

PDM-70

DIAGNOSTIC TEST
CZPMAC0

AH-9018C-MC

COPYRIGHT © 74-78

FICHE 1 OF 1

DEC 1978

digital

MADE IN USA

IDENTIFICATION

SEQ 0001

PRODUCT CODE: AC-9017C-MC
PRODUCT NAME: CZPMAC0 PDM70 DIAGNOSTIC TEST
DATE CREATED: 1-FEB-1978
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: EARL L. BOUSE/MIKE MITCHELL
REVISED BY: BILL SCHLITZKUS

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 MANUAL.

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

COPYRIGHT (C) 1974,1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

- 1.0 ABSTRACT
- 2.0 REQUIREMENTS (EQUIPMENT)
- 3.0 LOADING PROCEDURE
- 4.0 STARTING PROCEDURE
- 5.0 TELEPRINTER CONTROL SWITCHES
- 6.0 CONSOLE SWITCH SETTINGS
- 7.0 SERIAL I/O INPUT OPTION
- 8.0 DL11 ADDRESS SETUP PROCEDURE
- 9.0 MODULE ADDRESSING
- 10.0 MODULE ERROR REPORTING
- 11.0 SCOPE LOOPING
- 12.0 MODULE TEST PROGRAMS
- 13.0 USER AID ROUTINES

1.0 ABSTRACT

THIS IS A DESCRIPTION ON LOADING, USING AND INTERPRETING THE PDM70 DIAGNOSTIC PROGRAM. THE PROGRAM IS COMPRISED OF TWENTY-THREE KEYBOARD SELECTABLE TESTS WHICH TEST AND AID IN CHECKOUT OF THE PDM70 SYSTEM. THE PROGRAM IS STRUCTURED TO GIVE THE USER THE OPTION OF TESTING ANY OF THE MODULES COMPRISING THE PDM70 ON AN INDIVIDUAL OR SYSTEM TEST BASIS.

THE DIAGNOSTIC PROGRAM RESIDES IN A PDP-11 AND IS INTERFACED VIA A DL11 (ASYNCHRONOUS SERIAL LINE) TO THE PDM70. THE PDP-11 IS USED AS A COMBINATION CONTROL, SOURCE AND DESTINATION MODULE.

EACH MODULE TEST PROGRAM IS INDIVIDUALLY OUTLINED IN THIS WRITE-UP. THE SCOPE LOOPING TECHNIQUE AND MODULE ADDRESSING SCHEME IS IN GENERAL THE SAME FOR ALL MODULES WITH ANY UNIQUE CHARACTERISTICS POINTED OUT IN THE MODULE OUTLINE.

THE CONSOLE TELEPRINTER IS USED TO SELECT THE TEST PROGRAMS AND TO CONTROL THE DIAGNOSTIC. THE DIAGNOSTIC RUNS IN THREE MODES; MONITOR, WAIT AND RUN.

THE 'MONITOR MODE' IS ENTERED WHEN THE PROGRAM IS LOADED OR AT ANY TIME A NEW TEST IS TO BE SELECTED. HERE THE PROGRAM WAITS, DECODES AND THEN EXECUTES THE SELECTED TEST TYPED IN FROM THE KEYBOARD.

WHEN THE 'WAIT MODE' IS ENTERED THE PROGRAM HAS TO WAIT FOR ANY PARAMETERS (SUCH AS A MODULE ADDRESS) TO BE INPUTTED, A SIGNAL TO BE SCOPED OR TO STOP PROGRAM EXECUTION IF AN ERROR IS DETECTED.

THE 'RUN MODE' IS WHEN THE PROGRAM IS ACTUALLY EXECUTING A TEST PROGRAM.

THE TELEPRINTER KEYBOARD IS ALWAYS ACTIVE AND WILL RESPOND TO ANY KEYBOARD INPUT. ALL USERS RESPONSES ENTERED MUST END WITH A 'CR' (CARRIAGE RETURN) AND MAY NOT CONTAIN SPACES OR NULL CHARACTERS. 'RUBOUT' MAY BE USED TO ERASE ANY PREVIOUSLY ENTERED CHARACTERS. IF RUBOUT IS TYPED, THE ERASED CHARACTER WILL BE ECHOED BACK.

2.0 REQUIREMENTS (EQUIPMENT)

1. PDM70 MOTHER BOARD.
2. CLOCK MODULE (M7379-SET TO CORRESPOND TO THE DL11 FREQ.)
3. POWER SUPPLY
4. PDP-11 W/DL11 & 8K OF MEMORY
5. CONSOLE TELEPRINTER
6. PDM70 INTERFACE MODULE

A. THIS CAN EITHER BE A DF11 OR A SERIAL I/O MODULE (M7385)

3.0 LOADING PROCEDURE

1. USE STANDARD PROCEDURE FOR LOADING BINARY TAPES.

4.0 STARTING PROCEDURE

1. THE PROGRAM IS SELF STARTING WITH A RESTART ADDRESS OF '200'.

5.0 TELEPRINTER CONTROL SWITCHES

1. RETURN TO MONITOR (^C)*

TYPING A '^C' AT ANY TIME WILL ENABLE THE PROGRAM TO RETURN TO THE KEYBOARD MONITOR AND WAIT FOR A NEW TEST TO BE ENTERED.

2. CONTINUE (C)

IF A '^C' HAS BEEN TYPED, RETURNING CONTROL TO THE KEYBOARD MONITOR, AND THE USER WISHES TO RESTART THE LAST TEST HE WAS RUNNING, HE CAN SIMPLY TYPE 'C' CARRIAGE RETURN AND CONTINUE WITHOUT HAVING TO RE-TYPE THE TEST NAME.

3. RESTART (^R)*

TYPING A '^R' WILL ENABLE THE CURRENT TEST TO BE RESTARTED. IF A '^R' IS TYPED WHILE IN MONITOR MODE, THE ENTIRE TEST PROTOCOL IS RETYPED.

4. MODULE ADDRESS UPDATING (^A)*

TYPING A '^A' WHILE RUNNING ANY OF THE MODULE PROGRAMS WILL ENABLE A NEW MODULE ADDRESS TO BE ENTERED.

5. EXIT WAIT MODE (CR)

TYPING 'CR' WILL ENABLE THE PROGRAM TO CONTINUE FROM THE WAIT MODE.

* ALL CONTROL CHARACTERS ARE OBTAINED BY TYPING THE 'CTRL AND THE CHARACTER DESIGNATED' KEYS SIMULTANEOUSLY.

6. SUPPRESS PRINTING (^O)

TYPING A '^O' TELLS THE COMPUTER TO SUPPRESS THE REST OF THE TELEPRINTER OUTPUT. FOR INSTANCE, IF THE COMPUTER WAS TYPING OUT A MESSAGE AND THE USER KNEW WHAT THE MESSAGE WAS GOING TO BE, HE COULD TYPE A '^O' AND ENABLE THE PROGRAM TO CONTINUE WITHOUT WAITING FOR THE ENTIRE MESSAGE TO BE PRINTED.

6.0 CONSOLE SWITCH SETTINGS

WHEN THE EXERCISER TEST OR A DIAGNOSTIC TEST IS STARTED, THE PROGRAM WILL DETERMINE IF THE PROCESSOR HAS A HARDWARE SWITCH REGISTER (SWR). IF THERE IS NO HARDWARE SWR, THE PROGRAM WILL USE THE SOFTWARE SWR LOCATED AT ADDRESS 176. THE OPERATOR SHOULD SET UP LOC 176 BEFORE STARTING THE PROGRAM WITH THE APPROPRIATE VALUE.

SWITCH -----	FUNCTION -----
SW15=0	ENTER THE 'WAIT MODE' AND WAIT FOR 'CR' ON ERROR DETECTION
SW15=1	CONTINUE ON ERROR
SW14=0	CONTINUE ON TO NEXT SUBTEST
SW14=1	LOOP ON CURRENT SUBTEST
SW13=0	ENABLE PRINTOUTS
SW13=1	INHIBIT PRINTOUTS
SW12=0	NORMAL DL11 TRANSMISSION
SW12=1	ENTER THE 'WAIT MODE' AND WAIT FOR A 'CR' TO TRANSMIT EACH CHARACTER. AS EACH CHARACTER IS TRANSMITTED IT IS ALSO PRINTED.
SW11=0	NORMAL DL11 TRANSMISSION
SW11=1	TRANSMIT THE CURRENT CHARACTER UNTIL SW11 IS RESET TO '0'.
SW10=0	RUN THE ENTIRE MODULE TEST PROGRAM
SW10=1	INHIBIT THE MANUAL INTERVENTION TESTS IN THE MODULE TEST PROGRAM
SW09=0	NORMAL DL11 TRANSMISSION
SW09=1	INHIBIT TRANSMITTER DELAY

(SCOPE LOOPING AID)

G 1

SEQ 0006

NOTE: THE FUNCTIONS OF THE LOWER BITS (0-8) VARY IN USAGE AND
ARE OUTLINED IN THE APPLICABLE TEST DESCRIPTIONS. IN

V
N
A
E
S
A
R
R
T
T
C
T
O
C
M
C
A
.
1
T
M
F
I
D
C
T
C
T
1
1
T
S
S
E
E
S
I

GENERAL THOUGH, DATA SWITCHES '0-3' ARE USED IN THE EXERCISER TESTS TO ENABLE THE USER TO SELECT ANY PARTICULAR MODULE MODE. IN THESE CASES, THE PROGRAM ADDS A CODE OF '60' TO THE NUMBER READ FROM THE SWITCHES TO REPRESENT AN ASCII NUMBER.

IF THE PROGRAM IS USING THE SOFTWARE SWR, THE OPERATOR MAY CHANGE THE SWITCH SETTINGS FROM THE TTY. AFTER SELECTING A TEST OR TYPING 'C' IN THE MONITOR MODE, THE PROGRAM WILL OUTPUT AT THE TTY THE FOLLOWING MESSAGE

SWR=XXXXXX

NEW SWR

THE OPERATOR MAY THEN ENTER THE NEW VALUE. CARRIAGE RETURN ENTERS THE UPDATED VALUE. IF NO VALUE HAS BEEN ENTERED, THE SWITCH REGISTER VALUE REMAINS UNCHANGED.

WHILE SCOPE LOOPING ON SUBTEST, THE OPERATOR MAY INTERRUPT THE TEST TO CHANGE THE SWITCH SETTINGS BY TYPING CONTROL-G AT THE TTY. THE PROGRAM WILL OUTPUT AT THE TTY THE FOLLOWING MESSAGE

SWR XXXXXX

NEW SWR-

THE OPERATOR MAY THEN RESPOND AS DESCRIBED IN THE PRECEDING PARAGRAPH.

7.0 SERIAL I/O INPUT OPTION

AS MENTIONED IN THE ABSTRACT, THE PDM70 MODULES CAN BE TESTED IN TWO MODES: PER MODULE BASES OR SYSTEM TEST. IF THE MODULE IS TESTED INDIVIDUALLY A DF11 IS PLUGGED DIRECTLY INTO THE CONTROL SLOT OF THE PDM70 MOTHER BOARD. THIS ENABLES THE PDP-11 TO ACT AS A COMBINATION CONTROL, SOURCE AND DESTINATION MODULE. IN THIS CASE, THE SYSTEM CLOCK MUST BE SET TO CORRESPOND TO THE CLOCK FREQUENCY OF THE DL11.

WHEN THE MODULE IS TESTED IN A SYSTEM ENVIRONMENT, THE BASIC SYSTEM CONFIGURATION CONSISTS OF A : CONTROL, CLOCK, 'KGM' (KNOWN GOOD SERIAL INPUT/OUTPUT MODULE) AND A 'MUT' (MODULE UNDER TEST). THE 'KGM' SHOULD BE VERIFIED AS SUCH BY TESTING IT WITH THE M73851 TEST (REFER TO SECTION 12.13). THE 'KGM' CAN BE INSERTED IN ANY MODULE SLOT AND THEN CABLED TO THE DL11 OUTPUT OF THE PDP-11. THIS MODULE IS TO BE SET UP WITH THE 'D' JUMPER OUT AND THE 'L' JUMPER IN SO THAT IT IS INITIALIZED ON POWER UP. THE SYSTEM CLOCK MUST BE SET EITHER EQUAL TO OR GREATER THAN, THE INPUT DEVICES (E.G. DL11) BAUD RATE. PROGRAMS ARE THEN SENT FROM THE PDP-11 STORED IN THE CONTROL MODULE.

NOT OBVIOUS TO THE USER IS THE EXTRA ADDRESSING WHICH IS 'PADDED'

IN WHEN THE SERIAL I/O MODULE IS USED. THIS PADDING SERVES TWO FUNCTIONS. FIRST, IT FACILITATES LOADING A LEGAL PROGRAM INTO THE CONTROL MODULE 'FIFO' (FIRST-IN, FIRST-OUT BUFFER). THIS MEANS STARTING EACH PROGRAM WITH AN 'STX' AND ENDING IT WITH AN 'ETX'. PADDING ISN'T NECESSARY WHEN THE MODULE IS TESTED ON A MODULE BASIS. ALSO, EXTRA ADDRESSING MUST BE ADDED TO ADDRESS THE 'KGM'. THE PROGRAM HAS TO BE CERTAIN THAT THE 'KGM' NEVER LOSES CONTROL OF A PROGRAM SINCE THIS IS THE ONLY INTERFACE TO THE PDP-11. BY SETTING DATA SWITCH 12, THE USER CAN SINGLE STEP ANY MODULE TEST PROGRAM AND EXAMINE WHAT THIS PADDED PROGRAM LOOKS LIKE.

WHEN THE PROGRAM IS STARTED, IT ASKS IF A SERIAL I/O IS BEING USED. IF IT IS, TYPE 'YES' OR 'Y' CARRIAGE RETURN. IF IT'S NOT, TYPE 'NO', 'N' OR SIMPLY 'CR'. THIS PARAMETER CAN BE CHANGED AT ANY TIME BY TYPING A '^R' WHILE IN THE MONITOR MODE.

IF THE 'KGM' I/O IS BEING USED, THE PROGRAM WILL THEN ASK FOR THE ADDRESS OF THIS MODULE. THIS CAN BE ANY ADDRESS EXCEPT '17' WHICH FIT THE GUIDE LINES DESCRIBED IN SECTION 9.0 (MODULE ADDRESSING).

8.0 DL11 ADDRESS SETUP PROCEDURE

AFTER SETTING UP THE SERIAL I/O OPTION, THE PROGRAM PRINTS 'DL11 ADRS., VEC.?'. THIS ENABLES THE USER TO SELECT HIS OWN DL11 DEVICE AND VECTOR ADDRESSES. BY SIMPLY TYPING 'CR', THE DEFAULT RCSR ADDRESS OF '175610' AND VECTOR ADDRESS OF '300' ARE USED. IF THESE ADDRESSES ARE TO BE MODIFIED, TYPE THE RCSR ADDRESS AND THE VECTOR ADDRESS SEPERATED BY A COMMA.

THE USER SHOULD NOTE THAT BOTH THE DL11 AND THE SERIAL I/O MODULE ARE NORMALLY SFTUP FOR 7 BIT EVEN PARITY.

9.0 MODULE ADDRESSING

WHEN A MODULE PROGRAM IS SELECTED, THE PROGRAM REQUESTS THE MODULE ADDRESS BEFORE THE TEST IS RUN. THIS ADDRESS CAN BE ANY NUMBER FROM '0-17'*. THE ONLY RESTRICTION IS THAT IF THE SERIAL INPUT OPTION IS BEING USED, THESE TWO MODULE ADDRESSES MUST NOT CONFLICT. IF THEY DO, A NEW MODULE ADDRESS WILL BE REQUESTED. TYPING A '^A' AT ANY POINT WHILE A MODULE PROGRAM IS RUNNING WILL CAUSE THE PROGRAM TO REQUEST A NEW MODULE ADDRESS.

10.0 MODULE ERRORS

WHEN A MODULE ERROR IS DETECTED, THE FAILING SUBTEST NUMBER, M.A. (MEMORY ADDRESS) WHERE ERROR OCCURRED AND A DESCRIPTIVE MESSAGE OF THE FAILURE ARE TYPED OUT. THE PROGRAM THEN ENTERS THE 'WAIT MODE' UNTIL A 'CR' IS TYPED ENABLING THE PROGRAM TO CONTINUE.

WHEN AN ERROR IS DETECTED, THE 'M.A.' SHOULD BE USED TO LOCATE THE FAILING SUBTEST IN THE LISTING. HERE THE USER WILL FIND A WRITTEN DISCRPTION OF WHAT THE SUBTEST WAS ATTEMPTING TO DO. THE TEST CAN THEN BE ANALYZED AND THEN LOOPED IF NECESSARY UNTIL THE FAILURE IS FIXED.

WHEN A MODULE IS FAILING THE FIRST SUBTEST, IT IS A GOOD IDEA TO RE-CHECK THE MODULE TO MAKE SURE THAT IT WAS SET UP CORRECTLY WITH THE CORRECT SWITCH & JUMPER SETTINGS. THE IDEAL SITUATION, IF POSSIBLE, WOULD BE TO FIRST TEST A KNOWN GOOD MODULE.

11.0 SCOPE LOOPING

EACH MODULE ADDRESS TEST PROGRAM IS COMPRISED OF ANY NUMBER OF

* THE MODULE ADDRESS IS INTERPRETTED AS AN OCTAL VALUE.

INDIVIDUAL SUBTESTS. WHEN A MODULE PROGRAM IS RUN THESE SUBTESTS ARE RUN AS A WHOLE, OTHERWISE, WHEN ONE SUBTEST FINISHES THE NEXT SUBTEST IS EXECUTED.

THERE ARE TWO WAYS OF RUNNING ANY SELECTED SUBTEST: THE USER MAY RUN THE 'SUBX' ROUTINE (REFER TO SECTION 13.1) OR RUN THROUGH THE ENTIRE MODULE PROGRAM UNTIL THE SELECTED SUBTEST IS REACHED. IF THE LATTER METHOD IS USED, LOAD THE NUMBER OF THE SUBTEST TO BE LOOPED IN THE CONSOLE SWITCH REGISTER AND START THE MODULE PROGRAM. THE PROGRAM WILL TYPE 'SCOPE BREAK AT XXX' WHEN THE SUBTEST IS REACHED. NOW SET CONSOLE SWITCH '14' TO LOOP ON THE CURRENT SUBTEST AND THEN TYPE 'CR'. THE PROGRAM WILL THEN RUN THE SELECTED SUBTEST UNTIL SWITCH '14' IS RESET TO '0' ENABLING THE PROGRAM TO CONTINUE.

12.0 MODULE TEST PROGRAMS

THE FOLLOWING IS A LIST AND DESCRIPTION OF ALL THE MODULE PROGRAMS. IT SHOULD BE NOTED THAT IN THE PROGRAM TEST PROTOCOL EACH MODULE PROGRAM ENDS WITH A LETTER. THIS LETTER INDICATES THE TYPE OF TEST: A = ADDRESSING, C = CALIBRATION*, E EXERCISER, G = GAIN*, I = INTERFACE, R = REPEATIBILITY*.

THE MODULE ADDRESS TEST SHOULD BE RUN AND PROVED FULLY OPERATIONAL BEFORE RUNNING ANY OF THE OTHER TESTS. THIS TEST VERIFIES THAT THE MODULE CAN BE ADDRESSED AND THAT IT WORKS 'FUNCTIONALLY' IN ALL ITS INTENDED DATA MODES.

THE USER SHOULD REFER TO THE ENGINEERING SPECIFICATIONS TO VERIFY THAT THE SWITCHES AND JUMPERS ARE SET UP CORRECTLY BEFORE RUNNING ANY TESTS.

NOTE: BEFORE EACH MODULE TEST IT IS A GOOD PRACTICE TO CLEAR OUT THE PDM FIFO BY HITTING THE 'RESET' BUTTON ON THE FRONT PANEL.

***ALSO NOTE: IF THE PROGRAM IS USING THE SOFTWARE SWR, REFER TO SECTION 6.0.

12.1. M7380A, CONTROL MODULE TEST

THIS PROGRAM TAKES THE CONTROL MODULE THRU THE INITIALIZATION, ADDRESS AND DATA MODES RESPECTIVELY. INITIALLY, TWO PROGRAMS ARE STORED IN THE CONTROL MODULE 'FIFO'. THE SECOND PROGRAM IS HEADED WITH A 'DC4' SO IT WILL NOT BE RECIRCULATED. WITH THE FIRST PROGRAM IN THE DATA MODE, A '500' WORD RANDOM DATA BUFFER IS CIRCULATED THRU THE CONTROL MODULE. AFTER VERIFYING THE DATA, AN 'EOT' IS ISSUED. THIS ENABLES THE SECOND PROGRAM TO BE CALLED OUT. THE DATA MODE IS AGAIN CHECKED AND ANOTHER 'EOT' IS ISSUED ENABLING THE FIRST PROGRAM IS BE RE-CALLED. ONCE VERIFIED,

* APPLY TO THE A/D MODULE ONLY.

L 1

SEQ 0011

CZ
CZ

ANOTHER 'EOT' IS ISSUED. A CHECK IS THEN MADE THAT THE SECOND PROGRAM, HEADED WITH A 'DC4', NO LONGER EXISTS. THE 'FIFO' IS THEN REPROGRAMMED.

THIS PROGRAM CONSISTS OF '64' CHARACTERS ENABLING THE CONTROL 'FIFO' TO BE COMPLETELY FILLED. THE PROGRAM CONSISTS OF ONE SOURCE AND ONE DESTINATION ADDRESS. THE REMAINING 55 LOCATIONS ARE FILLED WITH RANDOM LITERAL CHARACTERS. THE PROGRAM IS THEN CALLED OUT AND VERIFIED.

THE LAST TEST CHECKS THE DELAY TIMES OF THE 'SYN' CHARACTER. THIS TEST REQUIRES A '110 BAUD' CONSOLE DEVICE SUCH AS A 'TTY' IN ORDER TO RUN. THE CRYSTAL CLOCK IN THE TTY IS USED TO TIME THE 'SYN' DELAYS. IF THE CONSOLE DEVICE IS NOT AVAILABLE, THIS TEST WILL NOT PASS. ALL THE DELAYS, 1-9, ARE TESTED IN ORDER. THE TESTS MAKES TWO CHECKS AT EACH DELAY. FIRST, THAT THE DELAY ISN'T TOO SHORT AND SECOND, THAT THE DELAY ISN'T TOO LONG.

THIS COMPLETES THE CONTROL MODULE TESTS. HOWEVER, IF DATA SW10 IS SET THE PROGRAM WILL ALSO TEST THE M7387 HARDWARE READIN MODULE (1). OTHERWISE, THE MESSAGE 'TEST COMPLETE' IS PRINTED AND THE PROGRAM WILL CONTINUE TO CYCLE THRU THE CONTROL TEST UNTIL STOPPED.

1. M7387, HARDWARE READ-IN MODULE

AS MENTIONED ABOVE, THIS TEST IS RUN IN CONJUNCTION WITH THE M7380A 1ST. THE TEST REQUIRES THE USER TO INSERT THE M7387 MODULE WITH A DIAGNOSTIC 'PROM' PROGRAM INTO SLOT 'P5' OF THE MOTHER BOARD.

AFTER THE MODULE HAS BEEN INSERTED, THE PDM70 SHOULD BE POWERED UP. THIS WILL ENABLE THE PROM PROGRAM TO BE READ OUT, STORED IN THE CONTROL MODULES FIFO, AND THEN EXECUTED.

THE PROM PROGRAM IS SETUP TO ADDRESS THE SERIAL I/O DESTINATION MODULE AND THEN SEND LITERAL DATA. AFTER VERIFYING THE DATA, THE MESSAGE 'PROM OK' IS TYPED. IF THIS MESSAGE IS NOT TYPED IMMEDIATELY AFTER POWER UP, NO DATA WAS EVER RECEIVED, THUS INDICATING AN ERROR CONDITION.

12.2. M7381A, BCD INPUT MODULE ADDRESS TEST

THIS TEST ADDRESSES THE 'BCD' MODULE IN ALL FOUR(4) DATA MODES VERIFYING INTERNAL AND EXTERNAL DEVICE FLAG OPERATION. IT IS SUGGESTED THAT THE M7381E TEST SHOULD BE RUN IF ANY DATA ERRORS ARE REPORTED. HERE THE USER CAN READILY IDENTIFY THE DATA ERROR BY THE TYPEOUT. THE CUSTOMER SWITCHES (WHICH SELECT HOW MANY DIGITS ARE READ) ARE TESTED BY THE PROGRAM REQUESTING UNIQUE SWITCH SETTINGS. SETTING DATA 'SW10' WILL INHIBIT THE MANUAL INTERVENTION TESTS. THIS MODULE HAS TO BE TESTED WITH THE 'L'

JUMPER OUT.

12.3. M7381E, BCD INPUT MODULE EXERCISER TEST

THIS PROGRAM CONTINUOUSLY LOOPS ADDRESSING THE BCD MODULE AND PRINTING THE RECEIVED DATA. DATA SWITCHES '0 & 1' ARE USED TO SELECT ANY ONE OF THE FOUR (4) 'BCD' DATA MODES. THE SWITCH SETTINGS MAY BE SET AND RESET ANY TIME. DATA SW13 CAN ALSO BE SET TO INHIBIT THE DATA PRINTOUT.

12.4. M7382A, BCD OUTPUT MODULE ADDRESSING TEST

THIS TEST IS COMPRISED OF A SERIES OF SUBTESTS WHICH OUTPUT KNOWN DATA TO THE 'BCD' OUTPUT MODULE. ONCE THE DATA IS TRANSMITTED, THE USER IS NOTIFIED OF THE TRANSMITTED PATTERN. THE PROGRAM THEN ENTERS THE 'WAIT' MODE ENABLING THE USER TO VERIFY THE DATA.

THE LAST SUBTEST REQUESTS FOR THE USER TO SCOPE FOR THE SIGNAL 'OUTPUT DONE H & L'. THE PROGRAM WILL INDEFINITELY HANG IN THIS SUBTEST UNTIL EITHER '^R' IS TYPED TO RESTART THE M7382A TEST OR '^C' IS TYPED TO RETURN TO THE MONITOR.

12.5. BCD I/O TEST

THIS IS AN EXERCISE TEST UTILIZING BOTH THE BCD 'INPUT & OUTPUT' MODULES. AN INCREMENTING BCD COUNT IS SENT TO THE OUTPUT MODULE AND WRAPPED AROUND VIA A SPECIAL CABLE TO THE INPUT MODULE. THE INPUT MODULE IS THEN ADDRESSED, ENABLING THE DATA TO BE READ. THE RECEIVED DATA IS VERIFIED AGAINST THE TRANSMITTED DATA. THIS TEST VERIFIES THAT ALL DATA LINES ARE GOOD AND THAT NO TWO LINES ARE SHORTED TOGETHER.

THE INPUT MODULE CAN BE SET UP TO USE EITHER INTERNAL OR EXTERNAL SYNC. IF EXTERNAL SYNC IS SELECTED, THE SYNC SIGNAL IS SUPPLIED FROM THE BCD OUTPUT MODULE VIA THE CABLE.

12.6. M7383A, A/D MODULE ADDRESS TEST

THIS TEST ADDRESSES THE A/D MODULE AND VERIFIES THE CORRECT DATA FORMAT IS RECEIVED FROM THE MODULE. THE EXTERNAL SYNC FUNCTION IS ALSO TESTED. IT SHOULD BE NOTED THAT THIS TEST MAKES NO ATTEMPT TO VERIFY WHETHER OR NOT THE A/D IS CONVERTING THE CORRECT VALUES.

12.7. M7383C, A/D CALIBRATION ROUTINE

THIS TEST RUNS IN A CONTINUOUS LOOP ADDRESSING THE A/D MODULE AND PRINTING THE CONVERSION VALUE. AFTER ACCEPTING THE MODULE ADDRESS, THE PROGRAM TYPES 'REMOTE DST.?'. THIS IS AN OPTION WHICH ENABLES THE USER TO SEND THE CONVERSION DATA TO A USER SELECTED DESTINATION, SUCH AS THE DISPLAY. IF THIS OPTION IS DESIRED, TYPE 'YES' OR 'Y' & 'CR'. A REQUEST WILL THEN MADE FOR THE ADDRESS OF THIS DESTINATION. DATA SWITCHES '0-3' ARE USED TO SELECT THE A/D CHANNEL TO BE CONVERTED. SETTING DATA SW13 WILL INHIBIT THE CONVERSION DATA PRINTOUT. ALL DATA SWITCHES MAY BE SET OR RESET AT ANY TIME.

CHANNEL SELECTION IS AS FOLLOWS:

DATA SW'S '0-1' SELECT 'INT. SYNC' ON CH.'S 0,1,2 OR 3
DATA SW'S '2' & '0-1' SELECT 'EXT SYNC' ON CH.'S 0,1,2 OR 3
DATA SW '3' SELECTS 'INT SYNC' CONVERSION ON ALL '4' CH.'S
DATA SW'S '2&3' SELECT 'EXT. SYNC' CONVERSION ON ALL '4' CH.'S

12.8. M7383G, A/D GAIN ACCURACY TEST

THIS TEST IS USED TO TEST THE GAIN ACCURACY OF THE A/D. FIVE SPECIFIC VOLTAGES AT A GAIN OF '1' ARE REQUESTED BY THE PROGRAM. WHEN THE VOLTAGE AND GAIN HAVE BEEN SUPPLIED, TYPE 'CR'. A SERIES OF ONE HUNDRED CONVERSIONS ARE THEN TAKEN AND AVERAGED. THIS AVERAGE IS THEN TESTED TO BE WITHIN '+ OR -' ONE COUNT FROM THE TRUE VOLTAGE VALUE FOR THAT SPECIFIED SETTING. IF IT IS NOT, THE LOW, AVERAGE AND HIGH VALUES OBTAINED ON THAT PARTICULAR GROUP OF CONVERSIONS ARE TYPED OUT. THE PROGRAM WILL THEN TAKE ANOTHER SERIES OF CONVERSIONS AND WILL CONTINUE DOING SO, UNTIL THE CORRECT VALUE IS RECEIVED. AT THAT POINT THE PROGRAM WILL REQUEST A NEW SETTING. DATA SWITCH '13' CAN BE SET TO INHIBIT THE ERROR DATA PRINTOUT.

