPOLY :CLEAR SOFT BCC
00000000 00000000 00000000 00000000 00000000 JSR PC,SYNLD :LOAD POLYNOMIAL
00000000 00000000 00000000 00000000 00000000 MOV R4,4(R1) :LOAD SILO WITH 2 SYNCS, SOM SET
00000000 00000000 00000000 00000000 00000000 ROMCLK :PORT4+CHAR
00000000 00000000 00000000 00000000 00000000 122110 :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
00000000 00000000 00000000 00000000 00000000 INC R4 :LOAD OUT DATA
00000000 00000000 00000000 00000000 00000000 MOV R4,4(R1) :INCREMENT TO NEXT CHARACTER
00000000 00000000 00000000 00000000 00000000 ROMCLK :PORT4+CHAR
00000000 00000000 00000000 00000000 00000000 :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

```



```

021432 122110
021434 005204
021436 010461 000004
021438 104414
021440 122110
021442 004737 026144
021444 104415 000032
021446 010537 021474
021448 104415 000001
021450 004537 027004
021452 000001
021454 000000
021456 000000
021458 000000
021500 103405
021502 004737 027240
021504 103006
021506 104022
021508 000404
021510 004737 027240
021512 103401
021514 104023
021522 104023
021524 006037 021474
021526 013737 027112 021476
021528 005203
021530 022703 000010
021532 001346
021534 005003
021536 005204
021538 022704 000400
021540 003404
021542 010461 000004
021544 104414
021546 122110
021548 005204
021550 022704 000400
021552 001327
021600 104400

```

```

122110 : LOAD OUT DATA
INC R4 : INCREMENT TO NEXT CHARACTER
MOV R4,4(R1) : PORT4+CHAR
ROMCLK : NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122110 : LOAD OUT DATA
JSR PC,OCOR : WAIT FOR OCOR
DATACLK,32 : CLOCK DATA
MOV R5,3$ : LOAD CHAR FOR SOFT CRC
DATACLK,1 : SHIFT BCC ONCE
JSR R5,SIMBCC : CALCULATE SOFT BCC
: SOFT SHIFT COUNT
: SOFT CHARACTER
: OLD SOFT BCC
BCS 5$ : BR IF SOFT BCC LSB IS SET
JSR PC,GETQI : GET HARDWARE RECEIVER BCC LSB
BCC 6$ : BR IF OK (CLEARED)
HLT 23$ : ERROR, BCC LSB WAS SET
BR 6$ : CONTINUE WITH TEST
JSR PC,GETQI : GET HARDWARE RECEIVER BCC LSB
BCS 6$ : BR IF OK (SET)
HLT 23 : ERROR, BCC LSB WAS CLEAR

6$:
ROR 3$ : SHIFT SOFT DATA
MOV CALBCC,4$ : LOAD OLD SOFT BCC
INC R3 : INCREMENT BIT COUNTER
CMP #10,R3 : DONE A FULL CHARACTER YET?
BNE 23$ : BR IF NO
CLR R3 : RESTART BIT COUNTER
INC R4 : INCREMENT DATA FOR SILO
CMP #400,R4 : DONE BINARY COUNT YET?
BLE 9$ : BR IF YES
MOV R4,4(R1) : PORT4+DATA
ROMCLK : NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122110 : LOAD OUT DATA
INC R5 : INCREMENT DATA
CMP #400,R5 : DONE BINARY PATTERN YET?
BNE 1$ : BR IF NO
SCOPE : SCOPE THIS TEST

```

```

:***** TEST 50 *****
: *TRANSMITTER DDCMP CRC TEST
: *THIS TEST TRANSMITS A FOUR CHARACTER MESSAGE WITH CRC
: *BOTH DATA AND THE BCC ARE VERIFIED IN THE BIT
: *WINDOW. THE FOUR CHARACTERS ARE 0,125,252,377
: *THE TRANSMITTER IS CHECKED FOR GOING TO A MARK STATE AFTER THE BCC
:*****

```

: TEST 50

```

021602 012737 000050 001226
021610 012737 022134 001216
021616 104412

```

```

TST50: MOV #50,TSTNO
MOV #TST51,NEXT
MSTCLR

```

```

:R1 CONTAINS BASE DMC11 ADDRESS
:MASTER CLEAR DMC11

```



```

4025 022032 004737 026112      69$: JSR    PC,GETSI      ;LOOK AT BIT WINDOW
4026 022036 103001                BCC    70$           ;BR IF OK (SPACE)
4027 022040 104014                HLT    14           ;ERROR, CRC WRONG (MARK)
4028 022042                70$: INC    R3           ;BUMP BIT COUNTER
4029 022044 005203                CMP    #20,R3      ;FINISHED BCC YET?
4030 022044 022703 000020        BNE    68$         ;BR IF NO
4031 022050 001357                CLR    R3          ;CLEAR BIT COUNTER
4032 022052 005003                ;CHECK TO SEE IF TRANSMITTER IS MARKING
4033 022054 104415 000001      2$: DATACLK, 1      ;CLOCK TRANSMITTER
4034 022060 004737 026112      JSR    PC,GETSI    ;LOOK AT WINDOW
4035 022064 103401                BCS    3$         ;IT SHOULD BE MARKING
4036 022066 104024                HLT    24         ;ERROR, BIT WAS A SPACE
4037 022070 005203                3$: INC    R3           ;BUMP BIT COUNTER
4038 022072 022703 000007      CMP    #7,R3      ;DONE YET
4039 022076 001366                BNE    2$         ;BR IF NO
4040 022100 104415 000010      DATACLK, 10      ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
4041 022104 005003                CLR    R3         ;CLEAR BIT COUNTER
4042 022106 104415 000001      4$: DATACLK, 1    ;SHIFT OUT NEXT BIT
4043 022112 004737 026112      JSR    PC,GETSI    ;LOOK AT BIT WINDOW
4044 022116 103401                BCS    +4         ;BR IF IT IS A MARK
4045 022120 104024                HLT    24         ;ERROR, TRANSMITTER IS NOT MARKING
4046 022122 005203                INC    R3         ;INC BIT COUNT
4047 022124 022703 000020      CMP    #20,R3    ;DONE YET?
4048 022130 001366                BNE    4$         ;BR IF NO
4049 022132 104400                5$: SCOPE          ;SCOPE THIS TEST

```

```

;***** TEST 51 *****
;*RECEIVER DDCMP CRC TEST
;*THIS TEST CLOCKS A FOUR CHARACTER MESSAGE WITH BCC
;*AND VERIFYS CORRECT DATA RECEPTION AND BCC MATCH
;*THE FOUR CHARACTER MESSAGE IS 0,125,252,377
;*****

```

: TEST 51

```

4064 022134 012737 000051 001226  TST51: MOV    #51,TSTNO
4065 022142 012737 022336 001216  MOV    #TST52,NEXT
4066 022150 104412                MSTCLR          ;R1 CONTAINS BASE DMC11 ADDRESS
4067 022152 012711 004000                MOV    #BIT11,(R1) ;MASTER CLEAR DMC11
4068 022156 012702 027674                MOV    #MESDAT,R2 ;SET LINE UNIT LOOP
4069 022162 012700 000004                MOV    #4,R0     ;LOAD POINTER TO DATA
4070 022166 004737 027256                JSR    PC,SYNLD  ;LOAD CHARACTER COUNT
4071 022172 004737 026276                JSR    PC,OUTRDY ;LOAD 2 SYNC IN OUT SILO
4072 022176 004537 027412                JSR    R5,MESLD  ;WAIT FOR OUTRDY
4073 022202 027674                MESDAT         ;LOAD SILO WITH 4 CHAR MESS
4074 022204 000004                4             ;ADDRESS OF MESSAGE
4075 022206 004737 027366                JSR    PC,EOM    ;NUMBER OF CHARACTERS
4076 022212 004737 026144                JSR    PC,OCOR   ;LOAD GARBAGE CHARACTER, WITH EOM SET
4077 022216 104415 000114                DATACLK,114    ;WAIT FOR OCOR
4078 022222 004737 026752                JSR    PC,INRDY ;CLOCK DATA
4079 022226 104414                3$: ROMCLK      ;WAIT FOR INRDY
4080                                ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

```

F07

00001	022230	021204			021204		:GET IN DATA
00002	022232	016104	000004		MOV	4(R1),R4	:PUT "FOUND" IN R4
00003	022236	112205			MOV	(R2)+,R5	:PUT "EXPECTED" IN R5
00004	022240	120504			CMPB	R5,R4	:COMPARE RECEIVED DATA
00005	022242	001401			BEQ	15	:BR IF OK
00006	022244	104010			HLT	10	:DATA ERROR
00007	022246	005300		15:	DEC	RO	:DEC CHARACTER COUNT
00008	022250	001364			BNE	35	:BR IF NOT DONE YET
00009							
00010							:CHECK TO SEE THAT IN BCC MATCH IS SET
00011	022252	004737	026752		JSR	PC,INRDY	:WAIT FOR INRDY
00012	022256	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
00013	022260	021204			021204		:GET FIRST HALF OF CRC
00014	022262	116137	000004	001252	MOV	4(R1),TEMP3	:PUT IN TEMP3
00015	022270	042737	177400	001252	BIC	#177400,TEMP3	:CLEAR HI BYTE
00016	022276	004737	026752		JSR	PC,INRDY	:WAIT FOR INRDY
00017	022302	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
00018	022304	021244			021244		
00019	022306	016104	000004		MOV	4(R1),R4	:PUT "FOUND" IN R4
00020	022312	042704	000376		BIC	#376,R4	:CLEAR UNWANTED BITS
00021	022316	012705	000001		MOV	#1,R5	:PUT "EXPECTED" IN R5
00022	022322	120504			CMPB	R5,R4	:IS IN BCC MATCH SET?
00023	022324	001401			BEQ	255	
00024	022326	104015			HLT	15	:IN BCC MATCH ERROR
00025	022330			255:			
00026	022332	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
00027	022332	021204			021204		:GET LAST HALF
00028	022334	104400		25:	SCOPE		:SCOPE THIS TEST
00029							
00030							
00031							
00032							
00033							
00034							
00035							
00036							
00037							
00038							
00039							
00040							
00041							
00042							
00043							
00044							
00045							
00046							
00047							
00048							
00049							
00050							
00051							
00052							
00053							
00054							
00055							
00056							
00057							
00058							
00059							
00060							
00061							
00062							
00063							
00064							
00065							
00066							
00067							
00068							
00069							
00070							
00071							
00072							
00073							
00074							
00075							
00076							
00077							
00078							
00079							
00080							
00081							
00082							
00083							
00084							
00085							
00086							
00087							
00088							
00089							
00090							
00091							
00092							
00093							
00094							
00095							
00096							
00097							
00098							
00099							
00100							

```

:***** TEST 52 *****
:DDCMP EOM FUNCTION TEST
:THIS TEST LOADS OUT SILO WITH: 2 SYNCS, 4 CHAR MESSAGE, EOM
:*4 CHARACTER MESS, EOM. THE DATA STREAM IS CHECKED TO BE
:*4 CHAR, BCC 4 CHAR, BCC MARKS. THIS TEST VERIFYS THAT
:*THE CHARCTERS LOADED WITH EOM SET ARE LOST
:*ALL DATA AND BCC'S ARE CHECKED IN THE BIT WINDOW
:*THE FOUR CHARACTER MESSAGE IS 0,125,252,377
:*RECEIVED DATA IS VERIFIED, AND IN BCC MATCH IS CHECKED
:*****

```

```

; TEST 52
-----
TST52: MOV #52,TSTNO
MOV #TST53,NEXT

MSTCLR ;R1 CONTAINS BASE DMC11 ADDRESS
;MASTER CLEAR DMC11

;LOAD OUT DATA SILO

MOV #BIT11,(R1) ;SET LINE UNIT LOOP
MOV #MESDAT,R4 ;LOAD POINTER TO DATA
CLR 105 ;CLEAR SOFT BCC
MOV #4,RO ;LOAD CHARACTER COUNT
JSR PC,SYNLD ;LOAD 2 SYNCS IN OUT SILO

```

4137	022400	004737	026276		JSR	PC,OUTRDY	:WAIT FOR OUTRDY
4138	022404	004537	027412		JSR	R5,MESLD	:LOAD SILO WITH 4 CHAR MESS
4139	022410	027674			MESDAT		:ADDRESS OF MESSAGE
4140	022412	000004			4		:NUMBER OF CHARACTERS
4141	022414	004737	027366		JSR	PC,EOM	:LOAD GARBAGE CHARACTER, WITH EOM SET
4142	022420	004537	027412		JSR	R5,MESLD	:LOAD FOUR MORE CHARACTERS
4143	022424	027674			MESDAT		:ADDRESS OF MESSAGE
4144	022426	000004			4		:NUMBER OF CHACTERS
4145	022430	004737	027366		JSR	PC,EOM	:SET EOM
4146	022434	004737	026144		JSR	PC,OCOR	:WAIT FOR OCOR
4147	022440	005003			CLR	R3	:CLEAR BIT COUNTER
4148	022442	104415	000022		DATACLK,	22	:CLOCK DATA
4149	022446	112405		12\$:	MOVB	(R4)+,R5	:LOAD R5 WITH CHAR
4150	022450	010502			MOV	R5,R2	:LOAD R2 WITH CHAR
4151							:CHECK FIRST FOUR CHARACTER MESSAGE
4152							:IN THE BIT WINDOW (0,125,252,377)
4153	022452	012737	120001	027110	MOV	#CRC16,XPOLY	:LOAD POLYNOMIAL
4154	022460	010537	022472		MOV	R5,67\$:LOAD SOFT CHAR FOR BCC
4155	022464	004537	027004		JSR	R5,SIMBCC	:CALCULATE SOFT BCC
4156	022470	000010			10		:SHIFT COUNT
4157	022472	000000		67\$:	0		:CHARACTER
4158	022474	000000		10\$:	0		:OLD BCC
4159	022476	013737	027112	022474	MOV	CALBCC,10\$:LOAD SOFT BCC FOR NEXT SHIFT
4160	022504	104415	000001		DATACLK,	1	:SHIFT DATA IN TO BIT WINDOW
4161	022510	106002		64\$:	RORB	R2	:SHIFT SOFT DATA
4162	022512	103005			BCC	65\$:BR IF A SPACE
4163	022514	004737	026112		JSR	PC,GETSI	:LOOK AT BIT WINDOW
4164	022520	103406			BCS	66\$:BR IF OK (MARK)
4165	022522	104006			HLT	6	:ERROR, BIT WINDOW WAS A SPACE
4166	022524	000404			BR	66\$:CONTINUE
4167	022526	004737	026112		JSR	PC,GETSI	:LOOK AT BIT WINDOW
4168	022532	103001		65\$:	BCC	66\$:BR IF OK (SPACE)
4169	022534	104006			HLT	6	:ERROR, BIT WINDOW WAS A MARK
4170	022536			66\$:			
4171	022536	005203			INC	R3	:BUMP BIT COUNTER
4172	022540	022703	000010		CMP	#10,R3	:DONE FULL 9 BITS YET
4173	022544	001357			BNE	64\$:BR IF NO
4174	022546	005003			CLR	R3	:CLEAR BIT COUNTER
4175	022550	005300			DEC	R0	:DEC CHARACTER COUNT
4176	022552	001335			BNE	12\$:BR IF NOT DONE YET
4177							:CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW
4178							
4179							
4180							
4181							
4182	022554	013700	027112		MOV	CALBCC,R0	:PUT BCC IN R0
4183	022560	104415	000001		DATACLK,	1	:SHIFT HARDWARE BCC
4184	022564	006000		68\$:	ROR	R0	:SHIFT SOFT BCC
4185	022566	103005			BCC	69\$:BR IF CARRY CLEAR
4186	022570	004737	026112		JSR	PC,GETSI	:LOOK AT BIT WINDOW
4187	022574	103406			BCS	70\$:BR IF OK (MARK)
4188	022576	104014			HLT	14	:ERROR, CRC WRONG (SPACE)
4189	022600	000404			BR	70\$:CONTINUE
4190	022602	004737	026112		JSR	PC,GETSI	:LOOK AT BIT WINDOW
4191	022606	103001		69\$:	BCC	70\$:BR IF OK (SPACE)
4192	022610	104014			HLT	14	:ERROR, CRC WRONG (MARK)

4193	022612			70\$:	INC R3	:BUMP BIT COUNTER
4194	022612	005203			CMP #20,R3	:FINISHED BCC YET?
4195	022614	022703	000020		BNE 68\$:BR IF NO
4196	022620	001357			CLR R3	:CLEAR BIT COUNTER
4197	022622	005003			MOV #4,R0	:RESET CHARACTER COUNTER
4198	022624	012700	000004		MOV #MESDAT,R4	:LOAD MESSAGE POINTER
4199	022630	012704	027674		CLR 11\$:CLR SOFT BCC
4200	022634	005037	022666		MOV (R4)+,R5	:LOAD CHAR IN R5
4201	022640	112405		13\$:	MOV R5,R2	:LOAD CHAR IN R2
4202	022642	010502				
4203						
4204						
4205						
4206						
4207	022644	012737	120001	027110	MOV #CRC16,XPOLY	:LOAD POLYNOMIAL
4208	022652	010537	022664		MOV R5,76\$:LOAD SOFT CHAR FOR BCC
4209	022656	004537	027004		JSR R5,SIMBCC	:CALCULATE SOFT BCC
4210	022662	000010			LD 10	:SHIFT COUNT
4211	022664	000000		76\$:	LD 0	:CHARACTER
4212	022666	000000		11\$:	LD 0	:OLD BCC
4213	022670	013737	027112	022666	MOV CALBCC,11\$:LOAD SOFT BCC FOR NEXT SHIFT
4214	022676	104415	000001		DATACLK, 1	:SHIFT DATA IN TO BIT WINDOW
4215	022702	106002		73\$:	RORB R2	:SHIFT SOFT DATA
4216	022704	103005			BCC 74\$:BR IF A SPACE
4217	022706	004737	026112		JSR PC,GETSI	:LOOK AT BIT WINDOW
4218	022712	103406			BCS 75\$:BR IF OK (MARK)
4219	022714	104006			HLT 6	:ERROR, BIT WINDOW WAS A SPACE
4220	022716	000404			BR 75\$:CONTINUE
4221	022720	004737	026112		JSR PC,GETSI	:LOOK AT BIT WINDOW
4222	022724	103001		74\$:	BCC 75\$:BR IF OK (SPACE)
4223	022726	104006			HLT 6	:ERROR, BIT WINDOW WAS A MARK
4224	022730			75\$:		
4225	022730	005203			INC R3	:BUMP BIT COUNTER
4226	022732	022703	000010		CMP #10,R3	:DONE FULL 8 BITS YET
4227	022736	001357			BNE 73\$:BR IF NO
4228	022740	005003			CLR R3	:CLEAR BIT COUNTER
4229	022742	005300			DEC R0	:DEC CHARACTER COUNT
4230	022744	001335			BNE 13\$:BR IF NOT DONE YET
4231						
4232						
4233						
4234	022746	013700	027112		MOV CALBCC,R0	:PUT BCC IN R0
4235	022752	104415	000001	77\$:	DATACLK, 1	:SHIFT HARDWARE BCC
4236	022756	006000			ROR R0	:SHIFT SOFT BCC
4237	022760	103005			BCC 78\$:BR IF CARRY CLEAR
4238	022762	004737	026112		JSR PC,GETSI	:LOOK AT BIT WINDOW
4239	022766	103406			BCS 79\$:BR IF OK (MARK)
4240	022770	104014			HLT 14	:ERROR, CRC WRONG (SPACE)
4241	022772	000404			BR 79\$:CONTINUE
4242	022774	004737	026112		JSR PC,GETSI	:LOOK AT BIT WINDOW
4243	023000	103001		78\$:	BCC 79\$:BR IF OK (SPACE)
4244	023002	104014			HLT 14	:ERROR, CRC WRONG (MARK)
4245	023004			79\$:		
4246	023004	005203			INC R3	:BUMP BIT COUNTER
4247	023006	022703	000020		CMP #20,R3	:FINISHED BCC YET?
4248	023012	001357			BNE 77\$:BR IF NO
4249	023014	005003			CLR R3	:CLEAR BIT COUNTER

42000
42001
42002
42003
42004
42005
42006
42007
42008
42009
42010
42011
42012
42013
42014
42015
42016
42017
42018
42019
42020
42021
42022
42023
42024
42025
42026
42027
42028
42029
42030
42031
42032
42033
42034
42035
42036
42037
42038
42039
42040
42041
42042
42043
42044
42045
42046
42047
42048
42049
42050
42051
42052
42053
42054
42055
42056
42057
42058
42059
42060
42061
42062
42063
42064
42065
42066
42067
42068
42069
42070
42071
42072
42073
42074
42075
42076
42077
42078
42079
42080
42081
42082
42083
42084
42085
42086
42087
42088
42089
42090
42091
42092
42093
42094
42095
42096
42097
42098
42099
42100
42101
42102
42103
42104