12.9. M7383R, A/D REPEATIBILITY TEST

THIS TEST TAKES A SERIES OF ONE HUNDRED CONVERSIONS ON A USER SELECTED CHANNEL. THE CONVERSIONS ARE AVERAGED AND THEN DISPLAYED IN A GRAPH FORMAT SHOWING THE REPEATIBILITY CHARACTERISTICS OF THE A/D. AFTER ACCEPTING THE MODULE ADDRESS, THE PROGRAM TYPES 'REMOTE DST.?'. THIS IS A OPTION WHICH ENABLES THE USER TO SEND THE COMPUTED A/D GRAPH TO A USER SELECTED DESTINATION. IF THIS OPTION IS DESIRED, TYPE 'YES' OR 'Y' & 'CR'. A REQUEST WILL THEN BE MADE FOR THE ADDRESS OF THE DESTINATION. WHEN STARTED, THE TEST REQUESTS A CHANNEL AND V.S.F (VERTICAL SCALE FACTOR). THE V.S.F. IS THE NUMBER OF CONVERSIONS, OF THE HUNDRED, TO BE AVERAGED TOGETHER TO REPRESENT ONE POINT ON THE GRAPH. THE V.S.F. CAN BE ANY NUMBER EVENLY DIVIDED INTO ONE HUNDRED. EACH POINT (REPRESENTEND AS AN ASTRICK) IS PLOTTED IN ITS RELATIONSHIP TO THE OVERALL AVERAGE OF THE HUNDRED CONVERSIONS. THE FOLLOWING IS AN EXAMPLE OF WHAT A GRAPH PRINTOUT MIGHT LOOK LIKE USING A V.S.F. OF 10; 10 POINTS,

EACH REPRESENTING THE AVERAGE OF '10' CONVERSIONS.

EXAMPLE:

VSF? 10
CH.? 1

```

-      *
-      *
-      *
-      *
-      *
-      *
-      *
-      *
-      *
-      *

```

+++++++ (SCALE=1MV/DIV)
-141 -150 -159

THE THREE NUMBERS AT THE BOTTOM OF THE SCALE (RIGHT TO LEFT) REPRESENT; THE LOWEST VALUE, THE OVERALL AVERAGE AND THE HIGHEST VALUE READ OF THE ONE HUNDRED CONVERSIONS. SINCE THE GRAPH ONLY SHOWS COUNTS '+ & -' 9 COUNTS FROM THE AVERAGE, AN OVERRANGE 'HI & LO' PRINTOUT WOULD RESULT IF ANY COUNTS FALL OUT OF THE 9 COUNT RANGE.

12.10. M7384A, D/A ADDRESSING TEST

THIS TEST STARTS BY ADDRESSING THE D/A MODULE USING MODES '8 & 9'. THE USER IS THEN REQUESTED TO SCOPE THE SIGNALS PROG 'L & H'. FIVE SPECIFIC VOLTAGE ARE THEN TRANSMITTED FROM THE D/A ON EACH CHANNEL. AFTER EACH VOLTAGE IS TRANSMITTED, A MESSAGE IS TYPED TELLING THE USER THE VOLTAGE AND CHANNEL. THE LAST SUBTEST CHECKS THE RECOVERY OF THE D/A. THIS IS DONE BY CONTINUOUSLY ADDRESSING THE DAC IN MODE 3 (BOTH CHANNELS). THE PROGRAM THEN ALTERNATLY OUTPUTS '0' VOLTS AND '9.5' VOLTS. THIS ENABLES A SQUARE WAVE OUTPUT FROM THE D/A. THE USER IS REQUESTED TO SCOPE BOTH CHANNEL OUTPUTS AND CHECK FOR A 5 U SECOND. RISE TIME.

THE PROGRAM WILL INDEFINITELY HANG IN THIS SUBTEST UNTIL RESTARTED OR EXITED.

12.11. M7384E, D/A EXERCISER TEST

THIS TEST ENABLES ANY USER SELECTED VALUE TO BE TRANSMITTED FROM THE D/A. WHEN SELECTED, THE TEST REQUESTS FOR TWO, THREE DIGIT VALUES (SEPARATED VIA COMMA'S) TO BE TYPED IN. THE FIRST VALUE IS THE ONLY ONE TRANSMITTED WHEN RUNNING ONE CHANNEL. IF BOTH

CHANNELS ARE SELECTED, THE FIRST VALUE WILL BE TRANSMITTED ON CHANNEL '0' (X DAC) AND THE SECOND VALUE WILL BE TRANSMITTED ON CHANNEL '1' (Y DAC). THE CHANNELS ARE SELECTED BY DATA SWITCHES '0 & 1' AND CAN BE SET AND RESET AT ANYTIME. SETTING DATA SWITCH '0' WILL SELECT CHANNEL '0'. SETTING DATA SWITCH '1' WILL SELECT CHANNEL 1 AND SETTING BOTH '0 & 1' WILL SELECT BOTH CHANNELS.

TYPING A '^R' WILL ENABLE FOR A NEW SET OF DAC VALUES TO BE ACCEPTED.

12.12. M7385A, SERIAL I/O ADDRESS TEST

THIS TEST CHECKS BOTH THE SOURCE AND DESTINATION PARTS OF THE SERIAL I/O. BY USING A SPECIAL WRAPPING CABLE, THE DESTINATION OUTPUTS TO THE SOURCE INPUT.

BEFORE TESTING, ALL 'ACTIVE' RECEIVER JUMPERS MUST BE INSERTED AND THE 'D' & 'L' 'MR' JUMPERS MUST BE OUT.

THIS TEST CHECKS ONLY THE 'EIA' OUTPUT OF THE MODULE. REFER TO THE M7385T TEST (12.14) FOR TESTING THE 'TTY' OUTPUT LOGIC.

IT SHOULD BE NOTED THAT WHEN THIS TEST IS RUN USING THE SERIAL I/O INPUT OPTION, THAT ONLY SUBTESTS '1,5 & 10' ARE EXECUTED. THIS MEANS THE TESTING ISN'T TESTED AS IT IS WHEN USING THE DF11 INTERFACE.

IT SHOULD ALSO BE NOTED THAT WHEN THE SERIAL INPUT OPTION IS USED, SUBTEST 5 RETURNS ONE HUNDRED AND TWENTY EIGHT CHARACTERS (128) TO THE DL11 RECEIVER INSTEAD OF '64'. THE FIRST '64' CHARACTERS OF THE BUFFER ARE RETURNED DIRECTLY FROM THE DESTINATION OF THE SERIAL INPUT MODULE. THE SECOND '64' CHARACTERS ARE THE CHARACTERS THAT WERE ACTUALLY BUFFERED IN THE 'FIFO' OF THE MODULE UNDER TEST.

12.13. M7385I, SERIAL I/O INTERFACE MODULE TEST

THIS TEST IS INTENDED TO VERIFY THAT THE SERIAL I/O MODULE USED AS THE PDP-11 INTERFACE IS FUNCTIONING CORRECTLY. THIS IS DONE BY REMOVING THE M7380 CONTROL MODULE (THUS ELIMINATING ONE UNKNOWN) AND JUMPING THE 'T & R' BUSES (F1D1 TO F1V2) TOGETHER. THE MODULE MUST HAVE THE 'D' JUMPER OUT AND THE 'L' JUMPER IN SO THAT IT IS INITIALIZED ON POWER UP. A PROGRAM IS THEN SENT TO ADDRESS THE DESTINATION PORTION OF THE MODULE. WHEN THIS TEST HAS BEEN RUN SUCCESSFULLY, THE CONTROL MODULE CAN BE RE-INSERTED AND VERIFIED BY RUNNING THE M7380A TEST (12.1).D

12.14. M7385T, SERIAL I/O TTL TEST

THIS TEST VERIFIES THAT THE TTL I/O SECTION OF THE SERIAL I/O MODULE IS FUNCTIONING CORRECTLY. IT REQUIRES THAT A TELEPRINTER BE CABLED TO THE MATIN LOCK OF THE SERIAL I/O. THIS COULD BE THE CONSOLE PRINTER ONCE THE TEST IS SELECTED. IF THE CONSOLE PRINTER IS USED, THE PROGRAM SHOULD BE HALTED BEFORE DISCONNECTING THE PRINTER AND THEN RE-STARTED AT THE 'TTLTST'* ADDRESS. ALL CHARACTERS THEN TRANSMITTED WILL BE RECEIVED BY THE SERIAL SOURCE AND WRAPPED AROUND (BY THE CONTROL MODULE OR COMPUTER IF THE DF11 IS USED) TO THE DESTINATION. HERE THE CHARACTER WILL BE TRANSMITTED BACK TO THE TELEPRINTER AND PRINTED. EFFECTIVELY AS FOR AS THE USER IS CONCERNED, THIS TEST ACTS LIKE A KEYBOARD ECHO TEST.

12.15. M7386A, KEYBOARD/DISPLAY MODULE ADDRESS TEST

IN ORDER TO RUN THIS TEST, THE 'W1' JUMPER MUST BE OUT. THE FIRST SUBTEST ADDRESSES THE KEYBOARD AND CHECKS FOR THE FORCED RETURN OF THE 'EOT'.

THE SECOND SUBTEST RUNS IN A CONTINUOUS LOOP ADDRESSING BOTH THE KEYBOARD & DISPLAY. WHEN THE USER STRIKES 'KEY REQUEST', THE KEYBOARD BECOMES BUS MASTER. ALL DATA THEN TRANSMITTED FROM THE KEYBOARD IS SENT TO THE DISPLAY (IF AVAILABLE). THIS DATA IS ALSO RECEIVED BY THE PDP-11 AND PRINTED.

IF 'EOT' IS STRUCK, THE KEYBOARD RELEASES THE BUS AND THE PROGRAM IS AGAIN LOOPED UNTIL THE NEXT 'KEY REQUEST'.

IF 'STX' IS STRUCK AND THE SERIAL INPUT OPTION IS BEING USED, THE MESSAGE 'RE-INITIALIZE THE PDM70' IS PRINTED. THE PROGRAM THEN ENTERS THE 'WAIT MODE' AND UPON RECEIVING A 'CR', WILL BEGIN RE-CYCLING THE SUBTEST.

IF 'ETX' IS STRUCK, THIS SUBTEST IS EXITED, AND THE NEXT SUBTEST IS ENTERED. UPON ENTERING THE NEXT SUBTEST, THE MESSAGE 'ENTERING THE DISPLAY TEST, RE-INITIALIZE THE PDM70' IS PRINTED. THE PROGRAM THEN ENTERS THE 'WAIT MODE' AND WAITS FOR 'CR'. UPON RECEIPT OF THE 'CR' THE SUBTEST STARTS DISPLAYING THE ENTIRE CHARACTER SET, ON CHARACTER AT A TIME ACROSS THE ENTIRE SCREEN. AFTER EACH CHARACTER IS DISPLAYED, A SOFTWARE DELAY IS EXECUTED. THIS DELAY ENABLES THE USER TO VIEW THE LINE BEFORE THE NEXT CHARACTER LINE IS DISPLAYED. AFTER THE ENTIRE CHARACTER SET HAS BEEN DISPLAYED, THE TEST ENTIRE TEST PROGRAM IS RESTARTED.

12.16. M7387A, PROM HARDWARE READ-IN MODULE

THIS PROGRAM MAY BE SELECTED AS A SEPERATE MODULE TEST, ALTHOUGH IT IS RUN AS PART OF THE M7380 CONTROL MODULE TEST. REFER TO

* REFERENCE THE LISTING FOR THE ADDRESS OF THIS 'TAG'.

PART 1 OF SECTION 12.1 FOR A COMPLETE TEST DESCRIPTION.

12.17. M7388A, CHARACTER I/O MODULE ADDRESS (IN-HOUSE) TEST

THIS TEST REQUIRES A SPECIAL WRAP-AROUND MODULE (AVAILABLE ONLY IN HOUSE) TO RUN THIS TEST. FOR FIELD TESTING THIS MODULE REFER TO THE M7388F (SECTION 12.18).

THE TEST USES THE SAME TEST PROGRAM AS THE SERIAL I/O MODULE (REFER TO SECTION 12.12). TO RUN THIS TEST, JUMPERS 'SO & SI' MUST BE IN AND THE 'D' & 'L' JUMPERS MUST BE OUT.

12.18 M7388F, CHARACTER I/O MODULE ADDRESS (FIELD) TEST.

THIS PROGRAM IS DESIGNED TO COMMUNICATE WITH THE FIELD SERVICE TESTER. THE FIRST SUBTEST ADDRESS THE MODULE IN MODE '0' AND CHECKS FOR THE FORCED 'EOT'. THE NEXT SUBTEST ADDRESS THE MODULE IN MODE '1' AND CHECKS THAT NO 'EOT' IS RETURNED. A REQUEST IS THEN MADE FOR THE USER TO INPUT DATA (VIA THE TESTER) TO THE MODULE. AS EACH CHARACTER IS RECEIVED, IT IS ECHOED TO PRINTER. THE PROGRAM WILL HANG IN THIS SUBTEST UNTIL 'EOT' IS RECEIVED, ENABLING IT TO ENTER THE NEXT SUBTEST. THE NEXT SUBTEST IS A 'FIFO' STORAGE TEST. IT REQUESTS FOR THE USER TO INPUT DATA (UP TO 63 CHARACTERS) AND AN 'EOT'. AFTER ALL THE DATA HAS BEEN TRANSMITTED, TYPE 'CR'. THE MODULE (SOURCE) IS THEN ADDRESSED IN MODE '0' ENABLING THE 'FIFO' DATA TO BE READ AND PRINTED.

THE NEXT SUBTEST LOADS '16', '4' CHARACTER DATA PATTERNS (A TOTAL OF 64 CHAR.'S) INTO THE DESTINATION 'FIFO'. THE USER IS THEN REQUESTED TO STROKE OUT THESE '64' CHARACTERS AND VERIFY THEM. THE '4' CHARACTERS PATTERN IS: ALL 1'S, ALL 0'S, ALTERNATE '1'S & 0'S' AND REVERSED ALTERNATE '1'S & 0'S'.

THE LAST SUBTEST ADDRESSES THE MODULE USING ALL THE WRONG MODULE ADDRESSES AND CHECKS THAT THE SOURCE ISN'T ENABLED. THIS SUBTEST IS NOT EXECUTED WHEN USING THE SERIAL INPUT OPTION.

12.19 M7377A, REMOTE SERIAL I/O TEST

THIS PROGRAM TESTS THE M7377 MODULE USING THE PDP-11 VIA THE DL-11 AS THE DESTINATION INPUT AND SOURCE OUTPUT.

THE FIRST SUBTEST ADDRESSES THE SOURCE PORTION OF THE MODULE AND CHECKS FOR FORCED RETURN OF EOT.

THE SECOND SUBTEST TRANSMITS A RANDOM BUFFER AND CHECKS THAT IT IS RETURNED CORRECTLY.

IN THE NEXT SUBTEST A 2ND RANDOM BUFFER IS TRANSMITTED AND THE

VARIABLE TERMINATOR OPTION IS CHECKED.

NEXT, THE SOURCE IS THEN ADDRESSED USING THE WRONG MODULE ADDRESSES AND CHECKED TO MAKE SURE IT DOESN'T BECOME ENABLED.

ETX AND STX ARE THEN USED TO VERIFY THAT ETX WILL CLEAR THE SOURCE AND STX WILL CLEAR THE DESTINATION.

A MANUAL INTERVENTION SUBTEST THEN REQUESTS THAT THE OPERATOR RESET THE MODULE ADDRESS TO '17'. DATA IS TRANSMITTED AND THE RECEIVED DATA IS VERIFIED.

THE LAST SUBTEST CHECKS THE TIMEOUT AND REMOTE TIMEOUT ABILITY OF THE MODULE. A NON-EXISTENT SOURCE IS ADDRESSED AND THE MODULE IS CHECKED TO SEE IF IT WILL TIME-OUT CORRECTLY.

THE TEST FOR THE M7377A MODULE WILL NOT RUN UNLESS THE OUTPUT OF THE M7377A (PDM70-JR) IS JUMPED BACK TO THE INPUT (PIN 2 TO 3 + PIN 5 TO 7 ON THE MATE'N'LOCK), AND THE TRANSMITTER AND RECEIVER CURRENT LOOP INTERFACES ARE SET UP (WITH SWITCHES) TO ONE BEING ACTIVE, AND THE OTHER PASSIVE. ALSO, JUMPER W5 ON THE M7377A HAS TO BE REMOVED, TO ALLOW 'EOT' TO BE TRANSMITTED TO THE RECEIVER.

12.20 M7378A, FOUNDATION MODULE TEST

THIS TEST SETS THE SERIAL I/O UP AS A SOURCE AND THE FOUNDATION MODULE AS A DESTINATION. A RANDOM BUFFER IS TRANSMITTED TO THE FOUNDATION MODULE VIA THE SERIAL I/O. THEN THE FOUNDATION MODULE IS ADDRESSED AS THE SOURCE AND THE SERIAL I/O IS ADDRESSED AS THE DESTINATION. THE DATA SHOULD BE RETURNED VIA THE 'WRAP-AROUND' CABLE FROM THE FOUNDATION MODULE TO THE SERIAL I/O.

THE NEXT SUBTESTS VERIFY THAT ADDRESS '17' WILL RETURN DATA CORRECTLY, THAT THE WRONG ADDRESSES WILL NOT RETURN DATA, AND THAT THE CUSTOMER DEFINED MODE FLIP FLOP WORKS CORRECTLY.

13.0 USER AID ROUTINES

13.1. SUBX

THIS ROUTINE ENABLES THE USER TO RUN ANY SELECTED MODULE ADDRESS SUBTEST WITHOUT RUNNING THE ENTIRE PROGRAM. WHEN 'SUBX' IS SELECTED IT ASKS FOR THE 'MEMORY ADDRESS' OF THE SUBTEST TO BE EXECUTED. THIS IS TO BE THE ADDRESS OF THE 'SCOPE' ARGUMENT BEGINNING THAT SUBTEST. IF A 'SUBX' ADDRESS HAD PREVIOUSLY BEEN SET UP, THE USER CAN SIMPLY TYPE 'CR' AND THE PREVIOUSLY SELECTED TEST WILL BE RE-ENTERED.

13.2. RECBUF

H 2

SEQ 0020

THIS ROUTINE ENABLES THE USER TO EXAMINE THE CONTENTS OF THE
DL11'S RECEIVER BUFFER. WHEN SELECTED, THIS ROUTINE PRINTS THE
CONTENTS OF THE BUFFER IN THE ORDER IT WAS RECEIVED. IF THE
BUFFER IS EMPTY, A MESSAGE IS TYPED TO THAT EFFECT.

CZ
CZ

IT SHOULD BE NOTED THAT ALL DATA RECEIVED FROM THE PDM70 IS STORED IN THIS BUFFER.

13.3. TRNBUF

THIS ROUTINE ENABLES THE USER TO EXAMINE THE DATA TRANSMITTED VIA THE DL11 TO THE PDM70. THE 'RECBUF' & 'TRNBUF' ROUTINES ARE ESPECIALLY USEFUL IN TRACKING DOWN A DATA FAILURE. BY COMPARING THE TWO BUFFERS, THE USER CAN SEE EXACTLY WHERE THE FAILURE OCCURRED AND PICK OUT ANY DESIRED DATA PATTERNS.

13.4. SEND

THIS ROUTINE ENABLES THE USER TO SEND HIS OWN PROGRAM TO THE PDM70. WHEN 'SEND' IS SELECTED AN ASTERISK IS PRINTED TO INDICATE THAT THE ROUTINE IS READY TO ACCEPT INPUT. AS EACH CHARACTER IS RECEIVED IT IS ECHOED BACK TO THE TELEPRINTER AND TRANSMITTED TO THE PDM70.

THIS ROUTINE IS RUN WITH THE DL11 RECEIVER ENABLED. THIS MEANS THAT THE USER CAN USE THE 'RECBUF' ROUTINE TO EXAMINE FOR ANY DATA RETURNED BY HIS PROGRAM.

13.5. RUN

THIS ROUTINE IS USED IN CONJUNCTION WITH THE SEND ROUTINE. WHEN 'RUN' IS SELECTED, IT WILL RE-TRANSMIT THE USER'S 'SEND' PROGRAM. IF THE SERIAL INPUT OPTION IS BEING USED, THE 'SEND' PROGRAM IS TRANSMITTED AND THEN THE PROGRAM ENTERS THE 'WAIT' MODE. IF THE SERIAL INPUT OPTION IS NOT BEING USED, THE SEND PROGRAM IS CONTINUOUSLY TRANSMITTED. IN THIS CASE, THE CONSOLE SWITCHES CAN BE USED TO INCORPORATE A DELAY TIME BEFORE THE PROGRAM IS RE-TRANSMITTED. NO PROGRAM DELAY IS ISSUED WITH ALL DATA SWITCHES DOWN. ALL DATA SWITCHES UP (EXCEPT 11 & 12)* REPRESENT A MAXIMUM PROGRAM DELAY. THE USER'S SEND PROGRAM CAN BE EXAMINED AT ANYTIME BY USING THE 'TRNBUF' ROUTINE.

'CONTROL C' WHICH IS NORMALLY USED TO RETURN TO THE MONITOR IS ECHOED AND TRANSMITTED AS AN 'ETX'. SO IN THE SEND ROUTINE, 'CONTROL E' IS USED TO ESCAPE AND RETURN TO THE MONITOR.

* REFER TO CONSOLE SWITCH SETTINGS (SECTION 6.) FOR SPECIFIC SWITCH FUNCTIONS.

```
1          .TITLE CZPMACO PDM70 DIAGNOSTIC TEST
2          .ENABLE ABS
3          .ENABLE AMA
4          :AC-9017C-MC
5          :COPYRIGHT 1974,1978
6          :REVISED:      SEPTEMBER 20,1974 ,FEBRJARY 1,1978
7          :DIGITAL EQUIPMENT CORP. MAYNARD MASS. 01754
8          :PROGRAMMER: EARL L. BOUSE
9          :              MIKE MITCHELL
10         :              BILL SCHLITZKUS
11
12         :SWITCH REGISTER DEFINITIONS AND FUNCTIONS:
13
14         100000      SW15=100000      ;=1, CONTINUE ON ERROR
15         040000      SW14=40000      ;=1, LOOP ON CURRENT TEST
16         020000      SW13=20000      ;=1, SUPPRESS ERROR TYPEOUT
17         010000      SW12=10000      ;=1, SINGLE STEP DL11 OUTPUT DATA.
18         004000      SW11=4000      ;=1, TRANSMIT SAME CHARACTER.
19         002000      SW10=2000      ;=1, INHIBIT MANUAL INTERVENTION
20         001000      SW09=1000      ;=1, INHIBIT TRANSMITTER DELAY
21         000400      SW08=400
22         000200      SW07=200
23         000100      SW06=100
24         000040      SW05=40
25         000020      SW04=20
26         000010      SW03=10
27         000004      SW02=4
28
29         :REGISTER DEFINITIONS
30
31         000000      R0=%0
32         000001      R1=%1
33         000002      R2=%2
34         000003      R3=%3
35         000004      R4=%4
36         000005      R5=%5
37         000006      SP %6
38         000007      PC=%7
39
40         :INSTRUCTIONS DEFINITIONS
41
42         005746      PUSH1SP=5746
43         005726      POP1SP=5726
44         024646      PUSH2SP=24646
45         022626      POP2SP=22626
46         000240      NOP=240
47         000002      X=2
48         000002      Y 2
```


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

.SBTTL ***** NOTES *****

:NOTES:

:

:PDM-70 MUST BE CLEARED BEFORE RUNNING ANY TEST.

:
:ON POWERUP, THE FIFO IS GENERALLY CLEARED (UNLESS 'N' MODULE IS PRESENT).
:IF, DURING THE COURSE OF RUNNING THIS DIAGNOSTIC, IT BECOMES
:NECESSARY TO RESTART A SUBTEST (FOR EXAMPLE, AFTER ENCOUNTERING AN ERROR),
:THE PDM70 FIFO SHOULD BE CLEARED OUT TO INSURE THAT GARBAGE WILL
:NOT ACCIDENTLY BE LEFT IN THE FIFO WHICH WOULD SUBSEQUENTLY
:GIVE AN ERRONOUS DATA ERROR.

:MODULE SETUP:

:*****
:MODULE UNDER TEST *M7379(CLOCK) * M7379 (CLOCK) *
:*****
: *M973 (CABLE) * M598 (COUPLER) *
:*****

:SYSTEM TEST SETUP (USING SERIAL I/O)

:*****
:MODULE UNDER TEST * M7379 (CLOCK) * M7379 *
:*****
: *M7380 (CONTROL MODULE) *
:*****

:*****
: SERIAL I/O CABLE *
:*****

:NOTE: JUMPER 'L' SHOULD BE IN TO ALLOW POWER UP TO ACCESS THE CONTROL MODULE.

```
90      ;LOAD TRAP ADDRESSES '0-1000' WITH THE 'IOT' TRAP
91      000000
92      000200
93      .REPT 200
94      .+2
95      .ENDR
96      .=20
97      000020 020440      ERTRAP      ;ERROR TRAP REPORTER ROUTINE.
98      000022 000340      340
99      000024 022722      PWRFAL      ;POWER FAIL HANDLER
100     000026 000340      340
101     000030 000030      .=30
102     000030 001200      EMTSRV      ;EMT TRAP, EMT DISPATCH SERVICE
103     000032 000340      340
104     000060 000060      .=60
105     000060 014716      XTTYIN      ;TELEPRINTER KEYBOARD ROUTINE
106     000062 000340      340
107     000176 000176      .=176
108     000176 000000      SWSWR: 0      ;SOFTWARE SWITCH REGISTER
109     000200 000200      .-200
110     000200 000137 001376      JMP MONITR      ;PROGRAM KEYBOARD MONITOR ROUTINE.
111
112      .SBTTL EMT TRAP EQUIVALENCE TABLE
113
114      104000      PRCNTR=EMT      ;SUBROUTINE TO PRINT CONTROL CHARACTER IN R1
115      104001      SCOPE=EMT+1      ;LOGIC TEST SCOPE SUBROUTINE
116      104002      SAVREG=EMT+2      ;SUBROUTINE TO SAVE 'R0-R5' ON STACK
117      104003      GETREG=EMT+3      ;SUBROUTINE TO GET 'R0-R5' FROM STACK
118      104004      DELAY=EMT+4      ;SUBROUTINE TO WAIT FOR DL11 RECV.
119      104005      RECVRO=EMT+5      ;SUBROUTINE TO SET UP THE DL11 O'S RECEIVER.
120      104006      LDCHRO=EMT+6      ;SUBROUTINE TO TRANSMIT A SINGLE CHAR. VIA DL 'O'
121      104007      LDPGMO=EMT+7      ;SUBROUTINE TO TRANSMIT THE DATA IN CALL+2 VIA DL 'O'
122      104010      TYPEIT=EMT+10      ;SUBROUTINE TO PRINT CHARACTER IN 'R1'
123      104011      RANDOM=EMT+11      ;SUBROUTINE TO CREATE A RANDOM DATA BUFFER.
124      104012      PRINT=EMT+12      ;SUBROUTINE TO PRINT ASCII MESSAGES.
125      104013      TTYIN=EMT+13      ;SUBROUTINE TO INPUT VIA KEYBOARD
126      104014      PRTOCT=EMT+14      ;SUBROUTINE TO PRINT A 6 DIGIT OCTAL NO.
127      104015      ASEMBL=EMT+15      ;SUBROUTINE TO ASSEMBLE CHARACTERS INTO OCTAL VALUE
128      104016      SPACE=EMT+16      ;SUBROUTINE TO PRINT SPACES
129      104017      TSTTKS=EMT+17      ;SUBROUTINE TO TEST FOR KEYBOARD FLAGS
130      104020      DELAYL=EMT+20      ;SUBROUTINE TO SETUP A LONG DISPLAY DELAY
131      104021      NULL=EMT+21      ;SUBROUTINE TO TRANSMIT A NULL PRINTER CHAR.
132      104022      MODERR=EMT+22      ;SUBROUTINE TO REPORT MODULE ERRORS.
133      104023      NULL1=EMT+23      ;SUBROUTINE TO TRANSMIT 12 NULL CHAR.'S.
134      104024      DESTIN=EMT+24      ;SUBROUTINE TO SETUP DESTINATION MODULE.
135      104025      SOURCE=EMT+25      ;SUBROUTINE TO SETUP A SOURCE MODULE
136      104026      ADDRES=EMT+26      ;SUBROUTINE TO REQUEST & SAVE MODULE ADDRESS
137      104027      ADCNVT=EMT+27      ;SUBROUTINE TO TAKE & STORE A/D CONVERSIONS
138      104030      BCDBIN=EMT+30      ;SUBROUTINE TO CONVERT 'BCD' TO BINARY
139      104031      AVERAG=EMT+31      ;SUBROUTINE TO AVERAGE 'N' NUMBERS
140      104032      CHANEL=EMT+32      ;SUBROUTINE TO REQUEST & STORE A/D CHANNEL
141      104033      BINDEC=EMT+33      ;SUBROUTINE TO CONVERT BINARY TO DEC.
142      104034      WAITGN=EMT+34      ;SUBROUTINE TO TEST GAIN ACCURACY
143      104035      SETUP=EMT+35      ;SUBROUTINE TO SETUP THE '*R' RESTART ADDR.
144      104036      NODLAY=EMT+36      ;SUBROUTINE TO INHIBIT TRANSMITTER DELAY
145      104037      PRTRBF=EMT+37      ;SUBROUTINE TO PRINT CONTENTS OF RECVR BUFFER
```

```
146 ;*****
147 ;SBTTL EMT DISPATCH SERVICE ROUTINE
148 ;ARGUMENT OF EMT IS EXTRACTED AND USED AS OFFSET TO OBTAIN POINTER
149 ;TO THE SELECTED SUBROUTINE.
150 ;*****
151
152      001200      001200      . = 1200
153 001200 011646      EMTSRV: MOV      (SP), -(SP)      ;GET PC FOR TO RETURN
154 001202 162716      SUB      #2, (SP)      ;PC OF EMT
155 001206 017616      MOV      @ (SP), (SP)      ;GET EMT
156 001212 005716      TST      (SP)      ;IS EMT VALID?
157 001214 001001      BNE      EMTOK
158 001216 000000      HALT
159 001220 006316      EMTOK: ASL      (SP)      ;INVALID EMT
160 001222 042716      BIC      #177001, (SP)      ;MULTIPLY EMT ARG BY '2'
161 001226 062716      ADD      #EMTTAB, (SP)      ;CLEAR UNWANTED BITS
162 001232 017616      MOV      @ (SP), (SP)      ;POINTER TO SUBROUTINE ADDRESS
163 001236 000136      JMP      @ (SP)+      ;SUBROUTINE ADDRESS
164 ;GO TO SUBROUTINE
165 ;EMT DISPATCH TABLE
166
167 001240 021376      EMTTAB: XPRCNT      ;SUBROUTINE TO PRINT CONTROL CHAR. IN R1.
168 001242 020636      XSCOPE      ;MODULE TEST SCOPE LOOP ROUTINE
169 001244 023026      XSAVRG      ;SUBROUTINE TO SAVE 'R1-R5' ON STACK
170 001246 023102      XGETRG      ;SUBROUTINE TO RETRIEVE 'R1-R5' FROM STACK
171 001250 022126      XDLAYL      ;SUBROUTINE TO WAIT FOR DATA FROM DL11 RECEIVER
172 001252 016130      XRECRO      ;SUBROUTINE TO SET UP DL 0'S RECEIVER.
173 001254 017302      XLDCHR      ;SUBROUTINE TO TRANSMIT A SINGLE CHAR
174 001256 017320      XLDADD      ;SUBROUTINE TO TRANSMIT DATA FROM ADDRESS IN CALL+2.
175 001260 021340      XTYPIT      ;SUBROUTINE TO PRINT CHARACTER IN 'R1'
176 001262 020776      XRANGN      ;SUBROUTINE TO CREATE A RANDOM DATA BUFFER.
177 001264 021650      XPRINT      ;SUBROUTINE TO PRINT ASCII MESSAGES.
178 001266 014716      XTTYIN      ;SUBROUTINE TO INPUT VAI KEYBOARD.
179 001270 022010      XOCTPR      ;SUBROUTINE TO PRINT A '6' DIGIT OCTAL NO.
180 001272 023156      XASEMB      ;SUBROUTINE TO ASSEMBLE A ONE WORD NO.
181 001274 021234      XSPACE      ;SUBROUTINE TO TYPE SPACES
182 001276 021264      TKSFLG      ;SUBROUTINE TO TEST FOR KEYBOARD FLAG.
183 001300 022126      XDLAYL      ;SUBROUTINE TO SET UP A LONG DELAY.
184 001302 021554      XNULL      ;SUBROUTINE TO ISSUE NULL CHARACTERS AFTER RESET''.
185 001304 020540      XERMES      ;SUBROUTINE TO REPORT MODULE ERRORS
186 001306 021564      XNULL1      ;SUBROUTINE TO TRANSPORT '12' NULL CHAR.'S
187 001310 017164      XDSTIN      ;SUBROUTINE TO SET UP A DESTINATION MODULE
188 001312 017144      XSOURCE      ;SUBROUTINE TO SET UP A SOURCE MODULE
189 001314 020374      XADRES      ;SUBROUTINE TO REQUEST & SAVE MODULE ADDRESS
190 001316 006670      XADCNT      ;SUBROUTINE TO TAKE & STORE A/D CONVERSIONS
191 001320 022430      XBCDBIN      ;SUBROUTINE TO CONVERT 'BCD' TO BINARY
192 001322 015702      XAVRAGE      ;SUBROUTINE TO AVERAGE 'N' NUMBERS
193 001324 020512      XCHANEL      ;SUBROUTINE TO REQUEST & STORE A/D CHANNEL.
194 001326 022574      XBINDEC      ;SUBROUTINE TO CONVERT BINARY TO DECIMALS.
195 001330 006474      XWATGN      ;SUBROUTINE TO TEST GAIN ACCURACY.
196 001332 021276      XSETUP      ;SUBROUTINE TO SETUP THE 'R' RESTART ADDR.
197 001334 021544      XNODLY      ;SUBROUTINE TO INHIBIT TRANSMITTED DELAY
198 001336 022246      XPRTRB      ;SUBROUTINE TO PRINT CONTENTS OF RECV. BUFFER
```

198				.SBTTL REGISTER ADDRESSES	
199					
200	001340	177776		PSW:	177776 ;ADDRESS OF PROCESSOR STATUS REG.
201	001342	177560		TKS:	177560 ;ADDRESS OF KEYBOARD STATUS REG.
202	001344	177562		TKB:	177562 ; " " BUFFER
203	001346	177564		TPS:	177564 ; " " PRINTER STATUS REG.
204	001350	177566		TPB:	177566 ; " " PRINTER BUFFER REG.
205	001352	177570		SWR:	177570 ; " " SWITCH REG.
206	001354	177571		SWRO:	177571 ; " " HIGH BYTE
207					
208				;DL11 REGISTER ADDRESSES	
209					
210	001356	175610		RCSR0:	175610 ;ADDRESS OF UNIT 0'S DL11 REC. CSR
211	001360	175612		RBUF0:	175612 ;ADDRESS OF UNIT 0'S DL11 REC. BUFFER
212	001362	175614		XCSR0:	175614 ;ADDRESS OF UNIT 0'S TRANS. CSR
213	001364	175616		XBUF0:	175616 ;ADDRESS OF UNIT 0'S DL11 TRANS. BUFFER
214	001366	000300		RINT0:	300 ;ADDRESS OF UNIT 0'S REC. VECTOR
215	001370	000302		RLVL0:	302
216					
217	001372	000304		XINT0:	304 ;ADDRESS OF UNIT 0'S DL11 TRANS. VECTOR
218	001374	000306		XLVL0:	306 ;ADDRESS OF UNIT 1'S DL11 TRANS. VECTOR
219					
220				.SBTTL DEFINITIONS OF THE 'PDM-70' CONTROL CHARACTERS.	
221					
222		000021		DC1=021	;ENAR = SOURCE
223		0C0022		DC2=022	;ENABLE DESTINATION
224		000023		DC3=023	;GO
225		000024		DC4=024	;DO NOT RECIRCULATE
226		000003		ETX=003	;END OF TEXT
227		000002		STX=002	;START OF TEXT
228		000026		SYN=026	;SYNCHRONIZE (DELAY)
229		000001		SOH=001	;START OF HEADER
230		000017		SI=017	;SHIFT IN.
231		000004		EOT=004	;END OF TRANSMISSION
232		000005		ENQ=005	;ENQUIRY.
233					
234				;*****	
235				.SBTTL KEYBOARD MONITOR	
236				;*****	
237					
238	001376	012706	001000	MONITR: MOV #1000, SP	;SET UP STACK
239	001402	012746	000340	MOV #340, -(SP)	;SET PROC. PRIORITY @7
240	001406	012746	001414	MOV #1\$, -(SP)	
241	001412	000002		RTI	
242	001414	012737	001430	000004 1\$: MOV #2\$, 4	;SETUP FOR TIMEOUT
243	001422	005777	177724	TST @SWR	;HARDWARE SWITCH REGISTER?
244	001426	000404		BR 3\$	;YES
245	001430	012737	000176	001352 2\$: MOV #SWSWR, SWR	;NO-USE SOFTWARE SWITCH REG.
246	001436	022626		POP2SP	;POP THE STACK
247	001440	012737	000006	000004 3\$: MOV #6, 4	;RESTORE TRAP CATCHER
248	001446	104021		NULL	
249	001450	000005		RESET	
250	001452	104021		NULL	
251	001454	005037	032144	CLR DLYSWH	;CLR SOFTWARE SW.
252	001460	005037	032122	CLR PRTSWH	;CLR SOFTWARE SW.
253	001464	005037	032150	CLR SFNDSW	;CLR SOFTWARE SW.

CZPMAC0 PDM70 DIAGNOSTIC TEST MACY11 30A(1052) 20-JAN-78 09:11 ^{B 3} PAGE 6
CZPMAC.P11 19-JAN-78 14:50 KEYBOARD MONITOR

SEQ 0027

254	001470	005737	032120	TST	MTRSWH	;PROGRAM BEEN INITIALIZED?
255	001474	001101		BNE	MONTR5	;YES
256	001476	005237	032120	INC	MTRSWH	;NO
257	001502	104012		PRINT		
258	001504	023353		TITLE		;PRINT PROGRAM HEADER

```

259
260
261
262
263
264 001506 005037 032132 MONTR1: CLR SIOSWH
265 001512 104012 PRINT
266 001514 023775 MESO ;TEXT 'IS INPUT VIA SERIAL I/O?
267 001516 104013 TTYIN ;WAIT FOR INPUT
268 001520 122737 000131 015330 CMPB #131,INBUF ;WAS 'Y' TYPED?
269 001526 001031 BNE MONT1A ;NO, SETUP DL11 INPUT
270 001530 104026 ADDRESS ;REQUEST SERIAL I/O ADDRESS
271 001532 110037 002214 MOVB R0,IADRS0 ;SET UP ALL ADDRESSES WHERE
272 001536 110037 002220 MOVB R0,IADRS1 ;SERIAL INTERFACE IS USED.
273 001542 110037 002224 MOVB R0,IADRS2
274 001546 110037 002230 MOVB R0,IADRS3
275 001552 110037 003120 MOVB R0,IADRS4
276 001556 110037 003124 MOVB R0,IADRS5
277 001562 110037 003132 MOVB R0,IADRS6
278 001566 110037 017547 MOVB R0,IADRS7
279 001572 110037 017643 MOVB R0,IADRS8
280 001576 110037 017202 MOVB R0,IADRS9
281 001602 110037 011167 MOVB R0,IADR10
282 001606 005237 032132 INC SIOSWH ;YES, SET SW.
283 001612 104012 MONT1A: PRINT
284 001614 027055 MFS63 ;REQUEST DL11 ADDRESS & VECTOR
285 001616 104015 ASIMBL ;WAIT AND DECODE
286 001620 005700 TST R0 ;WAS AN ADDRESS ENTERED?
287 001622 001416 BEQ MONTR3 ;NO, USE STANDARD ADDRESS.
288 001624 012702 001356 MOV #RCSR0,R2 ;SET UP TO LOAD ADDRESS
289 001630 012703 000004 MONTR2: MOV #4,R3
290 001634 010022 MOV R0,(R2)+
291 001636 062700 000002 ADD #2,R0 ;ADD '2' TO THE ADDRESS
292 001642 005303 DEC R3
293 001644 001373 BNE -10
294 001646 022702 001376 CMP #XLVL0+2,R2 ;LOADED VECTOR ADDRESSES?
295 001652 001402 BEQ MONTR3 ;YES, EXIT
296 001654 010400 MOV R4,R0
297 001656 000764 BR MONTR2
298 001660 012777 016740 177500 MONTR3: MOV #RECVER,@RINT0 ;SET UP RECEIVER SERVICE ADDRESS
299 ;RINT0=DL-11 VECTOR (300)
300 001666 012777 000200 177474 MOV #200,@RLVL0 ;BR LEVEL '4'
301 001674 104012 MONTR4: PRINT
302 001676 023415 HEADER ;PRINT TEST PROTOCOL
303 001700 012737 001506 032124 MONTR5: MOV #MONTR1,RVECTR ;SET UP THE 'RESTART' ADDR, POINTER
304 001706 104012 PRINT
305 001710 031706 DOT ;PRINT DOT TO INDICATE READY

```



```
306
307
308
309
310 001712 005037 015330
311 001716 104013
312 001720 022737 000103 015330
313 001726 001007
314 001730 005737 032224
315 001734 001455
316 001736 004737 023304
317 001742 000177 030256
318 001746 012701 023475
319 001752 005003
320 001754 012702 015330
321 001760 122711 000045
322 001764 001403
323 001766 122711 000040
324 001772 001002
325 001774 105721
326 001776 000766
327 002000 122122
328 002002 001022
329 002004 122711 000054
330 002010 001373
331 002012 006303
332 002014 005726
333 002016 016337 002076 032224
334 002024 016337 002076 032126
335 002032 062737 000004 032126
336 002040 004737 023304
337 002044 000173 002076
338 002050 005203
339 002052 122711 000100
340 002056 001404
341 002060 122721 000054
342 002064 001733
343 002066 000771
344
345 002070 104012
346 002072 031702
347 002074 000706
348

;*****
;THIS SUBROUTINE DECODES THE USER'S INPUT AND EXECUTES THE SELECTED TEST
;*****
DECODE: CLR      INBUF
        TTYIN
        CMP      #103,INBUF      ;CALL KEYBOARD ROUTINE
        BNE      DECOD1          ;WAS 'C' TYPED TO CONTINUE LAST TEST?
        TST      RESTR          ;NO, DECODE INPUT
        BEQ      NMATCH          ;YES, HAS A RESTART ADDR. BEEN SET UP?
        JSR      PC,      UPDAT1 ;NO, ILLEGAL ENTRY.
        JMP      @RESTR          ;CHECK FOR SOFTWARE SWR
        MOV      #TSTLST,R1      ;YES, RESTART LAST TEST
        CLR      R3              ;SET UP MESSAGE ADDR. POINTER
        MOV      #INBUF,R2       ;OFFSET REG.
        CMPB     #45,(R1)        ;SET UP TTY BUFFER POINTER
        BEQ      .+10
        CMPB     #40,(R1)        ;CHAR. = TO 'SPACE'
        BNE      .+6             ;NO
        TSTB     (R1)+           ;YES, SKIP CHAR.
        BR       RECYCL
MATCH:  CMPB     (R1)+,(R2)+      ;COMPARE BUFFERS
        BNE      FLUSH          ;NOT EQUAL, SET UP NEXT WORD
        CMPB     #54,(R1)
        BNE      MATCH          ;NO, COMPARE NEXT CHAR.
        ASL      R3              ;SET UP OFFSET
        POP1 SP
        MOV      TSTABL(R3),RESTR ;SET UP A RESTART ADDRESS
        MOV      TSTABL(R3),AVECTR
        ADD      #4,AVECTR      ;SET UP TO RE-ADDRESS MODULE
        JSR      PC,      UPDAT1 ;CHECK FOR SOFTWARE SWR
        JMP      @TSTABL(R3)    ;EXECUTE SELECTED TEST.
        INC      R3              ;INCREMENT OFFSET CNTR.
        CMPB     #100,(R1)      ;TEST FOR 'a'
        BEQ      NMATCH          ;YES, END OF MESSAGE.
        CMPB     #54,(R1)+      ;CHAR = COMMA?
        BEQ      RECYCLE        ;YES, COMPARE NEXT WORD.
        BR       FLUSH+2        ;NO, KEEP GOING.
        FLUSH:
        NMATCH: PRINT
        QMARK
        BR       DECODE
        ;ILLEGAL ENTRY, TYPE '?'
        ;GET NEW INPUT.
```

```
349
350
351
352
353 002076 002160
354 002100 003450
355 002102 004316
356 002104 004412
357 002106 004606
358 002110 005014
359 002112 005416
360 002114 005514
361 002116 006356
362 002120 007072
363 002122 010006
364 002124 010126
365 002126 011150
366 002130 011360
367 002132 011506
368 002134 003254
369 002136 012050
370 002140 012062
371 002142 012370
372 002144 013664
373 002146 021142
374 002150 022224
375 002152 022232
376 002154 022264
377 002156 022360

;*****
;TABLE FOR TESTS SELECTABLE VIA KEYBOARD
;*****

TSTABL: M7380A      ;CONTROL MODULE TEST
          M7381A      ;BCD INPUT MODULE ADDRESS TEST
          M7381E      ;BCD INPUT MODULE EXERCISER TEST
          M7382A      ;BCD OUTPUT MODULE ADDRESSING TEST
          BCDIO        ;BCD INPUT/OUTPUT EXERCISER TEST
          M7383A      ;A/D MODULE ADDRESS TEST
          M7383C      ;A/D MODULE CALIBRATION TEST
          M7383R      ;A/D REPEATIBILITY TEST
          M7383G      ;A/D GAIN ACCURACY TEST
          M7384A      ;D/A MODULE ADDRESS TEST
          M7384E      ;D/A OUTPUT MODULE EXERCISER TEST
          M7385A      ;SERIAL INPUT/OUTPUT MODULE ADDRESS TEST
          M7385I      ;SERIAL I/O INTERFACE TEST
          M7385T      ;SERIAL INPUT/OUTPUT TTL TEST
          M7386A      ;KEYBOARD/DISPLAY MODULE ADDRESS TEST
          M7387A      ;HARDWARE READ-IN MODULE TEST
          M7388A      ;CHARACTER I/O (IN-HOUSE) MODULE ADDRESS TEST
          M7388F      ;CHARACTER I/O (FIELD) MODULE ADDRESS TEST
          M7377A      ;REMOTE SERIAL MODULE TEST
          M7378A      ;FOUNDATION MODULE TEST
          SUBX          ;SUBTEST SELECTOR ROUTINE
          RECBUF        ;ROUTINE TO PRINT CONTENTS OF DL RECV BUFFER
          TRNBUF        ;ROUTINE TO PRINT CONTENTS OF DL TRNS BUFFER
          SEND          ;ROUTINE TO TRANSMIT CHAR.'S FROM TTY
          RUN           ;ROUTINE TO LOAD & RUN THE SEND PROGRAM
```



```

378 :*****
379 :SBTTL M7380 CONTROL MODULE TEST.
380 :THIS TEST COMPLETELY EXERCISES THE PDM-70 'CONTROL MODULE' USING THE
381 :PDP-11 AS THE MASTER 'SOURCE/DESTINATION' MODULE. THE TEST TAKES THE
382 :MODULE THRU THE INITIALIZATION, PROGRAM, ADDRESS AND DATA MODES RESPECTIVELY.
383 :*****
384
385 002160 104012 M7380A: PRINT
386 002162 024042 MEST ;TEXT 'CONTROL MODULE TEST'
387 002164 000240 NOP
388 002166 005037 032222 CLR LOPSWH
389 002172 104035 SETUP ;SET UP TEST PARAMETERS.
390 002174 005037 032144 CLR DLYSWH ;ENABLE TRANSMITTER DELAYS
391
392 :*****
393 :LOAD '2' PROGRAMS INTO THE CONTROLS 'FIFO' AND CHECK THAT
394 :THE CONTROL MODULE ENTERS THE ADDRESS MODE.
395 :*****
396
397 002200 000240 NOP
398 002202 000240 NOP
399 002204 104007 LDPGMO ;LOAD THE FOLLOWING PROGRAM.
400 002206 002212 PRGM1-1
401 002210 000412 BR TAGB
402 002212 002 .BYTE STX
403 002213 021 PRGM1: .BYTE DC1
404 002214 075 IADRS0: .BYTE 75
405 002215 001 .BYTE SOH
406 002216 061 .BYTE 61
407 002217 022 .BYTE DC2 ;ALERT DESTINATION
408 002220 075 IADRS1: .BYTE 75
409 002221 023 .BYTE DC3
410 002222 024 PRGM2: .BYTE DC4 ;START OF 2ND PROGRAM
411 002223 021 .BYTE DC1
412 002224 075 IADRS2: .BYTE 75
413 002225 001 .BYTE SOH
414 002226 061 .BYTE 61
415 002227 022 .BYTE DC2
416 002230 075 IADRS3: .BYTE 75
417 002231 061 .BYTE 61
418 002232 063 .BYTE 63
419 002233 023 .BYTE DC3
420 002234 003 END2: .BYTE ETX
421 002236 .EVEN
422
423 002236 005737 032132 TAGB: TST SIOSWH ;SERIAL I/O INPUT?
424 002242 001020 BNE TAGOB1 ;YES, JUST LOOK FOR DATA
425 002244 012701 002213 MOV #PRGM1,R1 ;NO, VERIFY 1ST PROGRAM
426 002250 005737 032222 TST LOPSWH ;LOOPING FROM LAST TEST?
427 002254 001401 BEQ .+4 ;NO, DON'T LOOK FOR 'STX'
428 002256 005301 DEC R1 ;YES, SET UP TO LOOK FOR 'STX'
429 002260 005237 032222 INC LOPSWH

```

```
430 002264 122221      CMP1:  CMPB    (R2)+,(R1)+      ;COMPARE RECV'D/TRANSMITTED DATA
431 002266 001403      BEQ      .+10              ;
432 002270 104022      MODERR                      ;RECV'D/TRANS ADDRESS DATA DIFFERENT
433 002272 030471      ERR2                        ;
434 002274 000412      BR        CT2              ;EXIT ON ERROR.
435 002276 122701 002222 CMPB    #PRGM2,R1        ;CHK'D ALL DATA?
436 002302 001370      BNE      CMP1              ;NO
437
438 ;AT THIS POINT THE MODULE SHOULD BE IN THE 'DATA MODE'
439 ;THIS NEXT SUBTEST SENDS THE CHAR. 'A' AND CHECKS
440 ;THAT IT IS RETURNED AS DATA.
441
442 002304 104006      TAGOB1: LDCHRO
443 002306 000101      'A
444 002310 122722 000101 CMPB    #'A',(R2)+      ;SEND CHAR. 'A'
445 002314 001402      BEQ      CT2              ;WAS 'A' RETURNED?
446 002316 104022      MODERR                      ;YES
447 002320 031365      ERR19                      ;MODULE DIDN'T ENTER DATA MODE
448
449 *****
450 ;THE CONTROL MODULE SHOULD NOW BE IN THE 'DATA MODE'. THE FOLLOWING
451 ;SUBTEST CREATES A RANDOM '500' WORD DATA BUFFER AND TRANSFERS IT TO THE
452 ;CONTROL MODULE. THIS DATA IS VERIFIED WHEN IT IS RECEIVED BACK FROM THE
453 ;CONTROL MODULE.
454 *****
455
456 002322 104001 000002 CT2:    SCOPE,2      ;TEST 2
457 002326 104011      RANDOM                      ;CREATE A RANDOM DATA BUFFER
458 002330 104007      LDPGMO                      ;TRANSMIT DATA FROM FOLLOWING ADDRESS.
459 002332 017670      TRNBFO
460 002334 012701 017670 MOV     #TRNBFO,R1
461 002340 122221      CMP2:  CMPB    (R2)+,(R1)+      ;REVC'D & TRANS DATA EQUAL?
462 002342 001403      BEQ      .+10              ;YES
463 002344 104022      MODERR                      ;RECV'D DATA DOESN'T EQUAL TRANS DATA
464 002346 030526      ERR3
465 002350 000411      BR        CT3
466 002352 005737 016220 TST     PARITY          ;PARITY ERROR FLAG SET?
467 002356 001403      BEQ      CT3A              ;NO, DATA GOOD
468 002360 104022      MODERR                      ;YES, PARITY ERROR ON LAST TRANSFER
469 002362 030677      ERR7
470 002364 000403      BR        .+10
471 002366 022701 020372 CT3A:  CMP     #TRNEND,R1      ;CHK'D WHOLE BUFFER?
472                                     ;CORRECTED 7/1/74.
473 002372 001362      BNE      CMP2
```

```

474
475
476
477
478
479
480
481 002374 104001 000003 CT3: SCOPE,3 ;TEST 3
482 002400 104006 LDCHRO
483 002402 000004 EOT ;TRANSMIT THE 'EOT' CHAR.
484 002404 012701 002222 MOV #PRGM2,R1
485 002410 122722 000004 CMPB #EOT,(R2)+ ;CHK THAT 'EOT' WAS RETURNED
486 002414 001403 BEQ .+10
487 002416 104022 MODERR
488 002420 030603 ERR5 ;EOT CHAR WASN'T RETURNED
489 002422 000422 BR CT4 ;EXIT ON ERROR
490 002424 005737 032132 TST SIOSWH ;SERIAL I/O INPUT?
491 002430 001010 BNE TAGOA ;YES, JUST VERIFY DATA
492 002432 122221 CMP3: CMPB (R2)+,(R1)+ ;COMPARE DATA OF THE SECOND PROGRAM
493 002434 001403 BEQ .+10
494 002436 104022 MODERR ;ADDRESS ERROR IN 2ND PROGRAM
495 002440 030544 ERR4
496 002442 000412 BR CT4
497 002444 122701 002234 CMPB #END2,R1 ;DONE
498 002450 001370 BNE CMP3 ;NO
499
500 ;SEND A CHAR. TO VERIFY THE 2ND PROGRAM IS
501 ;IN THE DATA MODE
502
503 002452 104006 TAGOA: LDCHRO
504 002454 000101 'A
505 002456 122722 000101 CMPB #'A,(R2)+ ;WAS THE 'A' RECV'D?
506 002462 001402 BEQ CT4 ;YES
507 002464 104022 MODERR ;2ND PROGRAM DIDN'T ENTER DATA MODE
508 002466 031321 ERR18

```

```

509
510
511
512
513
514
515 002470 104001 000004      CT4:  SCOPE,4          ;TEST 4
516 002474 104006              LDCHRO
517 002476 000004              EOT
518 002500 105722              TSTB (R2)+          ;'EOT' SHOULD RE-ADDRESS 1ST PROGRAM
519 002502 005737 032132      TST  SIOSWH         ;ADD '1' TO BUFFER POINTER
520 002506 001012              BNE  TAGOC          ;SERIAL I/O INPUT?
521 002510 012701 002213      MOV  #PRGM1,R1       ;YES, JUST CHECK DATA
522 002514 122122              CMP4: CMPB (R1)+,(R2)+
523 002516 001403              BEQ  .+10
524 002520 104022              MODERR              ;1ST PROGRAM DIDN'T RECIRCULATE
525 002522 030634              ERR6
526 002524 001012              BNE  CT5
527 002526 022701 002222      CMP  #PRGM2,R1
528 002532 001370              BNE  CMP4
529
530
531
532 002534 104006              ;SEND A CHAR. TO VERIFY THAT THE 1ST PROGRAM ENTERED THE DATA MODE
533 002536 000101              TAGOC: LDCHRO
534 002540 122722 000101      'A
535 002544 001402              CMPB #'A',(R2)+      ;WAS CHAR RETURNED?
536 002546 104022              BEQ  CT5             ;YES
537 002550 030634              MODERR              ;1ST PROGRAM DIDN'T RE-ENTER DATA MODE
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
  