```

023016 104415 000001
023026 004737 026112
023026 103401
023030 104024
023032 005203
023034 022703 000007
023040 001366
023042 104415 000010
023046 005003
023050 104415 000001
023054 004737 026112
023060 103401
023062 104024
023064 005203
023066 022703 000020
023072 001366

023074 104415 000001
023100 012703 000004
023104 012702 027674
023110 004737 026752
023114 104414
023116 021204
023120 016104 000004
023124 112205
023126 120504
023130 001401
023132 104010
023134 005303
023136 001364

023140 004737 026752
023144 104414
023146 021204
023150 116137 000004 001252
023156 042737 177400 001252
023164 004737 026752
023170 104414
023172 021244
023174 016104 000004
023200 042704 000376
023204 012705 000001
023210 120504
023212 001401
023214 104015
023216 104414
    
```

```

;CHECK TO SEE IF TRANSMITTER IS MARKING
2$: DATACLK, 1 ;CLOCK TRANSMITTER
    JSR PC,GETSI ;LOOK AT WINDOW
    BCS 3$ ;IT SHOULD BE MARKING
    HLT 24 ;ERROR, BIT WAS A SPACE
3$: INC R3 ;BUMP BIT COUNTER
    CMP #7,R3 ;DONE YET
    BNE 2$ ;BR IF NO
    DATACLK, 10 ;GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
    CLR R3 ;CLEAR BIT COUNTER
4$: DATACLK, 1 ;SHIFT OUT NEXT BIT
    JSR PC,GETSI ;LOOK AT BIT WINDOW
    BCS +4 ;BR IF IT IS A MARK
    HLT 24 ;ERROR, TRANSMITTER IS NOT MARKING
    INC R3 ;INC BIT COUNT
    CMP #20,R3 ;DONE YET?
    BNE 4$ ;BR IF NO

;CHECK TO SEE THAT FIRST FOUR CHARACTER MESSAGE
;WAS RECEIVED CORRECTLY (0,125,252,377)
40$: DATACLK, 1 ;GET LAST BIT IN RECEIVER
    MOV #4,R3 ;R3=CHARACTER COUNT
    MOV #MESDAT,R2 ;LOAD MESSAGE POINTER IN R2
    JSR PC,INRDY ;WAIT FOR INRDY
    ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
    021204
    MOV 4(R1),R4 ;PUT "FOUND" IN R4
    MOVB (R2)+,R5 ;PUT "EXPECTED" IN R5
    CPMB R5,R4 ;IS RECEIVED DATA CORRECT?
    BEQ 41$ ;BR IF YES
    HLT 10 ;RECEIVE DATA ERROR
41$: DEC R3 ;DEC CHARACTER COUNT
    BNE 40$ ;BR IF NOT DONE YET

;CHECK TO SEE THAT IN BCC MATCH IS SET
;AND THAT THE BCC WAS RECEIVED CORRECTLY
    JSR PC,INRDY ;WAIT FOR INRDY
    ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
    021204 ;GET FIRST HALF OF CRC
    MOVB 4(R1),TEMP3 ;PUT IN TEMP3
    BIC #177400,TEMP3 ;CLEAR HI BYTE
    JSR PC,INRDY ;WAIT FOR INRDY
    ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
    021244
    MOV 4(R1),R4 ;PUT "FOUND" IN R4
    BIC #376,R4 ;CLEAR UNWANTED BITS
    MOV #1,R5 ;PUT "EXPECTED" IN R5
    CPMB R5,R4 ;IS IN BCC MATCH SET?
    BEQ 50$
    HLT 15 ;IN BCC MATCH ERROR
50$: ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
    
```

```

4305 023220 021204
4306 023222 116137 000004 001251
4307 023230 042737 000377 001250
4308 023236 053737 001250 001252
4309 023244 023737 027112 001252
4310 023252 001401
4311 023254 104027

```

```

021204 :GET LAST HALF
MOVB 4(R1),TEMP2+1 :PUT IN TEMP2
BIC #377,TEMP2 :CLEAR LO BYTE
BIS TEMP2,TEMP3 :16 BIT BCC NOW IN TEMP3
CMP CALBCC,TEMP3 :IS IT CORRECT?
BEQ 42$ :BR IF OK
HLT 27

```

```

:CHECK TO SEE THAT SECOND FOUR CHARACTER MESSAGE
:WAS RECEIVED CORRECTLY (0,125,252,377)

```

```

4316 023256 012703 000004
4317 023262 012702 027674
4318 023266 004737 026752
4319 023272 104414
4320 023274 021204
4321 023276 016104 000004
4322 023302 112205
4323 023304 120504
4324 023306 001401
4325 023310 104010
4326 023312 005303
4327 023314 001364

```

```

42$: MOV #4,R3 :R3=CHARACTER COUNT
MOV #MESDAT,R2 :LOAD MESSAGE POINTER IN R2
43$: JSR PC,INRDY :WAIT FOR INRDY
ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021204
MOV 4(R1),R4 :PUT "FOUND" IN R4
MOVB (R2)+,R5 :PUT "EXPECTED" IN R5
CMPB R5,R4 :IS RECEIVED DATA CORRECT?
BEQ 44$ :BR IF YES
HLT 10 :RECEIVE DATA ERROR
44$: DEC R3 :DEC CHARACTER COUNT
BNE 43$ :BR IF NOT DONE YET

```

```

:CHECK TO SEE THAT IN BCC MATCH IS SET
:AND THAT THE BCC WAS RECEIVED CORRECTLY

```

```

4332 023316 004737 026752
4333 023322 104414
4334 023324 021204
4335 023326 116137 000004 001252
4336 023334 042737 177400 001252
4337 023342 004737 026752
4338 023346 104414
4339 023350 021244
4340 023352 016104 000004
4341 023356 042704 000376
4342 023362 012705 000001
4343 023366 120504
4344 023370 001401
4345 023372 104015
4346 023374
4347 023374 104414
4348 023376 021204
4349 023400 116137 000004 001251
4350 023406 042737 000377 001250
4351 023414 053737 001250 001252
4352 023422 023737 027112 001252
4353 023430 001401
4354 023432 104027
4355 023434 104400

```

```

JSR PC,INRDY :WAIT FOR INRDY
ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021204 :GET FIRST HALF OF CRC
MOVB 4(R1),TEMP3 :PUT IN TEMP3
BIC #177400,TEMP3 :CLEAR HI BYTE
JSR PC,INRDY :WAIT FOR INRDY
ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021244
MOV 4(R1),R4 :PUT "FOUND" IN R4
BIC #376,R4 :CLEAR UNWANTED BITS
MOV #1,R5 :PUT "EXPECTED" IN R5
CMPB R5,R4 :IS IN BCC MATCH SET?
BEQ 51$
HLT 15 :IN BCC MATCH ERROR
51$: ROMCLK :NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021204 :GET LAST HALF
MOVB 4(R1),TEMP2+1 :PUT IN TEMP2
BIC #377,TEMP2 :CLEAR LO BYTE
BIS TEMP2,TEMP3 :16 BIT BCC NOW IN TEMP3
CMP CALBCC,TEMP3 :IS IT CORRECT?
BEQ 5$ :BR IF OK
HLT 27
5$: SCOPE :SCOPE THIS TEST

```

```

:*****:***** TEST 53 *****
:*DDCMP EOM FUNCTION TEST
:*THIS TEST LOADS OUT SILO WITH: 2 SYNCs,4 CHAR MESSAGE,EOM

```


K07

```

4361 ;*SOM,4 CHAR MESS,EOM. THE DATA STREAM IS CHECKED TO BE
4362 ;*4 CHAR,BCC,4 CHAR,BCC,MARKS. THIS TEST VERIFYS THAT
4363 ;*THE CHARCTERS LOADED WITH EOM SET ARE LOST
4364 ;*ALSO THAT THE CHAR LOADED WITH SOM IS NOT IN THE BCC
4365 ;*ALL DATA AND BCC'S ARE CHECKED IN THE BIT WINDOW
4366 ;*THE FOUR CHARACTER MESSAGE IS 0,125,252,377
4367 ;*RECEIVED DATA IS VERIFIED, AND IN BCC MATCH IS CHECKED
4368 ;:*****
4369
4370 ; TEST 53
4371 ;-----
4372 023436 012737 000053 001226 TST53: MOV #53,TSTNO
4373 023444 012737 024636 001216 MOV #TST54,NEXT
4374
4375 023452 104412 MSTCLR ;R1 CONTAINS BASE DMC11 ADDRESS
4376 ;MASTER CLEAR DMC11
4377
4378 ;LOAD OUT DATA SILO
4379
4380 023454 012711 004000 MOV #BIT11,(R1) ;SET LINE UNIT LOOP
4381 023460 012704 027674 MOV #MESDAT,R4 ;LOAD POINTER TO DATA
4382 023464 005037 023600 CLR 10$ ;CLEAR SOFT BCC
4383 023470 012700 000004 MOV #4,R0 ;LOAD CHARACTER COUNT
4384 023474 004737 027256 JSR PC,SYNLD ;LOAD 2 SYNC'S IN OUT SILO
4385 023500 004737 026276 JSR PC,OUTRDY ;WAIT FOR OUTRDY
4386 023504 004537 027412 JSR R5,MESLD ;LOAD SILO WITH 4 CHAR MESS
4387 023510 027674 MESDAT ;ADDRESS OF MESSAGE
4388 023512 000004 4 ;NUMBER OF CHARACTERS
4389 023514 004737 027366 JSR PC,EOM ;LOAD GARBAGE CHARACTER, WITH EOM SET
4390 023520 004737 027336 JSR PC,SOM ;LOAD GARBAGE CHAR WITH SOM SET
4391 023524 004537 027412 JSR R5,MESLD ;LOAD FOUR MORE CHARACTERS
4392 023530 027674 MESDAT ;ADDRESS OF MESSAGE
4393 023532 000004 4 ;NUMBER OF CHACTERS
4394 023534 004737 027366 JSR PC,EOM ;SET EOM
4395 023540 004737 026144 JSR PC,OCOR ;WAIT FOR OCCr
4396 023544 005003 CLR R3 ;CLEAR BIT COUNTER
4397 023546 104415 000022 DATACLK,22 ;CLOCK DATA
4398 023552 112405 12$: MOVB (R4)+,R5 ;LOAD R5 WITH CHAR
4399 023554 010502 MOV R5,R2 ;LOAD R2 WITH CHAR
4400
4401 ;CHECK FIRST FOUR CHARACTER MESSAGE
4402 ;IN THE BIT WINDOW (0,125,252,377)
4403
4404 023556 012737 120001 027110 MOV #CRC16,XPOLY ;LOAD POLYNOMIAL
4405 023564 010537 023576 MOV R5,67$ ;LOAD SOFT CHAR FOR BCC
4406 023570 004537 027004 JSR R5,SIMBCC ;CALCULATE SOFT BCC
4407 023574 000010 10 ;SHIFT COUNT
4408 023576 000000 67$: 0 ;CHARACTER
4409 023600 000000 10$: 0 ;OLD BCC
4410 023602 013737 027112 023600 MOV CALBCC,10$ ;LOAD SOFT BCC FOR NEXT SHIFT
4411 023610 104415 000001 64$: DATACLK,1 ;SHIFT DATA IN TO BIT WINDOW
4412 023614 106002 RORB R2 ;SHIFT SOFT DATA
4413 023616 103005 BCC 65$ ;BR IF A SPACE
4414 023620 004737 026112 JSR PC,GETSI ;LOOK AT BIT WINDOW
4415 023624 103406 BCS 66$ ;BR IF OK (MARK)
4416 023626 104006 HLT 6 ;ERROR, BIT WINDOW WAS A SPACE
4417 023630 000404 BR 66$ ;CONTINUE
  
```

4417	023632	004737	026112	65\$:	JSR	PC,GETSI	:LOOK AT BIT WINDOW
4418	023636	103001			BCC	66\$:BR IF OK (SPACE)
4419	023640	104006			HLT	6	:ERROR, BIT WINDOW WAS A MARK
4420	023642			66\$:			
4421	023642	005203			INC	R3	:BUMP BIT COUNTER
4422	023644	022703	000010		CMP	#10,R3	:DONE FULL 8 BITS YET
4423	023650	001357			BNE	64\$:BR IF NO
4424	023652	005003			CLR	R3	:CLEAR BIT COUNTER
4425	023654	005300			DEC	R0	:DEC CHARACTER COUNT
4426	023656	001335			BNE	12\$:BR IF NOT DONE YET
4427							
4428							:CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW
4429							
4430	023660	013700	027112		MOV	CALBCC,R0	:PUT BCC IN R0
4431	023664	104415	000001	68\$:	DATACLK,	1	:SHIFT HARDWARE BCC
4432	023670	006000			ROR	R0	:SHIFT SOFT BCC
4433	023672	103005			BCC	69\$:BR IF CARRY CLEAR
4434	023674	004737	026112		JSR	PC,GETSI	:LOOK AT BIT WINDOW
4435	023700	103406			BCS	70\$:BR IF OK (MARK)
4436	023702	104014			HLT	14	:ERROR, CRC WRONG (SPACE)
4437	023704	000404			BR	70\$:CONTINUE
4438	023706	004737	026112	69\$:	JSR	PC,GETSI	:LOOK AT BIT WINDOW
4439	023712	103001			BCC	70\$:BR IF OK (SPACE)
4440	023714	104014			HLT	14	:ERROR, CRC WRONG (MARK)
4441	023716			70\$:			
4442	023716	005203			INC	R3	:BUMP BIT COUNTER
4443	023720	022703	000020		CMP	#20,R3	:FINISHED BCC YET?
4444	023724	001357			BNE	68\$:BR IF NO
4445	023726	005003			CLR	R3	:CLEAR BIT COUNTER
4446							
4447							:CHECK CHARACTER LOADED WITH SOM (000), IN THE BIT WINDOW
4448							
4449	023730	005005			CLR	R5	:CHARACTER LOADED WITH SOM
4450	023732	010502			MOV	R5,R2	:LOAD R2 WITH CHAR
4451	023734	104415	000001	32\$:	DATACLK,	1	:CLOCK TRANSMITTER
4452	023740	106002			RORB	R2	:SHIFT SOFT DATA
4453	023742	103005			BCC	30\$:BR IF SPACE
4454	023744	004737	026112		JSR	PC,GETSI	:LOOK AT BIT WINDOW
4455	023750	103406			BCS	31\$:BR IF OK (MARK)
4456	023752	104006			HLT	5	:ERROR, BIT WINDOW WAS A SPACE
4457	023754	000404			BR	31\$:CONTINUE
4458	023756	004737	026112	30\$:	JSR	PC,GETSI	:LOOK AT BIT WINDOW
4459	023762	103001			BCC	31\$:BR IF OK (SPACE)
4460	023764	104006			HLT	6	:ERROR, BIT WINDOW WAS A MARK
4461	023766	005203		31\$:	INC	R3	:BUMP BIT COUNTER
4462	023770	022703	000010		CMP	#10,R3	:DONE CHARACTER YET?
4463	023774	001357			BNE	32\$:BR IF NO
4464	023776	005003			CLR	R3	:RESET BIT COUNTER
4465	024000	012700	000004		MOV	#4,R0	:RESET CHARACTER COUNTER
4466	024004	012704	027674		MOV	#MESDAT,R4	:LOAD MESSAGE POINTER
4467	024010	005037	024042		CLR	11\$:CLR SOFT BCC
4468	024014	112405		13\$:	MOVB	(R4)+,R5	:LOAD CHAR IN R5
4469	024016	010502			MOV	R5,R2	:LOAD CHAR IN R2
4470							
4471							:CHECK SECOND MESSAGE IN THE BIT WINDOW (0,125,252,377)
4472							

M07

4473	024020	012737	120001	027110		MOV	#CRC16,XPOLY	:LOAD POLYNOMIAL
4474	024026	010537	024040			MOV	R5,76\$:LOAD SOFT CHAR FOR BCC
4475	024032	004537	027004			JSR	R5,SIMBCC	:CALCULATE SOFT BCC
4476	024036	000010				LD	10	:SHIFT COUNT
4477	024040	000000			76\$:	0		:CHARACTER
4478	024042	000000			11\$:	0		:OLD BCC
4479	024044	013737	027112	024042		MOV	CALBCC,11\$:LOAD SOFT BCC FOR NEXT SHIFT
4480	024052	104415	000001		73\$:	DATACLK,	1	:SHIFT DATA IN TO BIT WINDOW
4481	024056	106002				RORB	R2	:SHIFT SOFT DATA
4482	024060	103005				BCC	74\$:BR IF A SPACE
4483	024062	004737	026112			JSR	PC,GETSI	:LOOK AT BIT WINDOW
4484	024066	103406				BCS	75\$:BR IF OK (MARK)
4485	024070	104006				HLT	6	:ERROR, BIT WINDOW WAS A SPACE
4486	024072	000404				BR	75\$:CONTINUE
4487	024074	004737	026112		74\$:	JSR	PC,GETSI	:LOOK AT BIT WINDOW
4488	024100	103001				BCC	75\$:BR IF OK (SPACE)
4489	024102	104006				HLT	6	:ERROR, BIT WINDOW WAS A MARK
4490	024104				75\$:			
4491	024104	005203				INC	R3	:BUMP BIT COUNTER
4492	024106	022703	000010			CMP	#10,R3	:DONE FULL 8 BITS YET
4493	024112	001357				BNE	73\$:BR IF NO
4494	024114	005003				CLR	R3	:CLEAR BIT COUNTER
4495	024116	005300				DEC	R0	:DEC CHARACTER COUNT
4496	024120	001335				BNE	13\$:BR IF NOT DONE YET
4497								
4498								:CHECK BCC FOR PRECEDING MESSAGE IN THE BIT WINDOW
4499								
4500	024122	013700	027112			MOV	CALBCC,R0	:PUT BCC IN R0
4501	024126	104415	000001		77\$:	DATACLK,	1	:SHIFT HARDWARE BCC
4502	024132	006000				ROR	R0	:SHIFT SOFT BCC
4503	024134	103005				BCC	78\$:BR IF CARRY CLEAR
4504	024136	004737	026112			JSR	PC,GETSI	:LOOK AT BIT WINDOW
4505	024142	103406				BCS	79\$:BR IF OK (MARK)
4506	024144	104014				HLT	14	:ERROR, CRC WRONG (SPACE)
4507	024146	000404				BR	79\$:CONTINUE
4508	024150	004737	026112		78\$:	JSR	PC,GETSI	:LOOK AT BIT WINDOW
4509	024154	103001				BCC	79\$:BR IF OK (SPACE)
4510	024156	104014				HLT	14	:ERROR, CRC WRONG (MARK)
4511	024160				79\$:			
4512	024160	005203				INC	R3	:BUMP BIT COUNTER
4513	024162	022703	000020			CMP	#20,R3	:FINISHED BCC YET?
4514	024166	001357				BNE	77\$:BR IF NO
4515	024170	005003				CLR	R3	:CLEAR BIT COUNTER
4516								
4517								:CHECK TO SEE IF TRANSMITTER IS MARKING
4518								
4519	024172	104415	000001		2\$:	DATACLK,	1	:CLOCK TRANSMITTER
4520	024176	004737	026112			JSR	PC,GETSI	:LOOK AT WINDOW
4521	024202	103401				BCS	3\$:IT SHOULD BE MARKING
4522	024204	104024				HLT	24	:ERROR, BIT WAS A SPACE
4523	024206	005203			3\$:	INC	R3	:BUMP BIT COUNTER
4524	024210	022703	000007			CMP	#7,R3	:DONE YET
4525	024214	001357				BNE	2\$:BR IF NO
4526	024216	104415	000010			DATACLK,	10	:GIVE ENOUGH TICKS TO CLEAR OUT ACTIVE
4527	024222	005003				CLR	R3	:CLEAR BIT COUNTER
4528	024224	104415	000001		4\$:	DATACLK,	1	:SHIFT OUT NEXT BIT

4529	024230	004737	026112		JSR	PC,GETSI	:LOOK AT BIT WINDOW
4530	024234	103401			BOS	+4	:BR IF IT IS A MARK
4531	024236	104024			HLT	24	:ERROR, TRANSMITTER IS NOT MARKING
4532	024240	005203			INC	R3	:INC BIT COUNT
4533	024242	022703	000020		CMP	#20,R3	:DONE YET?
4534	024246	001366			BNE	4\$:BR IF NO
4535							
4536							
4537							
4538							
4539	024250	104415	000001				:CHECK TO SEE THAT FIRST FOUR CHARACTER MESSAGE
4540	024254	012703	000004				:WAS RECEIVED CORRECTLY (0,125,252,377)
4541	024260	012702	027674		DATACLK,	1	:GET LAST BIT IN RECEIVER
4542	024264	004737	026752	40\$:	MOV	#4,R3	:R3=CHARACTER COUNT
4543	024270	104414			MOV	#MESDAT,R2	:LOAD MESSAGE POINTER IN R2
4544	024272	021204			JSR	PC,INRDY	:WAIT FOR INRDY
4545	024274	016104	000004		ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4546	024300	112205			021204		
4547	024302	120504			MOV	4(R1),R4	:PUT "FOUND" IN R4
4548	024304	001401			MOVB	(R2)+,R5	:PUT "EXPECTED" IN R5
4549	024306	104010			CMPB	R5,R4	:IS RECEIVED DATA CORRECT?
4550	024310	005303		41\$:	BEQ	41\$:BR IF YES
4551	024312	001364			HLT	10	:RECEIVE DATA ERROR
4552					DEC	R3	:DEC CHARACTER COUNT
4553					BNE	40\$:BR IF NOT DONE YET
4554							:CHECK TO SEE THAT IN BCC MATCH IS SET
4555							:AND THAT THE BCC WAS RECEIVED CORRECTLY
4556	024314	004737	026752		JSR	PC,INRDY	:WAIT FOR INRDY
4557	024320	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4558	024322	021204			021204		:GET FIRST HALF OF CRC
4559	024324	116137	000004	001252	MOVB	4(R1),TEMP3	:PUT IN TEMP3
4560	024332	042737	177400	001252	BIC	#177400,TEMP3	:CLEAR HI BYTE
4561	024340	004737	026752		JSR	PC,INRDY	:WAIT FOR INRDY
4562	024344	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4563	024346	021244			021244		
4564	024350	016104	000004		MOV	4(R1),R4	:PUT "FOUND" IN R4
4565	024354	042704	000376		BIC	#376,R4	:CLEAR UNWANTED BITS
4566	024360	012705	000001		MOV	#1,R5	:PUT "EXPECTED" IN R5
4567	024364	120504			CMPB	R5,R4	:IS IN BCC MATCH SET?
4568	024366	001401			BEQ	50\$	
4569	024370	104015		50\$:	HLT	15	:IN BCC MATCH ERROR
4570	024372						
4571	024372	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4572	024374	021204			021204		:GET LAST HALF
4573	024376	116137	000004	001251	MOVB	4(R1),TEMP2+1	:PUT IN TEMP2
4574	024404	042737	000377	001250	BIC	#377,TEMP2	:CLEAR LO BYTE
4575	024412	053737	001250	001252	BIS	TEMP2,TEMP3	:16 BIT BCC NOW IN TEMP3
4576	024420	023737	027112	001252	CMP	CALBCC,TEMP3	:IS IT CORRECT?
4577	024426	001401			BEQ	45\$:BR IF OK
4578	024430	104027			HLT	27	
4579							
4580							:CHECK THAT CHARACTER LOADED WITH SOM WAS RECEIVED (000)
4581							
4582	024432	004737	026752	45\$:	JSR	PC,INRDY	:WAIT FOR INRDY
4583	024436	104414			ROMCLK		:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4584	024440	021204			021204		:GET RECEIVE DATA

000000 016104 000004
000000 005005
000000 120504
000000 001401
000000 104010

MOV 4(R1),R4 ;PUT "FOUND" IN R4
CLR R5 ;PUT "EXPECTED" IN R5
CMPB R5,R4 ;IS RECEIVED DATA CORRECT?
BEQ 42\$;BR IF YES
HLT 10 ;RECEIVE DATA ERROR

;CHECK TO SEE THAT SECOND FOUR CHARACTER MESSAGE
;WAS RECEIVED CORRECTLY (0,125,252,377)

000000 012703 000004
000000 005005 027674
000000 004737 026752
000000 104414
000000 021204
000000 016104 000004
000000 116137
000000 120504
000000 001401
000000 104010
000000 005303
000000 001364

42\$: MOV #4,R3 ;R3=CHARACTER COUNT
MOV #MESDAT,R2 ;LOAD MESSAGE POINTER IN R2
43\$: JSR PC,INRDY ;WAIT FOR INRDY
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021204
MOV 4(R1),R4 ;PUT "FOUND" IN R4
MOVB (R2)+,R5 ;PUT "EXPECTED" IN R5
CMPB R5,R4 ;IS RECEIVED DATA CORRECT?
BEQ 44\$;BR IF YES
HLT 10 ;RECEIVE DATA ERROR
44\$: DEC R3 ;DEC CHARACTER COUNT
BNE 43\$;BR IF NOT DONE YET

;CHECK TO SEE THAT IN BCC MATCH IS SET
;AND THAT THE BCC WAS RECEIVED CORRECTLY

000000 004737 026752
000000 104414
000000 021204
000000 116137 000004 001252
000000 042737 177400 001252
000000 004737 026752
000000 104414
000000 021244
000000 016104 000004
000000 042704 000376
000000 012705 000001
000000 120504
000000 001401
000000 104015

JSR PC,INRDY ;WAIT FOR INRDY
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021204 ;GET FIRST HALF OF CRC
MOVB 4(R1),TEMP3 ;PUT IN TEMP3
BIC #177400,TEMP3 ;CLEAR HI BYTE
JSR PC,INRDY ;WAIT FOR INRDY
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021244
MOV 4(R1),R4 ;PUT "FOUND" IN R4
BIC #376,R4 ;CLEAR UNWANTED BITS
MOV #1,R5 ;PUT "EXPECTED" IN R5
CMPB R5,R4 ;IS IN BCC MATCH SET?
BEQ 51\$
HLT 15 ;IN BCC MATCH ERROR

51\$: ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021204 ;GET LAST HALF
MOVB 4(R1),TEMP2+1 ;PUT IN TEMP2
BIC #377,TEMP2 ;CLEAR LO BYTE
BIS TEMP2,TEMP3 ;16 BIT BCC NOW IN TEMP3
CMP CALBCC,TEMP3 ;IS IT CORRECT?
BEQ 52\$;BR IF OK
HLT 27
52\$: SCOPE ;SCOPE THIS TEST

***** TEST 54 *****
*EMPTY SILO TEST
*LOAD SILO WITH 2 SYNCS, 4 CHAR MESSAGE, SINGLE CLOCK
*UNTIL THE SILO IS EMPTY, LOAD 4 MORE CHARACTERS IN THE
*SILO. GIVE MORE TICKS, AND VERIFY THAT ONLY THE FIRST