```

```
538
539
540 ;*****
541 ;THIS SUBTEST ISSUES ANOTHER 'EOT' CHARACTER. THIS SHOULD ENABLE THE
542 ;ADDRESS OF THE 1ST PROGRAM TO BE OUTPUT SINCE THE SECOND PROGRAM
543 ;CONTAINED A 'DC4' AND SHOULD HAVE BEEN FLUSHED.
544 ;*****
545 002552 104001 000005 CT5: SCOPE,5 ;TEST 5
546 002556 104006 LDCHRO
547 002560 000004 EOT ;'EOT' SHOULD ENABLE ADDRESS MODE
548 002562 122722 000004 CMPB #EOT,(R2)+ ;CHECK THAT 'EOT' WAS RETURNED
549 002566 001403 BEQ .+10
550 002570 104022 MODERR ;'EOT' CHAR. WASN'T RETURNED
551 002572 030603 ERR5
552 002574 000424 BR CT6 ;EXIT ON ERROR
553 002576 005737 032132 TST SIOSWH ;SERIAL I/O INPUT?
554 002602 001012 BNE TAGOD ;YES,
555 002604 012701 002213 MOV #PRGM1,R1
556 002610 122122 CMP5: CMPB (R1)+,(R2)+ ;CHECK RECV'D ADDR. AGAINST PROGRAM 1.
557 002612 001403 BEQ .+10 ;PROGRAM DIDN'T RECIRCULATE PROPERLY
558 002614 104022 MODERR
559 002616 030634 ERR6
560 002620 000412 BR CT6
561 002622 122701 002222 CMPB #PRGM2,R1
562 002626 001370 BNE CMP5
563 002630 104006 TAGOD: LDCHRO
564 002632 000101 'A
565 002634 122722 000101 CMPB #'A,(R2)+
566 002640 001402 BEQ .+6
567 002642 104022 MODERR
568 002644 030634 ERR6
```

```
569
570
571
572
573
574
575
576 002646 104001 000006
577 002652 104011
578 002654 012700 017670
579 002660 112720 000002
580 002664 112720 000021
581 002670 113720 002214
582 002674 112720 000001
583 002700 112720 000061
584 002704 112720 000022
585 002710 113720 002214
586 002714 112720 000017
587 002720 112737 000023 017767
588 002726 112737 000003 017770
589
590 002734 104007
591 002736 017670
592 002740 104004
593 002742 105722
594 002744 001003
595 002746 104022
596 002750 030471
597 002752 000417
598 002754 012701 017670
599 002760 005737 032132
600 002764 001402
601 002766 062701 000010
602 002772 122122
603 002774 001403
604 002776 104022
605 003000 030526
606 003002 000403
607 003004 022701 017767
608 003010 001370

:*****
:THIS TEST CHECKS THAT ALL '64' LOCATIONS OF THE CONTROLS 'FIFO' CAN
:BE ACCESSED. THIS IS DONE BY LOADING ONE '64' CHARACTER PROGRAM IN
:THE 'FIFO'. IN THIS PROGRAM, '56' CHARACTERS ARE RANDOM LITERAL
:CHARACTERS ENTERED UNDER AN 'SI' COMMAND.
:*****

CT6:  SCOPE,6          ;TEST 6
      RANDOM          ;CREATE A RANDOM DATA BUFFER
      MOV #TRNBFO,R0  ;SET UP TO LOAD AN ADDRESS ON THE DATA
      MOVB #STX,(R0)+ ;ENTER ADDRESS MODE
      MOVB #DC1,(R0)+ ;ALERT SOURCE IF SERIAL I/O IS OUT THERE.
      MOVB IADRS0,(R0)+;ADDRESS INPUTTED VIA USER
      MOVB #SOH,(R0)+ ;MODE '1'; WAIT FOR DATA
      MOVB #61,(R0)+
      MOVB #DC2,(R0)+ ;ALERT DESTINATION FOR SERIAL I/O
      MOVB IADRS0,(R0)+;ADDRESS INPUTTED VIA USER
      MOVB #SI,(R0)+  ;SEND '55' LITERAL CHARACTERS
      MOVB #DC3,TRNBFO+77;LOAD THE '64' CHAR.
      MOVB #ETX,TRNBFO+100;TERMINATE THE PROGRAM

      LDPGMO          ;SEND THE PROGRAM
      TRNBFO
      DELAY
      TSTB (R2)+      ;WAIT FOR DATA TO RETURN
      BNE .+10        ;WAS ANY DATA RETURNED?
      MODERR          ;YES
      ERR2            ;CONTROL MODULE DIDN'T RETURN ANY DATA
      BR CT7          ;EXIT ON ERROR
      MOV #TRNBFO,R1  ;SET UP TO VERIFY DATA
      TST SIOSWH      ;USING SERIAL I/O?
      BEQ .+6         ;NO. VERIFY ADDRESS AS WELL AS DATA
      ADD #10,R1      ;YES, MOVE POINTER TO VERIFY DATA ONLY
      CMPB (R1)+,(R2)+
      BEQ .+10
      MODERR          ;DATA ERROR
      ERR3
      BR CT7          ;EXIT ON ERROR
      CMP #TRNBFO+77,R1;DONE?
      BNE CMP6        ;NO

CMP6:
```



```
609
610
611      ;*****
612      ;AT THIS POINT THE PROGRAM, ADDRESS AND DATA MODES HAVE BEEN TESTED.
613      ;THIS SUBTEST ISSUES ANOTHER 'STX' CHARACTER TO GET THE CONTROL MODULE
614      ;BACK INTO THE PROGRAM MODE.
615      ;*****
616 003012 104001 000007      CT7:  SCOPE,7          ;TEST 7
617 003016 104006          LDCHRO
618 003020 000002          STX          ;ISSUE 'STX' TO RE-ENTER PROGRAM MODE
619 003022 122722 000002      CMPB  #STX,(R2)+
620 003026 001402          BEQ  .+6
621 003030 104022          MODERR      ;THE 'STX' CHARACTER WASN'T RETURNED
622 003032 030440          ERR1
623
624      ;*****
625      ;THIS SUBTESTS TESTS THE DELAY TIMES OF THE 'SYN' CHARACTER. ALL THE
626      ;DELAY TIMES OF '1-9' ARE TESTED IN ORDER. THE TEST MAKES '2' CHECKS
627      ;ON EACH TIME. FIRST IS THAT THE DELAY ISN'T TOO SHORT AND SECOND THAT
628      ;THE DELAY ISN'T TOO LONG. THIS TEST IS PERFORMED BY LOADING
629      ;'9' SEPARATE PROGRAMS AND STORING THEM IN THE CONTROL FIFO.
630      ;*****
631
632 003034 104001 000010      CT10:  SCOPE,10          ;TEST 10
633 003040 104036          NODLAY      ;INHIBIT TRANSMITTER DELAY
634 003042 012701 000001      MOV  #1,R1      ;SET UP DELAY TIMES (1-9).
635 003046 012702 000002      MOV  #2,R2      ;SHORT TIME DELAY COUNT.
636 003052 012703 000061      MOV  #61,R3     ;START DELAY WITH '1'.
637
638 003056 104005          TAGD:  RECVRO      ;ENABLE THE DL11 RECVR
639 003060 012746 000000      MOV  #0, -(SP)  ;ENABLE INTERRUPTS
640 003064 012746 003072      MOV  #1$, -(SP)
641 003070 000002          RTI
642 003072 005004          1$:  CLR  R4      ;CONTAINS THE ACTUAL DELAYS COUNTED
643 003074 110337 003127      MOVB R3,SYNTIM ;SET UP DELAY TIME FOR THIS LOOP
644 003100 110337 030766      MOVB R3,ERR9+16 ;PRINT DELAY TIME ON ERROR
645 003104 110337 031023      MOVB R3,ERR10+16
646 003110 104007          LDPGMO      ;LOAD THE FOLLOWING PROGRAM
647 003112 003116          .+4
648 003114 000411          BR  TAGF      ;GO HERE WHEN LOADED
649 003116          .BYTE  STX
650 003117          .BYTE  DC1
651 003120          .BYTE  75          ;MODIFIED BY USER
652 003121          .BYTE  SOH        ;MODE '0' AUTO 'EOT'
653 003122          .BYTE  60
654 003123          .BYTE  DC2
655 003124          .BYTE  75          ;MODIFIED BY USER
656 003125          .BYTE  SI         ;ENABLE DESTINATION
657 003126          .BYTE  SYN
658 003127          .BYTE  61          ;LOCATION MODIFIED ON EACH PASS.
659 003130          .BYTE  DC3
660 003131          .BYTE  DC1
661 003132          .BYTE  75          ;MODIFIED BY USER
662 003133          .BYTE  SOH
663 003134          .BYTE  61
664 003135          .BYTE  DC3
```

```
665 003136 003 .BYTE ETX
666 003140 .EVEN
667
668 003140 104023 TAGF: NULL1 ;1 SEC. TTY DELAY.
669 003142 005204 INC R4 ;INCREMENT DELAY COUNTER
670 003144 020401 CMP R4,R1 ;WAITED LONG ENOUGH?
671 003146 001410 BEQ TAGG ;YES, 'EOT' SHOULD BE BACK.
672 003150 020402 CMP R4,R2 ;CHECK FOR FAST RETURN?
673 003152 002372 BGE TAGF ;NO, EXECUTE NEXT DELAY
674 003154 005737 016224 TST RECEOT ;BACK?
675 003160 001767 BEQ TAGF ;NO, STILL OK
676 003162 104022 MODERR ;'SYN' DELAY TOO SHORT
677 003164 030750 ERR9 ;** CHECK 'W2' JUMPER IN?
678 003166 000416 BR TAGI+2 ;EXIT ON ERROR
679
680 003170 104023 TAGG: NULL1 ;GIVE IT AN EXTRA SEC.
681 003172 005737 016224 TST RECEOT ;SHOULD BE BACK HERE.
682 003176 001003 BNE TAGH ;HOORAY IT IS.
683 003200 104022 MODERR ;'SYN' DELAY TOO LONG
684 003202 031005 ERR10 ;** CHECK 'W2' JUMPER IN?
685 003204 000407 BR TAGI+2
686
687 003206 005201 TAGH: INC R1
688 003210 022701 000012 CMP #12,R1 ;TEST ALL TIMES?
689 003214 001403 BEQ TAGI+2 ;YES, EXIT
690 003216 005202 INC R2 ;NO, SET UP TO TEST NEXT TIME
691 003220 005203 INC R3 ;SET UP NEW 'SYN' COUNT
692 ;ENABLE NEXT PROGRAM
693 003222 000715 TAGI: BR TAGD ;LOAD THE NEW TIME DELAY PROGRAM
694 003224 012746 000340 MOV #340, -(SP)
695 003230 012746 003236 MOV #1$, -(SP)
696 003234 000002 RTI
697 003236 032777 002000 176106 1$: BIT #SW10,@SWR ;IS 'SW10' SET
698 003244 001471 BEQ CT12 ;NO, INHIBIT TESTING M7387
```



```

699
700
701
702
703
704
705
706
707
708
709 003246 104001 000011
710 003252 000401
711 003254 104035
712 003256 104012
713 003260 027100
714 003262 104005
715 003264 012746 000000
716 003270 012746 003276
717 003274 000002
718 003276 005712
719 003300 001776
720 003302 104004
721 003304 012746 000340
722 003310 012746 003316
723 003314 000002
724 003316 012701 003360
725 003322 122122
726 003324 001403
727 003326 104022
728 003330 030526
729 003332 000436
730
731 003334 022701 003427
732 003340 001370
733 003342 104012
734 003344 027166
735 003346 104013
736 003350 012737 000001 032222
737 003356 000424
738
739
740
741 003360 015
742 003361 012
743 003362 115
744 003363 067
745 003364 063
746 003365 070
747 003366 067
748 003367 040
749 003370 120
750 003371 122
751 003372 117
752 003373 115
753 003374 040
754 003375 122

:*****
:SBTTL M7387 READ-IN MODULE TEST
:THIS TEST IS RUN IN CONJUNCTION WITH THE M7380 CONTROL TEST. IT
:REQUESTS THE USER TO INSET THE M7387 READER MODULE AND THEN
:HIT THE RESET BUTTON. THIS WILL ENABLE THE PROM DIAGNOSTIC PROGRAM TO
:BE CALLED OUT. THIS PROGRAM IS THEN VERIFIED AND THE MESSAGE 'PROM OK'
:IS TYPED.
:*****

CT11: SCOPE,11 ;TEST 11
BR .+4
M7387A: SETUP ;ENTERED HERE IS M7387A IS TYPED
PRINT
MES64 ;TEXT 'INSERT M7387 MODULE'.
RECVRO ;ENABLE DL11
MOV #0, -(SP) ;ENABLE INTERRUPTS
MOV #1$, -(SP)
1$: TST (R2)
BEQ .-2 ;NO, WAIT
DELAY ;WAIT FOR DATA
MOV #340, -(SP) ;INHIBIT INTERRUPTS
MOV #2$, -(SP)
2$: MOV #PROMS,R1
CMP11: CMPB (R1)+,(R2)+ ;DATA OK?
BEQ .+10 ;YES
MODERR ;PROM DATA ERROR
ERR3
BR CT12 ;EXIT ON ERROR

CMP #PROMD,R1 ;CHECKED ALL DATA?
BNE CMP11 ;NO
PRINT ;YES
MES65 ;TEXT 'PROM OK'
TTYIN ;WAIT FOR 'CR' TO CONTINUE.
MOV #1,LOPSWH
BR CT12

:FOLLOWING IS THE DATA WHICH SHOULD BE READ FROM PROM
PROMS: .BYTE 15
.BYTE 12
.BYTE 'M
.BYTE 67
.BYTE 63
.BYTE 70
.BYTE 67
.BYTE 40
.BYTE 'P
.BYTE 'R
.BYTE 'C
.BYTE 'M
.BYTE 40
.BYTE 'R

```

755	003376	105	.BYTE	'E
756	003377	101	.BYTE	'A
757	003400	104	.BYTE	'D
758	003401	055	.BYTE	55
759	003402	111	.BYTE	'I
760	003403	116	.BYTE	'N
761	003404	040	.BYTE	40
762	003405	124	.BYTE	'T
763	003406	105	.BYTE	'E
764	003407	123	.BYTE	'S
765	003410	124	.BYTE	'T
766	003411	015	.BYTE	15
767	003412	012	.BYTE	12
768	003413	060	.BYTE	60
769	003414	061	.BYTE	61
770	003415	062	.BYTE	62
771	003416	063	.BYTE	63
772	003417	064	.BYTE	64
773	003420	065	.BYTE	65
774	003421	066	.BYTE	66
775	003422	067	.BYTE	67
776	003423	070	.BYTE	70
777	003424	071	.BYTE	71
778	003425	015	.BYTE	15
779	003426	012	.BYTE	12
780	003427	000	PROMD: .BYTE	0

.EVEN

:TEST COMPLETE

789	003430	104001	000012
790	003434	104012	
791	003436	024236	
792	003440	005237	032222
793	003444	000137	002172

CT12:	SCOPE,12	;TEST 12
	PRINT	
	MES7	
	INC	LOPSWH
	JMP	M7380A+12
		;TEXT 'TEST COMPLETE'
		;SET SW. TO LOOP PROGRAM
		;RESTART PROGRAM

```

794                                     :*****
795                                     :SBTTL M7381 BCD INPUT MODULE ADDRESS TEST
796                                     :*****
797
798 003450 104012 M7381A: PRINT
799 003452 025233 MES29
800 003454 104026 ADDRESS ;GET MODULE ADDRESS
801 003456 104012 BCDT0: PRINT ;TEST 'SET SW'S ALL ON
802 003460 025303 MES31
803 003462 025324 MES31A
804 003464 004737 023274 JSR PC, UPDATE ;CHECK FOR SOFTWARE SWR
805 003470 104035 SETUP ;SETUP TEST PARAMETERS
806
807                                     :*****
808                                     :THIS SUBTEST ADDRESSES THE MODULE IN MODE '0' AND CHECKS THAT THE
809                                     :MODULE ADDRESS, MODE AND CORRECT NUMBER OF DIGITS ARE RETURNED.
810                                     :*****
811
812 003472 000240 BCDT1: NOP
813 003474 000240 NOP
814 003476 112737 000060 017231 MOVB #60,SOH1 ;SET UP MODE '0'
815 003504 004737 017220 JSR PC,ADR SRC ;ADDRESS THE MODULE
816 003510 005737 016224 TST RECDT ;WAS 'EOT' RETURNED?
817 003514 001003 BNE .+10 ;YES, VERIFY DATA
818 003516 104022 MODERR ;NO, MODULE DIDN'T ENTER DATA MODE.
819 003520 031041 ERR11
820 003522 000432 BR BCDT2
821 003524 123722 032134 (MPB MODADR,(R2)+ ;RECEIVE CORRECT ADDRESS?
822 003530 001403 BEQ .+10 ;YES
823 003532 104022 MODERR ;RECEIVED WRONG MODULE ADDRESS
824 003534 030526 ERR3
825 003536 000424 BR BCDT2
826 003540 122722 000060 CMPB #60,(R2)+ ;RECEIVE CORRECT MODE?
827 003544 001403 BEQ .+10 ;YES
828 003546 104022 MODERR ;MODULE WAS ADDRESSED IN MODE '0'
829 003550 030526 ERR3
830 003552 000416 BR BCDT2
831 003554 122722 000077 CMP2A: CMPB #77,(R2)+ ;SHOULD READ ALL 1'S WITH INPUTS OPEN
832 003560 001403 BEQ .+10
833 003562 104022 MODERR ;DATA ERROR, SHOULD READ ALL 1'S
834 003564 030526 ERR3 ;WITH THE INPUTS OPEN.
835 003566 000410 BR BCDT2
836 003570 022702 016246 CMP #RECBF0+12,R2 ;DONE?
837 003574 001367 BNE CMP2A ;NO
838 003576 122722 000004 CMPB #EOT,(R2)+ ;WERE CORRECT NUMBER OF CHAR.'S RECEIVED?
839 003602 001402 BEQ .+6 ;YES
840 003604 104022 MODERR ;DIDN'T RECEIVE ALL DATA CHAR.'S
841 003606 030526 ERR3

```

```
842
843
844
845
846
847
848 003610 104001
849 003612 000002
850 003614 112737 000061 017231
851 003622 004737 017220
852 003626 122722 000004
853 003632 001402
854 003634 104022
855 003636 030603
856
857
858
859
860
861
862 003640 104001
863 003642 000003
864 003644 112737 000062 017231
865 003652 004737 017220
866 003656 122737 000004 016244
867 003664 001402
868 003666 104022
869 003670 030526
870
871
872
873
874
875 003672 104001
876 003674 000004
877 003676 112737 000063 017231
878 003704 004737 017220
879 003710 122722 000004
880 003714 001402
881 003716 104022
882 003720 030526
883

;*****
;THIS SUBTEST ADDRESSES THE MODULE FOR MODE '1' (EXT. SYNC) AND CHECKS
;THAT ONLY AN 'EOT' AND ONLY AN 'EOT' IS RECEIVED BACK.
;*****
BCDT2: SCOPE
2
MOVB #61,SOH1 ;SET UP MODE '1' 'EXT SYNC'
JSR PC,ADR SRC ;ADDRESS THE MODULE
CMPB #EOT,(R2)+ ;WAS 'EOT' RETURNED?
BEQ .+6 ;YES
MODERR ;'EXT SYNC' DIDN'T RETURN AN 'EOT'
ERR5

;*****
;THIS SUBTEST ADDRESSES THE MODULE IN MODE '2' AND CHECKS THAT ONLY
;DATA IS RECIEVED FROM THE MODULE.
;*****
BCDT3: SCOPE
3
MOVB #62,SOH1 ;SET UP MODE '2'
JSR PC,ADR SRC ;ADDRESS THE MODULE
CMPB #EOT,RECBF0+10 ;IS 'EOT' IN CORRECT PLACE
BEQ .+6 ;YES
MODERR ;ONLY DATA SHOULD BE TRANSMITTED IN MODE '2'
ERR3

;*****
;THIS SUBTEST ADDRESSES THE MODULE FOR MODE '3' (EXT. SYNC) AND CHECKS
;THAT ONLY AN 'EOT' AND ONLY 'EOT' IS RECEIVED BACK.
;*****
BCDT4: SCOPE
4
MOVB #63,SOH1 ;SET UP MODE '3'
JSR PC,ADR SRC ;ADDRESS MODULE
CMPB #EOT,(R2)+ ;WAS 'EOT' RETURNED?
BEQ .+6 ;YES
MODERR ;EXTERNAL &SYNC' DIDN'T RETURN AN 'EOT'
ERR3
```

```

884
885
886
887
888
889
890 003722 104001      BCDT5: SCOPE
891 003724 000005      5
892 003726 004737 005154 JSR PC,@#ADRSIT ;SUBROUTINE TO ADDRESS MODULE
893
894
895
896
897
898
899
900
901
902 003732 104001      BCDT6: SCOPE
903 003734 000006      6
904 003736 032777 002000 175406 BIT #SW10,@SWR ;SW SET?
905 003744 001520      BEQ BCDT11 ;NO, SKIP MANUAL TESTS.
906 003746 104012      PRINT
907 003750 025303      MES31
908 003752 025434      MES31D
909 003754 004737 023274 JSR PC, UPDATE ;CHECK FOR SOFTWARE SWR
910 003760 012737 003760 020774 MOV #.,RETURN ;RE-SET SCOPE LOOP POINTER
911 003766 104005      RECVRO
912 003770 112737 000062 017231 MOVB #62,SOH1 ;SET UP MODE '2'
913 003776 004737 017220 JSR PC,@#ADRSRC ;ADDRESS THE MODULE
914 004002 122722 000060 CMPB #60,(R2)+
915 004006 001403      BEQ .+10
916 004010 104022      MODERR
917 004012 030526      ERR3
918 004014 000403      BR BCDT7
919 004016 022702 016244 CMP #RECBF0+10,R2
920 004022 001367      BNE CMP2B
921
922
923
924
925
926
927
928
929 004024 104001      BCDT7: SCOPE
930 004026 000007      7
931 004030 104012      PRINT
932 004032 024316      MES10 ;TEXT 'RESET MODULE TO ADDR. '17'.
933 004034 104012      PRINT
934 004036 025303      MES31 ;SET CUST. SW.'S TO '0'
935 004040 025463      MES31E
936 004042 004737 023274 JSR PC, UPDATE ;CHECK FOR SOFTWARE SWR
937 004046 112737 000077 032134 MOVB #77,MODADR ;SET UP NEW MODULE ADDRESS.
938 004054 112737 000077 017227 MOVB #77,SRCADR
939 004062 012737 004062 020774 MOV #.,RETURN ;RE-SET SCOPE LOOP POINTER

```

940	004070	104005			RECVRO		;ENABLE THE DL11 RECVR.
941	004072	112737	000060	017231	MOVB	#60,SOH1	;SET UP MODE '0'
942	004100	004737	017220		JSR	PC,@ADRSRC	;ADDRESS THE MODULE
943	004104	005712			TST	(R2)	;WAS ANY DATA RETURNED?
944	004106	001003			BNE	+.10	;YES
945	004110	104022			MODERR		;DIDN'T ENTER DATA MODE
946	004112	031041			ERR11		
947	004114	000406			BR	BCDT10	;EXIT ON ERROR
948	004116	122737	000004	016236	CMPB	#EOT,RECBF0+2	;EOT SHOULD BE 3RD CHAR. BACK
949	004124	001402			BEQ	+.6	;OK, IT IS
950	004126	104022			MODERR		;DATA WASN'T INHIBITED
951	004130	030526			ERR3		

 :THIS SUBTEST REQUESTS THAT THE CUSTOMER SWITCHES BE SET TO ALTERNATE
 :ON & OFF AND CHECKS THAT ONE '4' CHARACTERS ARE RETURNED.
 :*****

958	004132	104001			BCDT10: SCOPE		
959	004134	000010			10		
960	004136	104012			PRINT		
961	004140	025303			MES31		
962	004142	025403			MES31C		
963	004144	004737	023274		JSR	PC, UPDATE	;CHECK FOR SOFTWARE SWR
964	004150	012737	004150	020774	MOV	#,RETURN	;RE-SET THE SCOPE LOOP POINTER
965	004156	104005			RECVRO		
966	004160	112737	000062	017231	MOVB	#62,SOH1	;SET UP MODE '2'
967	004166	004737	017220		JSR	PC,@ADRSRC	;ADDRESS THE MODULE
968	004172	022737	000004	016240	CMP	#EOT,RECBF0+4	;WHERE ONLY '4' CHAR.'S RETURNED
969	004200	001402			BEQ	+.6	;YES
970	004202	104022			MODERR		;ONLY '4' CHAR.'S SHOULD BE RETURNED
971	004204	030526			ERR3		


```

972
973
974
975
976
977
978 004206 104001
979 004210 000011
980 004212 104012
981 004214 026654
982 004216 104013
983 004220 112737 000061 017231
984 004226 004737 017220
985 004232 105737 016235
986 004236 001002
987 004240 104022
988 004242 031433
989
990
991
992
993
994
995
996
997 004244 104001
998 004246 000012
999 004250 104012
1000 004252 026654
1001 004254 104013
1002 004256 112737 000063 017231
1003 004264 004737 017220
1004 004270 105737 016235
1005 004274 001002
1006 004276 104022
1007 004300 031433

;*****
;THIS SUBTEST TESTS THE DEVICE FLAG IN MODE '1'. A REQUEST IS MADE
;FOR AN EXTERNAL SIGNAL TO BE SUPPLIED. THE 'BCD' INPUT MODULE
;IS THEN ADDRESSED AND CHECKS THAT DATA WAS RETURNED.
;*****

BCDT11: SCOPE
11
PRINT
MES60
TTYIN
MOV B #61,SOH1
JSR PC,ADR SRC
TST B RECBF0+1
BNE BCDT12
MODERR
ERR20

;TEXT 'SUPPLY AN EXTERNAL SYNC.
;WAIT FOR 'CR'
;SELECT MODE '1' ;WAIT FOR DEVICE FLAG.
;ADDRESS THE MODULE
;WAS ANY DATA RETURNED?
;YES, CHECK FORMAT
;NO DATA RETURNED WITH EXT. SYNC.

;*****
;THIS SUBTEST TEST THE DEVICE FLAG IN MODE '3'. A REQUEST IS MADE
;FOR AN EXTERNAL SIGNAL TO BE SUPPLIED. THE 'BCD' INPUT MODULE IS
;THEN ADDRESSED AND CHECKED THAT DATA WAS RETURNED.
;*****

BCDT12: SCOPE
12
PRINT
MES60
TTYIN
MOV B #63,SOH1
JSR PC,ADR SRC
TST B RECBF0+1
BNE BCDT13
MODERR
ERR20

;TEXT 'SUPPLY AN EXTERNAL SYNC.'
;WAIT FOR 'CR'
;SELECT MODE 3 WAIT FOR DEVICE FLAG
;ADDRESS THE MODULE
;WAS ANY DATA RETURNED?
;YES, VERIFY FORMAT
;NO DATA RETURNED WITH EXT. SYNC.

```

```

1008
1009
1010
1011
1012 004302 104001
1013 004304 000013
1014 004306 104012
1015 004310 024236
1016 004312 000137 003456
1017
1018
1019
1020
1021
1022
1023
1024
1025 004316 104012
1026 004320 025474
1027 004322 104026
1028 004324 104035
1029 004326 004737 021626
1030
1031 004332 104012
1032 004334 032032
1033 004336 104017
1034 004340 117700 175006
1035 004344 142700 000374
1036 004350 110037 017231
1037 004354 152737 000060 017231
1038 004362 104005
1039 004364 004737 017220
1040 004370 105737 016224
1041 004374 001775
1042 004376 032777 020000 174746
1043 004404 001354
1044 004406 104037
1045 004410 000752

;*****
;TEST COMPLETE
;*****

BCDT13: SCOPE
13
PRINT
MES7
JMP BCDT0 ;TEST COMPLETE

;*****
;SBTT. M7381 BCD INPUT EXERCISER TEST
;THIS TEST REQUESTS THE MODULE ADDRESS AND THEN CONTINUOUSLY
;ADDRESSES THE MODULE USING DATA SWITCHES '0 & 1' TO SELECT THE MODE.
;THE RECEIVED DATA IS THEN PRINTED ON THE TELETYPE.
;*****

M7381E: PRINT
MES32
ADDRESS ;GET MODULE ADDRESS
SETUP ;SETUP THE '^R' ADDRESS
JSR PC,TTYENB ;ENABLE INTERRUPTS

M381E1: PRINT
CRLF
M381E2: TSTTKS ;CHECK FOR KEYBOARD FLAG
MOV B @SWR,R0 ;GET MODE FROM SW.'S
BIC B #374,R0 ;CLR UN-WANTED BITS
MOV B R0,SOH1 ;SET UP THE MODE
BIS B #60,SOH1

;ENABLE THE DL11 RECV.
JSR PC,@ADRSRC ;ADDRESS THE MODULE
TST B RECEOT ;HAS 'EOT' RETURNED?
BEQ .-4 ;NO, WAIT IT OUT
BIT #SW13,@SWR ;INHIBIT PRINTOUT?
BNE M381E2 ;YES,
PRTRBF ;PRINT RECV. DATA
BR M381E2

```



```
1046 :*****
1047 :SBTTL M7382 BCD OUTPUT MODULE ADDRESS TEST
1048 :*****
1049
1050 M7382A: PRINT
1051 MES33
1052 ADDRESS ;GET THE MODULE ADDRESS
1053 004420 104035 SETUP ;SET UP TEST PARAMETERS
1054
1055 :*****
1056 :THIS TEST ADDRESSED THE 2ND OUTPUT MODULE AND TRANSMITS '8' DIGITS
1057 :OF DATA AS '77'. THIS SHOULD CAUSE ALL THE OUTPUT LINES TO BE HIGH
1058 :*****
1059
1060 004422 000240 OBCTD1: NOP
1061 004424 000240 NOP
1062 004426 004737 017234 JSR PC,ADRST ;ADDRESS DESTINATION
1063
1064 004432 104007 LDPMO ;TRANSMIT THE FOLLOWING DATA
1065 004434 004440 .+4
1066 004436 000405 BR TAG3A
1067 004440 077 .BYTE 77 ;1ST DIGIT
1068 004441 077 .BYTE 77
1069 004442 077 .BYTE 77
1070 004443 077 .BYTE 77
1071 004444 077 .BYTE 77
1072 004445 077 .BYTE 77
1073 004446 077 .BYTE 77
1074 004447 077 .BYTE 77 ;LAST DIGIT
1075 004450 004 .BYTE EOT
1076 004452 .EVEN
1077
1078 004452 104012 TAG3A: PRINT ;TEXT 'EXAMINE OUTPUT
1079 004454 025557 MES34
1080 004456 025612 MES35 ;CHECK FOR ALL LOGIC 1'S
1081 004460 104013 TTYIN ;WAIT FOR CHECK
```

1082			
1083			
1084			
1085			
1086			
1087	004462	104001	
1088	004464	000002	
1089	004466	004737	017234
1090	004472	104007	
1091	004474	004500	
1092	004476	000405	
1093	004500	065	
1094	004501	065	
1095	004502	065	
1096	004503	065	
1097	004504	065	
1098	004505	065	
1099	004506	065	
1100	004507	065	
1101	004510	004	
1102		004512	
1103			
1104	004512	104012	
1105	004514	025557	
1106	004516	026064	
1107	004520	104013	
1108			
1109			
1110			
1111			
1112			
1113	004522	104001	
1114	004524	000003	
1115	004526	004737	017234
1116			
1117	004532	104007	
1118	004534	004540	
1119	004536	000405	
1120	004540	072	
1121	004541	072	
1122	004542	072	
1123	004543	072	
1124	004544	072	
1125	004545	072	
1126	004546	072	
1127	004547	072	
1128	004550	004	
1129		004552	
1130	004552	104012	
1131	004554	025557	
1132	004556	025631	
1133	004560	026064	
1134	004562	104013	

```

:*****
:THIS SUBTEST SHOULD CAUSE ALTERNATE 'HI & LO 'S' TO BE SEEN ON THE OUTPUT
:*****

```

```

08CDT2: SCOPE
        2
        JSR      PC,ADRST      ;ADDRESS DESTINATION
        LDPMO    ;TRANSMIT THE FOLLOWING DATA
        .+4
        BR       TAG3B
        .BYTE    65            ;1ST DIGIT
        .BYTE    65
        .BYTE    65
        .BYTE    65
        .BYTE    65
        .BYTE    65
        .BYTE    65
        .BYTE    65
        .BYTE    65            ;LAST DIGIT
        .BYTE    EOT
        .EVEN

```

TAG3B: PRINT
MES34
MES40B
TTYIN :WAIT FOR CHECK

```

*****
;THIS SUBTEST SHOULD CAUSE ALTERNATE 'HI & LO'S' TO BE SEEN ON THE OUTPUT
;IN REVERSE OF THOSE IN THE PREVIOUS SUBTEST.
*****

```

```

OBCDT3: SCOPE
3
JSR      PC,ADRST      ;ADDRESS DESTINATION

LDPMO    ;TRANSMIT THE FOLLOWING DATA
.+4
BR        TAG3C
.BYTE    72
.BYTE    72
.BYTE    72
.BYTE    72
.BYTE    72
.BYTE    72
.BYTE    72
.BYTE    72
.BYTE    72
.BYTE    EOT
.EVEN
TAG3C:   PRINT
MES34
MES37
MES40B
TTYIN    ;WAIT FOR CHECK

```

```
1135 :*****
1136 :THIS SUBTEST CONTINUOUSLY ADDRESSES THE MODULE ENABLING THE USER TO SCOPE
1137 :FOR THE SIGNAL 'OUTPUT DONE 'H & L'.
1138 :*****
1139
1140 004564 104001 OBCDT4: SCOPE
1141 004566 000004 4
1142 004570 104012 PRINT
1143 004572 025643 MES38 ;TEXT SCOPE FOR OUTPUT DONE
1144 004574 004737 017234 TAG3D: JSR PC,@#ADRST ;ADDRESS DESTINATION
1145
1146 004600 104006 LDCHRO
1147 004602 000004 EOT
1148 004604 000773 BR TAG3D ;CYCLE UNTIL RESTARTED
1149
1150
1151
1152 .SBTTL BCD I/O TEST
1153 :*****
1154 :BCD INPUT/OUTPUT EXERCISER TEST
1155 :THIS TEST USES BOTH THE BCD 'INPUT&OUTPUT' MODULES. AN INCREMENTING
1156 :'BCD' COUNT IS SENT TO THE OUTPUT MODULE AND WRAPPED AROUND VIA A
1157 :SPECIAL CABLE TO THE INPUT MODULE. THE DATA RECEIVED FROM THE INPUT MODULE
1158 :IS THEN VERIFIED AGAINST THE TRANSMITTED DATA. THE INPUT MODULE CAN
1159 :BE SETUP TO USE EITHER INTERNAL OR EXTERNAL SYNC. IF EXTERNAL SYNC IS
1160 :SELECTED, THIS SIGNAL IS SUPPLIED FROM THE SIGNAL ON THE BCD OUTPUT
1161 :MODULE KNOWN AS OUTPUT DATA H.
1162 :*****
1163
1164 004606 104012 BCDIO: PRINT
1165 004610 026143 MES43 ;TEXT 'BCD I/O TEST'
1166 004612 104026 ADDRESS ;GET THE MODULE ADDRESS
1167 004614 104012 PRINT
1168 004616 026116 MES42 ;TEXT 'INT OR EXT SYNC.?'
1169 004620 104013 TTYIN ;WAIT FOR INPUT
1170 004622 122737 000111 015330 CMPB #111,INBUF ;'I' FOR INTERNAL?
1171 004630 001404 BEQ .+12 ;YES, SET UP FOR INT. SYNC.
1172 004632 112737 000061 017231 MOVB #61,SOH1 ;NO, SET UP FOR EXT. SYNC.
1173 004640 000403 BR .+10
1174 004642 112737 000060 017231 MOVB #60,SOH1
1175 004650 104035 SETUP ;SET UP TEST PARAMETERS
```

```
1176 ;*****
1177 ;THIS SUBTEST OUTPUTS A DATA PATTERN OF '60-77' TO EACH 'BCD' OUTPUT,
1178 ;ONE AT A TIME. THIS PATTERN IS THEN READ BACK BY THE BCD INPUT
1179 ;MODULE AND COMPARED AGAINST THE OUTPUTTED DATA. THIS TEST WILL VERIFY
1180 ;THAT EACH OUTPUT LINE CAN BE ADDRESSED AND THAT NO TWO OUTPUTS ARE
1181 ;SHORTED TOGETHER.
1182 ;*****
1183
1184 004652 000240 BCDI01: NOP
1185 004654 000240 NOP
1186 004656 012700 004712 MOV #DATA1,R0 ;SET UP DATA TABLE TO TRANSMIT ALL 0'S.
1187 004662 112720 000060 MOVB #60,(R0)+
1188 004666 022700 004722 CMP #DATA2,R0 ;DONE?
1189 004672 001373 BNE .-10 ;NO
1190 004674 012701 004712 MOV #DATA1,R1 ;SET UP DATA POINTER
1191
1192 004700 004737 017234 TAG4A: JSR PC,@#ADR DST ;ADDRESS DESTINATION
1193 004704 104007 LDPGMO ;TRANSMIT DATA
1194 004706 004712 .+4
1195 004710 000405 BR TAG4B ;GO HERE WHEN DONE
1196 004712 060 DATA1: .BYTE 60
1197 004713 060 .BYTE 60
1198 004714 060 .BYTE 60
1199 004715 060 .BYTE 60
1200 004716 060 .BYTE 60
1201 004717 060 .BYTE 60
1202 004720 060 .BYTE 60
1203 004721 060 .BYTE 60
1204 004722 004 DATA2: .BYTE EOT
1205 004724 .EVEN
1206
1207 004724 104005 TAG4B: RECVRO ;ENABLE THE DL11 RECVR.
1208 004726 004737 017220 JSR PC,@#ADR SRC ;ADDRESS BCD INPUT
1209 004732 104004 DELAY ;GIVE 'EM TIME TO READ THE DATA.
1210 004734 012702 004712 MOV #DATA1,R2 ;SET UP TO VERIFY DATA
1211 004740 012703 016236 MOV #RECBF0+2,R3
1212 004744 122223 TAG4C: CMPB (R2)+,(R3)+ ;DATA EQUAL?
1213 004746 001403 BEQ .+10 ;YES
1214 004750 104022 MODERR ;INPUT DATA DOESN'T EQUAL DATA OUTPUT
1215 004752 030526 ERR3
1216 004754 000414 BR TAG4E+2 ;EXIT ON ERROR
1217 004756 022702 004722 CMP #DATA2,R2 ;DONE?
1218 004762 001370 BNE TAG4C ;NO, COMPARE NEXT BYTE
1219
1220 004764 105211 TAG4D: INCB (R1) ;UPDATE DATA PATTERN
1221 004766 122711 000100 CMPB #100,(R1) ;DONE ALL CODES FOR THIS OUTPUT?
1222 004772 001342 BNE TAG4A ;NO, TRANSMIT NEXT PATTERN
1223 004774 112721 000060 MOVB #60,(R1)+ ;YES, RESET IT TO '60'.
1224 005000 022701 004722 CMP #DATA2,R1 ;DONE WITH TEST?
1225 005004 001367 TAG4E: BNE TAG4D ;NO, START NEXT OUTPUT TEST
1226 005006 104012 PRINT
1227 005010 024236 MES7 ;TEST COMPLETE
1228 005012 000717 BR BCDI01 ;RESTART TEST
```

```

1229 ;*****
1230 ;SBTTL M7383 A/D INPUT MODULE ADDRESS TEST
1231 ;THIS TEST IS USED TO VERIFY THAT THE A/D MODULE CAN BE ADDRESS
1232 ;AND THAT IT WILL RETURN DATA ON COMPLETION OF A CONVERSION.
1233 ;*****
1234
1235 005014 104012 M7383A: PRINT
1236 005016 024364 MES11 ;TEXT 'A/D ADDRESSING TEST.'
1237 005020 104026 ADDRESS ;GET MODULE ADDRESS
1238 005022 104035 ADT0: SETUP ;SET UP TEST PARAMETERS
1239
1240 ;*****
1241 ;THE FOLLOWING SUBTEST ADDRESSES THE A/D MODULE AND VERIFIES THAT
1242 ;DATA AND 'EOT' ARE RETURNED BY THE MODULE
1243 ;*****
1244
1245 005024 000240 ADT1: NOP
1246 005026 000240 NOP
1247 005030 112737 000063 017231 MOVB #63,SOH1 ;PROGRAM CH.'3'
1248 005036 004737 017220 JSR PC,@ADRSRC ;ADDRESS MODULE
1249
1250 005042 000240 TAG2A: NOP
1251 005044 105737 016235 TSTB RECBF0+1 ;DATA RETURNED?
1252 005050 001003 BNE .+10 ;YES
1253 005052 104022 MODERR ;MODULE DIDN'T ENTER DATA MODE
1254 005054 031041 ERR11
1255 005056 000414 BR ADT2 ;EXIT ON ERROR
1256 005060 005737 016224 TST RECEOT ;WAS 'EOT' RETURNED?
1257 005064 001003 BNE .+10 ;YES
1258 005066 104022 MODERR ;MODULE DIDN'T RETURN 'EOT'
1259 005070 030603 ERR5
1260 005072 000406 BR ADT2 ;EXIT ON ERROR
1261 005074 122737 000004 016244 CMPB #EOT,RECBF0+10 ;CORRECT NO. OF CHAR.'S RETURNED?
1262 005102 001402 BEQ .+6 ;YFS
1263 005104 104022 MODERR ;DIDN'T RECV. CORRECT NO. OF CHAR.'S.
1264 005106 030526 ERR3
1265
1266 ;*****
1267 ;THIS SUBTEST ADDRESSES THE A/D MODULE USING MODE '8' AND TESTS
1268 ;THAT THE CORRECT NUMBER OF CHARACTER ARE RECEIVED BACK.
1269 ;*****
1270
1271 005110 104001 ADT2: SCOPE
1272 005112 000002 2
1273 005114 112737 000070 017231 MOVB #70,SOH1 ;PROGRAM MODE '8'
1274 005122 004737 017220 JSR PC,@ADRSRC ;ADDRESS MODULE
1275
1276 005126 122737 000004 016274 TAG2B: CMPB #EOT,RECBF0+40 ;'EOT' SHOULD BE RETURNED HERE
1277 005134 001402 BEQ .+6 ;OK
1278 005136 104022 MODERR ;MODULE DIDN'T RETURN '4' CH.'S OF DATA
1279 005140 030526 ERR3

```

```
1280
1281
1282
1283
1284
1285
1286 005142 104001
1287 005144 000003
1288 005146 004737 005154
1289 005152 000437
1290
1291 005154 005737 032132
1292 005160 001033
1293 005162 112737 000060 005211
1294 005170 123737 032134 005211
1295 005176 001416
1296 005200 104005
1297 005202 104025
1298 005204 005210
1299 005206 000403
1300
1301 005210 021
1302 005211 060
1303 005212 001
1304 005213 060
1305 005214 023
1306 005215 000
1307
1308 005216 005712
1309 005220 001405
1310 005222 113737 005211 031166
1311 005230 104022
1312 005232 031124
1313
1314 005234 105237 005211
1315 005240 122737 000100 005211
1316 005246 001350
1317 005250 000207

:*****
:THIS SUBTEST ADDRESSES THE 'A/D' USING THE WRONG MODULE ADDRESSES
:AND TESTS THAT THE MODULE ISN'T ENABLED.
:*****

ADT3: SCOPE
      3
      JSR PC,ADRSIT
      BR ADT4

ADRSIT: TST SIOSWH ;USING THE SERIAL INPUT?
        BNE TAG2H ;YES, INHIBIT RUNNING THIS TEST.
        MOV #60,ADCHX1 ;SET UP 1ST ADDRESS TO BE TESTED
TAG2C: CMPB MODADR,ADCHX1 ;EQUAL TO SELECTED ADDR.?
        BEQ TAG2F ;YES, SELECT NEXT. ADDR.
        RECVRO
        SOURCE
        .+4
        BR TAG2G

ADCHX1: .BYTE DC1 ;ALERT MODULE
        .BYTE 60 ;ADDRESS MODIFIED FROM '60-77'
        .BYTE SOH
        .BYTE 60
        .BYTE DC3
        .BYTE 0 ;ENABLE MODULE

TAG2G: TST (R2) ;WAS ANY DATA RETURNED
        BEQ TAG2F ;NO
        MOVB ADCHX1,ERR13A ;MODULE WAS ENABLE WITH ILLEGAL ADDR.
        MODERR
        ERR13

TAG2F: INCB ADCHX1 ;UPDATE MODULE ADDRESS
        CMPB #100,ADCHX1 ;DONE?
        BNE TAG2C ;NO
TAG2H: RTS PC
```



```
1318
1319
1320
1321
1322
1323
1324 005252 104001
1325 005254 000004
1326 005256 032777 002000 174066
1327 005258 001450
1328 005260 104012
1329 005270 024316
1330 005272 104013
1331 005274 012737 005274 020774
1332 005302 112737 000077 017227
1333 005310 104005
1334 005312 112737 000064 017231
1335 005320 004737 017220
1336 005324 005712
1337 005326 001403
1338 005330 104022
1339 005332 031475
1340 005334 000422
1341 005336 104012
1342 005340 026654
1343 005342 012746 000000
1344 005346 012746 005354
1345 005352 000002
1346 005354 000001
1347 005356 012746 000340
1348 005362 012746 005370
1349 005366 000002
1350 005370 104004
1351 005372 005712
1352 005374 001002
1353 005376 104022
1354 005400 031433
1355
1356
1357
1358
1359
1360 005402 104001
1361 005404 000005
1362 005406 104012
1363 005410 024236
1364 005412 000137 005022
```

```
*****
;THIS SUBTEST CHECKS THAT THE A/D MODULE WILL WORK UNDER EXTERNAL SYNC.
;THE MODULE IS ADDRESSED AND THEN A REQUEST IS MADE FOR AN EXTERNAL
;SYNC SIGNAL TO BE SUPPLIED.
*****
ADT4:  SCOPE
      4
      BIT    #SW10,@SWR      ;SW. '10' SET?
      BEQ    ADT5+4          ;NO, INHIBIT THIS TEST
      PRINT
      MES10                  ;TEXT 'RESET MODULE ADDR.
      TTYIN                  ;WAIT FOR SETUP
      MOV     #.,RETURN      ;RESET SCOPE LOOP POINTER
      MOVB    #77,SRADR      ;SET UP ADDRESS '17'
      RECVRO                  ;ENABLE THE DL11 RECEIVER
      MOVB    #64,SOM1       ;EXT SYNC; CHANNEL '0'
      JSR     PC,@ADRSRC     ;ADDRESS MODULE
      TST     (R2)           ;MAKE SURE NO DATA WAS RETURNED
      BEQ     .+10
      MODERR                  ;EXT SYNC CONVERSION TOOK PLACE
      ERR21                  ;WITH NO EXT. SYNC SUPPLIED.
      BR      ADT5           ;EXIT ON ERROR
      PRINT
      MES60                  ;TEXT 'SUPPLY EXT. SYNC SIGNAL'.
      MOV     #0, -(SP)      ;ENABLE INTERRUPTS
      MOV     #1$, -(SP)
      RTI
1$:   WAIT
      MOV     #340, -(SP)    ;INHIBIT INTERRUPTS
      MOV     #2$, -(SP)
      RTI
2$:   DELAY
      TST     (R2)           ;WAIT FOR DATA
      BNE     .+6            ;WAS A DATA RETURNED
      MODERR                  ;YES
      ERR20                  ;NO DATA WAS RETURNED WITH
                              ;EXTERNAL SYNC.
*****
;TEST COMPLETE
*****
ADT5:  SCOPE
      5
      PRINT
      MES7
      JMP     ADT0          ;TEXT 'TEST COMPLETE'
                              ;RE-START TEST.
```

```
1365 :*****
1366 :SBTTL M7383 A/D CALIBRATION ROUTINE
1367 :THIS ROUTINE TAKES CONTINUOUS CONVERSION USING DATA SW'S '0-4' IN OCTAL
1368 :WEIGHT TO SELECT THE CHANNEL TO BE CONVERTED AND THEN PRINTS THE CONVERTED VALUE
1369 :
1370 :CHANNEL SELECTION IS AS FOLLOWS:
1371 :
1372 :DATA SW'S '0-1' SELECT 'INT. SYNC' ON CH.'S 0,1,2 OR 3
1373 :DATA SW '2' & '0-1' SELECT 'EXT SYNC' ON CH.'S 0,1,2 OR 3
1374 :DATA SW '3' ONLY SELECTS 'INT SYNC' CONVERSION ON ALL '4' CH.'S
1375 :DATA SW'S '2&3' SELECT 'EXT. SYNC' CONVERSION ON ALL '4' CH.'S
1376 :*****
1377
1378 005416 104012 M7383C: PRINT
1379 005420 024411 MES12 ;TEXT 'A/D CALIBRATION ROUTINE'
1380 005422 104026 ADDRESS ;GET MODULE ADDRESS
1381 005424 004737 021442 JSR PC,REMOTE ;CHECK FOR REMOTE DESTINATION
1382 005430 104035 SETUP ;SET UP THE '^R' RESTART ADDRESS
1383 005432 012701 000001 MOV #1,R1 ;SET UP FOR '1' CONVERSION
1384 005436 104036 NODLAY ;SET TRANS. DELAY INHIBIT SW.
1385 005440 104012 CALBT1: PRINT
1386 005442 032032 CRLF
1387 005444 104017 CALBT2: TSTTKS ;CHECK FOR KEYBOARD FLAG
1388 005446 117703 173700 MOVB @SWR,R3 ;GET CHANNEL
1389 005452 142703 000300 BICB #300,R3 ;CLR UN-WANTED BITS
1390 005456 152703 000060 BISB #60,R3 ;CONVERT TO ASCII
1391 005462 110337 017231 MOVB R3,SOH1 ;SET UP TO CONVERT CH.
1392 005466 104027 ADCNVT ;CONVERT
1393 005470 032777 020000 173654 BIT #SW13,@SWR ;INHIBIT TYPEOUT?
1394 005476 001362 BNE CALBT2 ;YES, TAKE NEXT CONVERSION
1395 005500 004737 021502 JSR PC,SETRMT ;CHK FOR AND SET UP REMOTE DST.
1396
1397 005504 104037 PRTRBF ;PRINT RECV'D DATA
1398 005506 004737 021524 JSR PC,CLRMOTE ;CLEAR REMOTE DESTINATION
1399 005512 000754 BR CALBT2
```



```
1400 ;*****
1401 ;SBTTL M7383 A/D REPEATABILITY TEST
1402 ;THIS TEST REQUESTS FOR A CHANNEL AND A V.S.F (VERTICAL SCALE FACTOR) TO
1403 ;BE INPUTTED FROM THE TELETYPE. A SERIES OF '100' CONVERSIONS A THEN TAKEN,
1404 ;AVERAGED AND THEN THE RESULT IS DISPLAYED IN A HISTOGRAM FORMAT ON
1405 ;THE TELETYPE.
1406 ;*****
1407
1408 005514 104012 M7383R: PRINT
1409 005516 024574 MES15 ;TEXT 'A/D REPEATABILITY TEST'.
1410 005520 104026 ADDRESS ;GET THE MODULE ADDRESS
1411 005522 004737 021442 JSR PC,REMOTE ;CHECK FOR REMOTE DESTINATION
1412 005526 104035 SETUP ;SET UP RESTART ADDR. POINTER
1413 005530 104036 NODLAY ;SET TRANS. DELAY INHIBIT SW.
1414 005532 104012 REPT0A: PRINT
1415 005534 024624 MES16 ;REQUEST 'VSF'
1416 005536 104013 TTYIN
1417 005540 104030 BCDBIN ;CONVERT INPUT TO BINARY
1418 005542 005737 022564 TST BCDTAB ;VSF=07
1419 005546 001771 BEQ REPT0A ;YES, ILLEGAL ENTRY
1420 005550 013737 022564 032174 MOV BCDTAB,KSTOR1 ;SAVE INPUT
1421 005556 005037 016112 CLR HIDIVR
1422 005562 005037 016116 CLR HIDIVD
1423 005566 013737 032174 016110 MOV KSTOR1,LODIVR ;SET UP TO DIVIDE 'VSF' TO GET NO. OF AVG.'S
1424 005574 012737 000144 016114 MOV #100.,LODIVD
1425 005602 004737 016016 JSR PC,DIVIDE
1426 005606 005737 016126 TST REMAIN ;IS NUMBER LEGAL?
1427 005612 001347 BNE REPT0A ;NO, REQUEST NEW 'VSF'
1428 005614 013737 016122 032202 MOV QUOTENT,KSTOR4 ;YES, SAVE IT
1429 005622 013737 032174 032204 MOV KSTOR1,KSTOR5
1430 005630 006337 032204 ASL KSTOR5
1431 005634 104032 CHANEL ;REQUEST CHANNEL.
1432 005636 012701 000144 REPT0: MOV #100.,R1 ;SET UP TO TAKE '100' CONVERSIONS
1433 005642 012702 017670 MOV #TRNBFO,R2 ;SET UP TO SAVE CONVERTED VALUE
1434 005646 104027 REPT1: ADCNVT ;TAKE 100 CONVERSION
1435 005650 104031 AVERAGE ;AVERAGE THEM
1436 005652 012700 016120 MOV #LOW,R0 ;SET UP TO SAVE VALUES
1437 005656 012703 032302 MOV #AVGTAB,R3 ;SAVE AVERAGE HERE
1438 005662 012704 000003 MOV #3,R4
1439 005666 012023 MOV (R0)+,(R3)+ ;SAVE AVG HIGH & LOW
1440 005670 005304 DEC R4 ;SAVED ALL VALUES
1441 005672 001375 BNE .-4 ;NO
1442 005674 013700 016122 MOV QUOTENT,R0 ;SET UP AVERAGE
1443 005700 062700 000011 ADD #9.,R0 ;CALCULATE AVERAGE +9 VALUE
1444 005704 010037 032176 MOV R0,KSTOR2 ;SAVE IT
1445 005710 162700 000022 SUB #18.,R0 ;CALCULATE AVERAGE -9 VALUE
1446 005714 010037 032200 MOV R0,KSTOR3 ;SAVE IT
1447 005720 013704 032202 MOV KSTOR4,R4 ;SETUP TO AVERAGE OUT 'VSF'
1448 005724 013701 032174 MOV KSTOR1,R1 ;SET UP TO TAKE 'X' AVERAGES
1449 005730 022701 000001 CMP #1,R1 ;VSF =1?
1450 005734 001412 BEQ REPT3 ;YES, NO AVERAGING NEEDED
1451 005736 104031 REPT2: AVERAGE ;DO IT
1452 005740 013723 016122 MOV QUOTENT,(R3)+ ;SAVE VALUE
1453 005744 063702 032204 ADD KSTOR5,R2 ;SET BUFFER POINTER TO PICK UP NEXT GROUP
1454 005750 005304 DEC R4 ;DONE
1455 005752 001371 BNE REPT2 ;NO
```

```

1456 005754 012702 032310      MOV      #AVGTAB+6,R2      ;SET UP TO CATEGORIZE AVERAGES
1457 005760 000402              BR          .+6
1458 005762 012702 017670      REPT3:  MOV      #TRNBF0,R2      ;FOR VSF OF '1' USE ACTUAL VALUES
1459 005766 012700 032226      MOV      #ORLOW,R0      ;SET UP TO CLR COUNT BUFFER
1460 005772 005020              CLR      (R0)+      ;CLR BUFFER
1461 005774 022700 032302      CMP      #ORHIGH+2,R0      ;DONE?
1462 006000 001374              BNE      .-6      ;NO
1463 006002 013700 032202      MOV      KSTOR4,R0      ;KSTOR4 CONTAINS VSF
1464 006006 010001              MOV      R0,R1
1465 006010 021237 032176      REPT4:  CMP      (R2),KSTOR2      ;IS VALUE > AVG. +9?
1466 006014 003403              BLE      .+10      ;NO
1467 006016 005237 032300      INC      ORHIGH      ;YES, VALUE OUT OF RANGE
1468 006022 000414              BR          REPT5
1469 006024 021237 032200      CMP      (R2),KSTOR3      ;IS VALUE < AVG. -9?
1470 006030 002003              BGE      .+10      ;YES
1471 006032 005237 032226      INC      ORLOW      ;NO, OUT OF RANGE
1472 006036 000406              BR          REPT5
1473 006040 011203              MOV      (R2),R3      ;GET VALUE TO WORK ON IT
1474 006042 163703 032200      SUB      KSTOR3,R3      ;OBTAIN OFFSET
1475 006046 006303              ASL      R3
1476 006050 005263 032230      INC      MINUS9(R3)      ;INCREMENT CNTR
1477 006054 005722      REPT5:  TST      (R2)+      ;INCREMENT POINTER
1478 006056 005300              DEC      R0      ;DONE?
1479 006060 001353              BNE      REPT4      ;NO
1480 006062 004737 021502      JSR      PC,SETRMT      ;CHK FOR AND SET UP REMOTE DST.
1481
1482      ;*****
1483      ;AT THIS POINT THE AVERAGES HAVE BEEN TAKEN AND CATEGORIZED. THE
1484      ;NEXT SECTION DISPLAYS THE COUNTS IN A HISTOGRAM FORMAT.
1485      ;*****
1486
1487 006066 012702 032230      REPT6:  MOV      #MINUS9,R2      ;SET UP COUNT TABLE
1488 006072 005003              CLR      R3
1489 006074 020122              CMP      R1,(R2)+      ;SCAN TABLE FOR CURRENT COUNT
1490 006076 001407              BEQ      REPT7      ;COUNT FOUND, PRINT IT
1491 006100 005203              INC      R3
1492 006102 022702 032300      REPT6A: CMP      #ORHIGH,R2      ;SCANNED WHOLE TABLE?
1493 006106 001372              BNE      REPT6+6      ;NO, CONTINUE
1494 006110 005301      REPT6B: DEC      R1      ;YES, CHECKED ALL COUNTS?
1495 006112 001365              BNE      REPT6      ;NO, RE-SCAN TABLE
1496 006114 000422              BR          REPT9      ;TYPE FINAL DATA
1497
1498 006116 104012      REPT7:  PRINT
1499 006120 031712      DASH
1500 006122 010337 021262      REPT8:  MOV      R3,SPACEX      ;ANY SPACES TO BE TYPED?
1501 006126 001401              BEQ      .+4      ;NO, PRINT ASTRICK
1502 006130 104016              SPACE      ;YES, PRINT SPACE
1503 006132 005342              DEC      -(R2)      ;SUBTRACT '1' FROM COUNT
1504 006134 005722              TST      (R2)+
1505 006136 104012              PRINT
1506 006140 031700      ASTRIC
1507 006142 005003      CLR      R3
1508 006144 022702 032300      REPT8A: CMP      #ORHIGH,R2      ;DONE CURRENT SCAN?
1509 006150 001757              BEQ      REPT6B      ;YES, EXIT
1510 006152 020122              CMP      R1,(R2)+      ;NO, IS THIS COUNT EQUAL?
1511 006154 001762              BEQ      REPT8      ;YES, PRINT IT