```

46700 024636 012737 000054 001226
46701 024644 012737 025070 001216
46702 024652 104412
46703 024660 012711 004000
46704 024668 012702 027674
46705 024676 012700 000004
46706 024684 004737 027256
46707 024692 004737 026276
46708 024700 004537 027412
46709 024708 027674
46710 024716 004737 026144
46711 024724 104415 000065
46712 024732 004537 027412
46713 024740 027674
46714 024748 000004
46715 024756 004737 026144
46716 024764 104415 000005
46717 024772 104414
46718 024780 027276 000040 000004
46719 024788 001401
46720 024796 104034
46721 024804 104415 000041
46722 024812 004737 026752
46723 024820 104414
46724 024828 021204
46725 024836 016104 000004
46726 024844 112205
46727 024852 120504
46728 024860 001401
46729 024868 104010
46730 024876 005300
46731 024884 001364
46732 024892 004737 026752
46733 024900 104414
46734 024908 021204
46735 024916 016104 000004
46736 024924 012705 000377
46737 024932 120504
46738 024940 001401
46739 024948 104010
46740 024956 004737 026752
46741 024964 104414
46742 024972 021204
46743 024980 016104 000004
46744 024988 012705 000177
46745 024996 120504
46746 025004 001401
46747 025012 104010

```

```

: *4 CHARACTER MESSAGE WAS RECEIVED AND THAT RTS IS CLEAR
: *****
:
: TEST 54
:-----
TST54: MOV #54,TSTNO
MOV #TST55,NEXT

MSTCLR ;R1 CONTAINS BASE DMC11 ADDRESS
MOV #BIT11,(R1) ;MASTER CLEAR DMC11
MOV #MESDAT,R2 ;SET LINE UNIT LOOP
MOV #4,R0 ;R2 POINTS TO MESSAGE
JSR PC,SYNLD ;R0 = CHAR COUNT
JSR PC,OUTRDY ;LOAD SILO WITH TWO SYNCs
JSR R5,MESLD ;WAIT FOR OUTRDY
MESDAT ;LOAD MESSAGE IN SILO
4 ;START OF MESSAGE
JSR PC,OCOR ;CHARACTER COUNT
DATACLK, 65 ;WAIT FOR OCOR
JSR R5,MESLD ;CLOCK DATA (EMPTY SILO)
MESDAT ;PUT MORE CHARACTERS IN SILO
4

JSR PC,OCOR ;CLOCK UNTIL RTS IS CLEARED
DATACLK, 5 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
ROMCLK ;GET RTS
021264 ;IS IT CLEAR?
BIT #BITS,4(R1) ;BR IF YES
BEG 5$ ;ERROR, RTS NOT CLEAR
HLT 34 ;CLOCK XMITTER SOME MORE
;OK LETS CHECK WHAT WAS RECEIVED
DATACLK, 41 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
JSR PC,INRDY ;GET RECEIVE DATA
ROMCLK ;PUT IT IN R4
021204 ;R5 = "EXPECTED"
MOV 4(R1),R4 ;IS DATA CORRECT?
MOVB (R2)+,R5 ;BR IF OK
CMPB R5,R4 ;DATA ERROR
BEG 2$ ;DEC CHAR COUNT
HLT 10 ;BR IF NOT DONE YET
;WAIT FOR INRDY
DEC R0 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
BNE 1$ ;GET RECEIVE DATA
;PUT IT IN "FOUND"
JSR PC,INRDY ;R5 = "EXPECTED"
ROMCLK ;SHOULD SEE 377
021204 ;BR IF OK
MOV 4(R1),R4 ;ERROR, TRANSMITTER DID NOT ABORT
MOV #377,R5 ;WAIT FOR INRDY
CMPB R5,R4 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
BEG 4$ ;GET RECEIVE DATA
HLT 10 ;PUT IT IN "FOUND"
;R5 = "EXPECTED"
JSR PC,INRDY ;SHOULD SEE 177
ROMCLK ;BR IF OK
021204 ;ERROR, TRANSMITTER DID NOT ABORT
MOV 4(R1),R4
MOV #177,R5
CMPB R5,R4
BEG 10$
HLT 10

```

47067	025066			
47068	025066	104400		
47069				
47070				
47071				
47072				
47073				
47074				
47075				
47076				
47077				
47078				
47079	025070	012737	000055	001226
4710	025076	012737	025206	001216
4711				
4712	025104	104412		
4713	025106	012702	000012	
4714	025112	012711	004000	
4715	025116	012761	000020	000004
4716	025124	104414		
4717	025126	122113		
4718	025130	004737	027256	
4719	025134	004737	026276	
4720	025140	004537	027412	
4721	025144	027674		
4722	025146	000004		
4723	025150	004737	026144	
4724	025154	104415	000073	
4725	025160	104414		
4726	025162	021244		
4727	025164	016104	000004	
4728	025170	042704	000257	
4729	025174	005005		
4730	025176	120504		
4731	025200	001401		
4732	025202	104035		
4733	025204	104400		
4734				
4735				
4736				
4737				
4738				
4739				
4740				
4741				
4742				
4743				
4744				
4745				
4746				
4747				
4748	025206	012737	000056	001226
4749	025214	012737	025574	001216
4750				
4751	025222	104412		
4752	025224	032737	040000	001366

10:

SCOPE ;SCOPE THIS TEST

```

***** TEST 55 *****
*HALF DUPLEX TEST
*SET LINE UNIT LOOP AND HALF DUPLEX, SEND SYNCs AND A
*MESSAGE. VERIFY THAT IN-ACTIVE AND IN-READY ARE CLEAR
*****

```

TEST 55

TST55:

```

-----
MOV #55,TSTNO
MOV #TST56,NEXT

```

```

;R1 CONTAINS BASE DMC11 ADDRESS
;MASTER CLEAR DMC11
;SAVE R2 FOR TYPEOUT
;SET LINE UNIT LOOP
;LOAD PORT4
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;SET H/D BIT
;LOAD 2 SYNCs
;WAIT FOR OUTRDY
;LOAD 4 CHAR MESSAGE
;ADDRESS OF MESSAGE
;CHARACTER COUNT
;WAIT FOR OCOR
;SEND MESSAGE
;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
;READ LU-12
;PUT "FOUND" IN R4
;CLEAR UNWANTED BITS
;R5 = "EXPECTED"
;IN-ACTIVE AND IN-RDY SHOULD BE CLEAR
;BR IF OK
;ERROR BOTH ARE NOT CLEAR

```

18:

SCOPE

```

***** TEST 56 *****
*DDCMP CABLE DATA TEST
*THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
*4 SYNCs, 16 CHAR,EOM, 16 CHAR,EOM, 16 CHAR,EOM
*THE 16 CHARACTERS INCLUDE A FLOATING ONE AND ZERO
*THE DATA IS TRANSMITTED OVER THE CABLE USING THE INTERNAL CLOCK
*RECEIVED DATA IS VERIFIED AS IS IN BCC MATCH
*LOOP-BACK CONNECTOR MUST BE ON TO RUN THIS TEST
*****

```

TEST 56

TST56:

```

-----
MOV #56,TSTNO
MOV #TST57,NEXT

```

```

;R1 CONTAINS BASE DMC11 ADDRESS
;MASTER CLEAR DMC11
;SKIP TEST IF NO

```

E08

4753	025232	001557			BEQ 3\$: LOOPBACK CONNECTOR ON
4754	025234	012711	004000		MOV #BIT11,(R1)	: SET LINE UNIT LOOP
4755	025240	004737	027256		JSR PC,SYNLD	: LOAD 2 SYNC
4756	025244	004737	027256		JSR PC,SYNLD	: LOAD 2 MORE SYNC
4757	025250	012737	120001	027110	MOV #CRC16,XPOLY	: LOAD POLYNOMIAL FOR SOFT CRC CALC
4758	025256	005037	025306		CLR 6\$: CLEAR OLD BCC
4759	025262	012703	000020		MOV #16.,R3	: CHARACTER COUNT
4760	025266	012702	027700		MOV #FLTDAT,R2	: R2= POINTER
4761	025272	112237	025304	7\$:	MOVB (R2)+,5\$: LOAD CHAR FOR SOFT BCC CALC.
4762	025276	004537	027004		JSR R5,SIMBCC	: CALC SOFT BCC
4763	025302	000010			LD	: SHIFT COUNT
4764	025304	000000		5\$:	D	: CHARACTER
4765	025306	000000		5\$:	D	: OLD BCC
4766	025310	013737	027112	025306	MOV CALBCC,6\$: LOAD OLD BCC
4767	025316	005303			DEC R3	: DEC COUNT
4768	025320	001364			BNE 7\$: BR IF NOT DONE YET
4769	025322	004537	027412		JSR R5,MESLD	: LOAD SILO
4770	025326	027700			FLTDAT	: MESSAGE ADDRESS
4771	025330	000020			LD	: CHARACTER COUNT
4772	025332	004737	027366		JSR PC,EOM	: LOAD AN EOM
4773	025336	004537	027412		JSR R5,MESLD	: LOAD SILO
4774	025342	027700			FLTDAT	: MESSAGE ADDRESS
4775	025344	000020			LD	: CHARACTER COUNT
4776	025346	004737	027366		JSR PC,EOM	: LOAD AN EOM
4777	025352	004537	027412		JSR R5,MESLD	: LOAD SILO
4778	025356	027700			FLTDAT	: MESSAGE ADDRESS
4779	025360	000020			LD	: CHARACTER COUNT
4780	025362	004737	027366		JSR PC,EOM	: LOAD AN EOM
4781	025366	004737	026144		JSR PC,OCOR	: WAIT FOR OCOR
4782	025372	005011			CLR (R1)	: CLEAR LINE UNIT LOOP
4783	025374	012700	000003		MOV #3,R0	: R0 = MESSAGE COUNT
4784	025400	012703	000020		MOV #16.,R3	: R3= CHARACTER COUNT
4785	025404	012702	027700		MOV #FLTDAT,R2	: LOAD MESSAGE POINTER IN R2
4786	025410	004737	026752	1\$:	JSR PC,INRDY	: WAIT FOR INRDY
4787	025414	104414			ROMCLK	: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4788	025416	021204			021204	: GET DATA FROM IN SILO
4789	025420	016104	000004		MOV 4(R1),R4	: PUT CHARACTER IN "FOUND"
4790	025424	112205			MOVB (R2)+,R5	: PUT "EXPECTED" IN R5
4791	025426	120504			CMPB R5,R4	: IS RECEIVED DATA CORRECT
4792	025430	001401			BEQ 2\$: BR IF OK
4793	025432	104025			HLT 2\$: DATA ERROR
4794	025434			2\$:		
4795	025434	005303			DEC R3	: DEC CHARACTER COUNT
4796	025436	001364			BNE 1\$: BR IF NOT DONE THIS MESSAGE
4797	025440	012703	000020		MOV #16.,R3	: RESET CHARACTER COUNT
4798						
4799						
4800						: CHECK TO SEE THAT IN BCC MATCH IS SET
4801						: AND THAT THE BCC WAS RECEIVED CORRECTLY
4802	025444	004737	026752		JSR PC,INRDY	: WAIT FOR INRDY
4803	025450	104414			ROMCLK	: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4804	025452	021204			021204	: GET FIRST HALF OF CRC
4805	025454	116137	000004	001252	MOVB 4(R1),TEMP3	: PUT IN TEMP3
4806	025462	042737	177400	001252	BIC #177400,TEMP3	: CLEAR HI BYTE
4807	025470	004737	026752		JSR PC,INRDY	: WAIT FOR INRDY
4808	025474	104414			ROMCLK	: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

F08

```

025476 021244
025500 016104 000004
025504 042704 000376
025510 012705 000001
025514 120504
025516 001401
025520 104015
025522 104414
025524 021204
025526 116137 000004 001251
025534 042737 000377 001250
025542 053737 001250 001252
025550 023737 027112 001252
025556 001401
025560 104027
025562 012702 027700
025566 005300
025570 001307
025572 104400
  
```

```

021244
MOV 4(R1),R4 ;PUT "FOUND" IN R4
BIC #376,R4 ;CLEAR UNWANTED BITS
MOV #1,R5 ;PUT "EXPECTED" IN R5
CMPB R5,R4 ;IS IN BCC MATCH SET?
BEQ 25$ ;IN BCC MATCH ERROR
HLT 15

25$: ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021204 ;GET LAST HALF
MOVB 4(R1),TEMP2+1 ;PUT IN TEMP2
BIC #377,TEMP2 ;CLEAR LO BYTE
BIS TEMP2,TEMP3 ;16 BIT BCC NOW IN TEMP3
CMP CALBCC,TEMP3 ;IS IT CORRECT?
BEQ 4$ ;BR IF OK
HLT 27

4$: MOV #FLTDAT,R2 ;RESET MESSAGE POINTER
DEC R0 ;DECREMENT COUNTER
BNE 1$ ;BR IF NOT DONE

3$: SCOPE ;SCOPE THIS TEST
  
```

```

***** TEST 57 *****
*DDCMP CABLE DATA TEST
*THIS TEST LOADS OUT SILO WITH THE FOLLOWING:
*4 SYNCs, 59 DATA CHARACTERS, EOM WITH GARBAGE CHARACTER
*THE DATA IS TRANSMITTED OVER THE CABLE USING THE INTERNAL CLOCK
*RECEIVED DATA IS VERIFIED AS IS IN BCC MATCH
*LOOP-BACK CONNECTOR MUST BE ON TO RUN THIS TEST
*****
  
```

TEST 57