```

Line	Address	Hex	Label	Operation	Comments	
1512	006156	005203		INC R3	;NO, INC. SPACE CNTR.	
1513	006160	000771		BR REPT8A		
1514	006162	113701	016236	REPT9: MOV# RECBF0+2,R1	;SAVE GAIN SETTING	
1515	006166	122701	000114	CMPB #'L,R1	;RUNNING WITH LOW GAIN?	
1516	006172	001003		BNE .+10	;NO	
1517	006174	012737	031753	006236	MOV#X1MV,REPT10	
1518	006202	122701	000115	CMPB #'M,R1	;RUNNING WITH MEDIUN GAIN?	
1519	006206	001003		BNE .+10	;NO	
1520	006210	012737	031757	006236	MOV#X100UV,REPT10	
1521	006216	122701	000110	CMPB #'H,R1	;RUNNING WITH HIGH GAIN	
1522	006222	001003		BNE .+10	;NO	
1523	006224	012737	031765	006236	MOV#X10UV,REPT10	
1524	006232	104012		PRINT SCALE	;PRINT HORIZONTAL SCALE HEADER	
1525	006234	031715		REPT10: 0		
1526	006236	000000		XDIV		
1527	006240	031772		JSR PC,REPT13	;PRINT SUMMARY	
1528	006242	004737	006320	REPT11: MOV ORLOW,R5		
1529	006246	013705	032226	ADD ORHIGH,R5	;WERE ANY COUNTS OUT OF RANGE?	
1530	006252	063705	032300	BEQ REPT12	;NO, RE-CYCLE TEST	
1531	006256	001412		PRINT XLOW	;TEXT 'OR-LOW	
1532	006260	104012		MOV ORLOW,R2	;PRINT COUNTS LOW	
1533	006262	032003		BINDEC		
1534	006264	013702	032226	PRINT XHIGH		
1535	006270	104033		MOV ORHIGH,R2		
1536	006272	104012		BINDEC	;PRINT COUNTS 'HI'	
1537	006274	032014		REPT12: PRINT CRLF2		
1538	006276	013702	032300	JSR PC,CLRMOTE	;CLEAR REMOTE DESTINATION	
1539	006302	104033		RPT12A: JMP REPT0		
1540	006304	104012		REPT13: MOV #3,R3	;SET UP PRINT LO-HI & AVG. VALUES	
1541	006306	032034		MOV #AVGTAB,R1	;SET UP BUFFER POINTER	
1542	006310	004737	021524	REPT14: MOV (R1)+,R2	;GET VALUE	
1543	006314	000137	005636	JSR PC,POSTIR	;CONVERT & PRINT IT	
1544				DEC R3		
1545	006320	012703	000003	BNE .+4		
1546	006324	012701	032302	RTS PC		
1547	006330	012102		MOV #2,SPACEX		
1548	006332	004737	007040			
1549	006336	005303				
1550	006340	001001				
1551	006342	000207				
1552	006344	012737	000002	021262		
1553	006352	104016				
1554	006354	000765		BR REPT14		
1555						

```
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566 006356 104012
1567 006360 024641
1568 006362 104026
1569 006364 104035
1570 006366 005037 032222
1571 006372 104032
1572
1573
1574
1575 006374 104012
1576 006376 024666
1577 006400 104012
1578 006402 024714
1579 006404 104034
1580 006406 000114
1581 006410 007625
1582
1583
1584
1585 006412 104012
1586 006414 024730
1587 006416 104034
1588 006420 000114
1589 006422 000011
1590
1591
1592
1593 006424 104012
1594 006426 025014
1595 006430 024714
1596 006432 104034
1597 006434 000114
1598 006436 004226
1599

;*****
;SBTTL M7383 A/D GAIN ACCURACY TEST
;THIS TEST REQUESTS OF A SERIES OF VOLTAGES A SPECIFIED GAIN SETTINGS
;TO BE SUPPLIED TO THE 'A/D'. A SERIES OF A HUNDRED CONVERSIONS ARE TAKEN
;AT EACH OF THESE SETTINGS AND AVERAGED OUT. THIS AVERAGE IS THEN TESTED
;TO BE WRITTEN '+ OR -' A COUNT FROM THE TRUE VOLTAGE VALUE FOR THAT
;SPECIFIED SETTING.
;*****

M7383G: PRINT
          MES18                      ;TEXT 'A/D GAIN TEST'
          ADDRESS
          SETUP                      ;SET UP RESTART ADDR. POINTER
          CLR      LOPSWH            ;REQUEST & STORE CH. TO BE TESTED.
          CHANEL

;TEST '+1.990V' AT 'LOW' GAIN

          PRINT
          MES19                      ;TEXT 'SUPPLY +1.990V'
          PRINT
          MES20                      ;TEXT 'AT LOW GAIN'
          WAITGN
          'L                          ;LOW GAIN
          7625                       ;TRUE VOLTAGE VALUE + OFFSET

;TEST -1.990V AT 'LOW' GAIN

          PRINT
          MES21                      ;SWITCH VOLTAGE NEG.
          WAITGN
          'L
          11                          ;TRUE VOLTAGE VALUE + OFFSET

;TEST +.1990V AT LOW GAIN

          PRINT
          MES24                      ;TEXT 'SUPPLY' +.1990V'
          MES20                      ;TEXT 'SUPPLY +.1990V'
          WAITGN
          'L                          ;GAIN MED.
          4226
```

```
1600      ;TEST '-.1990V AT 'LOW' GAIN
1601
1602      006440 104012      PRINT
1603      006442 024730      MES21      ;TEXT 'SWITCH VOLTAGE NEG.'
1604      006444 104034      WAITGN
1605      006446 000114      'L
1606      006450 003410      3410
1607
1608      ;TEXT '0.0V' AT LOW GAIN
1609
1610      006452 104012      PRINT
1611      006454 025072      MES25      ;TEXT 'SUPPLY +0.000V.'
1612      006456 104034      WAITGN
1613      006460 000114      'L
1614      006462 003717      3717
1615      ;*****
1616      ;TEST COMPLETE
1617      ;*****
1618      006464 104012      PRINT
1619      006466 024236      MES7      ;TEST COMPLETE
1620      006470 000735      BR      M7383G+6      ;RE-START TEST
1621      006472 000240      NOP
1622
1623      ;*****
1624      ;SBTTL M7383 A/D GAIN AVERAGING SUBROUTINE
1625      ;THIS SUBROUTINE WAITS FOR 'CR' THEN TAKES AND AVERAGES '100' A/D CONVERSIONS.
1626      ;THIS COMPUTED AVERAGE IS COMPARED AGAINST THE TRUE VOLTAGE VALUE FOR A
1627      ;SPECIFIED SETTING. THE AVERAGE IS PRINTED OUT IF FOUND TO BE MORE THAN '+ OR -'
1628      ;1 COUNT FROM THE AVERAGE
1629      ;*****
1630
1631      006474 017603 000000      XWATGN: MOV      @ (SP),R3      ;PICK UP GAIN CODE FROM CALL +2
1632      006500 062716 000002      A'      #2,(SP)
1633      006504 017604 000000      ML      @ (SP),R4      ;PICK UP TRUE VOLTAGE VALUE
1634      006510 104013      WAITG1: TTYIN      ;WAIT FOR 'CR' TO CONTINUE
1635      006512 012701 000001      MOV      #1,R1
1636      006516 104027      ADCNVT
1637      006520 120337 016236      CMPB      R3,RECBF0+2      ;IS GAIN CODE CORRECT?
1638      006524 001403      BEQ      .+10      ;YES
1639      006526 104012      PRINT      ;NO, TELL HIM ABOUT IT
1640      006530 025113      MES26
1641      006532 000766      BR      WAITG1      ;WAIT FOR SETUP
1642      006534 012701 000144      MOV      #100.,R1      ;SET UP TO TAKE '100' CONVERSIONS
```

```

1643 006540 012702 017670      WAITG2: MOV      #TRNBF0,R2      ;SAVE THEM HERE
1644 006544 104027                ADCNV      ;TAKE THE CONVERSIONS
1645 006546 104031                AVERAGE   ;AVERAGE THEM
1646 006550 013702 016122      MOV      QUOENT,R2
1647 006554 020402                CMP      R4,R2      ;AVERAGE = TRUE VALUE?
1648 006556 001441                BEQ      GANEXT      ;YES, EXIT
1649 006560 005204                INC      R4
1650 006562 020402                CMP      R4,R2      ;AVERAGE TRUE VALUE +1?
1651 006564 001436                BEQ      GANEXT      ;YES, EXIT
1652 006566 162704 000002      SUB      #2,R4
1653 006572 020402                CMP      R4,R2      ;AVERAGE = TRUE VALUE -1?
1654 006574 001432                BEQ      GANEXT      ;YES, EXIT
1655 006576 032777 020000 172546 WAITG3: BIT      #SW13,@SWR      ;NO, PRINT INHIBIT SW. SET?
1656 006604 001355                BNE      WAITG2      ;YES
1657 006606 032777 020000 172536 BIT      #SW13,@SWR      ;SW SET?
1658 006614 001351                BNE      WAITG2      ;YES, INHIBIT ERROR TYPEOUT.
1659 006616 005737 032222      TST      LOPSWH      ;NO, HAS ERROR HEADER BEEN TYPED?
1660 006622 001004                BNE      .+12
1661 006624 005237 032222      INC      LOPSWH
1662 006630 104012                PRINT
1663 006632 025134                MES27
1664 006634 104012                PRINT
1665 006636 032032                CRLF
1666 006640 104002                SAVREG
1667 006642 012703 000003      MOV      #3,R3
1668 006646 012701 016120      MOV      #LOW,R1
1669 006652 004737 006330      JSR      PC,REPT14
1670 006656 104003                GETREG
1671 006660 000727                BR      WAITG2
1672
1673 006662 0627'6 000002      GANEXT: ADD      #2,(SP)
1674 006666 000002                RTI
1675

```



```
1676      ;*****
1677      ;SBTTL A/D CONVERSION ROUTINE
1678      ;THIS ROUTINE TAKES AN A/D CONVERSION AND ENABLES
1679      ;DL 0'S RECVR TO ACCEPT & STORE THE CONVERTED VALUE
1680      ;*****
1681
1682      006670 104002      XADCNT: SAVREG      ;SAVE REG.'S
1683      006672 012705 017670      MOV      #TRNBF0,R5      ;SAVE CONVERTED VALUES HERE
1684      006676 012746 000000      MOV      #0, -(SP)      ;ENABLE INTERRUPTS
1685      006702 012746 006710      MOV      #ADCT0, -(SP)
1686      006706 000002      RTI
1687      006710 104005      ADCT0: RECVRO      ;ENABLE THE DL11 RECVR
1688      006712 004737 017220      JSR      PC, @#ADRSRC      ;ADDRESS MODULE
1689
1690      006716 005737 016224      ADCT1: TST      RECEOT      ;WAS 'EOT' RETURNED?
1691      006722 001775      BEQ      .-4      ;NO, WAIT FOR CONVERT
1692      006724 012703 016237      MOV      #RECBF0+3,R3      ;SET UP ADDRESS TO PICK UP SIGN
1693      006730 012704 016240      MOV      #PECBF0+4,R4      ;SET UP ADDRESS TO PICK UP DATA
1694      006734 005000      CLR      R0
1695      006736 012437 015330      ADCT2: MOV      (R4)+, INBUF      ;SET UP NO. TO BE CONVERTED
1696      006742 011437 015332      MOV      (R4), INBUF+2
1697      006746 012737 000004 032214      MOV      #4, CHRCNT
1698      006754 104030      BCDBIN      ;CONVERT VALUE TO BINARY
1699      006756 013715 022564      MOV      BCDTAB, (R5)      ;SAVE IT
1700      006762 122713 000053      CMPB      #53, (R3)      ;VALUE POS.?
1701      006766 001401      BEQ      .+4      ;YES, LEAVE AS IS
1702      006770 005415      NEG      (R5)      ;NO,
1703      006772 063725 032130      ADD      OFFSET, (R5)+      ;ADD OFFSET
1704      006776 132737 000010 017231      BITB      #10, SOH1      ;CONVERTING ALL '4' CH.'S?
1705      007004 001411      BEQ      ADCT3      ;NO, EXIT
1706      007006 005200      INC      R0
1707      007010 022700 000004      CMP      #4, R0      ;SAVED ALL VALUES?
1708      007014 001405      BEQ      ADCT3      ;YES, EXIT
1709      007016 062703 000010      ADD      #10, R3      ;NO, PICK UP NEXT ADDRESSES
1710      007022 062704 000006      ADD      #6, R4
1711      007026 000743      BR      ADCT2
1712      007030 005301      ADCT3: DEC      R1
1713      007032 003326      BGT      ADCT0      ;TAKE NEXT CONVERSION
1714      007034 104003      GETREG
1715      007036 000002      RTI      ;NO, EXIT
1716
1717      ;*****
1718      ;SUBROUTINE TO CONVERT THE VALUE IN 'R2' BACK TO A 'TRUE' A/D VALUE
1719      ;AND PRINT IT IN DECIMAL AS EITHER '+ OR-'.
1720      ;*****
1721
1722      007040 104002      POSTIT: SAVREG
1723      007042 012701 000053      MOV      #53, R1      ;SET UP TO PRINT '+'
1724      007046 163702 032130      SUB      OFFSET, R2      ;SUBTRACT OFFSET TO OBTAIN REAL VALUE.
1725      007052 100003      BPL      .+10      ;VALUE POS.?
1726      007054 005402      NEG      R2      ;NO, COMPLIMENT IT
1727      007056 012701 000055      MOV      #55, R1      ;NO, SET UP TO PRINT '-'
1728      007062 104010      TYPEIT
1729      007064 104033      BINDEC      ;TYPE VALUE
1730      007066 104003      GETREG
1731      007070 000207      RTS      PC
```

```

1732      ;*****
1733      ;SBTTL M7384 D/A ADDRESSING TEST
1734      ;*****
1735
1736      M7384A: PRINT
1737      MES45      ;TEXT 'D/A ADDRESSING TEST'
1738      ADDRESS    ;GET THE MODULE ADDRESS
1739      007100 104035  SETUP      ;SETUP TEST PARAMETERS
1740
1741      ;*****
1742      ;THIS SUBTEST ADDRESSES THE D/A MODULE SENDS A CODE OF '70' (MODE 8)
1743      ;THIS SHOULD ENABLE THE SIGNAL 'PROG L' TO BE LOW UNTIL THE 2ND CHAR.
1744      ;IS SENT TO THE MODULE.
1745      ;*****
1746
1747      007102 000240      DAT1:  NOP
1748      007104 000240      NOP
1749      007106 004737 017234 JSR      PC,ADRST      ;ADDRESS THE MODULE
1750      007112 104006      LDCHRO
1751      007114 000070      70      ;SEND THE CHAR. '8'
1752      007116 104012      PRINT
1753      007120 026242      MES46      ;TEXT 'SCOPE FOR 'PROG L' HI'
1754      007122 104013      TTYIN      ;WAIT FOR 'CR' TO CONTINUE
1755      007124 104006      LDCHRO      ;SEND 'EOT'
1756      007126 000004      EOT
1757      007130 104012      PRINT
1758      007132 026305      MES47      ;SCOPE FOR 'PROG L HI & FLOP L LO'
1759      007134 104013      TTYIN
1760
1761      ;*****
1762      ;THIS SUBTEST 1ST ADDRESSES THE MODULE AND SENDS MODE '9' TO SET
1763      ;THE 'FLOP' FLOP. THEN THE MODULE IS RE-ADDRESSED AND SENDS MODE '8'
1764      ;TO CLR THE 'FLOP' FLOP.
1765      ;*****
1766
1767      007136 104001      DAT2:  SCOPE
1768      007140 000002      2
1769      007142 004737 017234 JSR      PC,ADRST      ;ADDRESS MODULE
1770      007146 104007      LDPGM0
1771      007150 007154      .+4
1772      007152 000401      BR      .+4
1773      007154 071      .BYTE 71      ;SEND CHAR. '9'
1774      007155 004      .BYTE EOT
1775
1776      007156 104012      PRINT
1777      007160 026352      MES48      ;SCOPE FOR 'FLOP L' HI'
1778      007162 104013      TTYIN
1779
1780      007164 004737 017234 JSR      PC,ADRST      ;RE-ADDRESS MODULE
1781      007170 104007      LDPGM0
1782      007172 007176      .+4
1783      007174 000401      BR      .+4
1784      007176 070      .BYTE 70      ;SEND CHAR. '8'
1785      007177 004      .BYTE EOT
1786
1787      007200 104012      PRINT
  
```


1788 007202 026401
1789 007204 104013

MES49
TTYIN

;SCOPE FOR 'FLOP L' LO

1790

1791

1792

1793

1794

1795 007206 104001

1796 007210 000003

1797 007212 012737 030061 007776

1798 007220 012737 030060 010000

1799 007226 004737 007746

1800 007232 104012

1801 007234 026430

1802 007236 026514

1803 007240 104013

1804

1805

1806

1807

1808

1809 007242 104001

1810 007244 000004

1811 007246 012737 030461 007776

1812 007254 012737 030461 010000

1813 007262 004737 007746

1814 007266 104012

1815 007270 026430

1816 007272 026522

1817 007274 104013

1818

1819

1820

1821

1822

1823 007276 104001

1824 007300 000005

1825 007302 012737 031061 007776

1826 007310 012737 031062 010000

1827 007316 004737 007746

1828 007322 104012

1829 007324 026430

1830 007326 026530

1831 007330 104013

DATST3: SCOPE

3

MOV #30061,DATA3 ;CH. '0' 0 VOLTS

MOV #30060,DATA4

JSR PC,DAOUT ;SEND DATA

PRINT

MES50

MES52

TTYIN

DATST4: SCOPE

4

MOV #30461,DATA3

MOV #30461,DATA4

JSR PC,DAOUT

PRINT

MES50

MES53

TTYIN

DATST5: SCOPE

5

MOV #31061,DATA3

MOV #31062,DATA4

JSR PC,DAOUT

PRINT

MES50

MES54

TTYIN

```
1832
1833
1834 :*****
1835 :THIS SUBTEST OUTPUTS 4.44 VOLTS TO CH. '0'.
1836 :*****
1837 007332 104001 DATST6: SCOPE
1838 007334 000006 6
1839 007336 012737 032061 007776 MOV #32061,DATA3
1840 007344 012737 032064 010000 MOV #32064,DATA4
1841 007352 004737 007746 JSR PC,DAOUT
1842 007356 104012 PRINT
1843 007360 026430 MES50
1844 007362 026536 MES55
1845 007364 104013 TTYIN
1846
1847 :*****
1848 :THIS SUBTEST OUTPUTS 8.88 VOLTS TO CH. '0'.
1849 :*****
1850
1851 007366 104001 DATST7: SCOPE
1852 007370 000007 7
1853 007372 012737 034061 007776 MOV #34061,DATA3
1854 007400 012737 034070 010000 MOV #34070,DATA4
1855 007406 004737 007746 JSR PC,DAOUT
1856 007412 104012 PRINT
1857 007414 026430 MES50
1858 007416 026544 MES56
1859 007420 104013 TTYIN
1860
1861 :*****
1862 :THIS SUBTEST OUTPUTS 0.00 VOLTS TO CH. '1'
1863 :*****
1864
1865 007422 104001 DATS10: SCOPE
1866 007424 000010 10
1867 007426 012737 030062 007776 MOV #30062,DATA3
1868 007434 012737 030060 010000 MOV #30060,DATA4
1869 007442 004737 007746 JSR PC,DAOUT
1870 007446 104012 PRINT
1871 007450 026462 MES51
1872 007452 026514 MES52
1873 007454 104013 TTYIN
```

```

1874
1875
1876      :*****
1877      :THIS SUBTEST OUTPUTS 1.11 VOLTS TO CH '1'.
1878      :*****
1879      DATS11: SCOPE
1880      007456 104001      11
1881      007460 000011      MOV      #30462,DATA3
1882      007462 012737 030462 007776      MOV      #30461,DATA4
1883      007470 012737 030461 010000      JSR      PC,DAOUT
1884      007476 004737 007746      PRINT
1885      007502 104012      MESS1
1886      007504 026462      MESS3
1887      007506 026522      TTYIN
1888      007510 104013
1889
1890      :*****
1891      :THIS SUBTEST OUTPUTS 2.22 VOLTS TO CH. '1'.
1892      :*****
1893      DATS12: SCOPE
1894      007512 104001      12
1895      007514 000012      MOV      #31062,DATA3
1896      007516 012737 031062 007776      MOV      #31062,DATA4
1897      007524 012737 031062 010000      JSR      PC,DAOUT
1898      007532 004737 007746      PRINT
1899      007536 104012      MESS1
1900      007540 026462      MESS4
1901      007542 026530      TTYIN
1902      007544 104013
1903
1904      :*****
1905      :THIS SUBTEST OUTPUTS 4.44 VOLTS TO CH. '1'.
1906      :*****
1907      DATS13: SCOPE
1908      007546 104001      13
1909      007550 000013      MOV      #32062,DATA3
1910      007552 012737 032062 007776      MOV      #32064,DATA4
1911      007556 012737 032064 010000      JSR      PC,DAOUT
1912      007560 004737 007746      PRINT
1913      007572 104012      MESS1
1914      007574 026462      MESS5
1915      007576 026536      TTYIN
1916      007600 104013
1917
1918      :*****
1919      :THIS SUBTEST OUTPUTS 8.88 VOLTS TO CH. '1'.
1920      :*****
1921      DATS14: SCOPE
1922      007602 104001      14
1923      007604 000014      MOV      #34062,DATA3
1924      007606 012737 034062 007776      MOV      #34070,DATA4
1925      007614 012737 034070 010000      JSR      PC,DAOUT
1926      007622 004737 007746      PRINT
1927      007626 104012      MESS1
1928      007630 026462      MESS6
1929      007632 026544      TTYIN
1930      007634 104013

```

```
1928
1929
1930
1931
1932
1933
1934
1935
1936 007636 104001
1937 007640 000015
1938 007642 005037 032144
1939 007646 104012
1940 007650 026705
1941 007652 012737 030063 007776
1942 007660 012737 030060 010000
1943 007666 012737 032471 010002
1944 007674 012737 002060 010004
1945 007702 004737 007746
1946
1947 007706 012737 034463 007776
1948 007714 012737 030065 010000
1949 007722 012737 030060 010002
1950 007730 012737 002060 010004
1951 007736 004737 007746
1952 007742 000743
1953 007744 000000
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963 007746 122737 000063 007776
1964 007754 001403
1965 007756 112737 000004 010002
1966 007764 004737 017234
1967
1968 007770 104007
1969 007772 007776
1970 007774 000207
1971
1972 007776 000000
1973 010000 000000
1974 010002 000000
1975 010004 000000

;*****
;THIS SUBTEST OUTPUTS 0.00 VOLTS TO CH. '0' & 9.5 VOLTS TO CH. '1' AND
;THEN THIS SUBTEST RUNS IN A CONTINUOUS LOOP UNTIL EITHER
;'^R' IS TYPED TO RESTART THE TEST OR '^C' IS TYPED TO
;RETURN TO THE MONITOR.
;*****

DATS15: SCOPE
15
CLR      DLYSWH      ;ENABLE TRANSMITTER DELAY
PRINT
MES61
DAT15A: MOV      #30063,DATA3      ;MODE '3' CH. '0'
MOV      #30060,DATA4
MOV      #32471,DATA5      ;CH. '1'
MOV      #2060,DATA6      ;SEND 'EOT' WITH 'LSB'
JSR      PC,DAOUT
MOV      #34463,DATA3      ;MODE '3' CH. '0'
MOV      #30065,DATA4
MOV      #30060,DATA5      ;CH. '1'
MOV      #2060,DATA6      ;SEND 'EOT' WITH 'LSB'
JSR      PC,DAOUT
BR      DAT15A
HALT

;*****
;M7384 ADDRESS TEST COMPLETE
;*****

;*****
;ROUTINE TO OUTPUT A PRE-LOAD DATA VALUE TO THE D/A CONVERTER.
;*****

DAOUT:  CMPB      #63,DATA3      ;OUTPUTTING BOTH CH.'S?
BEQ      .+10      ;YES
MOVB      #EOT,DATA5      ;NO, TERMINATE AFTER '3' CHAR.'S
JSR      PC,ADRST      ;ADDRESS THE MODULE

LDPGMO
.+4
RTS      PC      ;TRANSMIT THE DATA

DATA3:  0      ;LOW BYTE=MODE, HI BYTE=MSB
DATA4:  0      ;HI BYTE=LSB
DATA5:  0      ;LO BYTE='EOT' OR 'MSB' OF CH. '2'
DATA6:  0
```

```
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991 010006 104012
1992 010010 026552
1993 010012 104026
1994 010014 104035
1995 010016 104012
1996 010020 026576
1997 010022 104013
1998 010024 022737 000007 032214
1999 010032 001371
2000 010034 012701 015330
2001 010040 012702 007777
2002 010044 112122
2003 010046 112122
2004 010050 112122
2005 010052 122721 000054
2006 010056 001357
2007 010060 111103
2008 010062 112122
2009 010064 112122
2010 010066 112122
2011 010070 112722 000004
2012
2013 010074 012701 007776
2014 010100 117711 171246
2015 010104 142711 000310
2016 010110 152711 000060
2017 010114 110337 010002
2018 010120 004737 007746
2019 010124 000763

;*****
;SBTTL M7384 D/A EXERCISER TEST
;THIS TEST ENABLES ANY VALUE THE USER TYPES IN ON THE TELEPRINTER TO BE
;OUTPUTTED FROM THE D/A. WHEN SELECTED, THE TEST REQUESTS FOR TWO THREE DIGIT VALUES
;(SEPARATED VIA COMMA'S) TO BE TYPED IN. THE FIRST VALUE IS THE ONLY ONE
;OUTPUTTED WHEN RUNNING ONLY ONE CHANNEL. IF BOTH CHANNELS ARE SELECTED
;THE FIRST VALUE WILL BE OUTPUTTED ON CHANNEL '0' (X DAC) AND THE
;SECOND VALUE WILL BE OUTPUTTED ON CHANNEL '1' (Y DAC). THE CHANNELS
;ARE SELECTED BY DATA SWITCHES '0 & 1' AND MAY BE SET AND RESET
;AT ANYTIME. SETTING DATA SWITCH '0' WILL SELECTED CHANNEL '0'. SETTING
;DATA SWITCH '1' WILL SELECT CHANNEL 1 AND SETTING BOTH '0 & 1' WILL
;ENABLE BOTH CHANNELS.
;*****

M7384E: PRINT
MESS7
ADDRESS
SETUP
TAG4F: PRINT
MESS8
TTYIN
CMP #7,CHRCNT
BNE TAG4F
MOV #INBUF,R1
MOV #DATA3+1,R2
MOVB (R1)+,(R2)+
MOVB (R1)+,(R2)+
MOVB (R1)+,(R2)+
CMPB #54,(R1)+
BNE TAG4F
MOVB (R1),R3
MOVB (R1)+,(R2)+
MOVB (R1)+,(R2)+
MOVB (R1)+,(R2)+
MOVB #EOT,(R2)+

;D/A EXERCISER TEST
;GET AND SETUP MODULE ADDRESS
;SET UP TEST PARAMETERS

;REQUEST THE D/A VALUES
;GET 'EM
;WERE '7' CHARACTERS INPUTTED?
;NO, ASK 'EM AGAIN
;SET UP TO SAVE THEM

;SAVE 'MSB' OF CH. '0'

;SAVE 'LSB'
;DIGIT BETTER BE THE COMMA
;NO, ILLEGAL INPUT
;SAVE THE 'MSB' OF 2ND WORD

;TERMINATE WITH 'EOT'

TAG4G: MOV #DATA3,R1
MOVB @SWR,(R1)
BICB #310,(R1)
BISB #60,(R1)
MOVB R3,DATA5
JSR PC,@DAOUT
BR TAG4G

;SET UP SAVE SWITCHES
;CLR UNWANTED BITS
;MAKE NO. BCD
;RESTORE 'MSB' OF CH. 2 EACH TIME
;SEND THE DATA
```

```
2020
2021      .SBTTL M7385 (SERIAL) & M7388 (CHAR.) I/O ADDRESS TEST
2022      :*****
2023      :THIS TEST EXERCISES THE 'M7385' MODULE USING THE PDP-11 VIA THE DL11
2024      :AS THE DESTINATION INPUT AND THE SOURCE OUTPUT
2025      :*****
2026
2027      M7385A: PRINT
2028      010126 104012      MES3      ;TEXT 'M7385 MODULE TEST'.
2029      010130 024127      ADDRESS    ;GET MODULE ADDRESS
2030      010132 104026
2031      010134 110037 010637
2032      010140 110037 010643
2033      010144 110037 010723
2034      010150 110037 010727
2035      010154 005037 032222
2036      010160 104035      CLR      LOPSWH
2037      SETUP      ;SET UP TEST PARAMETERS.
2038
2039      :*****
2040      :THIS SUBTEST ADDRESSES THE 'SOURCE' PORTION OF THE MODULE USING
2041      :MODE '0' AND TESTS FOR THE FORCED RETURN OF THE 'EOT'.
2042      :*****
2043      ST7385: NOP
2044      010162 000240      NOP
2045      010164 000240      MOV#60,SOH1 ;SET UP MODE '0'
2046      010166 112737 000060 017231 JSR PC,ADRSRC ;ADDRESS THE MODULE
2047      010174 004737 017220
2048      010200 022712 000004      CMP #EOT,(R2) ;WAS IT RETURNED?
2049      010204 001402      BEQ .+6 ;YES
2050      010206 104022      MODERR ;'EOT' WASN'T FORCED OUT BY SOURCE
2051      010210 030603      ERR5
2052      010212 005737 032132      TST SIOSWH ;SERIAL INPUT
2053      010216 001106      BNE SD5 ;YES, GO TO TEST '5'.
2054
2055      :*****
2056      :THIS SUBTEST ADDRESSES THE SOURCE IN MODE '1' AND CHECKS THAT THE
2057      :'EOT' ISN'T FORCED.
2058      :*****
2059      SD2: SCOPE,2
2060      010220 104001 000002      MOV#61,SOH1 ;SET UP MODE '1'
2061      010224 112737 000061 017231 JSR PC,ADRSRC ;ADDRESS MODULE
2062      010232 004737 017220
2063      010236 005712      TST (R2) ;WAS ANY DATA RETURNED?
2064      010240 001402      BEQ SD3 ;NO-OK
2065      010242 104022      MODERR ;ILLEGAL DATA TRANSFER VIA SOURCE
2066      010244 030724      ERR8
2067
```



```

2068 ;*****
2069 ;AT THIS POINT THE SOURCE MODULE IS ADDRESSED WAITING FOR DATA.
2070 ;THIS SUBTEST ADDRESSES THE DESTINATION MODULE AND TRANSFERS DATA
2071 ;TO THE SOURCE AND CHECKS THAT IT IS RETURNED.
2072 ;*****
2073
2074 010246 104001 000003 SD3: SCOPE,3
2075 010252 004737 017234 JSR PC,ADRST ;ADDRESS DESTINATION
2076 010256 005712 TST (R2) ;HAS ANY DATA RETURNED?
2077 010260 001403 BEQ .+10 ;NO, OK
2078 010262 104022 MODERR ;NO DATA HAS YET BEEN TRANSFERED
2079 010264 030724 ERR8
2080 010266 000436 BR TAG1D+2 ;EXIT ON ERROR
2081
2082 010270 104011 RANDOM ;CREATE A RANDOM DATA BUFFER
2083 010272 012737 010272 020774 MOV #.,RETURN ;RE-SET SCOPE LOOP ADDR.
2084 010300 104005 RECVRO ;ENABLE DL 0'S RECVR
2085 010302 104007 LDPGMO ;TRANSFER '500' WORDS TO SOURCE VIA DEST.
2086 010304 017670 TRNBFO
2087 010306 005737 016234 TST RECBFO ;WAS ANY DATA RECV'D?
2088 010312 001003 BNE .+10 ;YES, VERIFY IT
2089 010314 104022 MODERR ;NO DATA WAS RECV'D BACK FROM SOURCE
2090 010316 031041 ERR11
2091 010320 000421 BR TAG1D+2 ;EXIT ON ERROR
2092 010322 005737 016220 TST PARITY ;WAS PARITY ERROR DETECTED?
2093 010326 001402 BEQ .+6 ;NO, VERIFY DATA
2094 010330 104022 MODERR ;DATA PARITY ERROR
2095 010332 030677 ERR7
2096 010334 012701 016234 MOV #RECBFO,R1 ;SET UP TO COMPARE RECV'D DATA
2097 010340 012702 017670 MOV #TRNBFO,R2 ;AGAINST TRANSMITTED DATA
2098 010344 022122 CMP1A: CMP (R1)+,(R2)+ ;DATA MATCH?
2099 010346 001403 BEQ .+10 ;YES, CONTINUE
2100 010350 104022 MODERR ;RECEIVED DATA DOESN'T MATCH TRANSMITTED DATA
2101 010352 030526 ERR3
2102 010354 000403 BR TAG1D+2
2103 010356 022702 020372 CMP #TRNEND,R2 ;DONE?
2104 010362 001370 TAG1D: BNE CMP1A ;NO
  
```

```

2105 ;*****
2106 ;AT THIS POINT DATA HAS BEEN TRANSFERED TO THE DESTINATION AND
2107 ;RECEIVED BACK FROM THE SOURCE. THIS SUBTEST TRANSFERS AN 'EOT' FOLLOWED
2108 ;BY DATA TO VERIFY THAT THE 'EOT' CLEARS THE SOURCE & DESTINATION.
2109 ;*****
2110
2111 010364 104001 000004 SD4: SCOPE,4
2112 010370 104006 LDCHRO ;TRANSMIT 'EOT'
2113 010372 000004 EOT
2114 010374 104007 LDPGM0 ;FOLLOW 'EOT' WITH SOME DATA
2115
2116 010376 010402 .+4
2117 010400 000402 BR TAG1E
2118
2119 010402 101 .BYTE 'A ;SEND A COUPLE OF DATA CHAR.'S
2120 010403 102 .BYTE 'B
2121 010404 000 .BYTE 0 ;TERMINATE
2122 010406 .EVEN
2123
2124 010406 005737 016224 TAG1E: TST RECEOT ;WAS 'EOT' RECV'D?
2125 010412 001003 BNE .+10 ;YES
2126 010414 104022 MODERR ;'EOT' WASN'T RETURNED
2127 010416 030603 ERR5
2128 010420 000405 BR TAG1F+2 ;EXIT ON ERROR
2129
2130 010422 022712 000004 CMP #EOT,(R2) ;WAS 'EOT' ONLY CHAR. RETURNED?
2131 010426 001402 BEQ .+6 ;YES
2132 010430 104022 MODERR ;ILLEGAL DATA TRANSFER
2133 010432 030724 TAG1F: ERR8
  
```



```

2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148 010434 104001 000005
2149 010440 012746 000000
2150 010444 012746 010452
2151 010450 000002
2152 010452 104011
2153 010454 112737 000004 017767
2154 010462 005037 017770
2155 010466 004737 017234
2156
2157 010472 104007
2158 010474 017670
2159
2160 010476 112737 000060 017231
2161 010504 004737 017220
2162 010510 005737 016224
2163 010514 001775
2164
2165 010516 012701 017670
2166 010522 122122
2167 010524 001403
2168 010526 104022
2169 010530 030526
2170 010532 000420
2171 010534 020127 017770
2172 010540 001370
2173 010542 005737 032132
2174 010546 001412
2175 010550 105737 017771
2176 010554 001103
2177 010556 105237 017771
2178 010562 022737 000002 016224
2179 010570 001374
2180 010572 000751

*****
;FIFO CHARACTER STORAGE TEST
;THIS SUBTEST ADDRESSES THE DESTINATION MODULE THEN TRANSMITS
;'63' AND AN 'EOT'. THE SOURCE MODULE IS THEN ADDRESSED
;AND IT SHOULD TRANSMIT THESE CHARACTERS BACK TO THE PDP-11.
;IT SHOULD BE NOTED THAT WHEN THIS TEST IS RUN USING THE
;SERIAL INPUT OPTION, ONE HUNDRED AND TWENTY-EIGHT (128)
;CHARACTERS WILL BE RETURNED TO THE DL11 RECEIVER. THE FIRST
;'64' CHARACTERS ARE RECEIVED BACK FROM THE SERIAL INPUT
;DESTINATION, AND THE SECOND '64' CHARACTERS ARE THE CHARACTERS
;THAT WERE ACUTALLY STORED IN THE 'FIFO' OF THE MODULE UNDER TEST.
*****

SD5:  SCOPE,5
      MOV    #0,      -(SP)    ;ENABLE INTERRUPTS
      MOV    #1$,    -(SP)
      RTI
1$:   RANDOM
      MOV    #EOT,TRNBFO+77    ;CREATE A RANDOM DATA BUFFER
      CLRB   TRNBFO+100        ;TERMINATE BUFFER AFTER '64' BYTES
      JSR    PC,ADRST         ;TERMINATE BUFFER
                                ;ADDRESS DESTINATION MODULE
TAG1H: LDPMO
      TRNBFO
                                ;TRANSMIT DATA
TAG1L: MOV    #60,SOH1
      JSR    PC,ADRSRC
      TST    RECEOT
      BEQ    .-4
                                ;SET UP FOR MODE '0'
                                ;ADDRESS SOURCE
                                ;RECEIVED ALL DATA BACK?
                                ;NO, WAIT FOR 'EOT'
CMP1C: MOV    #TRNBFO,R1
      CMP1B: CMPB (R1)+,(R2)+
                                ;TO TRANSMITTED DATA
                                ;DATA MATCH?
      BEQ    .+10
                                ;YES
                                ;RECV'D DATA NOT EQUAL TO TRANS. DATA
      MODERR
      ERK3
      BR     SD6
                                ;EXIT ON ERROR
      CMP    R1,#TRNBFO+100
                                ;DONE?
      BNE    CMP1B
                                ;NO
      TST    SIOSWH
                                ;USING THE SERIAL I/O INPUT?
      BEQ    SD6
                                ;NO, CHECK ONLY '64' CHAR.'S
      TSTB   TRNBFO+101
                                ;YES, HAVE WE CHK'D '128' CHAR.'S?
      BNE    SD10
                                ;YES, EXIT
      INCB   TRNBFO+101
                                ;NO, CHK NEXT '64' CHAR.'S FROM 'FIFO'
      CMP    #2,RECEOT
                                ;RECEIVED ALL DATA FROM FIFO?
      BNE    .-6
                                ;NO, WAIT FOR 'EOT'
      BR     CMP1C
                                ;DO IT.

```

```

2181
2182
2183
2184
2185
2186
2187 010574 104001 000006
2188 010600 012746 000340
2189 010604 012746 010612
2190 010610 000002
2191 010612 005737 032132
2192 010616 001062
2193 010620 004737 005154
2194
2195
2196
2197
2198
2199
2200 010624 104001 000007
2201 010630 104007
2202 010632 010636
2203 010634 000404
2204 010636 021
2205 010637 061
2206 010640 001
2207 010641 061
2208 010642 022
2209 010643 061
2210 010644 023
2211 010645 003
2212
2213 010646 104006
2214 010650 000102
2215
2216 010652 122722 000003
2217 010656 001403
2218 010660 104022
2219 010662 031257
2220 010664 000435
2221
2222 010666 105722
2223 010670 001403
2224 010672 104022
2225 010674 031171
2226 010676 000430

;*****
;THIS SUBTEST ADDRESSES THE 'SOURCE' USING THE WRONG MODULE ADDRESSES
;AND TESTS THAT THE SOURCE ISN'T ENABLED.
;*****

SD6:  SCOPE,6
      MOV #340, -(SP) ;INHIBIT INTERRUPTS
      MOV #1$, -(SP)
      RTI
1$:   TST SIOSWH ;USING SERIAL INPUT OPTION?
      BNE SD10 ;YES, SKIP THE NEXT TEST.
      JSR PC,@ADRSIT

;*****
;THIS SUBTEST CHECKS THAT 'ETX' WILL CLEAR THE SOURCE AND THAT 'STX'
;WILL CLEAR THE DESTINATION
;*****

SD7:  SCOPE,7
      LDPMO ;ADDRESS MODULE
      .+4
      BR TAG1K
STADR7: .BYTE DC1 ;ALERT SOURCE
        .BYTE 61
        .BYTE SOH
        .BYTE 61 ;MODE '1'
        .BYTE DC2 ;ALERT DESTINATION
STADR8: .BYTE 61
        .BYTE DC3 ;ENABLE MODULE
        .BYTE ETX ;CLR SOURCE

TAG1K: LDCHRO ;SEND A DATA CHAR.
      'B

      CMPB #ETX,(R2)+ ;WAS 'ETX' RETURNED?
      BEQ .+10 ;YES
      MODERR ;'ETX' WASN'T RETURNED
      ERR16
      BR TAG1W ;EXIT ON ERROR

      TSTB (R2)+ ;WAS ANY OTHER DATA RECV'D?
      BEQ .+10 ;NO-OK
      MODERR ;ETX DIDN'T CLR SOURCE
      ERR14
      BR TAG1W ;EXIT ON ERROR

```

```

2227      ;NOW CLR DESTINATION
2228
2229      010700 104007      LDPGMO
2230      010702 010706      .+4
2231      010704 000402      BR      TAG1S
2232      010706      002      .BYTE      STX      ;CLR DESTINATION
2233      010707      101      .BYTE      'A      ;SEND SOME DATA
2234      010710      130      .BYTE      'X
2235      010711      000      .BYTE      0      ;TERMINATE
2236
2237      ;NOW RE-ADDRESS SOURCE & DESTINATION AND EXAMINE DATA
2238
2239      010712 104005      TAG1S:  RECVRO
2240      010714 104007      LDPGMO      ;RE-ADDRESS SOURCE
2241      010716 010722      .+4
2242      010720 000404      BR      TAG1T
2243
2244      010722      021      .BYTE      DC1      ;ALERT SOURCE
2245      010723      061      STADR9: .BYTE      61
2246      010724      001      .BYTE      SOH
2247      010725      061      .BYTE      61      ;MODE '1'
2248      010726      022      .BYTE      DC2      ;ALERT DESTINATION
2249      010727      061      STAD10: .BYTE      61
2250      010730      023      .BYTE      DC3      ;ENABLE MODULE
2251      010731      000      .BYTE      0
2252
2253      010732 005737 016226      TAG1T:  TST      RECSTX      ;WAS 'STX' RETURNED?
2254      010736 001003      BNE      .+10      ;YES
2255      010740 104022      MODERR      ;'STX' WASN'T RECV'D FROM DEST.
2256      010742 030440      ERR1
2257      010744 000405      BR      TAG1W      ;EXIT ON ERROR
2258
2259      010746 105737 016236      TSTB      RECBF0+2      ;WAS 'STX' THE ONLY DATA RECV'D
2260      010752 001402      BEQ      .+6      ;YES
2261      010754 104022      MODERR      ;'STX' DIDN'T CLR DEST.
2262      010756 031073      ERR12
2263
2264      ;SEND AN 'EOT' TO CLR MODULE
2265
2266      010760 104006      TAG1W:  LDCHRO
2267      010762 000004      EOT      ;CLR MODULE

```

```
2268 ;*****
2269 ;THIS SUBTEST REQUESTS THE OPERATOR TO RE-SET THE MODULE ADDRESS TO '17'.
2270 ;AND INSERT THE STRAP TO INHIBIT THE 'EOT' FROM BEING TRANSMITTED.
2271 ;IF DATA 'SW10' IS NOT SET THIS MANUAL INTERVENTION TEST IS SKIPPED.
2272 ;*****
2273
2274 010764 104001 000010 SD10: SCOPE,10
2275 010770 032777 002000 170354 BIT #SW10,@SWR ;SW10 SET?
2276 010776 001456 BEQ TAG1P ;NO, TYPE TEST COMPLETE
2277 011000 104012 PRINT
2278 011002 024460 MES14
2279 011004 024316 MES10 ;TEXT 'RE-SET MODULE ADDRESS TO '17'.
2280 011006 104013 TTYIN ;WAIT FOR 'CR' TO CONTINUE
2281
2282 011010 012737 011010 020774 TAG1Q: MOV #.,RETURN ;RE-SET SCOPE LOOP ADDRESS POINTER
2283 011016 112737 000077 032134 MOVB #77,MODADR ;SET UP FOR ADDR. '17'
2284 011024 112737 000077 017227 MOVB #77,SRCADR
2285 011032 112737 000077 017277 MOVB #77,DSTADR
2286 011040 104005 RECVRO
2287 011042 004737 017234 JSR PC,ADR DST ;ENABLE DL 0'S RECVR.
2288 ;ADDRESS DEST. MODULE
2289 011046 104007 TAG1R: LDPGMO ;SEND SOME DATA
2290 011050 011054 .+4
2291 011052 000402 BR TAG1U
2292 011054 101 .BYTE 'A' ;SEND DATA
2293 011055 102 .BYTE 'B'
2294 011056 004 .BYTE EOT ;TERMINATE
2295 011060 011060 .EVEN
2296 011060 104005 TAG1U: RECVRO ;CLR & RESET BUFFER
2297 011062 112737 000060 017231 MOVB #60,SOH1 ;SET UP FOR MODE '0'
2298 011070 004737 017220 JSR PC,ADR SRC ;ADDRESS THE SOURCE
2299
2300 011074 022712 041101 TAG1Z: CMP #41101,(R2) ;WAS THE 'A & B' RETURNED?
2301 011100 001403 BEQ .+10 ;YES
2302 011102 104022 MODERR ;MODULE WASN'T ADDRESS W/ '17'
2303 011104 031041 ERR11
2304 011106 000405 BR SD11 ;EXIT ON ERROR
2305 011110 005737 016224 TST RECEOT ;WAS 'EOT' STRAPPED OUT?
2306 011114 001402 BEQ .+6 ;YES
2307 011116 104022 MODERR ;'EOT' WASN'T STRAPPED OUT
2308 011120 031223 ERR15
2309
2310 ;*****
2311 ;TEST COMPLETE
2312 ;*****
2313
2314 011122 104001 000011 SD11: SCOPE,11
2315 011126 104012 PRINT
2316 011130 024527 MES14A ;TEXT 'REMOVE STAP'
2317 011132 104026 ADDRESS ;SET UP NEW MODULE ADDRESS
2318 011134 104012 TAG1P: PRINT
2319 011136 024236 MES7 ;TEXT 'TEST COMPLETE'
2320 011140 113700 032134 MOVB MODADR,R0
2321 011144 000137 010134 JMP M385A1 ;RESTART TEST
```

```
2322
2323      ;*****
2324      ;SBTTL M7385 SERIAL I/O INTERFACE TEST
2325      ;THIS TEST IS USED TO TEST THAT THE SERIAL I/O INTERFACE MODULE IS FUNCTIONING
2326      ;CORRECTLY. TO RUN THIS TEST THE 'L' JUMPER MUST BE INSERTED ON THE M7385
2327      ;SO AS TO BE INITIALIZED ON POWER UP. REMOVE THE CONTROL MODULE AND
2328      ;TIE THE 'T&R' BUSES TOGETHER.
2329      ;*****
2330      M7385I: PRINT                ;TEXT 'M7385 SERIAL INTERFACE TEST
                MES66
                SETUP
2331      011150 104012
2332      011152 027223
2333      011154 104035
2334
2335      ;*****
2336      ;THIS TEST SIMPLY ADDRESSES THE DESTINATION PORTION OF THE MODULE WHICH
2337      ;WILL ENABLE A CLOSED LOOP FOR DATA BEING SENT TO THE SOURCE.
2338      ;*****
2339
2340      011156 104011      TEST1:  RANDOM                ;CREATE A RANDOM DATA BUFFER.
2341      011160 104007      LDPGM0                ;ADDRESS DESTINATION
2342      011162 011166      .+4
2343      011164 000402      BR          TST1A
2344      011166          022      .BYTE  DC2                ;ALERT THE DESTINATION
2345      011167          060      IADR10: .BYTE  60          ;MODIFIED BY USER
2346      011170          017      .BYTE  SI                ;ENABLE DESTINATION
2347      011171          000      .BYTE  0                ;TERMINATE
2348      .EVEN
2349
2350      011172 005712      TST1A:  TST      (R2)                ;HAS ANY DATA RETURNED?
2351      011174 001403      BEQ          .+10                ;NO, OK
2352      011176 104022      MODERR                ;NO DATA HAS YET BEEN TRANSFERED
2353      011200 030724      ERR8
2354      011202 000434      BR          TEST2                ;EXIT ON ERROR
2355
2356      011204 104011      RANDOM                ;CREATE A RANDOM DATA BUFFER
2357      011206 012737 011206 020774      MOV      #.,RETURN  ;RE-SET SCOPE LOOP ADDR.
2358      011214 104005      RECVRO                ;ENABLE DL 0'S RECVR
2359      011216 104007      LDPGM0                ;TRANSFER '500' WORDS TO SOURCE VIA DEST.
2360      011220 017670      TRNBFO
2361      011222 005712      TST      (R2)                ;WAS ANY DATA RECV'D?
2362      011224 001003      BNE          .+10                ;YES, VERIFY IT
2363      011226 104022      MODERR                ;NO DATA WAS RECV'D BACK FROM SOURCE
2364      011230 031041      ERR11
2365      011232 000420      BR          TEST2                ;EXIT ON ERROR
2366      011234 005737 016220      TST      PARITY          ;WAS PARITY ERROR DETECTED?
2367      011240 001403      BEQ          .+10                ;NO, VERIFY DATA
2368      011242 104022      MODERR                ;DATA PARITY ERROR
2369      011244 030677      ERR7
2370      011246 000412      BR          TEST2
2371      011250 012701 017670      MOV      #TRNBFO,R1      ;AGAINST TRANSMITTED DATA
```

```

2372 011254 022122          CMPT1A: CMP      (R1)+,(R2)+      ;DATA MATCH?
2373 011256 001403          BEQ      .+10                ;YES, CONTINUE
2374 011260 104022          MODERR                      ;RECEIVED DATA DOESN'T MATCH TRANSMITTED DATA
2375 011262 030526          ERR3
2376 011264 000403          BR      TEST2
2377 011266 022701 020372  CMP      #TRNEND,R1          ;DONE?
2378 011272 001370          BNE      CMPT1A              ;NO
2379
2380 ;*****
2381 ;AT THIS POINT DATA HAS BEEN TRANSFERED TO THE DESTINATION AND
2382 ;RECEIVED BACK FROM THE SOURCE. THIS SUBTEST TRANSFERS AN 'EOT' FOLLOWED
2383 ;BY DATA TO VERIFY THAT THE 'EOT' CLEARS THE SOURCE & DESTINATION.
2384 ;*****
2385
2386 011274 104001          TEST2: SCOPE
2387 011276 000002          2
2388 011300 104006          LDCHRO                      ;TRANSMIT 'EOT'
2389 011302 000004          EOT
2390 011304 104007          LDPGMO                      ;FOLLOW 'EOT' WITH SOME DATA
2391 011306 011312          .+4
2392 011310 000402          BR      TST2A
2393
2394 011312 101             .BYTE  'A                      ;SEND A COUPLE OF DATA CHAR.'S
2395 011313 102             .BYTE  'B
2396 011314 000             .BYTE  0                      ;TERMINATE
2397 011316 011316          .EVEN
2398
2399 011316 005737 016224  TST2A: TST      RECEOT          ;WAS 'EOT' RECV'D?
2400 011322 001003          BNE      .+10                ;YES
2401 011324 104022          MODERR                      ;'EOT' WASN'T RETURNED
2402 011326 030603          ERR5
2403 011330 000405          BR      TST2B+2              ;EXIT ON ERROR
2404
2405 011332 022712 000004  CMP      #EOT,(R2)          ;WAS 'EOT' ONLY CHAR. RETURNED?
2406 011336 001402          BEQ      .+6                ;YES
2407 011340 104022          MODERR                      ;ILLEGAL DATA TRANSFER
2408 011342 030724          TST2B: ERR8
2409
2410 ;*****
2411 ;TEST COMPLETE
2412 ;*****
2413
2414 011344 104001          TEST3: SCOPE
2415 011346 000003          3
2416 011350 104012          PRINT
2417 011352 024236          MES7                          ;TEXT 'TEST COMPLETE
2418 011354 000137 001376  JMP      MONITR              ;RETURN TO MONITOR