```

TST57: MOV #57,TSTNO ;R1 CONTAINS BASE DMC11 ADDRESS
MOV #.EOP,NEXT ;MASTER CLEAR DMC11

MSTCLR ;SKIP TEST IF NO
BIT #BIT14,STAT1 ;LOOPBACK CONNECTOR ON
BEQ 3$ ;SET LINE UNIT LOOP
MOV #BIT11,(R1) ;LOAD 2 SYNCs
JSR PC,SYNLD ;LOAD 2 MORE SYNCs
JSR PC,SYNLD ;LOAD POLYNOMIAL FOR SOFT CRC CALC
MOV #CRC16,XPOLY ;CLEAR OLD BCC
CLR 6$ ;CHARACTER COUNT
MOV #59,R3 ;R2= POINTER
MOV #MESDAT,R2 ;LOAD CHAR FOR SOFT BCC CALC.
MOV (R2)+,5$ ;CALC SOFT BCC
MOVB (R2)+,5$ ;SHIFT COUNT
JSR R5,SIMBCC ;CHARACTER
10 ;OLD BCC
0 ;LOAD OLD BCC
0 ;DEC COUNT
MOV CALPCC,6$ ;BR IF NOT DONE YET
DEC R3 ;LOAD SILO
BNE 7$ ;MESSAGE ADDRESS
JSR R5,MESLD
MESDAT
  
```

4865	025716	000073			59.		: CHARACTER COUNT
4866	025720	004737	027366		JSR	PC, EOM	: LOAD AN EOM
4867	025724	004737	026144		JSR	PC, OCOR	: WAIT FOR OCOR
4868	025730	005011			CLR	(R1)	: CLEAR LINE UNIT LOOP
4869	025732	012700	000073		MOV	#59, R0	: R0= CHARACTER COUNT
4870	025736	012702	027674		MOV	#MESDAT, R2	: LOAD MESSAGE POINTER IN R2
4871	025742	004737	026752		JSR	PC, INRDY	: WAIT FOR INRDY
4872	025746	104414			ROMCLK		: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4873	025750	021204			021204		: GET DATA FROM IN SILO
4874	025752	016104	000004		MOV	4(R1), R4	: PUT CHARACTER IN "FOUND"
4875	025756	112205			MOVB	(R2)+, R5	: PUT "EXPECTED" IN R5
4876	025760	120504			CMPB	R5, R4	: IS RECEIVED DATA CORRECT
4877	025762	001401			BEQ	2\$: BR IF OK
4878	025764	104025			HLT	25	: DATA ERROR
4879	025766						
4880	025766	005300			DEC	R0	: DECREMENT COUNTER
4881	025770	001364			BNE	1\$: BR IF NOT DONE
4882							
4883							
4884							
4885							
4886	025772	004737	026752		JSR	PC, INRDY	: WAIT FOR INRDY
4887	025776	104414			ROMCLK		: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4888	026000	021204			021204		: GET FIRST HALF OF CRC
4889	026002	116137	000004	001252	MOVB	4(R1), TEMP3	: PUT IN TEMP3
4890	026010	042737	177400	001252	BIC	#177400, TEMP3	: CLEAR HI BYTE
4891	026016	004737	026752		JSR	PC, INRDY	: WAIT FOR INRDY
4892	026022	104414			ROMCLK		: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4893	026024	021244			021244		
4894	026026	016104	000004		MOV	4(R1), R4	: PUT "FOUND" IN R4
4895	026032	042704	000376		BIC	#376, R4	: CLEAR UNWANTED BITS
4896	026036	012705	000001		MOV	#1, R5	: PUT "EXPECTED" IN R5
4897	026042	120504			CMPB	R5, R4	: IS IN BCC MATCH SET?
4898	026044	001401			BEQ	25\$	
4899	026046	104015			HLT	15	: IN BCC MATCH ERROR
4900	026050						
4901	026050	104414			ROMCLK		: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4902	026052	021204			021204		: GET LAST HALF
4903	026054	116137	000004	001251	MOVB	4(R1), TEMP2+1	: PUT IN TEMP2
4904	026062	042737	000377	001250	BIC	#377, TEMP2	: CLEAR LO BYTE
4905	026070	053737	001250	001252	BIS	TEMP2, TEMP3	: 16 BIT BCC NOW IN TEMP3
4906	026076	023737	027112	001252	CMP	CALBCC, TEMP3	: IS IT CORRECT?
4907	026104	001401			BEQ	3\$: BR IF OK
4908	026106	104027			HLT	27	
4909	026110	104400					
4910							
4911		00300					
4912		00400					
4913		00500					
4914		00600					
4915	026112	00700					
4916		00800					
4917		00900					
4918		01000					
4919		01100					
4920	026112	104414			ROMCLK		: NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4921	026114	021364			021364		: PORT4+LU 17

1\$:

2\$:

25\$:

3\$:

; SUBROUTINES

GETSI:

; CHECK TO SEE THAT IN BCC MATCH IS SET
 ; AND THAT THE BCC WAS RECEIVED CORRECTLY

; THIS SUBROUTINE READS LU 17, AND PUTS IT INTO NITCH.
 ; NITCH IS ROTATED LEFT UNTILL THE SI BIT IS IN CARRY

4921	026116	017737	153270	026142	01400
4922	026124	106137	026142		01500
4923	026130	106137	026142		01600
4924	026134	106137	026142		01700
4925	026140	000207			01800
4926	026142	000000			01900
4927					02000
4928					02100
4929	026144				02200
4930					02300
4931					02400
4932	026144	104414			
4933	026146	021364			02600
4934	026150	032777	000020	153234	02700
4935	026156	001772			02800
4936	026160	000207			02900
4937					03000
4938					03100
4939	026162				03200
4940					03300
4941					03400
4942					03500
4943					03600
4944	026162	013637	001246		03700
4945	026166	062746	000002		03800
4946	026172	012761	000026	000004	03900
4947	026200	104414			
4948	026202	122114			04100
4949	026204	004737	026276		04200
4950	026210	012761	000001	000004	04300
4951	026216	104414			
4952	026220	122111			04500
4953	026222	012761	000026	000004	04600
4954	026230	104414			
4955	026232	122110			04800
4956	026234	005337	001246		04900
4957	026240	001361			05000
4958	026242	004737	026276		05100
4959	026246	005061	000004		05200
4960	026252	104414			
4961	026254	122111			05400
4962	026256	012761	000301	000004	05500
4963	026264	104414			
4964	026266	122110			05700
4965	026270	004737	026144		05800
4966	026274	000207			05900
4967					06000
4968					06100
4969	026276				06200
4970					06300
4971					06400
4972	026276	005037	001256		06500
4973	026302				06600
4974	026302	104414			
4975	026304	021224			06800
4976	026306	032777	000020	153076	06900

```

MOV      @DMP04,NITCH      ;STORE LU 17
ROLB    NITCH
ROLB    NITCH
ROLB    NITCH              ;PUT SI IN THE CARRY BIT
RTS     PC
NITCH:  0
OCOR:   ;THIS SUBROUTINE SPINS ON OCOR
ROMCLK  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021364  ;PORT4+LU 17
BIT     #BIT4,@DMP04      ;IS OCOR SET?
BEQ     OCOR              ;BR IF NO
RTS     PC                ;OK OCOR IS SET, GO BACK

SYNC:   ;THIS SUBROUTINE LOADS THE SILO WITH THE NUMBER OF SYNC
        ;CHARACTERS PASSED TO IT IN THE WORD AFTER THE JSR CALL
        ;AND A NON-SYNC CHARACTER (301)
MOV     @ (SP)+,TEMP1     ;GET COUNT
ADD     #,-(SP)          ;ADJUST STACK
MOV     #26,4(R1)        ;LOAD PORT4
ROMCLK  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122114  ;LOAD SYNC REGISTER
1$:     JSR     PC,OUTRDY  ;WAIT FOR OUTRDY
MOV     #1,4(R1)         ;LOAD PORT4
ROMCLK  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122111  ;SET SOM
MOV     #26,4(R1)        ;LOAD PORT4
ROMCLK  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122110  ;LOAD OUT DATA
DEC     TEMP1            ;ALL DONE?
BNE     1$              ;BR IF NOT
JSR     PC,OUTRDY       ;WAIT FOR OUTRDY
CLR     4(R1)           ;LOAD PORT4
ROMCLK  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122111  ;SET SOM
MOV     #301,4(R1)       ;LOAD PORT4
ROMCLK  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122110  ;LOAD OUT DATA
JSR     PC,OCOR         ;WAIT FOR OCOR
RTS     PC

OUTRDY: ;THIS SUBROUTINE SPINS ON OUT READY
1$:     CLR     TEMP5     ;CLEAR TIMER
ROMCLK  ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
021224  ;PORT4+LU11
BIT     #BIT4,@DMP04    ;IS OUT RDY SET?

```

4977	026314	001004		07000	BNE	2\$:BR IF YES
4978	026316	005237	001256	07100	INC	TEMP5		:INC TIMER
4979	026322	001367		07200	BNE	1\$:KEEP CHECKING IF NOT DONE
4980	026324	104036		07300	HLT	36		:ERROR, OUT READY NOT SET
4981	026326	000207		07400	RTS	PC		
4982				07500				
4983				07600				
4984	026330			07700				
4985				07800				
4986				07900				
4987				08000				
4988	026330	013637	001250	08100	MOV	3(SP)+,TEMP2		:GET CHARACTER
4989	026334	062746	000002	08200	ADD	#2,-(SP)		:ADJUST STACK
4990	026340	012737	000003	08300	MOV	#3,TEMP1		:SET FOR 3 SYNC
4991	026346	012761	000026	08400	MOV	#26,4(R1)		:LOAD PORT4
4992	026354	104414			ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4993	026356	122114		08600	122114			:LOAD SYNC REGISTER
4994	026360	004737	026276	08700	JSR	PC,OUTRDY		:WAIT FOR OUTRDY
4995	026364	012761	000001	08800	MOV	#1,4(R1)		:LOAD PORT4
4996	026372	104414			ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
4997	026374	122111		09000	122111			:SET SOM
4998	026376	012761	000026	09100	MOV	#26,4(R1)		:LOAD PORT4
4999	026404	104414			ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5000	026406	122110		09300	122110			:LOAD OUT DATA
5001	026410	005337	001246	09400	DEC	TEMP1		:ALL DONE?
5002	026414	001361		09500	BNE	1\$:BR IF NOT
5003	026416	004737	026276	09600	JSR	PC,OUTRDY		:WAIT FOR OUTRDY
5004	026422	013761	001250	09700	MOV	TEMP2,4(R1)		:LOAD PORT4
5005	026430	104414			ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5006	026432	122110		09900	122110			:LOAD OUT DATA
5007	026434	004737	026144	10000	JSR	PC,OCOR		:WAIT FOR OCOR
5008	026440	000207		10100	RTS	PC		
5009				10200				
5010				10300				
5011	026442			10400				
5012				10500				
5013				10600				
5014	026442	013637	001250	10700	MOV	3(SP)+,TEMP2		:GET CHARACTER
5015	026446	062746	000002	10800	ADD	#2,-(SP)		:ADJUST STACK
5016	026452	004737	026276	10900	JSR	PC,OUTRDY		:WAIT FOR OUTRDY
5017	026456	013761	001250	11000	MOV	TEMP2,4(R1)		:LOAD PORT4
5018	026464	104414			ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5019	026466	122110		11200	122110			:LOAD OUT DATA
5020	026470	004737	026276	11300	JSR	PC,OUTRDY		:WAIT FOR OUTRDY
5021	026474	104414			ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5022	026476	122110		11500	122110			:LOAD GARBAGE CHAR
5023	026500	004737	026144	11600	JSR	PC,OCOR		:WAIT FOR OCOR
5024	026504	000207		11700	RTS	PC		
5025				11800				
5026				11900				
5027	026506			12000				
5028				12100				
5029				12200				
5030				12300				
5031	026506	012737	000073	12400	MOV	#73,TEMP2		:LOAD COUNT
5032	026514	005737	026746	12500	TST	SCHAR		:FIRST TIME HERE?

2\$:

CHAR:

:THIS SUBROUTINE LOADS THE SILO WITH 3 SYNC
 :AND THE CHARACTER PASSED TO IT.

1\$:

CHARSD:

:THIS SUBROUTINE LOADS THE SILO WITH THE CHARACTER PASSED TO IT.

SILOLD:

:THIS SUBROUTINE FILLS THE OUT SILO
 : WITH A BINARY COUNT PATTERN

5033	026520	100470			12600	BMI	4\$:BR IF BITSTUFF
5034	026522	001032			12700	BNE	2\$:BR IF NO
5035	026524	062737	000002	001250	12800	ADD	#2,TEMP2		:ADD 2 TO CHARACTER COUNT
5036	026532	012737	000003	001246	12900	MOV	#3,TEMP1		:SET FOR 3 SYNCs
5037	026540	012761	000026	000004	13000	MOV	#26,4(R1)		:LOAD PORT4
5038	026546	104414				ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5039	026550	122114			13200	122114			:LOAD SYNC REGISTER
5040	026552	004737	026276		13300	1\$:	JSR	PC,OUTRDY	:WAIT FOR OUTRDY
5041	026556	012761	000001	000004	13400	MOV	#1,4(R1)		:LOAD PORT4
5042	026564	104414				ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5043	026566	122111			13600	122111			:SET SOM
5044	026570	012761	000026	000004	13700	MOV	#26,4(R1)		:LOAD PORT4
5045	026576	104414				ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5046	026600	122110			13900	122110			:LOAD OUT DATA
5047	026602	005337	001246		14000	DEC	TEMP1		:ALL DONE?
5048	026606	001361			14100	BNE	1\$:BR IF NOT
5049	026610	004737	026276		14200	2\$:	JSR	PC,OUTRDY	:WAIT FOR OUTRDY
5050	026614	013761	026746	000004	14300	MOV	SCHAR,4(R1)		:LOAD PORT4
5051	026622	104414				ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5052	026624	122110			14500	122110			:LOAD OUT DATA
5053	026626	005737	026750		14600	TST	STUFLG		:BITSTUFF???
5054	026632	001407			14700	BEQ	6\$:BR IF NO
5055	026634	013737	026746	026646	14800	MOV	SCHAR,5\$:IT IS SLD SO CHECK BITSTUFFING
5056	026642	004537	027474		14900	JSR	R5,STFFCL		:ADD ANY BIT STUFF CLOCK TICKS
5057	026646	000000			15000	5\$:	0		:CHARACTER
5058	026650	000010			15100	10			:CHIFT COUNT
5059	026652	005237	026746		15200	6\$:	INC	SCHAR	:NEXT CHARACTER
5060	026656	022737	000400	026746	15300	CMP	#400,SCHAR		:ALL DONE?
5061	026664	001403			15400	BEQ	3\$		
5062	026666	005337	001250		15500	DEC	TEMP2		:DECREMENT COUNT
5063	026672	001346			15600	BNE	2\$:BR IF NOT DONE
5064	026674	004737	026144		15700	3\$:	JSR	PC,OCOR	:WAIT FOR OCOR
5065	026700	000207			15800	RTS	PC		
5066	026702	005037	026746		15900	4\$:	CLR	SCHAR	:START PATTERN AT ZERO
5067	026706	012737	177777	026750	16000	MOV	#-1,STUFLG		:SET BITSTUFF FLAG
5068	026714	005037	027672		16100	CLR	BITCON		:CLEAR STUFF COUNT
5069	026720	062737	000002	001250	16200	ADD	#2,TEMP2		:ADD 2 TO CHARACTER COUNT
5070	026726	012761	000001	000004	16300	MOV	#1,4(R1)		:SET BIT0 IN PORT4
5071	026734	104414				ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5072	026736	122111			16500	122111			:SET SOM!
5073	026740	104414				ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5074	026742	122110			16700	122110			:LOAD GARBAGE CHAR
5075	026744	000721			16800	BR	2\$:GO LOAD SILO
5076	026746	000000			16900	SCHAR:	0		
5077	026750	000000			17000	STUFLG:	0		
5078					17100				
5079					17200				
5080	026752				17300	INRDY:			
5081					17400				:THIS SUBROUTINE SPINS ON INRDY
5082					17500				
5083	026752	005037	001246		17600	1\$:	CLR	TEMP1	
5084	026756				17700				
5085	026756	104414				ROMCLK			:NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5086	026760	021244			17900	021244			:PORT4+LU12
5087	026762	032777	000020	152422	18000	BIT	#BIT4,3DMP04		:IS INRDY SET?
5088	026770	001004			18100	BNE	2\$:BR IF YES

K08

5089	026772	005237	001246	18200		INC	TEMP1	; INC DELAY
5090	026776	001367		18300		BNE	1\$; TRY AGAIN
5091	027000	104037		18400		HLT	37	; ERROR, NO INRDY
5092	027002	000207		18500	2\$:	RTS	PC	; RETURN
5093				18600				
5094				18700				
5095	027004					SIMBCC:		
5096						; THIS SUBROUTINE CALCULATES THE CRC USING POLYNOMIAL GIVEN		
5097						; IN XPOLY. THE CORRECT CRC IS RETURNED IN CALBCC, AND THE		
5098						; STATE OF THE LSB OF THE BCC IS RETURNED IN THE C BIT.		
5099								
5100	027004	010046				MOV	RO, -(SP)	; SAVE RO ON STACK
5101	027006	012537	001246			MOV	(R5)+, TEMP1	; TEMP1 = SHIFT COUNT
5102	027012	012537	001250			MOV	(R5)+, TEMP2	; TEMP2 = CHARACTER
5103	027016	012537	027112			MOV	(R5)+, CALBCC	; CALBCC = OLD BCC
5104	027022	013700	027112		1\$:	MOV	CALBCC, RO	; PUT OLD BCC IN RO
5105	027026	000241				CLC		
5106	027030	006037	027112			ROR	CALBCC	; SHIFT OLD BCC
5107	027034	006037	001250			ROR	TEMP2	; SHIFT CHARACTER
5108	027040	005500				ADC	RO	; ADD CHAR CARRY TO OLD BCC
5109	027042	006000				ROR	RO	; PUT BIT0 TO CARRY BIT
5110	027044	103011				BCC	2\$; CARRY IS FEEDBACK BIT
5111	027046	013700	027110			MOV	XPOLY, RO	; IF FEEDBACK = 1
5112	027052	043700	027112			BIC	CALBCC, RO	; EXCLUSIVLY OR XPOLY TO CALBCC
5113	027056	043737	027110	027112		BIC	XPOLY, CALBCC	
5114	027064	050037	027112			BIS	RO, CALBCC	
5115	027070	005337	001246		2\$:	DEC	TEMP1	; DEC SHIFT COUNT
5116	027074	001352				BNE	1\$; BR IF NOT DONE
5117	027076	013700	027112			MOV	CALBCC, RO	; PUT RESULT IN RO
5118	027102	006000				ROR	RO	; SHIFT BIT0 TO CARRY
5119	027104	012600				MOV	(SP)+, RO	; RESTORE RO
5120	027106	000205				RTS	RS	; RETURN
5121	027110	000000				XPOLY: 0		
5122	027112	000000				CALBCC: 0		
5123		000200				LRC8=200		
5124		120001				CRC16=120001		
5125		102010				CRC.CCITT=102010		
5126								
5127								
5128	027114			19300		BCCLD:		
5129				19400		; THIS SUBROUTINE LOADS THE OUT SILO WITH 2 SYNCs		
5130				19500		; WITH SOM SET, AND ONE CHARACTER PASSED TO IT		
5131				19600		; WITH THE SOM BIT CLEAR (ENABLE CRC)		
5132				19700				
5133	027114	013637	001250	19800		MOV	2(SP)+, TEMP2	; GET CHARACTER
5134	027120	062746	000002	19900		ADD	#2, -(SP)	; ADJUST STACK
5135	027124	012737	000002	20000		MOV	#2, TEMP1	; SET FOR 2 SYNCs
5136	027132	012761	000026	20100		MOV	#26, 4(R1)	; LOAD PORT4
5137	027140	104414				ROMCLK		; NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5138	027142	122114		20300		122114		; LOAD SYNC REGISTER
5139	027144	004737	026276	20400	1\$:	JSR	PC, OUTRDY	; WAIT FOR OUTRDY
5140	027150	012761	000001	20500		MOV	#1, 4(R1)	; LOAD PORT4
5141	027156	104414				ROMCLK		; NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5142	027160	122111		20700		122111		; SET SOM
5143	027162	012761	000026	20800		MOV	#26, 4(R1)	; LOAD PORT4
5144	027170	104414				ROMCLK		; NEXT WORD IS INSTRUCTION, ROMCLK PC=5304

L08

5145	027172	122110		21000		122110	;LOAD OUT DATA
5146	027174	005337	001246	21100		DEC	TEMP1 ;ALL DONE?
5147	027200	001361		21200		BNE	1\$;BR IF NOT
5148	027202	004737	026276	21300		JSR	PC,OUTRDY ;WAIT FOR OUTRDY
5149	027206	013761	001250	21400	000004	MOV	TEMP2,4(R1) ;LOAD PORT4
5150	027214	104414				ROMCLK	122110 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5151	027216	122110		21600		122110	;LOAD OUT DATA
5152	027220	004737	026144	21700		JSR	PC,OCOR ;WAIT FOR OCOR
5153	027224	000207		21800		RTS	PC
5154				21900			
5155				22000			
5156	027226					GETQO:	
5157							;THIS SUBROUTINE READS THE STATE OF THE TRANSMIT
5158							;BCC LSB AND PUTS IT IN THE CARRY BIT
5159							
5160	027226	104414				ROMCLK	;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5161	027230	021364				021364	;PORT4+LU-17
5162	027232	106177	152154			ROLB	@DMP04 ;PUT QO IN CARRY
5163	027236	000207				RTS	PC ;RETURN
5164							
5165							
5166	027240					GETQI:	
5167							;THIS SUBROUTINE READS THE STATE OF THE RECEIVE
5168							;BCC LSB AND PUTS IT IN THE CARRY BIT
5169							
5170	027240	104414				ROMCLK	;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5171	027242	021364				021364	;PORT4+LU-17
5172	027244	106177	152142			ROLB	@DMP04 ;PUT QO IN CARRY
5173	027250	106177	152136			ROLB	@DMP04 ;PUT QI IN CARRY
5174	027254	000207				RTS	PC ;RETURN
5175							
5176							
5177	027256			22300		SYNLD:	
5178				22400			;THIS SUBROUTINE LOADS OUT SILO WITH
5179				22500			;2 SYNC CHARACTERS WITH SOM SET
5180				22600			
5181	027256	012737	000002	22700	001246	MOV	#2,TEMP1 ;LOAD COUNTER FOR 2 SYNC
5182	027264	012761	000026	22800	000004	MOV	#26,4(R1) ;PORT4+26
5183	027272	104414				ROMCLK	122114 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5184	027274	122114		23000		122114	;LOAD SYNC REG
5185	027276	004737	026276	23100	000004	1\$: JSR	PC,OUTRDY ;WAIT FOR OUTRDY
5186	027302	012761	000001	23200		MOV	#1,4(R1) ;LOAD PORT4
5187	027310	104414				ROMCLK	122111 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5188	027312	122111		23400		122111	;SET SOM
5189	027314	012761	000026	23500	000004	MOV	#26,4(R1) ;PORT+26
5190	027322	104414				ROMCLK	122110 ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
5191	027324	122110		23700		122110	;LOAD OUT DATA WITH SYNC
5192	027326	005337	001246	23800		DEC	TEMP1 ;DECREMENT COUNTER
5193	027332	001361		23900		BNE	1\$;BR IF NOT DONE
5194	027334	000207		24000		RTS	PC ;RETURN
5195				24100			
5196				24200			
5197	027336			24300		SOM:	
5198				24400			;THIS SUBROUTINE LOADS SOM AND OUT DATA WITH A
5199				24500			;GARBAGE CHARACTER (0)
5200				24600			

MO8

5201	027336	004737	026276	000000	24700
5202	027342	012761	000001	000004	24800
5203	027350	104414			25000
5204	027352	122111			25100
5205	027354	005061	000004		25300
5206	027360	104414			25400
5207	027362	122110			25500
5208	027364	000207			25600
5209					25700
5210					25800
5211	027366				25900
5212					26000
5213					26100
5214					26200
5215	027366	004737	026276	000004	26400
5216	027372	012761	000002	000004	26600
5217	027400	104414			26700
5218	027402	122111			26800
5219	027404	104414			26900
5220	027406	122110			27000
5221	027410	000207			27100
5222					27200
5223					27300
5224	027412				27400
5225					27500
5226					27600
5227					27700
5228					27800
5229	027412	010046			27900
5230	027414	012500			28100
5231	027416	012537	001246		28200
5232	027422	004737	026276		28300
5233	027426	112061	000004		28400
5234	027432	104414			28500
5235	027434	122110			28600
5236	027436	005337	001246		28700
5237	027442	001367			28800
5238	027444	004737	026144		28900
5239	027450	012600			29000
5240	027452	000205			29100
5241					29200
5242	027454				29300
5243					29500
5244					29700
5245					29800
5246					29900
5247	027454	012761	000200	000004	30000
5248	027462	104414			30100
5249	027464	122112			30200
5250	027466	104414			
5251	027470	122111			
5252	027472	000207			
5253					
5254					
5255	027474				
5256					

EOM:

```

JSR PC,OUTRDY ;WAIT FOR OUTRDY
MOV #1,4(R1) ;PORT4+1
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122111 ;SET SOM
CLR 4(R1) ;CLEAR DATA CHAR
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122110 ;LOAD GARBAGE CHARACTER
RTS PC ;RETURN
  
```

```

;THIS SUBROUTINE LOADS EOM AND OUT DATA WITH A
;GARBAGE CHARACTER (2) TO ENABLE TRANSMISSION OF BCC
  
```

```

JSR PC,OUTRDY ;WAIT FOR OUTRDY
MOV #2,4(R1) ;PORT4+2
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122111 ;SET EOM
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122110 ;LOAD GARBAGE CHARACTER
RTS PC ;RETURN
  
```

MESLD:

```

;THIS SUBROUTINE LOADS SILO WITH MESSAGE
;THE FIRST ARGUMENT IS THE ADDRESS OF THE MESSAGE
;THE SECOND ARGUMENT IS THE NUMBER OF CHARACTERS IN THE MESSAGE
  
```

1\$:

```

MOV RO, -(SP) ;SAVE RO
MOV (R5)+, RO ;RO=MESSAGE POINTER
MOV (R5)+, TEMP1 ;TEMP1=CHARACTER COUNT
JSR PC,OUTRDY ;WAIT FOR OUT RDY
MOV B (R0)+, 4(R1) ;LOAD PORT4 WITH CHARACTER
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122110 ;LOAD OUT DATA SILO
DEC TEMP1 ;DEC CHAR COUNT
BNE 1$ ;BR IF NOT DONE
JSR PC,OCOR ;WAIT FOR OCOR
MOV (SP)+, RO ;RESTORE RO
RTS R5 ;RETURN
  
```

CLR10:

```

;THIS SUBROUTINE SETS IN CLR AND OUT CLR TO
;CLEAR THE TRANSMIT AND RECEIVE BCC REGISTERS
  
```

```

MOV #BIT7,4(R1) ;LOAD PORT4
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122112 ;SET IN CLR!
ROMCLK ;NEXT WORD IS INSTRUCTION, ROMCLK PC=5304
122111 ;SET OUT CLR!
RTS PC ;RETURN
  
```

STFFCL:

```

;THIS SUBROUTINE ADDS ANY NECESSARY BIT STUFF CLOCK TICKS
  
```



```

5257          30300           ;FIRST ARGUMENT IS CHAR, SECOND ARGUMENT IS SHIFT COUNT.
5258          30400
5259 027474 010046          30500      MOV      RO, -(SP)           ;SAVE RO
5260 027476 012500          30600      MOV      (R5)+, RO         ;PUT CHAR IN RO
5261 027500 012537 001252  30700      MOV      (R5)+, TEMP3     ;PUT SHIFT COUNT IN TEMP3
5262 027504 106000          30800 1$:  RORB    RO                 ;LOOK AT NEXT BIT
5263 027506 103403          30900      BCS     2$                ;BR IF A MARK
5264 027510 005037 027672  31000      CLR     BITCON            ;IT WAS A SPACE, CLEAR 1'S COUNTER
5265 027514 000412          31100      BR      3$                ;CONTINUE
5266 027516 005237 027672  31200 2$:  INC     BITCON            ;INC CONSECUTIVE 1'S COUNTER
5267 027522 022737 000005 027672  31300      CMP     #5, BITCON        ;IS IT 5 YET?
5268 027530 001004          31400      BNE     3$                ;BR IF NO
5269 027532 005037 027672  31500      CLR     BITCON            ;YES! SO START AGAIN
5270 027536 104415 000001  31600      DATACLK, 1              ;GIVE EXTRA TICK TO STUFF ZERO
5271 027542 005337 001252  31700 3$:  DEC     TEMP3             ;DEC SHIFT COUNT
5272 027546 001356          31800      BNE     1$                ;BR IF NOT DONE
5273 027550 012600          31900      MOV     (SP)+, RO         ;RESTORE RO
5274 027552 000205          32000      RTS     R5                ;RETURN
5275          32100
5276          32200
5277 027554          32300 STFFCK:
5278          32400           ;THIS SUBROUTINE CHECKS TO SEE IF TRANSMITTER
5279          32500           ;IS STUFFING ZEROS WHEN IT SHOULD. FIRST ARGUMENT
5280          32600           ;IS THE CHARACTER, SECOND ARGUMENT IS SHIFT COUNT.
5281          32700
5282 027554 010046          32800      MOV     RO, -(SP)         ;SAVE RO
5283 027556 012500          32900      MOV     (R5)+, RO         ;PUT CHAR IN RO
5284 027560 012537 001252  33000      MOV     (R5)+, TEMP3     ;PUT SHIFT COUNT IN TEMP3
5285 027564 106000          33100 1$:  RORB    RO                 ;SHIFT OUT NEXT BIT
5286 027566 103403          33200      BCS     2$                ;BR IF IT IS A MARK
5287 027570 005037 027672  33300      CLR     BITCON            ;IT WAS A SPACE, CLEAR 1'S COUNTER
5288 027574 000416          33400      BR      3$                ;CONTINUE
5289 027576 005237 027672  33500 2$:  INC     BITCON            ;INC CONSECUTIVE I'S COUNTER
5290 027602 022737 000005 027672  33600      CMP     #5, BITCON        ;5 IN A ROW YET?
5291 027610 001010          33700      BNE     3$                ;BR IF NO
5292 027612 005037 027672  33800      CLR     BITCON            ;YES, SO START OVER
5293 027616 104415 000001  33900      DATACLK, 1              ;EXTRA TICK TO STUFF ZERO
5294 027622 004737 026112  34000      JSR     PC, GETSI         ;LOOK AT WINDOW
5295 027626 103001          34100      BCC     3$                ;IS IT A ZERO, BR IF YES
5296 027630 104030          34200      HLT     30                ;NO, ERROR ZERO WAS NOT STUFFED
5297 027632 005337 001252  34300 3$:  DEC     TEMP3             ;DEC SHIFT COUNT
5298 027636 001352          34400      BNE     1$                ;BR IF NOT DONE
5299 027640 012600          34500      MOV     (SP)+, RO         ;RESTORE RO
5300 027642 000205          34600      RTS     R5                ;RETURN
5301          34700
5302          34800
5303 027644          34900 CTSPLY:
5304          35000           ;THIS SUBROUTINE WASTES TIME UNTIL CTS SETS,
5305          35100           ;BUT HOPEFULLY NOT SO LONG THAT THE SILO RUNS OUT
5306          35200
5307 027644 010046          35300      MOV     RO, -(SP)         ;SAVE RO
5308 027646 012700 000032  35400      MOV     #32, RO          ;LOAD RO WITH COUNT
5309 027652 027777 151326 151324  35500 1$:  CMP     @TKCSR, @TKCSR    ;WASTE TIME
5310 027660 005300          35600      DEC     RO                 ;DECREMENT COUNTER
5311 027662 001373          35700      BNE     1$                ;DO IT AGAIN IF NOT = 0
5312 027664 012600          35800      MOV     (SP)+, RO         ;RESTORE RO

```

ADDRESS	PC	RTS	PC	OPERATION
027566	000207		35900	;RETURN
000000	000176		36000	
000000	000000		36100	
000000	000000		36200	FLAG: ↑B<0111.110> ;FLAG CHARACTER
000000	000000		36300	BITCON: 0
000000	000000		36400	MESDAT: .BYTE 0,125,252,377
000000	000000		36500	FLTDAT: .BYTE 1,2,4,10,20,40,100,200,376,375,373,367,357,337,277,177
000000	000000		36600	STUFDT: .BYTE 100,140,160,170,3,300,174,176,177,1
000000	000000		36700	.BYTE 363,347,317,200,0,377,377,377,200,37
000000	000000		36800	.EVEN
000000	000000		00300	EM1: .ASCIZ <377>/LINE UNIT INITIALIZATION TEST/
000000	000000		00400	EM2: .ASCIZ <377>/LINE UNIT REGISTER READ/ONLY TEST/
000000	000000		00500	EM3: .ASCIZ <377>/LINE UNIT REGISTER WRITE/READ TEST/
000000	000000		00600	EM4: .ASCIZ <377>/LINE UNIT INTERNAL CLOCK FAILURE/
000000	000000		00700	EM5: .ASCIZ <377>/TRANSMITTER DATA ERROR/
000000	000000		00800	EM6: .ASCIZ <377>/RECEIVER TEST/
000000	000000		00900	EM7: .ASCIZ <377>/RECEIVER DATA ERROR/
000000	000000		01000	EM10: .ASCIZ <377>/MODEM SIGNAL ERROR/
000000	000000		01100	EM11: .ASCIZ <377>/TRANSMITTER CRC ERROR/
000000	000000		01200	EM12: .ASCIZ <377>/RECEIVER CRC ERROR/
000000	000000		01300	EM13: .ASCIZ <377>/IN BCC MATCH ERROR (LU REG 12)/
000000	000000		01400	EM14: .ASCIZ <377>/TRANSMITTER FAILED TO GO TO MARK STATE/
000000	000000		01500	EM15: .ASCIZ <377>/CABLE DATA TEST/
000000	000000		01600	EM16: .ASCIZ <377>/FLAG ERROR/
000000	000000		01700	EM17: .ASCIZ <377>/TRANSMITTER FAILED TO STUFF A ZERO/
000000	000000		01800	EM20: .ASCIZ <377>/SWITCH PAC TEST/
000000	000000		01900	EM21: .ASCIZ <377>/ABORT ERROR/
000000	000000		02000	EM22: .ASCIZ <377>/TRANSMITTER ERROR/
000000	000000		02100	EM23: .ASCIZ <377>/HALF DUPLEX TEST/
000000	000000		02200	EM24: .ASCIZ <377>/OUT READY NOT SET/
000000	000000		02300	EM25: .ASCIZ <377>/IN READY NOT SET/
000000	000000		02400	
000000	000000		02500	DH1: .ASCIZ <377>/EXPECTED FOUND/
000000	000000		02600	DH2: .ASCIZ <377>/EXPECTED FOUND LU-REGISTER/
000000	000000		02700	DH3: .ASCIZ <377>/CHARACTER BIT THAT FAILED/
000000	000000		02800	DH4: .ASCIZ <377>/CORRECT CRC BIT THAT FAILED/
000000	000000		02900	DH5: .ASCIZ <377>/EXPECTED FOUND SHIFT/
000000	000000		03000	DH6: .ASCIZ <377>/EXPECTED FOUND CHARACTER SHIFT/
000000	000000		03100	DH7: .ASCIZ <377>/BLOCK END NOT SET/
000000	000000		03200	DH10: .ASCIZ <377>/RTS DID NOT CLEAR/
000000	000000		03300	.EVEN
000000	000000		03400	
000000	000000		03500	DT1: 2
000000	000000		03600	.BYTE 3,7

031256	001272		03700	SAVR5	
031256.0	003	002	03800	.BYTE	3,2
031256.2	001270		03900	SAVR4	
031256.4	000003		04000	DT2:	3
031256.6	003	007	04100	.BYTE	3,7
031270	001272		04200	SAVR5	
031272	003	010	04300	.BYTE	3,10
031274	001270		04400	SAVR4	
031276	003	002	04500	.BYTE	3,2
031300	001264		04600	SAVR2	
031302	000002		04700	DT3:	2
031304	003	017	04800	.BYTE	3,17
031306	001272		04900	SAVR5	
031310	002	002	05000	.BYTE	2,2
031312	001266		05100	SAVR3	
031314	000002		05200	DT4:	2
031316	006	021	05300	.BYTE	6,21
031320	027112		05400	CALBOC	
031322	002	002	05500	.BYTE	2,2
031324	001266		05600	SAVR3	
031326	000003		05700	DT5:	3
031328	001	011	05800	.BYTE	1,11
031330	001300		05900	ZERO	
031332	001	011	06000	.BYTE	1,11
031334	001302		06100	ONE	
031336	002	002	06200	.BYTE	2,2
031338	001260		06300	SAVR0	
031340	000003		06400	DT6:	3
031342	001	011	06500	.BYTE	1,11
031344	001302		06600	ONE	
031346	001	011	06700	.BYTE	1,11
031348	001300		06800	ZERO	
031350	002	002	06900	.BYTE	2,2
031352	001260		07000	SAVR0	
031354	000004		07100	DT7:	4
031356	001	011	07200	.BYTE	1,11
031358	001300		07300	ZERO	
031360	001	011	07400	.BYTE	1,11
031362	001302		07500	ONE	
031364	003	007	07600	.BYTE	3,7
031366	001272		07700	SAVR5	
031400	002	001	07800	.BYTE	2,1
031402	001266		07900	SAVR3	
031404	000004		08000	DT10:	4
031406	001	011	08100	.BYTE	1,11
031410	001302		08200	ONE	
031412	001	011	08300	.BYTE	1,11
031414	001300		08400	ZERO	
031416	003	007	08500	.BYTE	3,7
031420	001272		08600	SAVR5	
031422	002	001	08700	.BYTE	2,1
031424	001266		08800	SAVR3	
031426	000002		08900	DT11:	2
031430	003	007	09000	.BYTE	3,7
031432	027670		09100	FLAG	
031434	002	002	09200	.BYTE	2,2

031436	001266	
031440	000002	
031442	006	004
031444	027112	
031446	006	002
031450	001252	

09300
09400
09500
09600
09700
09800
09900
10000
10100
10200
10300
10400
10500
10600
10700
10800
10900
11000
11100
11200
11300
11400
11500
11600
11700
11800
11900
12000
12100
12200
12300
12400
12500
12600
12700
12800
12900
13000
13100
13200
13300
13400
13500
13600
13700
13800
13900
14000
14100
14200
14300
14400
14500
14600
14700
14800

SAVR3
DT12:
2
.BYTE 6.4
CALBCC
.BYTE 6.2
TEMP3

031454	000000
031456	000000
031458	000000
031460	000000
031462	027744
031464	030750
031466	031264
031468	030002
031470	030750
031472	031264
031474	030045
031476	030750
031478	031264
031480	030111
031482	000000
031484	000000
031486	030152
031488	030750
031490	031264
031492	030152
031494	031006
031496	031302
031498	030203
031500	030727
031502	031264
031504	030222
031506	030727
031508	031264
031510	030247
031512	030727
031514	031264
031516	030272
031518	031104
031520	031326
031522	030322
031524	031104
031526	031226
031528	030272
031530	031044
031532	031214
031534	030346
031536	000000
031538	000000
031540	030272
031542	031104
031544	031344
031546	030322
031548	031104
031550	031344

.ERRTAB:
0
0
0
FM1
DH2
:HLT 1
DT2
FM2
DH2
:HLT 2
DT2
FM3
DH2
:HLT 3
DT2
FM4
:HLT 4
FM5
DH2
:HLT 5
DT2
FM5
:HLT 6
DH3
DT3
FM6
:HLT 7
DH1
DT1
FM7
:HLT 10
DH1
DT1
FM10
:HLT 11
DH1
DT1
FM11
:HLT 12
DH5
DT5
FM12
:HLT 13
DH5
DT5
FM11
:HLT 14
DH4
DT4
FM13
:HLT 15
0
0
FM11
:HLT 16
DH5
DT6
FM12
:HLT 17
DH5
DT6



031612	030273	14900	EM11		
031614	031136	15000	DH6	:HLT	20
031616	031266	15100	DT7		
031618	030273	15200	EM11		
031620	031136	15300	DH6	:HLT	21
031622	031404	15400	DT10		
031624	030222	15500	EM12		
031626	031136	15600	DH6	:HLT	22
031628	031266	15700	DT7		
031630	030222	15800	EM12		
031632	031136	15900	DH6	:HLT	23
031634	031404	16000	DT10		
031636	030406	16100	EM14		
031638	000000	16200	0	:HLT	24
031640	000000	16300	0		
031642	030456	16400	EM15		
031644	030727	16500	DH1	:HLT	25
031646	031252	16600	DT1		
031648	030477	16700	EM16		
031650	031006	16800	DH3	:HLT	26
031652	031426	16900	DT11		
031654	030322	17000	EM12		
031656	030727	17100	DH1	:HLT	27
031658	031440	17200	DT12		
031660	030513	17300	EM17		
031662	000000	17400	0	:HLT	30
031664	000000	17500	0		
031666	030557	17600	EM20		
031668	030727	17700	DH1	:HLT	31
031670	031252	17800	DT1		
031672	030500	17900	EM21		
031674	031204	18000	DH7	:HLT	32
031676	000000	18100	0		
031678	030600	18200	EM21		
031680	031006	18300	DH3	:HLT	33
031682	031302	18400	DT3		
031684	030615	18500	EM22		
031686	031227	18600	DH10	:HLT	34
031688	000000	18700	0		
031690	030640	18800	EM23		
031692	030750	18900	DH2	:HLT	35
031694	031264	19000	DT2		
031696	030662	19100	EM24		
031698	000000	19200	0	:HLT	36
031700	000000	19300	0		
031702	030705	19400	EM25		
031704	000000	19500	0	:HLT	37
031706	000000	19600	0		
031708	030203	19700	EM6		
031710	030750	19800	DH2	:HLT	40
031712	031264	19900	DT2		
031714	030153	20000	EM5		
031716	031104	200100	DH5	:HLT	41
031718	031326	200200	DT5		
031720	030346	200300	EM13		
031722	030727	200400	DH1	:HLT	42

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 110
DZDMEA P11 SUBROUTINES

031772 031252

20500 D71

031774 000001

20600
20700
20800 CORMAX:
21300 .END

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 114
DZDMEA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

DMS102	001522	884#
DMS103	001532	889#
DMS104	001542	894#
DMS105	001552	899#
DMS106	001562	904#
DMS107	001572	909#
DMS110	001602	914#
DMS111	001612	919#
DMS112	001622	924#
DMS113	001632	929#
DMS114	001642	934#
DMS115	001652	939#
DMS116	001662	944#
DMS117	001672	949#
DMS200	001504	875#
DMS201	001514	880#
DMS202	001524	885#
DMS203	001534	890#
DMS204	001544	895#
DMS205	001554	900#
DMS206	001564	905#
DMS207	001574	910#
DMS210	001604	915#
DMS211	001614	920#
DMS212	001624	925#
DMS213	001634	930#
DMS214	001644	935#
DMS215	001654	940#
DMS216	001664	945#
DMS217	001674	950#
DMS300	001506	876#
DMS301	001516	881#
DMS302	001526	886#
DMS303	001536	891#
DMS304	001546	896#
DMS305	001556	901#
DMS306	001566	906#
DMS307	001576	911#
DMS310	001606	916#
DMS311	001616	921#
DMS312	001626	926#
DMS313	001636	931#
DMS314	001646	936#
DMS315	001656	941#
DMS316	001666	946#
DMS317	001676	951#
DMTLVL	001402	860#
DMTVEC	001400	859#
DM.END	001700	953#
DM.MAP	001500	793#
DT1	031252	5335#
DT10	031404	5335#
DT11	031426	5335#
DT12	031440	5335#
DT2	031264	5335#
DT3	031302	5335#

1890*	1891*								
1888*	1889*	1890							
1942									
872#	1078	1131	1141	1235	1860	1862	1940	1945	2157

DT4	031314	5335#																		
DT5	031326	5335#																		
DT6	031344	5335#																		
DT7	031362	5335#																		
EM1	027744	5335#																		
EM10	030247	5335#																		
EM11	030273	5335#																		
EM12	030322	5335#																		
EM13	030346	5335#																		
EM14	030406	5335#																		
EM15	030456	5335#																		
EM16	030477	5335#																		
EM17	030513	5335#																		
EM2	030002	5335#																		
EM20	030557	5335#																		
EM21	030600	5335#																		
EM22	030615	5335#																		
EM23	030640	5335#																		
EM24	030662	5335#																		
EM25	030705	5335#																		
EM3	030045	5335#																		
EM4	030111	5335#																		
EM5	030153	5335#																		
EM6	030203	5335#																		
EM7	030222	5335#																		
EM	027366	3980	4076	4141	4145	4388	4393	4772	4776	4780	4866	5211#								
ERCT00	001704	960#																		
ERCT01	001710	963#																		
ERCT02	001714	966#																		
ERCT03	001720	969#																		
ERCT04	001724	972#																		
ERCT05	001730	975#																		
ERCT06	001734	978#																		
ERCT07	001740	981#																		
ERCT10	001744	984#																		
ERCT11	001750	987#																		
ERCT12	001754	990#																		
ERCT13	001760	993#																		
ERCT14	001764	996#																		
ERCT15	001770	999#																		
ERCT16	001774	1002#																		
ERCT17	002000	1005#																		
ERR	002612	1182	1192#	1196																
ERRCNT	001232	761#	1320	1347	1659*	1875*														
ERRFLG	001325	800#	1076*	1306*	1368*	1609*	1622	1636*	1694*											
ERRMSG	005100	1619*	1637	1640#																
ERRPC	002702	1198	1214#																	
ERTAB0	005224	1634	1668#																	
EXIT =	000205	694#																		
EXITER	005160	1654	1659#																	
FLAG	027670	5316#	5335																	
FLOAT	002536	1175#	1181																	
FLTDAT	027700	4760	4770	4774	4778	4785	4825	5320#												
FY	002562	1183#	1187																	
GETQ1	027240	3581	3585	3656	3660	3731	3735	3806	3810	3927	3931	5166#								
GETQ0	027226	3552	3556	3627	3631	3702	3706	3777	3781	3961	3965	5156#								



K09

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 116
 DZDMEAR.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

GETSI	026112	2754	2758	2805	2809	2856	2860	2907	2911	2966	2970	4000	4004	4021
		4025	4037	4046	4165	4169	4186	4190	4216	4220	4237	4241	4253	4262
		4413	4417	4434	4438	4454	4458	4483	4487	4504	4508	4520	4529	4915#
		5294												
HALTS	005130	1605	1651#											
HILIM	004274	1448*	1475	1493#										
ICOUNT	001222	757#	1366	1371*										
INBUF	007322	1418	1454	1773#										
INCHAR	007666	1801	1829#											
INIFLG	001324	799#	1102	1122	1129*									
INRDY	026752	3317	3361	3398	3406	3414	4079	4092	4097	4275	4289	4294	4318	4332
		4337	4542	4556	4561	4582	4596	4610	4615	4671	4681	4689	4786	4802
		4807	4871	4886	4891	5080#								
INSTR=	104404	821#	1469											
INSTR =	104403	819#	1899	1951	1964	1973	2040	2049						
INSTR2	004074	1425	1437#											
INTTY	012042	1984	2001	2012	2028	2207#								
KMCM	007264	1211	1758#											
LIMITS	004222	1464	1475#											
LINE	006742	1758#	2041											
LOBITS	004300	1450*	1479	1495#	1496									
LOCK	001220	756#	1370*	1386	1388	1628	2328*	2344*	2370*	2386*	2412*	2435*	2470*	2490*
		3533*	3565*	3608*	3640*	3683*	3715*	3758*	3790*					
LOKFLG	001326	801#												
LOLIM	004272	1447*	1477	1492#										
LPCNT	001224	758#	1365*	1366	1369*									
LRC8 =	000200	5123#												
LSTERR	001234	762#	1085*	1305*	1606	1608*	1695*							
LUTYPE=	***** U	599	5095											
MASKX	001244	770#												
MASTEK	006066	1630	1758#											
MCRLF	005574	1403	1526	1626	1627	1635	1758#	1898	1960					
MCSRX	006016	1310	1758#											
MDATA	007426	1556	1566	1777#										
MEMLIM	001304	786#	1273*											
MEPASS	005635	1309	1758#											
MERRPC	006143	1633	1758#											
MERRX	006043	1316	1758#											
MERR2	005662	1758#	2142											
MERR3	005731	1246	1758#											
MESDAT	027674	3501	3972	3978	4069	4074	4133	4139	4143	4199	4274	4317	4380	4386
		4391	4466	4541	4595	4651	4656	4661	4721	4854	4864	4870	5318#	
MESLD	027412	3500	3977	4073	4138	4142	4385	4390	4655	4660	4720	4769	4773	4777
		4863	5224#											
MILK	001322	794#	1079*	1318	1858*	1863*	1867							
MLOCK	005767	1287	1758#											
MNEW	006070	1241	1758#											
MODU	006630	1758#	2011											
MPASSX	006032	1314	1758#											
MPFAIL	005577	1692	1758#											
MOM	005570	1433	1758#	1922	1997	2007	2020	2034						
MR	005657	1296	1758#	1916										
MRESET=	004000	694#												
MSTCLR=	104412	833#	1697	2277	2301	2330	2372	2414	2472	2523	2545	2567	2601	2634
		2678	2736	2787	2838	2889	2941	2999	3027	3055	3083	3112	3160	3197
		3234	3271	3308	3349	3393	3437	3487	3535	3610	3685	3760	3833	3899

MTITLE	001000	3967	4067	4128	4375	4649	4712	4751	4845											
MTSTN	006054	734#	1106																	
MTSTPC	005755	1631	1758#	1900																
MVECX	006024	1758#																		
NEXT	001216	1312	1758#																	
		755#	1372	1664	2228*	2252*	2275*	2299*	2327*	2369*	2411*	2469*	2521*	2543*						
		2565*	2599*	2632*	2676*	2734*	2785*	2836*	2887*	2939*	2997*	3025*	3053*	3081*						
		3110*	3158*	3195*	3232*	3269*	3306*	3347*	3391*	3435*	3485*	3532*	3607*	3682*						
		3757*	3831*	3897*	3965*	4065*	4126*	4373*	4647*	4710*	4749*	4843*								
NITCH	026142	4921*	4922*	4923*	4924*	4926#														
NOACT	007100	1118	1758#	1852																
NODEV	002606	1167	1190#																	
NUM	006374	1758#	1952																	
OCOR	026144	2641	2685	2747	2798	2849	2900	2959	3852	3918	3981	4077	4146	4394						
		4658	4663	4723	4781	4867	4929#	4935	4965	5007	5023	5064	5152	5238						
OK	002600	1163	1166	1185	1188#	1213														
ONE	001302	785#	5335																	
OUTRDY	026276	2738	2743	2789	2794	2840	2845	2891	2896	2946	2950	2955	2978	3976						
		4072	4137	4384	4654	4719	4949	4958	4969#	4994	5003	5016	5020	5040						
		5049	5139	5148	5185	5201	5215	5232												
PACT00	001702	959#																		
PACT01	001706	962#																		
PACT02	001712	965#																		
PACT03	001716	968#																		
PACT04	001722	971#																		
PACT05	001726	974#																		
PACT06	001732	977#																		
PACT07	001736	980#																		
PACT10	001742	983#																		
PACT11	001746	986#																		
PACT12	001752	989#																		
PACT13	001756	992#																		
PACT14	001762	995#																		
PACT15	001766	998#																		
PACT16	001772	1001#																		
PACT17	001776	1004#																		
PARAM =	104405	823#	1901	1953	1966	1975	2042	2051												
PARAM1	004142	1453#	1470																	
PARBIT=	040000	694#																		
PARERR	004216	1456	1458	1460	1469#	1476	1478	1480												
PASCNT	001230	760#	1307*	1308	1319	1344	1874*													
PC =%	000007	656#	1107*	1121*	1329*	1352*	1383*	1598*	1801*	1825*	1836*	1909	1909*	1984*						
		2001*	2012*	2028*	2197*	2214*	2641*	2685*	2738*	2743*	2747*	2754*	2758*	2789*						
		2794*	2798*	2805*	2809*	2840*	2845*	2849*	2856*	2860*	2891*	2896*	2900*	2907*						
		2911*	2946*	2950*	2955*	2959*	2966*	2970*	2978*	3002*	3030*	3058*	3086*	3115*						
		3163*	3200*	3237*	3274*	3314*	3317*	3330*	3358*	3361*	3374*	3395*	3399*	3406*						
		3414*	3537*	3542*	3552*	3556*	3566*	3571*	3581*	3585*	3612*	3617*	3627*	3631*						
		3641*	3646*	3656*	3660*	3687*	3692*	3702*	3706*	3716*	3721*	3731*	3735*	3762*						
		3767*	3777*	3781*	3791*	3796*	3806*	3810*	3840*	3852*	3861*	3865*	3906*	3918*						
		3927*	3931*	3975*	3976*	3980*	3981*	4000*	4004*	4021*	4025*	4037*	4046*	4071*						
		4072*	4076*	4077*	4079*	4092*	4097*	4136*	4137*	4141*	4145*	4146*	4165*	4169*						
		4186*	4190*	4216*	4220*	4237*	4241*	4253*	4262*	4275*	4289*	4294*	4318*	4332*						
		4337*	4383*	4384*	4388*	4389*	4393*	4394*	4413*	4417*	4434*	4438*	4454*	4458*						
		4483*	4487*	4504*	4508*	4520*	4529*	4542*	4556*	4561*	4582*	4596*	4610*	4615*						
		4653*	4654*	4658*	4663*	4671*	4681*	4689*	4718*	4719*	4723*	4755*	4756*	4772*						
		4776*	4780*	4781*	4786*	4802*	4807*	4849*	4850*	4866*	4867*	4871*	4886*	4891*						

		2790*	2795*	2819	2839*	2841*	2846*	2870	2890*	2892*	2897*	2921	2942*	2947*
		2951*	2956*	2979*	3000*	3007	3028*	3035	3056*	3063	3084*	3091	3114*	3121
		3128*	3133	3140	3162*	3169	3178	3199*	3206	3215	3236*	3243	3252	3273*
		3280	3289	3313*	3320	3354*	3355*	3364	3394*	3401	3409	3417	3442*	3443*
		3449	3458*	3464	3492*	3493*	3497*	3504*	3508	3536*	3611*	3686*	3761*	3834*
		3841*	3845*	3849*	3879*	3900*	3907*	3911*	3915*	3945*	3571*	4068*	4082	4095
		4100	4132*	4278	4292	4297	4306	4321	4335	4340	4349	4379*	4545	4559
		4564	4573	4585	4599	4613	4618	4627	4650*	4667	4674	4684	4692	4714*
		4715*	4727	4754*	4782*	4789	4805	4810	4819	4848*	4868*	4874	4889	4894
		4903	4946*	4950*	4953*	4959*	4962*	4991*	4995*	4998*	5004*	5017*	5037*	5041*
		5044*	5050*	5070*	5136*	5140*	5143*	5149*	5182*	5186*	5189*	5202*	5205*	5216*
		5233*	5247*											
R2	=%000002	651#	1173*	1176	1190	1508	1517*	1791	1795*	1812*	1813*	1814*	1815*	1820
		1824*	1857*	1874	1875	1940*	1941*	1942	1945*	1972*	1981*	1994*	2009*	2022*
		2023	2025*	2036*	2038*	2048*	2057*	2058	2076*	2087*	2096*	2100*	2111*	2112
		2115*	2118*	2119	2157*	2170	2181*	2183*	2185*	2186*	2193*	2231*	2254*	2278*
		2302*	2331*	2373*	2415*	2473*	2609*	2643*	2653*	2691*	2701*	2711*	2749*	2752*
		2900*	2803*	2851*	2854*	2902*	2905*	2962*	2964*	3001*	3029*	3057*	3085*	3113*
		3161*	3198*	3235*	3272*	3311*	3321	3325*	3326	3352*	3365	3369*	3370	3985*
		3998*	4069*	4083	4150*	4163*	4202*	4214*	4274*	4279	4317*	4322	4398*	4411*
		4450*	4452*	4469*	4481*	4541*	4546	4595*	4600	4651*	4675	4713*	4760*	4761
		4785*	4790	4825*	4854*	4855	4870*	4875						
R3	=%000003	652#	1170*	1172*	1180	1190	1412	1419*	1429*	1432*	1434	1438*	1507	1518*
		1529	1544*	1545*	1546*	1547	1556*	1557*	1562*	1565*	1571*	1792	1802	1804
		1806	1808	1811*	1815	1823*	1831*	1834	1835*	1985	1987	1990*	1993*	1994
		2002	2004	2013	2015	2017	2029	2031	2209*	2212	2213*	2748*	2762*	2763
		2799*	2813*	2814	2850*	2864*	2865	2901*	2915*	2916	2943*	2961*	2974*	2975
		3312*	3328*	3331*	3353*	3372*	3375*	3835*	3872*	3873	3875*	3901*	3938*	3939
		3941*	3982*	4008*	4009	4011*	4029*	4030	4032*	4040*	4041	4044*	4049*	4050
		4147*	4173*	4174	4176*	4194*	4195	4197*	4224*	4225	4227*	4245*	4246	4248*
		4256*	4257	4260*	4265*	4266	4273*	4283*	4316*	4326*	4395*	4421*	4422	4424*
		4442*	4443	4445*	4461*	4462	4464*	4491*	4492	4494*	4512*	4513	4515*	4523*
		4524	4527*	4532*	4533	4540*	4550*	4594*	4604*	4759*	4767*	4784*	4795*	4797*
		4853*	4861*											
R4	=%000004	653#	1131*	1132	1133	1135	1136	1137	1176*	1177	1179	1413	1418*	1422*
		1423*	1424	1431*	1435	1437*	1445	1454*	1455	1457	1459	1461*	1462	1463
		1484*	1495*	1489*	1506	1519*	1530	1541*	1544	1549*	1551*	1553*	1570*	1613*
		1614*	1615*	1616*	1617*	1618*	1619	1620	1621	1793	1796*	1810*	1818	1822*
		1989*	1991*	2168*	2169	2177*	2179	2187	2234*	2235*	2237	2257*	2258*	2260
		2281*	2282*	2284	2305*	2306*	2310*	2312	2338*	2339*	2340	2351*	2352*	2353
		2380*	2381*	2382	2393*	2394*	2395	2426*	2427*	2428	2447*	2448*	2449	2482*
		2483	2500*	2501	2526*	2528	2548*	2550	2612*	2613*	2615	2646*	2647*	2649
		2656*	2657*	2659	2694*	2695*	2697	2704*	2705*	2707	2714*	2716	2944*	2951
		2954*	2956	2977*	2979	3007*	3008*	3010	3035*	3036*	3038	3063*	3064*	3066
		3091*	3092*	3094	3121*	3122*	3124	3133*	3134*	3136	3140*	3141*	3143	3169*
		3170*	3172	3178*	3180	3206*	3207*	3209	3215*	3217	3243*	3244*	3246	3252*
		3254	3280*	3281*	3283	3289*	3291	3320*	3322	3364*	3366	3401*	3403	3409*
		3411	3417*	3419	3449*	3450*	3455	3464*	3465*	3470	3508*	3509*	3514	3836*
		3841	3844*	3845	3848*	3849	3876*	3877	3879	3902*	3907	3910*	3911	3914*
		3915	3942*	3943	3945	3972*	3984	4082*	4084	4100*	4101*	4103	4133*	4149
		4199*	4201	4278*	4280	4297*	4298*	4300	4321*	4323	4340*	4341*	4343	4360*
		4397	4466*	4468	4545*	4547	4564*	4565*	4567	4585*	4587	4599*	4601	4618*
		4619*	4621	4674*	4676	4684*	4686	4692*	4694	4727*	4728*	4730	4789*	4791
		4810*	4811*	4813	4874*	4876	4894*	4895*	4897					
R5	=%000005	654#	1157*	1160	1168	1205	1232	1234*	1394	1395*	1401	1406	1408*	1444
		1446*	1447	1448	1449	1450	1451	1452	1453*	1462*	1465*	1466*	1467*	1475

DD FORM 13 SEP 76 13:18 PAGE 120
MACY11 27 (733) CROSS REFERENCE TABLE -- USER SYMBOLS
DD FORM 1 P11

Table with 12 columns of data, including numerical values, symbols, and alphanumeric strings. The columns contain various alphanumeric characters and numbers, likely representing a cross-reference table for user symbols. The data is organized in a grid-like structure with multiple rows and columns.

OMEND	13														
OMFRNT	13														
FLT	13														
ABORT	13														
AUTO	1142														
BCC	3595	3670	3745												
BTNCR	3886														
BTNWI	1770														
BUFTTE	4734	4829													
CDATA	2121	3133	3140	3169	3178	3206	3215	3242	3252	3280	3289	3401	3409		
CLOCK	3464	4100	4297	4340	4554	4618	4810	4894							
COMP	3566	3612	3641	3687	3716	3762	3791								
CRC	3900														
CRCSH	1841														
CYCLE	4634														
EMPTY	1398														
POP	4910														
INI	2416	2436	2474	2491											
LAG	4599														
FLOAT	4599														
GETPA	2986	3014	3042	3070											
HALF	2400	2458													
HEADF	2217	2241	2264	2288											
INACT	2316														
INIT	2358														
LINE1	3378														
LU1	4089	4285	4328	4552	4606	4798	4892								
LU12															
LU17															
MARHI															
MARK															
MATCH															
MOCK															
MODEM															
MSG	3423														
MULT	1758														
PATTE	3295	3334													
PFAIL	1674														
QOQT	5156	5166													
QUEST	1951	1964	1973	2040	2049										
RAMCL	1702														
RCLK	1705	1708	1745	1750	2232	2255	2279	2303	2333	2335	2346	2348	2375	2377	
	2388	2390	2420	2422	2441	2443	2477	2479	2495	2497	2524	2546	2570	2575	2577
	2606	2610	2637	2639	2644	2654	2681	2683	2688	2692	2702	2712	2740	2745	2766
	2791	2796	2817	2842	2847	2868	2893	2898	2919	2948	2952	2957	2980	2985	2993
	3061	3089	3119	3129	3131	3167	3176	3204	3213	3241	3250	3278	3287	3318	3325

H10

DZDME MACY11 27(732) 22-SEP-76 13:18 PAGE 127
DZDME.P11 CROSS REFERENCE TABLE -- MACRO NAMES

	3399	3407	3415	3444	3447	3459	3462	3494	3498	3505	3842	3846	3850	3880
	3316	3316	3346	4080	4093	4098	4107	4276	4290	4295	4304	4319	4333	4338
	4557	4557	4562	4571	4583	4597	4611	4616	4625	4665	4672	4682	4690	4716
	4787	4803	4808	4817	4872	4887	4892	4901	4919	4932	4947	4951	4954	4960
	4974	4992	4996	4999	5005	5018	5021	5038	5042	5045	5051	5071	5073	5085
	5141	5144	5150	5160	5170	5183	5187	5190	5203	5206	5217	5219	5234	5248
STORC	405													
STORC	208	3147	3184	3221	3258									
STORC	104													
STORC	509													
STORC	1780													
STORC	2510	2532												
STORC	3971	4068	4132	4379										
STORC	3952	4110	4356											
STRANW	3990	4155	4206	4403	4473									
STRANI	2587	2620	2664											
STRAPDE	813	815	817	819	821	823	825	827	829	831	833	835	837	839
STSTN	2225	2249	2272	2296	2324	2366	2408	2466	2518	2540	2562	2596	2629	2673
	2273	2283	2284	2296	2324	2366	2408	2466	2518	2540	2562	2596	2629	2673
	4707	4746	4840	4829	3604	3679	3754	3828	3894	3962	4062	4123	4170	4644
SVARIA	732													
SWINDO	2217	2272	2283	2296	2324	2366	2408	2466	2518	2540	2562	2596	2629	2673
XZ	2217	2272	2283	2296	2324	2366	2408	2466	2518	2540	2562	2596	2629	2673
	2273	2283	2284	2296	2324	2366	2408	2466	2518	2540	2562	2596	2629	2673
	4707	4746	4840	4829	3604	3679	3754	3828	3894	3962	4062	4123	4170	4644
SZEROS	2217	2272	2283	2296	2324	2366	2408	2466	2518	2540	2562	2596	2629	2673
	2273	2283	2284	2296	2324	2366	2408	2466	2518	2540	2562	2596	2629	2673
	4707	4746	4840	4829	3604	3679	3754	3828	3894	3962	4062	4123	4170	4644

ADC	1857	5108													
ADD	1177	1186	1188	1396	1415	1486	1533	1546	1591	1615	1618	1722	1732	1743	1858
ASL	1859	1892	1884	1897	1889	1891	2126	2183	4945	4989	5015	5035	5069	5134	
BCC	1465	1465	1467	1569	1614	1616	1812	1813	1814	1990	2188	2189	2190	2191	
	2167	2753	2759	2804	2810	2855	2861	2906	2912	2965	2971	3553	3582	3628	3657
	3703	3732	3778	3807	3862	3928	3999	4005	4020	4026	4164	4170	4185	4191	4215
	4221	4236	4242	4412	4418	4433	4439	4453	4459	4482	4488	4503	4509	5110	5295
BCS	2755	2806	2857	2908	2967	3551	3557	3580	3586	3626	3632	3655	3661	3701	3707
	3730	3736	3776	3782	3805	3811	3860	3866	3926	3932	4001	4022	4038	4047	4166
	4187	4217	4238	4254	4263	4414	4435	4455	4484	4505	4521	4530	5263	5286	
BEQ	1095	1105	1110	1123	1127	1134	1159	1166	1185	1240	1285	1326	1355	1364	1385
	1387	1425	1456	1464	1561	1600	1607	1623	1629	1638	1643	1647	1652	1663	1711
	1724	1748	1788	1803	1805	1807	1809	1865	1894	2003	2005	2014	2016	2018	2030
	2032	2083	2099	2110	2123	2131	2178	2238	2261	2285	2309	2313	2341	2354	2383
	2396	2429	2450	2484	2502	2529	2551	2582	2616	2650	2660	2698	2708	2717	2769
	2808	2871	2922	3011	3039	3067	3095	3125	3137	3144	3173	3181	3210	3218	3247
	3296	3284	3292	3323	3327	3367	3371	3404	3412	3420	3441	3453	3456	3468	3471
	3491	3512	3515	4085	4104	4281	4301	4310	4324	4344	4353	4548	4568	4577	4588
	4602	4622	4631	4668	4677	4687	4695	4731	4753	4792	4814	4823	4847	4877	4898
	4907	4935	5054	5061											
BGT	1460														
BHI	1476	1986													
BIC	1179	1540	1545	1590	1617	1811	1835	1870	1917	1993	2038	2186	2192	2213	2235
	2258	2282	2306	2310	2311	2339	2352	2381	2394	2419	2425	2427	2440	2446	2448
	2513	2647	2657	2695	2705	3008	3036	3064	3092	3122	3134	3141	3170	3207	3244
	3281	3450	3454	3465	3509	3513	4096	4101	4293	4298	4307	4336	4341	4350	4560
	4565	4574	4614	4619	4628	4728	4806	4811	4820	4890	4895	4904	5112	5113	
BICB	1422	1461	1716	1727	1735										
BIS	1815	1994	2009	2022	2025	2036	2073	2076	2091	2093	2096	2100	2106	2108	2111
	2181	2185	2193	3469	4308	4351	4575	4629	4821	4905	5114				
BISB	1462	1715	1720	1726	1733	2115	2118								
BIT	1112	1126	1162	1165	1239	1285	1294	1354	1361	1384	1399	1599	1604	1660	1662
	1710	1723	1747	1752	1864	1893	1947	2098	2102	2109	2153	2308	2572	2581	2768
	2819	2870	2921	3438	3440	3452	3467	3488	3490	3511	4667	4752	4846	4934	4976
	5087														
BITB	1479														
BLE	3878	3944													
BLO	1478	1988													
BLOS	1245	1367	2164												
BLT	1458														
BMT	1125	5033													
BNE	1084	1103	1113	1117	1161	1163	1169	1181	1193	1206	1233	1258	1271	1281	1295
	1322	1362	1398	1400	1407	1430	1480	1488	1529	1555	1559	1564	1568	1605	1625
	1651	1691	1737	1753	1755	1786	1790	1819	1851	1861	1897	1910	1912	1914	1921
	1942	1948	1992	2060	2066	2069	2075	2085	2095	2103	2128	2138	2154	2176	2434
	2456	2489	2508	2573	2575	2584	2764	2815	2866	2917	2976	2984	3029	3273	3439
	3489	3517	3563	3592	3638	3667	3713	3742	3788	3817	3874	3884	3940	3950	4010
	4013	4031	4042	4051	4088	4175	4178	4196	4226	4229	4247	4258	4267	4284	4327
	4423	4426	4444	4463	4493	4496	4514	4525	4534	4551	4505	4680	4769	4796	4827
	4862	4881	4957	4977	4979	5002	5034	5048	5063	5088	5090	5116	5147	5193	5237
	5268	5272	5291	5298	5311										
BPL	1115	1358	1402	1405	1421	1427	1602	1654	1830	1833	2208	2211			
BR	1096	1120	1128	1140	1164	1171	1187	1189	1209	1213	1236	1248	1276	1283	1290
	1356	1360	1436	1468	1470	1681	1816	1954	1918	1923	1995	1998	2008	2021	2024
	2036	2037	2077	2097	2101	2145	2180	2184	2757	2809	2859	2910	2969	3022	3076
	3555	3584	3630	3659	3705	3734	3780	3809	3864	3930	4003	4024	4168	4189	4219

CLC	4240	4416	4437	4457	4486	4507	5075	5265	5288									
CLR	1548	1550	1552	1855	2134	2432	2454	2487	2506	5105								
	1075	1082	1085	1111	1155	1156	1157	1242	1255	1266	1268	1305	1369	1370	1453			
	1689	1695	1696	1795	1810	1821	1941	1944	1946	2064	2067	2071	2089	2120	2124			
	2125	2129	2143	2182	2230	2259	2345	2350	2387	2392	2568	2577	2696	2706	2748			
	2799	2850	2901	2943	2944	2945	2961	3009	3065	3135	3142	3179	3309	3310	3311			
	3350	3351	3352	3354	3442	3458	3466	3504	3538	3541	3567	3570	3613	3616	3642			
	3645	3688	3691	3717	3720	3763	3766	3792	3795	3835	3836	3837	3838	3875	3901			
	3902	3903	3904	3941	3973	3982	4011	4032	4044	4134	4147	4176	4197	4200	4227			
	4248	4260	4381	4395	4424	4445	4449	4464	4467	4494	4515	4527	4586	4729	4758			
CLRB	4782	4852	4868	4959	4972	5066	5068	5083	5205	5264	5269	5287	5292					
CMP	1076	1077	1306	1368	1565	1609	1694											
	1083	1094	1097	1104	1184	1212	1244	1256	1257	1270	1277	1280	1366	1475	1477			
	1606	1651	1734	1785	1787	1789	1802	1804	1806	1808	1860	1908	1911	1913	1920			
	1942	1985	1987	2002	2004	2013	2015	2017	2023	2029	2031	2074	2082	2084	2122			
	2127	2162	2163	2194	2763	2814	2865	2916	2975	2983	3326	3370	3562	3591	3637			
	3666	3712	3741	3787	3816	3873	3877	3883	3939	3943	3949	4009	4030	4041	4050			
	4174	4195	4225	4246	4257	4266	4309	4352	4422	4443	4462	4492	4513	4524	4533			
CMPB	4576	4630	4822	4906	5060	5267	5290	5309										
	1190	1424	1455	1457	1459	1463	1538	2094	2237	2260	2284	2312	2340	2353	2382			
	2395	2428	2449	2483	2501	2528	2550	2615	2649	2659	2697	2707	2716	3010	3038			
	3066	3094	3124	3136	3143	3172	3180	3209	3217	3246	3254	3283	3291	3322	3366			
	3403	3411	3419	3455	3470	3514	4084	4103	4280	4300	4323	4343	4547	4567	4587			
	4601	4621	4676	4686	4694	4730	4791	4813	4876	4897								
COM	1129	2437	2453	2492	2505													
DEC	1321	1429	1554	1567	1736	1754	1991	2059	2137	3328	3372	3516	4012	4087	4177			
	4228	4283	4326	4425	4495	4550	4604	4679	4767	4795	4826	4861	4880	4956	5001			
	5047	5062	5115	5146	5192	5236	5271	5297	5310									
DECB	1487	1558	1563															
EMT	673																	
HALT	707	1119	1243	1247	1251	1657	1680	1725	1853	2144								
INC	1178	1307	1365	1659	1690	1878	1880	1963	2121	2136	2574	2583	2762	2813	2864			
	2915	2954	2974	2977	2982	3325	3369	3546	3575	3621	3650	3696	3725	3771	3800			
	3844	3848	3872	3876	3882	3910	3914	3938	3942	3948	4008	4029	4040	4049	4173			
	4194	4224	4245	4256	4265	4421	4442	4461	4491	4512	4523	4532	4978	5059	5089			
	5266	5289																
INCB	2175																	
JMP	730	1297	1335	1376	1593	1666	1698	1927	1949	2061	2141							
JSR	1107	1121	1329	1352	1383	1598	1801	1984	2001	2012	2028	2641	2685	2738	2743			
	2747	2754	2758	2789	2794	2798	2805	2809	2840	2845	2849	2856	2860	2891	2896			
	2900	2907	2911	2946	2950	2955	2959	2966	2970	2978	3002	3030	3058	3086	3115			
	3163	3200	3237	3274	3314	3317	3330	3358	3361	3374	3395	3398	3406	3414	3500			
	3537	3542	3547	3552	3556	3566	3571	3576	3581	3585	3612	3617	3622	3627	3631			
	3641	3646	3651	3656	3660	3687	3692	3697	3702	3706	3716	3721	3726	3731	3735			
	3762	3767	3772	3777	3781	3791	3796	3801	3806	3810	3840	3852	3856	3861	3865			
	3906	3918	3922	3927	3931	3975	3976	3977	3980	3981	3992	4000	4004	4021	4025			
	4037	4046	4071	4072	4073	4076	4077	4079	4092	4097	4136	4137	4138	4141	4143			
	4145	4146	4157	4165	4169	4186	4190	4208	4216	4220	4237	4241	4253	4262	4273			
	4289	4294	4318	4332	4337	4383	4384	4385	4388	4389	4390	4393	4394	4405	4413			
	4417	4434	4438	4454	4458	4475	4483	4487	4504	4508	4520	4529	4542	4556	4561			
	4582	4596	4610	4615	4653	4654	4655	4658	4660	4663	4671	4681	4689	4718	4719			
	4720	4723	4755	4756	4762	4769	4772	4773	4776	4777	4780	4781	4786	4802	4807			
	4849	4850	4856	4863	4866	4867	4871	4886	4891	4949	4958	4965	4994	5003	5007			
	5016	5020	5023	5040	5049	5056	5064	5139	5148	5152	5185	5201	5215	5232	5238			
	5294																	
MOV	1071	1072	1073	1074	1078	1079	1080	1081	1086	1087	1089	1090	1091	1092	1093			

	1098	1099	1100	1101	1108	1131	1132	1133	1135	1136	1137	1141	1153	1154	1158
	1167	1170	1172	1173	1174	1182	1196	1201	1202	1234	1235	1237	1238	1249	1250
	1252	1253	1254	1263	1264	1265	1267	1273	1274	1275	1282	1284	1288	1289	1291
	1292	1293	1309	1318	1319	1320	1324	1325	1334	1353	1359	1371	1372	1373	1375
	1388	1394	1395	1408	1412	1413	1414	1418	1419	1428	1431	1432	1434	1435	1437
	1438	1444	1445	1446	1447	1448	1449	1452	1454	1484	1485	1489	1490	1501	1505
	1506	1507	1508	1509	1510	1515	1516	1517	1518	1519	1520	1527	1528	1529	1530
	1531	1532	1534	1537	1541	1543	1544	1556	1569	1570	1571	1572	1573	1586	1588
	1592	1608	1611	1613	1619	1620	1621	1656	1664	1665	1679	1686	1687	1688	1704
	1721	1731	1742	1791	1792	1793	1794	1796	1817	1820	1822	1823	1824	1831	1834
	1862	1863	1866	1867	1868	1869	1871	1872	1873	1874	1875	1876	1877	1879	1881
	1883	1886	1888	1890	1907	1909	1915	1925	1926	1940	1945	1950	1959	1972	1981
	1989	2062	2063	2070	2072	2078	2079	2080	2081	2087	2088	2090	2092	2104	2105
	2107	2113	2114	2116	2117	2132	2133	2139	2140	2146	2155	2156	2157	2158	2159
	2160	2161	2165	2168	2169	2170	2171	2172	2173	2174	2179	2187	2195	2209	2212
	2227	2228	2231	2234	2236	2251	2252	2254	2257	2274	2275	2278	2281	2283	2298
	2299	2302	2305	2307	2326	2327	2328	2331	2332	2337	2344	2368	2369	2370	2373
	2374	2379	2386	2410	2411	2412	2415	2416	2418	2424	2435	2436	2439	2445	2468
	2469	2470	2473	2474	2476	2481	2490	2491	2494	2499	2520	2521	2526	2542	2543
	2548	2564	2565	2598	2599	2602	2603	2609	2612	2614	2631	2632	2635	2636	2643
	2646	2648	2653	2656	2658	2675	2676	2679	2680	2687	2691	2694	2701	2704	2711
	2714	2715	2733	2734	2737	2739	2742	2744	2749	2784	2785	2788	2790	2793	2795
	2800	2835	2836	2839	2841	2844	2846	2851	2886	2887	2890	2892	2895	2897	2902
	2938	2939	2942	2947	2951	2956	2962	2979	2996	2997	3000	3001	3007	3024	3025
	3028	3029	3035	3037	3052	3053	3056	3057	3063	3080	3081	3084	3085	3091	3093
	3109	3110	3113	3114	3121	3123	3128	3133	3140	3157	3158	3161	3162	3169	3171
	3178	3194	3195	3198	3199	3206	3208	3215	3216	3231	3232	3235	3236	3243	3245
	3252	3253	3268	3269	3272	3273	3280	3282	3289	3290	3305	3306	3312	3313	3320
	3321	3331	3346	3347	3353	3355	3364	3365	3375	3390	3391	3394	3401	3402	3409
	3410	3417	3418	3434	3435	3443	3449	3451	3464	3484	3485	3492	3493	3497	3503
	3508	3510	3531	3532	3533	3536	3539	3540	3561	3565	3568	3569	3590	3606	3607
	3608	3611	3614	3615	3636	3640	3643	3644	3665	3681	3682	3683	3686	3689	3690
	3711	3715	3718	3719	3740	3756	3757	3758	3761	3764	3765	3786	3790	3793	3794
	3815	3830	3831	3834	3839	3841	3845	3849	3854	3871	3879	3896	3897	3900	3905
	3907	3911	3915	3920	3937	3945	3964	3965	3971	3972	3974	3985	3990	3991	3996
	4017	4064	4065	4068	4069	4070	4082	4100	4102	4125	4126	4132	4133	4135	4150
	4155	4156	4161	4182	4198	4199	4202	4206	4207	4212	4233	4273	4274	4278	4297
	4299	4316	4317	4321	4340	4342	4372	4373	4379	4380	4382	4398	4403	4404	4409
	4430	4450	4465	4466	4469	4473	4474	4479	4500	4540	4541	4545	4564	4566	4585
	4594	4595	4599	4618	4620	4646	4647	4650	4651	4652	4674	4684	4685	4692	4693
	4709	4710	4713	4714	4715	4727	4748	4749	4754	4757	4759	4760	4766	4783	4784
	4785	4789	4797	4810	4812	4825	4842	4843	4848	4851	4853	4854	4860	4869	4870
	4874	4894	4896	4921	4944	4946	4950	4953	4962	4988	4990	4991	4995	4998	5004
	5014	5017	5031	5036	5037	5041	5044	5050	5055	5067	5070	5100	5101	5102	5103
	5104	5111	5117	5119	5133	5135	5136	5140	5143	5149	5181	5182	5186	5189	5202
	5216	5229	5230	5231	5239	5247	5259	5260	5261	5273	5282	5283	5284	5299	5307
	5308	5312													
MOV B	1176	1323	1406	1422	1450	1451	1535	1536	1542	1547	1557	1562	1603	1636	2048
	2057	2338	2351	2380	2393	2426	2447	2482	2500	2527	2549	3984	4083	4095	4149
	4201	4279	4292	4306	4322	4335	4349	4397	4468	4546	4559	4573	4600	4613	4627
	4675	4761	4790	4805	4819	4855	4875	4889	4903	5233					
NOP	1288	1289	1330	1331	1332	1333									
RESET	1304	1327	1939												
ROL	1856	2135													
ROL B	2433	2455	2488	2507	4922	4923	4924	5162	5172	5173					
ROR	1549	1551	1553	2166	3560	3589	3635	3664	3710	3739	3785	3814	3870	3936	4019

	4184	4235	4432	4502	5106	5107	5109	5118							
RORB	2752	2803	2854	2905	2964	3998	4163	4214	4411	4452	4481	5262	5285	1756	2147
RTI	1191	1389	1409	1439	1491	1511	1521	1574	1667	1712	1717	1728	1738		
	2196														
RTS	1825	1836	2197	2214	4925	4936	4966	4981	5008	5024	5065	5092	5120	5153	5163
	5174	5194	5208	5221	5240	5252	5274	5300	5313						
SUB	1272	1278	1279	1587	1612	2177									
TRAP	813	815	817	819	821	823	825	827	829	831	833	835	837	839	841
TST	1109	1116	1160	1168	1175	1180	1183	1192	1205	1232	1269	1386	1397	1624	1628
	1637	1642	1646	1653	1818	1850	1896	1919	2058	2065	2068	2112	2119	2130	5032
	5053														
TSTB	1102	1114	1122	1124	1357	1363	1401	1404	1420	1426	1560	1601	1622	1829	1832
	2207	2210													
.ASCII	734	1758													
.ASCIZ	734	1758	5335												
.BLKW	787	788	789	790	873	874	875	876	878	879	880	881	883	884	885
	886	888	889	890	891	893	894	895	896	898	899	900	901	903	904
	905	906	908	909	910	911	913	914	915	916	918	919	920	921	923
	924	925	926	928	929	930	931	933	934	935	936	938	939	940	941
	943	944	945	946	948	949	950	951	1739						
.BYTE	799	800	801	802	1215	1218	1220	1222	1223	1224	1225	1226	1227	1228	1229
	1230	1337	1340	1343	1346	1669	1672	1700	1759	1761	1763	1765	1767	1839	1905
	1906	1957	1958	1970	1971	1979	1980	2046	2047	2055	2056	2150	5318	5320	5326
	5330	5335													
.ENABL	599	622													
.END	5335														
.ENDC	724	1957	1970	1979	2046	2055	2216	2220	2223	2224	2225	2229	2230	2231	2237
	2244	2247	2248	2249	2253	2254	2259	2260	2267	2270	2271	2272	2276	2277	2278
	2284	2291	2294	2295	2296	2300	2301	2302	2312	2319	2322	2323	2324	2328	2329
	2330	2331	2361	2364	2365	2366	2370	2371	2372	2373	2403	2406	2407	2408	2412
	2413	2414	2415	2426	2435	2447	2457	2461	2464	2465	2466	2470	2471	2472	2473
	2482	2490	2500	2509	2513	2516	2517	2518	2522	2523	2524	2528	2535	2538	2539
	2540	2544	2545	2546	2550	2557	2560	2561	2562	2566	2567	2568	2590	2594	2595
	2596	2600	2601	2602	2609	2612	2615	2619	2623	2627	2628	2629	2633	2634	2635
	2643	2646	2649	2659	2663	2667	2671	2672	2673	2677	2678	2679	2687	2691	2694
	2697	2707	2720	2724	2726	2729	2730	2731	2735	2736	2737	2762	2775	2777	2780
	2781	2782	2786	2787	2788	2813	2826	2828	2831	2832	2833	2837	2838	2839	2864
	2877	2879	2882	2883	2884	2888	2889	2890	2915	2928	2934	2935	2936	2940	2941
	2942	2974	2989	2992	2993	2994	2998	2999	3000	3010	3017	3020	3021	3022	3026
	3027	3028	3038	3045	3048	3049	3050	3054	3055	3056	3066	3073	3076	3077	3078
	3082	3083	3084	3094	3101	3105	3106	3107	3111	3112	3113	3118	3124	3136	3143
	3147	3150	3153	3154	3155	3159	3160	3161	3166	3172	3180	3184	3187	3190	3191
	3192	3196	3197	3198	3203	3209	3217	3221	3224	3227	3228	3229	3233	3234	3235
	3240	3246	3254	3258	3261	3264	3265	3266	3270	3271	3272	3277	3283	3291	3295
	3298	3301	3302	3303	3307	3308	3309	3314	3316	3337	3342	3343	3344	3348	3349
	3350	3358	3360	3381	3386	3387	3388	3392	3393	3394	3403	3411	3419	3426	3430
	3431	3432	3436	3437	3438	3452	3455	3467	3470	3477	3480	3481	3482	3486	3487
	3488	3520	3523	3527	3528	3529	3533	3534	3535	3536	3540	3541	3542	3545	3546
	3559	3560	3566	3569	3570	3571	3574	3575	3588	3589	3595	3598	3599	3602	3604
	3608	3609	3610	3611	3615	3616	3617	3620	3621	3634	3635	3641	3644	3645	3646
	3649	3650	3663	3664	3670	3673	3677	3678	3679	3683	3684	3685	3686	3690	3691
	3692	3695	3696	3709	3710	3716	3719	3720	3721	3724	3725	3738	3739	3745	3748
	3752	3753	3754	3758	3759	3760	3761	3765	3766	3767	3770	3771	3784	3785	3791
	3794	3795	3796	3799	3800	3813	3814	3820	3823	3826	3827	3828	3832	3833	3834
	3839	3854	3856	3868	3870	3889	3892	3893	3894	3899	3899	3900	3905	3920	3922
	3934	3936	3955	3960	3961	3962	3966	3967	3968	3974	3981	3986	4008	4033	4052

4056	4060	4061	4062	4066	4067	4068	4069	4077	4079	4100	4103	4106	4109	4113
4115	4117	4121	4122	4123	4127	4128	4129	4135	4142	4146	4151	4173	4198	4201
4224	4249	4285	4300	4303	4309	4312	4343	4346	4352	4355	4359	4361	4363	4368
4369	4370	4374	4375	4376	4382	4389	4390	4394	4399	4421	4446	4465	4468	4491
4516	4567	4570	4576	4579	4590	4621	4624	4630	4633	4637	4642	4643	4644	4648
4649	4650	4698	4702	4705	4706	4707	4711	4712	4713	4737	4739	4740	4744	4745
4746	4750	4751	4752	4758	4769	4773	4777	4781	4813	4816	4822	4825	4828	4832
4835	4838	4839	4840	4844	4845	4846	4852	4867	4897	4900	4906	4909	4910	5119
5128	5335													
673	804	1231	1758	1769	5334	5335								
792	723	1956	1969	1978	2045	2054	2216	2217	2220	2223	2228	2229	2230	2236
599	2241	2244	2247	2252	2253	2254	2259	2264	2267	2270	2275	2276	2277	2278
2237	2284	2288	2291	2294	2299	2300	2301	2302	2307	2308	2316	2319	2322	2327
2283	2330	2331	2358	2361	2364	2369	2371	2372	2373	2400	2403	2406	2411	2413
2329	2415	2426	2424	2447	2456	2458	2461	2464	2469	2471	2472	2473	2482	2489
2414	2508	2510	2513	2516	2521	2522	2523	2524	2525	2532	2535	2538	2543	2544
2500	2546	2547	2554	2557	2560	2565	2566	2567	2568	2587	2590	2594	2599	2600
2545	2602	2608	2609	2614	2619	2620	2623	2627	2632	2633	2634	2635	2641	2643
2601	2653	2658	2663	2664	2667	2671	2676	2677	2678	2679	2685	2687	2691	2696
2648	2706	2711	2721	2724	2726	2729	2734	2735	2736	2737	2762	2772	2775	2777
2701	2785	2786	2787	2788	2813	2823	2826	2828	2831	2836	2837	2838	2839	2864
2780	2877	2879	2882	2887	2888	2889	2890	2915	2925	2928	2934	2939	2940	2941
2874	2996	2989	2992	2997	2998	2999	3000	3009	3014	3017	3020	3025	3026	3027
2942	3037	3042	3045	3048	3053	3054	3055	3056	3065	3070	3073	3076	3081	3082
3028	3084	3093	3098	3101	3105	3110	3111	3112	3113	3118	3123	3124	3128	3135
3083	3142	3143	3147	3150	3153	3158	3159	3160	3161	3166	3171	3172	3176	3179
3136	3184	3187	3190	3195	3196	3197	3198	3203	3208	3209	3213	3216	3217	3221
3180	3227	3232	3233	3234	3235	3240	3245	3246	3250	3253	3254	3258	3261	3264
3224	3270	3271	3272	3277	3282	3283	3287	3290	3291	3295	3298	3301	3306	3307
3269	3309	3313	3334	3337	3340	3342	3347	3348	3349	3350	3354	3378	3381	3386
3308	3392	3393	3394	3402	3403	3410	3411	3418	3419	3423	3426	3430	3435	3436
3391	3438	3451	3452	3455	3466	3467	3474	3477	3480	3485	3486	3487	3488	3520
3437	3527	3532	3534	3535	3536	3539	3540	3542	3545	3546	3554	3560	3566	3568
3523	3571	3574	3575	3583	3589	3595	3598	3602	3607	3609	3610	3611	3614	3615
3569	3620	3621	3629	3635	3641	3643	3644	3646	3649	3650	3658	3664	3670	3673
3617	3682	3684	3685	3686	3689	3690	3692	3695	3696	3704	3710	3716	3718	3719
3677	3724	3725	3733	3739	3745	3748	3752	3757	3759	3760	3761	3764	3765	3767
3721	3771	3779	3785	3791	3793	3794	3796	3799	3800	3808	3814	3820	3823	3826
3770	3832	3833	3834	3839	3854	3863	3870	3886	3889	3892	3897	3898	3899	3900
3831	3920	3929	3936	3952	3955	3960	3965	3966	3967	3968	3972	3974	3975	3981
3905	4017	4033	4052	4053	4056	4060	4065	4066	4067	4068	4069	4071	4077	4091
3990	4102	4103	4109	4110	4113	4114	4115	4116	4121	4126	4127	4129	4129	4133
4100	4136	4142	4146	4155	4182	4198	4201	4206	4233	4295	4287	4297	4299	4300
4135	4309	4312	4330	4340	4342	4343	4346	4352	4356	4359	4360	4361	4362	4368
4303	4374	4375	4376	4380	4382	4383	4389	4394	4403	4430	4446	4468	4473	4500
4373	4554	4564	4566	4567	4570	4576	4579	4608	4618	4620	4621	4624	4630	4634
4552	4641	4642	4647	4648	4649	4650	4699	4702	4705	4710	4711	4712	4713	4734
4637	4739	4744	4749	4750	4751	4752	4754	4759	4773	4777	4781	4800	4810	4812
4737	4816	4822	4829	4832	4834	4835	4838	4843	4844	4845	4846	4848	4852	4853
4813	4894	4896	4897	4900	4906	5095	5117	5335						
4884	723	1957	1978	2045	2054	2216	2220	2223	2228	2229	2230	2237	2244	2247
	2252	2253	2254	2259	2270	2275	2276	2278	2284	2291	2294	2299	2300	2302
.IFF	2312	2319	2322	2327	2328	2329	2331	2364	2369	2370	2371	2373	2403	2406
	2411	2412	2413	2415	2434	2457	2461	2465	2469	2470	2471	2473	2509	2513
	2517	2521	2522	2524	2526	2528	2535	2539	2543	2544	2546	2547	2549	2561

	2565	2566	2568	2590	2595	2599	2600	2602	2609	2612	2615	2619	2623	2628	2632
	2633	2635	2641	2643	2649	2659	2667	2672	2676	2677	2679	2685	2694	2696	2706
	2724	2730	2734	2735	2737	2742	2743	2751	2762	2775	2781	2785	2786	2788	2793
	2794	2802	2813	2826	2832	2836	2837	2839	2844	2845	2853	2864	2877	2883	2887
	2898	2890	2895	2896	2904	2915	2928	2935	2939	2940	2942	2950	2961	2963	2974
	2989	2993	2997	2998	3000	3009	3017	3021	3025	3026	3028	3038	3045	3049	3053
	3054	3056	3065	3073	3077	3081	3082	3084	3094	3101	3106	3110	3111	3113	3118
	3124	3135	3142	3147	3150	3153	3154	3158	3159	3161	3166	3172	3176	3179	3187
	3190	3191	3195	3196	3198	3203	3209	3213	3217	3224	3227	3228	3232	3233	3235
	3240	3246	3250	3254	3261	3264	3265	3269	3270	3272	3277	3283	3287	3291	3298
	3302	3306	3307	3309	3310	3313	3316	3337	3343	3347	3348	3350	3351	3358	3360
	3381	3387	3391	3392	3394	3403	3411	3419	3426	3431	3435	3436	3438	3442	3452
	3466	3477	3481	3485	3486	3488	3503	3523	3528	3532	3533	3534	3536	3540	3541
	3555	3559	3566	3569	3570	3583	3587	3598	3603	3607	3608	3609	3611	3615	3630
	3634	3641	3644	3658	3662	3673	3678	3682	3683	3684	3686	3690	3705	3709	3716
	3719	3733	3737	3748	3753	3757	3758	3759	3761	3765	3780	3784	3791	3794	3808
	3812	3823	3827	3831	3832	3834	3854	3864	3868	3889	3893	3897	3898	3900	3920
	3929	3933	3955	3961	3965	3966	3968	3974	3976	3981	4033	4052	4056	4061	4065
	4066	4068	4069	4072	4077	4100	4103	4106	4109	4113	4115	4117	4121	4122	4126
	4127	4129	4135	4137	4142	4146	4198	4201	4285	4300	4303	4309	4343	4346	4352
	4359	4361	4362	4363	4368	4369	4373	4374	4376	4382	4384	4389	4394	4465	4468
	4567	4570	4576	4579	4621	4624	4630	4637	4641	4643	4647	4648	4650	4652	4664
	4681	4702	4706	4710	4711	4713	4737	4739	4740	4745	4749	4750	4752	4754	4758
	4759	4769	4773	4777	4781	4783	4795	4813	4816	4822	4832	4835	4839	4843	4844
	4846	4848	4852	4853	4863	4867	4869	4880	4897	4900	4906	4910	5128		
.IFT	2216	2527	2549	2587	3115	3163	3200	3237	3274	3315	3359	3442	3477	3546	3558
	3560	3575	3587	3589	3621	3633	3635	3650	3662	3664	3696	3708	3710	3725	3737
	3739	3771	3783	3785	3800	3812	3814	3856	3867	3922	3933	3955	3981	3982	3984
	4008	4018	4029	4033	4052	4068	4077	4078	4079	4109	4118	4142	4147	4149	4173
	4183	4194	4198	4224	4234	4245	4249	4268	4306	4312	4349	4364	4389	4395	4397
	4421	4431	4442	4446	4491	4501	4512	4516	4535	4573	4579	4627	4650	4654	4664
	4681	4755	4781	4785	4819	4828	4849	4852	4869	4882	4903	5119			
.IFTF	599	2216	2526	2548	2737	2742	2743	2751	2788	2793	2794	2802	2839	2844	2845
	2853	2890	2895	2896	2904	2942	2950	2961	2963	3113	3161	3198	3235	3272	3309
	3310	3350	3351	3428	3442	3443	3479	3492	3505	3545	3546	3555	3574	3575	3584
	3620	3621	3630	3649	3650	3659	3695	3696	3705	3724	3725	3734	3770	3771	3780
	3799	3800	3809	3854	3864	3920	3930	3955	3974	3976	3981	3983	3990	4017	4018
	4019	4029	4033	4070	4072	4077	4078	4091	4100	4116	4118	4135	4137	4146	4148
	4155	4182	4183	4184	4194	4206	4233	4234	4235	4245	4249	4288	4297	4331	4340
	4362	4365	4382	4384	4394	4396	4403	4430	4431	4432	4442	4473	4500	4501	4502
	4512	4516	4555	4564	4609	4618	4642	4650	4654	4665	4741	4754	4759	4781	4785
	4801	4810	4826	4835	4848	4852	4853	4863	4867	4870	4880	4885	4894	5117	
.IIF	2228	2229	2235	2237	2238	2252	2253	2258	2260	2261	2275	2276	2282	2284	2285
	2299	2300	2306	2312	2327	2328	2369	2370	2411	2412	2417	2419	2421	2422	2423
	2424	2425	2427	2430	2431	2432	2437	2440	2442	2443	2444	2445	2446	2448	2451
	2452	2453	2464	2469	2470	2475	2477	2478	2479	2480	2481	2482	2483	2485	2486
	2487	2492	2495	2496	2497	2498	2499	2500	2501	2503	2504	2505	2516	2521	2522
	2538	2543	2544	2560	2565	2566	2594	2599	2600	2613	2615	2616	2627	2632	2633
	2647	2649	2650	2671	2676	2677	2695	2697	2729	2734	2735	2780	2785	2786	2831
	2836	2837	2882	2887	2888	2934	2939	2940	2992	2997	2998	3020	3025	3026	3048
	3053	3054	3076	3081	3082	3105	3110	3111	3116	3122	3134	3141	3153	3158	3159
	3164	3165	3170	3179	3190	3195	3196	3201	3202	3207	3216	3227	3232	3233	3238
	3239	3244	3253	3264	3269	3270	3275	3276	3281	3290	3301	3306	3307	3342	3347
	3348	3386	3391	3392	3402	3410	3418	3430	3435	3436	3450	3465	3480	3485	3486
	3527	3532	3533	3565	3594	3602	3607	3608	3640	3669	3677	3682	3683	3715	3744
	3752	3757	3758	3790	3819	3826	3831	3832	3892	3897	3898	3960	3965	3966	4060

LIST	MACRO	SECT	BTTL	TITLE
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20
21	21	21	21	21
22	22	22	22	22
23	23	23	23	23
24	24	24	24	24
25	25	25	25	25
26	26	26	26	26
27	27	27	27	27
28	28	28	28	28
29	29	29	29	29
30	30	30	30	30
31	31	31	31	31
32	32	32	32	32
33	33	33	33	33
34	34	34	34	34
35	35	35	35	35
36	36	36	36	36
37	37	37	37	37
38	38	38	38	38
39	39	39	39	39
40	40	40	40	40
41	41	41	41	41
42	42	42	42	42
43	43	43	43	43
44	44	44	44	44
45	45	45	45	45
46	46	46	46	46
47	47	47	47	47
48	48	48	48	48
49	49	49	49	49
50	50	50	50	50
51	51	51	51	51
52	52	52	52	52
53	53	53	53	53
54	54	54	54	54
55	55	55	55	55
56	56	56	56	56
57	57	57	57	57
58	58	58	58	58
59	59	59	59	59
60	60	60	60	60
61	61	61	61	61
62	62	62	62	62
63	63	63	63	63
64	64	64	64	64
65	65	65	65	65
66	66	66	66	66
67	67	67	67	67
68	68	68	68	68
69	69	69	69	69
70	70	70	70	70
71	71	71	71	71
72	72	72	72	72
73	73	73	73	73
74	74	74	74	74
75	75	75	75	75
76	76	76	76	76
77	77	77	77	77
78	78	78	78	78
79	79	79	79	79
80	80	80	80	80
81	81	81	81	81
82	82	82	82	82
83	83	83	83	83
84	84	84	84	84
85	85	85	85	85
86	86	86	86	86
87	87	87	87	87
88	88	88	88	88
89	89	89	89	89
90	90	90	90	90
91	91	91	91	91
92	92	92	92	92
93	93	93	93	93
94	94	94	94	94
95	95	95	95	95
96	96	96	96	96
97	97	97	97	97
98	98	98	98	98
99	99	99	99	99
100	100	100	100	100

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0
 * DZDMEA.SEG/SOL/CRF/PAGNUM/NL:TOC=IPLJTL.MAC,DZDMEA.P11
 RUN-TIME: 52 74 11 SECONDS
 RUN-TIME RATIO: 654/137=4.7
 CORE USED: 32K (63 PAGES)