```


2419
2420
2421
2422
2423 011360 104012
2424 011362 027037
2425 011364 104026
2426 011366 110037 011441
2427 011372 110037 011445
2428 011376 104035
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441 011400 012746 000340
2442 011404 012746 011412
2443 011410 000002
2444 011412 104005
2445 011414 012746 000000
2446 011420 012746 011426
2447 011424 000002
2448 011426 005237 032146
2449 011432 104025
2450 011434 011440
2451 011436 000404
2452 011440 021
2453 011441 061
2454 011442 001
2455 011443 061
2456 011444 022
2457 011445 061
2458 011446 023
2459 011447 000
2460
2461
2462 011450 005037 032146
2463 011454 105712
2464 011456 001776
2465 011460 005737 032132
2466 011464 001004
2467 011466 111237 011474
2468 011472 104006
2469 011474 000000
2470 011476 122722 000004
2471 011502 001736
2472 011504 000761
2473
2474

```
*****
;SBTTL M7385 TTL I/O TEST
*****

M7385T: PRINT
        MES62                ;TEXT 'TTL I/O TEST'
        ADDRESS              ;GET THE MODULE ADDRESS
        MOVB R0,TTLAD1       ;SET UP MODULE ADDRESS
        MOVB R0,TTLAD2
        SETUP

*****
;THIS TEST VERIFIES THAT THE TTL I/O SECTION OF THE SERIAL I/O MODULE
;IS FUNCTIONING CORRECTLY. IT REQUIRES FOR A TELEPRINTER TO BE CABLED TO
;THE MATON LOCK OF THE SERIAL I/O (THIS COULD BE THE CONSOLE PRINTER ONCE
;THE TEST IS SELECTED). ALL CHARACTERS THEN INPUTTED WILL BE RECEIVED BY
;THE SERIAL SOURCE AND WRAPPED AROUND (BY THE CONTROL MODULE OR
;COMPUTER IF DF11 IS USED) TO THE DESTINATION. HERE THE CHARACTER WILL BE
;TRANSMITTED BACK TO THE TELEPRINTER AND PRINTED. EFFECTIVELY, AS FAR AS
;THE USER IS CONCERNED, THIS TEST ACTS LIKE A KEYBOARD ECHO TEST.
*****

TTLTST: MOV #340, -(SP)      ;INHIBIT INTERRUPTS
        MOV #1$, -(SP)
        RTI
1$:      RECVRO              ;ENABLE DL11 RECVR.
        MOV #0, -(SP)       ;ENABLE INTERRUPTS
        MOV #2$, -(SP)
        RTI
2$:      INC DSTSWH
        SOURCE
        .+4
        BR TAG7A            ;ADDRESS THE MODULE
TTLAD1: .BYTE DC1            ;ADDRESS MODIFIED BY USER
        .BYTE 61            ;MODE 1, WAIT FOR DATA
        .BYTE SOH
        .BYTE 61
TTLAD2: .BYTE DC2            ;ALERT DEST.
        .BYTE 61            ;ADDRESS MODIFIED BY USER
        .BYTE DC3
        .BYTE 0
        .EVEN

TAG7A:   CLR DSTSWH
        TSTB (R2)            ;DATA READY?
        BEQ .-2              ;NO
        TST SIOSWH           ;USING SERIAL I/O
        BNE TAG7B+2          ;YES, TEST ONLY FOR EOT
        MOVB (R2),TAG7B      ;NO, SET UP TO TRANSMIT CHAR.
        LDCHRO
TAG7B:   0
        CMPB #EOT,(R2)+      ;CHAR. = 'EOT'?
        BEQ TTLTST           ;YES, RE-ADDRESS MODULE
        BR TAG7A             ;NO, WAIT FOR NEXT CHAR.
```

```
2475
2476
2477
2478
2479
2480 011506 104012
2481 011510 025726
2482 011512 104026
2483 011514 110037 011623
2484 011520 110037 011625
2485 011524 104035
2486 011526 005037 032144
2487
2488
2489
2490
2491
2492
2493 011532 000240
2494 011534 000240
2495 011536 004737 017220
2496 011542 022712 000004
2497 011546 001402
2498 011550 104022
2499 011552 030603
2500
2501
2502
2503
2504
2505
2506
2507 011554 104001
2508 011556 000002
2509 011560 104036
2510 011562 012746 000340
2511 011566 012746 011574
2512 011572 000002
2513 011574 012746 000000
2514 011600 012746 011606
2515 011604 000002
2516 011606 104005
2517 011610 005237 032146
2518 011614 104025
2519 011616 011622
2520 011620 000403
2521 011622 021
2522 011623 060
2523 011624 022
2524 011625 060
2525 011626 023
2526 011627 000
2527
2528
2529 011630 005037 032146
2530 011634 105712
```

```
*****
;SBTTL M7386 KEYBOARD/DISPLAY MODULE ADDRESS TEST
*****
M7386A: FINT
        MES39
        ADDRESS          ;GET THE MODULE ADDRESS
        MOV      R0,KEYAD1 ;SET IT UP
        MOV      R0,KEYAD2 ;SET UP TEST PARAMETERS
KEYT0:  SETUP
        CLR      DLYSWH   ;ENABLE TRANSMITTER DELAY
*****
;THIS SUBTEST ADDRESSES THE KEYBOARD MODULE AND CHECKS FOR THE
;AUTOMATIC RETURN OF AN 'EOT'.
*****
KEYT1:  NOP
        NOP
        JSR      PC,ADRSRC ;ADDRESS THE MODULE
        CMP      #EOT,(R2) ;WAS 'EOT' RETURNED?
        BEQ      KEYT2     ;YES
        MODERR   ;MODULE DIDN'T RETURN 'EOT'
        ERR5
*****
;THIS SUBTEST ADDRESSES BOTH THE KEYBOARD & THE DISPLAY. THE DATA
;FROM THE KEYBOARD IS DISPLAYED AND ALSO PRINTED OUT ON THE TELETYPE.
;THE TELETYPE OUTPUT CAN BE ELIMINATED BY SETTING DATA SW15.
*****
KEYT2:  SCOPE
        2
        NODLAY
        MOV      #340, -(SP) ;INHIBIT TRANSMITTER DELAY
        MOV      #1$, -(SP) ;INHIBIT INTERRUPTS
        RTI
1$:     MOV      #0, -(SP)   ;ENABLE INTERRUPTS
        MOV      #2$, -(SP)
        RTI
2$:     RECVRO   ;ENABLE DL11 RECEIVER
        INC      DSTSWH
        SOURCE   ;ADDRESS THE MODULE
        .+4
        BR       TAG6A
        .BYTE   DC1      ;ALERT SOURCE
KEYAD1: .BYTE   60
        .BYTE   DC2      ;ALERT DESTINATION
KEYAD2: .BYTE   60
        .BYTE   DC3      ;ENABLE MODULE.
        .BYTE   0
        .EVEN
TAG6A:  CLR      DSTSWH
        TSTB     (R2)      ;DATA READY?
```



```
2531 011636 001776      BEQ      .-2      ;NO, WAIT
2532 011640 005737 032132  TST      SIOSWH    ;USING THE CONTROL MODULE?
2533 011644 001004      BNE      TAG6B+2  ;YES
2534 011646 111237 011654  MOV      (R2),TAG6B ;NO,SET UP TO SEND CHAR TO DISPLAY
2535 011652 104006      LDCHRO
2536 011654 000000      TAG6B: 0
2537 011656 122712 000004  CMP      #EOT,(R2) ;REC. 'EOT'?
2538 011662 001737      BEQ      KEYT2+6 ;YES, RE-ADDRESS MODULE
2539 011664 111201      MOV      (R2),R1 ;GET DATA
2540 011666 032777 020000 167456 BIT      #SW13,@SWR ;INHIBIT PRINTOUT?
2541 011674 001002      BNE      TAG6C    ;YES
2542 011676 004737 015434  JSR      PC,PDMSET ;NO, TYPE IT
2543 011702 122712 000003  TAG6C: CMP      #ETX,(R2) ;REC. AN 'ETX'?
2544 011706 001412      BEQ      KEYT3    ;YES, RUN DISPLAY TEST
2545 011710 122722 000002  CMP      #STX,(R2)+ ;REC. AN 'STX'?
2546 011714 001345      BNE      TAG6A    ;NO, RE-ADDRESS MODULE
2547 011716 005737 032132  TST      SIOSWH    ;YES, USING SERIAL INPUT?
2548 011722 001742      BEQ      TAG6A    ;NO, 'STX' IS LEGAL
2549 011724 104012      PRINT
2550 011726 027473      MES73A    ;TEXT 'RE-INITIALIZE PDM70.'
2551 011730 104013      TTYIN
2552 011732 000713      BR      KEYT2+6 ;WAIT FOR SETUP
2553                                     ;RESTART TEST
```

```
2554                                     ;*****
2555                                     ;THIS SUBTEST IS ENTERED UPON RECEIPT OF AN 'ETX' FROM THE KEYBOARD
2556                                     ;IN THE PREVIOUS SUBTEST. THIS TEST THEN ADDRESSES THE DISPLAY
2557                                     ;AND DISPLAYS THE ENTIRE DISPLAY CHARACTER SET ONE CHARACTER
2558                                     ;AT A TIME. EACH CHARACTER IS DISPLAYED ACROSS THE ENTIRE SCREEN
2559                                     ;FOR APPROXIMATELY ONE SECOND.
2560                                     ;*****
```

```
2561
2562 011734 104001      KEYT3: SCOPE
2563 011736 000003      3
2564 011740 104012      PRINT
2565 011742 027440      MES73    ;TEXT 'DISPLAY TEST'
2566 011744 104013      TTYIN    ;WAIT FOR 'CR'
2567 011746 104036      NODLAY    ;INHIBIT TRANSMITTER DELAY
2568 011750 012737 011750 020774 MOV      #.,RETURN ;RESET SCOPE LOOP POINTER
2569 011756 012737 000040 012004  TAG6E: MOV      #40,TAG6D+2 ;START OFF WITH DISPLAYING SPACES.
2570 011764 104005      RECVR0    ;ENABLE DL11 RECV.
2571 011766 004737 017234  JSR      PC,ADDRST ;ADDRESS THE DESTINATION
2572 011772 012702 000040  MOV      #32.,R2 ;DISPLAY '32' CHAR./LINE
2573 011776 104006      LDCHRO
2574 012000 000212      TAG6D: 212 ;SEND 'LF' TO CLEAR SCREEN
2575 012002 104006      LDCHRO
2576 012004 000040      40 ;MODIFIED TO CHAR. BEING DISPLAYED.
2577 012006 005302      DEC      R2 ;DISPLAYED 32 CHAR.'S?
2578 012010 001374      BNE      TAG6D    ;NO, LOAD NEXT CHAR.
2579 012012 104006      LDCHRO ;YES
2580 012014 000004      EOT ;CLEAR DESTINATION
2581 012016 104004      DELAY ;DELAY SO USER CAN VIEW SCREEN
2582 012020 104004      DELAY
2583 012022 005237 012004      INC      TAG6D+2 ;SETUP NEXT CHAR.
2584 012026 022737 000140 012004  CMP      #140,TAG6D+2 ;DISPLAYED ALL CHAR'S.?
2585 012034 001353      BNE      TAG6E    ;NO,
```

```
2586
2587 ;*****
2588 ;TEST COMPLETE
2589 ;*****
2590
2591 012036 104001 KEYT4: SCOPE
2592 012040 000004 4
2593 012042 104012 PRINT
2594 012044 024236 MES7 ;TEXT 'TEST COMPLETE'
2595 012046 000626 BR KEYT0
2596
2597 ;*****
2598 ;SBTTL M7388 CHARACTER I/O MODULE ADDRESS (IN-HOUSE) TEST
2599 ;THIS TEST USES THE SAME TEST AS THE SERIAL I/O THE
2600 ;TEST HEADER IS TYPED HERE AND THEN THE PROGRAM GOES TO THE
2601 ;SERIAL I/O TESTS TO EXERCISE THE MODULE
2602 ;THIS IS DESIGNATED AS AN IN-HOUSE TEST SINCE A SPECIAL
2603 ;WRAP-A-ROUND MODULE IS REQUIRED TO RUN THE TEST.
2604 ;*****
2605
2606 012050 104012 M7388A: PRINT
2607 012052 026161 MES44 ;TEXT 'CHAR. I/O ADDRESS TEST'
2608 012054 027323 MES69 ;TEXT '(IN-HOUSE)'
2609 012056 000137 010132 JMP @M7385A+4
2610
2611 ;*****
2612 ;SBTTL M7388F CHARACTER I/O MODULE ADDRESS (FIELD) TEST
2613 ;THIS TEST REQUIRES FOR THE FIELD SERVICE TESTER BE CONNECTED TO THE
2614 ;INPUT /OUTPUT OF THE CHARACTER I/O MODULE. THE PROGRAM THEN SENDS
2615 ;SPECIFIC DATA AND THEN REQUESTS THE USER TO VERIFY (WITH HIS TESTER) THIS
2616 ;DATA. THE PROGRAM ALSO REQUESTS THE USER TO INPUT DATA WHICH WILL
2617 ;IN TURN BE PRINTER ON THE CONSOLE DEVICE.
2618 ;*****
2619
2620 012062 104012 M7388F: PRINT
2621 012064 026161 MES44 ;TEXT 'CHARACTER I/O ADDRESS TEST'
2622 012066 027337 MES70 ;TEXT '(FIELD)'.
2623 012070 104026 ADDRESS ;GET THE MODULE ADDRESS
2624 012072 104035 SETUP ;SET UP TEST PARAMETERS
2625
2626 ;*****
2627 ;THIS SUBTEST ADDRESSES THE SOURCE IN MODE '0' AND CHECKS FOR A
2628 ;FORCED 'EOT'.
2629 ;*****
2630
2631 012074 000240 CHART1: NOP
2632 012076 000240 NOP
2633 012100 112737 000060 017231 MOVB #60,SOH1 ;SET UP MODE '0'
2634 012106 004737 017220 JSR PC,ADRSRC ;ADDRESS THE SOURCE
2635 012112 022712 000004 CMP #EOT,(R2) ;WAS 'EOT' RETURNED?
2636 012116 001402 BEQ CHART2 ;YES
2637 012120 104022 MODERR ;'EOT' WASN'T FORCED BY SOURCE
2638 012122 030603 ERR5
```

```
2639
2640
2641
2642
2643
2644
2645
2646
2647 012124 104001
2648 012126 000002
2649 012130 112737 000061 017231
2650 012136 004737 017220
2651 012142 005712
2652 012144 001403
2653 012146 104022
2654 012150 030724
2655 012152 000405
2656 012154 004737 021626
2657 012160 104012
2658 012162 027520
2659 012164 027350
2660 012166 105712
2661 012170 001776
2662 012172 111201
2663 012174 004737 015434
2664 012200 122722 000004
2665 012204 001370
2666
2667
2668
2669
2670
2671
2672
2673
2674 012206 104001
2675 012210 000003
2676 012212 104012
2677 012214 027534
2678 012216 027350
2679 012220 005237 032150
2680 012224 004737 021626
2681 012230 000001
2682 012232 000776
2683 012234 005037 032150
2684 012240 112737 000060 017231
2685 012246 004737 017220
2686 012252 104037

*****
:THIS SUBTEST ADDRESSES THE SOURCE IN MODE '1' AND CHECKS THAT THE
:'EOT' ISN'T FORCED. IT THEN REQUESTS THE USER TO INPUT DATA TO THE MODULE.
:THE INPUTTED DATA WILL BE ECHOED TO THE PRINTER UNTIL AND 'EOT' IS RECEIVED.
:THIS WILL ENABLE THE PROGRAM TO CONTINUE ON TO THE NEXT SUBTEST.
*****

CHART2: SCOPE
2
MOVB #61,SOH1 ;SET UP FOR MODE '1'
JSR PC,ADRSRC ;ADDRESS THE SOURCE
TST (R2) ;WAS ANY DATA RETURNED?
BEQ .+10 ;NO-OK
MODERR ;ILLEGAL DATA TRANSFER
ERR8
BR TAG8A ;PRINT THE RECEIVED DATA
JSR PC,TTYENB ;ENABLE INTERRUPTS
PRINT
MES74 ;TEXT 'ECHO TEST'
MES71 ;TEXT'' INPUT DATA, TERMINATE TEST W/EOT''
TAG8A: TSTB (R2) ;WAIT FOR DATA
BEQ .-2
MOVB (R2),R1
JSR PC,PDMSET ;PRINT IT
CMPB #EOT,(R2)+ ;WAS 'EOT' RECEIVED?
BNE TAG8A

*****
:THIS IS A 'FIFO' STORAGE TEST. IT REQUESTS THE USER TO INPUT DATA (UP TO 63)
:CHARACTERS) AND AN 'EOT'. AFTER THE USER HAS INPUTTED ALL HIS DATA, TYPE 'CR'.
:THE TEST THEN ADDRESSES THE MODULE IN MODE '0' AND THEN PRINTS THE RECEIVED
:DATA WHICH WAS STORED IN THE SOURCE 'FIFO'.
*****

CHART3: SCOPE
3
PRINT
MES75 ;TEXT 'STORAGE TEST'
MES71 ;TEXT'' INPUT DATA & TERMINATE W/EOT''
INr SENDSW ;SET UP TO RETURN ON TTY INTERRUPT
JSR PC,TTYENB ;ENABLE INTERRUPTS
WAIT ;WAIT FOR RECV. INTERRUPTS
BR .-2 ;TTY INTERRUPTS RETURN .+2
CLR SENDSW
MOVB #60,SOH1 ;SET UP FOR MODE '0'
JSR PC,ADRSRC ;ADDRESS THE MODULE
PRTRBF ;PRINT CONTENTS OF THE RECV. BUFFER
```

```
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696 012254 104001
2697 012256 000004
2698 012260 012701 000016
2699 012264 012702 017670
2700 012270 112722 000377
2701 012274 112722 000200
2702 012300 112722 000125
2703 012304 112722 000252
2704 012310 005301
2705 012312 001366
2706 012314 012712 000004
2707 012320 012737 012320 020774
2708 012326 004737 017234
2709 012332 104007
2710 012334 017670
2711 012336 104012
2712 012340 027413
2713 012342 104013
2714
2715
2716
2717
2718
2719
2720 012344 104001
2721 012346 000005
2722 012350 004737 005154
2723
2724
2725
2726
2727
2728 012354 104001
2729 012356 000006
2730 012360 104012
2731 012362 024236
2732 012364 000137 012072
2733
2734

;*****
;THIS SUBTEST LOAD '16','4' CHARACTER DATA PATTERNS (TOTAL OF 64 CHAR.'S)
;INTO THE DESTINATION 'FIFO'. THE USER IS THEN REQUESTED TO STROBE OUT
;THESE '64' CHARACTERS AND VERIFY THEM.
;THE '4' CHARACTER PATTERN IS: ALL 1'S, ALL 0'S, ALTERNATE '1&0'S', AND
;REVERSED ALTERNATE '1&0'S'.
;*****

CHART4: SCOPE
4
MOV #16,R1 ;SET UP THE CHARACTER PATTERN
MOV #TRNBFO,R2 ;SAVE IT IN TRANSMITTER BUFFER
TAG88A: MOVB #377,(R2)+ ;ALL 1'S
MOVB #200,(R2)+ ;ALL 0'S
MOVB #125,(R2)+ ;ALTERNATE '1&0'S'
MOVB #252,(R2)+ ;REVERSED ALTERNATE '1&0'S'
DEC R1 ;LOAD '16' PATTERN'S?
BNE TAG88A ;NO
MOV #EOT,(R2) ;TERMINATE W/EOT
MOV #.,RETURN ;RESET SCOPE LOOP POINTER
JSR PC,ADRST ;ADDRESS DESTINATION
LDPGMO ;TRANSMIT THE '64' CHARACTERS
TRNBFO
PRINT
MES72 ;TEXT 'EXAMINE '64' CHARACTERS
TTYIN ;WAIT FOR 'CR'

;*****
;THIS SUBTEST ADDRESSES THE 'SOURCE' USING ALL THE WRONG MODULE
;ADDRESSES AND CHECKS THAT THE SOURCE ISN'T ENABLED.
;*****

CHART5: SCOPE
5
JSR PC,ADRSIT ;DO IT

;*****
;TEST COMPLETE
;*****

CHART6: SCOPE
6
PRINT
MES7 ;TEXT 'TEST COMPLETE'
JMP M7388F+10

.SBTTL M7377A REMOTE SERIAL I/O TEST
```

```
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747 012370 104012  
2748 012372 027726  
2749 012374 104026  
2750 012376 110037 013103  
2751 012402 110037 013125  
2752 012406 110037 013165  
2753 012412 110037 013170  
2754 012416 110037 013207  
2755 012422 110037 013467  
2756 012426 110037 013531  
2757 012432 110037 013577  
2758 012436 110037 013617  
2759 012442 104035  
2760 012444 005037 032222  
2761  
2762  
2763  
2764  
2765  
2766  
2767 012450 104001 000001  
2768 012454 112737 000062 017231  
2769 012462 004737 017220  
2770  
2771 012466 104004  
2772 012470 022712 000004  
2773 012474 001402  
2774 012476 104022  
2775 012500 030603  
2776
```

```
*****  
:M7377A REMOTE SERIAL I/O TEST  
*****  
  
*****  
:THIS TEST EXERCISES THE 'M7377' MODULE USING THE PDP-11 VIA THE DL11  
:AS THE DESTINATION INPUT AND THE SOURCE OUTPUT  
*****  
  
M7377A: PRINT          :TEXT 'M7377A REMOTE SERIAL I/O TEST'.  
          MES80  
          ADDRESS      :GET MODULE ADDRESS OF MODULE UNDER TEST  
  
M7377B: MOV8 R0,STDR7  
          MOV8 R0,STDR8  
          MOV8 R0,STDR9  
          MOV8 R0,STDR10  
          MOV8 R0,STDR11  
          MOV8 R0,STDR12  
          MOV8 R0,STDR13  
          MOV8 R0,STDR14  
          MOV8 R0,STDR15  
  
M377A1: SETUP  
          CLR LOPSWH  
  
*****  
:THIS SUBTEST ADDRESSES THE 'SOURCE' PORTION OF THE MODULE USING  
:MODE '2' AND TESTS FOR THE FORCED RETURN OF THE 'EOT'.  
*****  
  
ST7377: SCOPE,1  
          MOV8 #62,SOH1 :SET UP MODE '2'  
          JSR PC,ADRSRC :ADDRESS THE MODULE  
  
          DELAY  
          CMP #EOT,(R2) :WAS IT RETURNED?  
          BEQ SD4A :YES  
          MODERR :'EOT' WASN'T FORCED OUT BY SOURCE  
          ERR5
```

```

2777
2778
2779 012502 104001 000002 SD4A: SCOPE,2
2780 ;*****
2781 ;FIFO CHARACTER STORAGE TEST
2782 ;THIS SUBTEST ADDRESSES THE DESTINATION MODULE THEN TRANSMITTS
2783 ;'63' AND AN 'EOT'. THE SOURCE MODULE IS THEN ADDRESSED
2784 ;AND IT SHOULD TRANSMIT THESE CHARACTERS BACK TO THE PDP-11.
2785 ;IT SHOULD BE NOTED THAT WHEN THIS TEST IS RUN USING THE
2786 ;SERIAL INPUT OPTION, ONE HUNDRED AND TWENTY-EIGHT (128)
2787 ;CHARACTERS WILL BE RETURNED TO THE PDP-11 RECEIVER. THE FIRST
2788 ;'64' CHARACTERS ARE RECEIVED BACK FROM THE SERIAL INPUT
2789 ;DESTINATION, AND THE SECOND '64' CHARACTERS ARE THE CHARACTERS
2790 ;THAT WERE ACTUALLY STORED IN THE FIFO OF THE MODULE UNDER TEST.
2791 ;*****
2792
2793
2794
2795 ;NOTE: THE CONTENTS OF THE RECEIVER BUFFER ARE:
2796 ;LOCATIONS 1-62 (1-75 BASE 8) ARE XMITTED/RCVD CHARACTERS.
2797 ;LOC 63: XMITTED/RCVD EOT (76 BASE 8)
2798 ;LOC 64: (77 BASE 8)
2799 ;LOC 65: TERMINATE IF 1,INITIALLY SET TO 0 (2ND BUFFER SWITCH)
2800
2801
2802
2803
2804 012506 012746 000000 S5: MOV #0, -(SP) ;ENABLE INTERRUPTS
2805 012512 012746 012520 MOV #1$, -(SP)
2806 012516 000002 RTI
2807 012520 104011 1$: RANDOM ;CREATE A RANDOM DATA BUFFER
2808 012522 112737 000004 017767 MOVB #EOT,TRNBFO+77 ;TERMINATE BUFFER AFTER '63' BYTES
2809 012530 005037 017770 CLR TRNBFO+100 ;TERMINATE BUFFER
2810 012534 004737 017234 JSR PC,ADRST ;ADDRESS DESTINATION MODULE
2811
2812 012540 104007 TG1H: LDPGMO ;TRANSMIT DATA
2813 012542 017670 TRNBFO
2814
2815 012544 112737 000062 017231 TG1L: MOVB #62,SOH1 ;SET UP FOR MODE '2'
2816 012552 104020 DELAYL ;WAIT FOR THE DATA
2817 012554 104020 DELAYL
2818 012556 104020 DELAYL
2819 012560 004737 017220 JSR PC,ADRSRC ;ADDRESS SOURCE
2820 012564 005737 016224 TST RECEOT ;RECEIVED ALL DATA BACK?
2821 012570 001775 BEQ .-4 ;NO, WAIT FOR 'EOT'
2822
2823 ;NOTE: HANGS HERE WAITING FOR EOT
2824
2825
2826 012572 012701 017670 CP1C: MOV #TRNBFO,R1 ;TO TRANSMITTED DATA
2827 012576 122122 CP1B: CMPB (R1)+,(R2)+ ;DATA MATCH?
2828 012600 001403 BEQ .+10 ;YES
2829 012602 104022 MODERR ;RECV'D DATA NOT EQUAL TO TRANS. DATA
2830 012604 030526 ERR3
2831 012606 000420 BR S5B ;EXIT ON ERROR
2832 012610 120127 017770 CMPB R1,#TRNBFO+100 ;DONE?

```


2833	012614	001370		BNE	CP1B	:NO
2834	012616	005737	032132	TST	S10SWH	:USING THE SERIAL I/O INPUT?
2835	012622	001412		BEQ	S5B	:NO, CHECK ONLY '63' CHAR.'S
2836	012624	105737	017771	TSTB	TRNBFO+101	:YES, HAVE WE CHK'D '128' CHAR.'S?
2837	012630	001007		BNE	S5B	:YES, EXIT
2838	012632	105237	017771	INCB	TRNBFO+101	:NO, CHK NEXT '63' CHARACTERS FROM FIFO
2839	012636	022737	000002 016224	CMP	#2,RECEOT	:RECEIVED ALL DATA FROM FIFO?
2840	012644	001374		BNE	.-6	:NO, WAIT FOR 'EOT'
2841	012646	000751		BR	CP1C	:DO IT.
2842						

CZPMAC0 PDM70 DIAGNOSTIC TEST MACY11 30A(1052) 20-JAN-78 09:11^{I 7} PAGE 65
CZPMAC.P11 19-JAN-78 14:50 M7377A REMOTE SERIAL I/O TEST

SEQ 0086

2843 012650 005737 032132
2844 012654 001402
2845 012656 000137 013644

SSB: TST SIOSWH
BEQ SD5A
JMP TAG1PD

;USING SERIAL I/O? (SYSTEM TEST)?
;YES, SKIP THE FOLLOWING TEST.


```
2846
2847      ;*****
2848      ;THIS TEST CHECKS VARIABLE TERMINATORS BY REQUESTING
2849      ;THAT THE MODULE BE CHANGED TO MODE 2 AND CHECKING THAT
2850      ;THE VARIABLE TERMINATOR EVOKES
2851      ;A TRANSFER.
2852
2853      ;62 CHARACTERS +DEFINED VARIABLE TERMINATOR ARE XMITTED
2854      ;TO THE MODULE.
2855
2856      ;THIS ADDRESSES THE DESTINATION MODULE THEN TRANSMITTS
2857      ;62 CHARACTERS FOLLOWED BY THE CUSTOMER SELECTED TERMINATOR.
2858      ;THE SOURCE MODULE IS THEN ADDRESSED
2859      ;AND IT SHOULD TRANSMIT THESE CHARACTERS BACK TO THE PDP-11.
2860      ;IT SHOULD BE NOTED THAT WHEN THIS TEST IS RUN USING THE
2861      ;SERIAL INPUT OPTION, ONE HUNDRED AND TWENTY-EIGHT (128)
2862      ;CHARACTERS WILL BE RETURNED TO THE DL11 RECEIVER. THE FIRST
2863      ;'64' CHARACTERS ARE RECEIVED BACK FROM THE SERIAL INPUT
2864      ;DESTINATION, AND THE SECOND '64' CHARACTERS ARE THE CHARACTERS
2865      ;THAT WERE ACTUALLY STORED IN THE 'FIFO' OF THE MODULE UNDER TEST.
2866      ;*****
2867      ;
2868      ;
2869      ;THIS TEST CAN ONLY BE CHECKED IF WE ARE NOT USING THE SERIAL I/O MODULE
2870      ;FROM THE PDP-11 TO THE PDM-70.
2871
2872
2873
2874
2875
2876      ;THE REMOTE SERIAL I/O MODULE HAS 4 MODES:
2877      ;
2878      ;MODE:                FUNCTION:
2879
2880      ;0                    CLEAR ALL MODE FUNCTIONS
2881      ;1                    TIME-OUT MODE
2882      ;2                    VARIABLE TERMINATOR MODE
2883      ;4                    REMOTE POWER CLEAR
2884      ;7                    ENABLE ALL FUNCTIONS
2885
2886      ;IMPORTANT:  NOTE THAT THIS SUBTEST WILL 'HANG' IF EOT IS NOT RETURNED
2887
2888
2889
2890      ;NOTE THAT THE REMOTE SERIAL I/O ALWAYS RESPONDS TO 'EOT'
2891      ;IN ALL MODES, BUT ONLY RESPONDS TO VARIABLES IN MODE 2.
2892
2893
2894
2895      012662 104012      SD5A:  PRINT                ;TEXT 'SELECT 12 (LF) ON SWITCH V (CR)'.
2896      012664 027767      MES81
2897      012666 104013      TTYIN                ;WAIT FOR CR.
2898      012670 104001 000003      SCOPE,3
2899      012674 112737 000062 017231      MOV      #62,SOH1
2900
2901      012702 012746 000000      MOV      #0,      -(SP)      ;USE MODE 2
2901                                ;ENABLE INTERRUPTS
```

```

2902 012706 012746 012714      MOV      #1$,      -(SP)
2903 012712 000002              RTI
2904 012714 104011      1$:      RANDOM          ;CREATE A RANDOM DATA BUFFER
2905 012716 005037 017766      CLR      TRNBFO+76          ;CLR HIGH BYTE
2906 012722 012737 002012 017766      MOV      #2012,TRNBFO+76 ;VARIABLE TERMINATOR=LINEFEED.
2907                                ;EOT AFTER LF GETS STRAPPED OUT.
2908                                ;INTO THE LOW BYTE.
2909                                ;NOTE THAT AN EOT WILL BE RETURNED AFTER THE LINEFEED...
2910 012730 005037 017770      CLR      TRNBFO+100        ;TERMINATE BUFFER
2911
2912 012734 004737 017234      JSR      PC,ADRST          ;ADDRESS DESTINATION MODULE
2913
2914 012740 104007      TAG1HA: LDPGMO          ;TRANSMIT DATA
2915 012742 017670      TRNBFO
2916
2917 012744 004737 017220      TAGILA: JSR      PC,ADRSRC        ;ADDRESS SOURCE
2918 012750 005737 016224      TST      RECEOT          ;RECEIVED ALL DATA BACK?
2919 012754 001775      BEQ      .-4              ;NO, WAIT FOR 'EOT'
2920
2921                                ;NOTE: HANGS HERE WAITNG FOR AN EOT....
2922
2923
2924
2925                                ;DATA PLUS AN EOT SHOULD BE RETURNED.
2926
2927 012756 012701 017670      CMP1CA: MOV      #TRNBFO,R1          ;TO TRANSMITTED DATA
2928 012762 122122      CMP1BA: CMPB      (R1)+,(R2)+          ;DATA MATCH?
2929 012764 001403      BEQ      CMP1DA          ;YES
2930 012766 104022      MODERR          ;RECV'D DATA NOT EQUAL TO TRANS. DATA
2931 012770 030526      ERR3
2932 012772 000420      BR      SD6A          ;EXIT ON ERROR
2933 012774 020127 017766      CMP1DA: CMP      R1,#TRNBFO+76      ;DONE?
2934
2935                                ;NOTE: DON'T TRY TO COMPARE THE 'EOT'....
2936 013000 001370      BNE      CMP1BA          ;NO
2937 013002 005737 032132      TST      SIOSWH          ;USING THE SERIAL I/O INPUT?
2938 013006 001412      BEQ      SD6A          ;NO, CHECK ONLY '64' CHAR.'S
2939 013010 105737 017771      TSTB     TRNBFO+101        ;YES, HAVE WE CHK'D '128' CHAR.'S?
2940 013014 001010      BNE      SD6B          ;YES, EXIT
2941 013016 105237 017771      INCB     TRNBFO+101        ;NO, CHK NEXT '64' CHAR.'S FROM 'FIFO'
2942 013022 022737 000002 016224      CMP      #2,RECEOT        ;RECEIVED ALL DATA FROM FIFO?
2943 013030 001374      BNE      .-6              ;NO, WAIT FOR 'EOT'
2944 013032 000751      BR      CMP1CA          ;DO IT.
2945 013034 000240      SD6A:      NOP
2946
2947
2948

```

```
2949
2950
2951
2952
2953
2954
2955 013036 104001 000004
2956 013042 012746 000340
2957 013046 012746 013054
2958 013052 000002
2959 013054 005737 032132
2960 013060 001074
2961 013062 004737 005154
2962
2963
2964
2965
2966
2967 013066 104001 000005
2968 013072 104005
2969 013074 104007
2970 013076 013102
2971 013100 000402
2972 013102 022
2973 013103 067
2974 013104 023
2975 013105 003
2976
2977 013106 104007
2978 013110 013114
2979 013112 000401
2980 013114 102
2981 013115 004
2982
2983 013116 104007
2984 013120 013124
2985 013122 000402
2986 013124 021
2987 013125 067
2988 013126 023
2989 013127 000
2990
2991
2992
2993 013130 122722 000003
2994 013134 001403
2995 013136 104022
2996 013140 031257
2997 013142 000443
2998
2999 013144 105722
3000 013146 001403
3001 013150 104022
3002 013152 031171
3003 013154 000436

;*****
;THIS SUBTEST ADDRESSES THE 'SOURCE' USING THE WRONG MODULE ADDRESSES
;AND TESTS THAT THE SOURCE ISN'T ENABLED.
;*****

SD6B:  SCOPE,4
      MOV #340, -(SP) ;INHIBIT INTERRUPTS
      MOV #1$, -(SP)
      RTI
1$:    TST SIOSWH ;USING SERIAL INPUT OPTION?
      BNE SD10A ;YES, SKIP THE NEXT TEST.
      JSR PC,@ADRSIT

;*****
;THIS SUBTEST CHECKS THAT 'ETX' WILL CLEAR THE SOURCE AND THAT 'STX'
;WILL CLEAR THE DESTINATION
;*****

SD7A:  SCOPE,5
      RECVRO
      LDPGMO ;ADDRESS MODULE
      .+4
      BR TG1KA
      .BYTE DC2 ;ALERT DESTIN
STDR7: .BYTE 67 ;SEND THE ETX TO CLEAR THE SOURCE.
      .BYTE DC3
      .BYTE ETX
      ;SEND THE 'B' AS DATA.

TG1KA: LDPGMO
      .+4
      BR TG1LA
      .BYTE 'B'
      .BYTE EOT
      ;THIS EOT SHOULD CLEAR THE DESTINATION.

TG1LA: LDPGMO
      .+4
      BR TAG1KA
      .BYTE DC1 ;ALERT SOURCE
STDR8: .BYTE 67
      .BYTE DC3 ;ENABLE MODULE TO RECEIVE ANY DATA.
      .BYTE 0
      .EVEN
      ;ONLY 'ETX' SHOULD BE RETURNED.

TAG1KA: CMPB #ETX,(R2)+
      BEQ .+10 ;WAS 'ETX' RETURNED?
      MODERR ;YES
      ERR16 ;'ETX' WASN'T RETURNED
      BR SD10A ;EXIT ON ERROR

      TSTB (R2)+
      BEQ .+10 ;WAS ANY OTHER DATA RECV'D?
      MODERR ;NO-OK
      ERR14 ;ETX DIDN'T CLR SOURCE
      BR SD10A ;EXIT ON ERROR
```

```
3004
3005      ;REMEMBER TO CLEAR THE 'B' AND 'EOT' THAT ARE IN THE BUFFER.
3006
3007      TAG1SA: LDPGM0
3008      .+4
3009      BR TAG1SB
3010      .BYTE DC1      ;SEND THE 'B' & 'EOT' OUT OF FIFO.
3011      STDR9: .BYTE 61
3012      .BYTE DC3
3013      .BYTE DC2
3014
3015      STDR10: .BYTE 61      ;NOW RE-ENABLE THE DESTINATION.
3016      .BYTE DC3
3017      .BYTE STX
3018      .BYTE 'A'
3019      .BYTE 'X'
3020      .BYTE 0
3021      .EVEN
3022      ;NOW RE-ADDRESS SOURCE & DESTINATION AND EXAMINE DATA
3023
3024      TAG1SB: RECVRO
3025      LDPGM0      ;RE-ADDRESS SOURCE
3026      .+4
3027      BR TAG1TA
3028
3029      STDR11: .BYTE DC1      ;ALERT SOURCE
3030      .BYTE 61
3031      .BYTE DC3
3032      .BYTE 0
3033      .EVEN
3034
3035      TAG1TA: TST RECSTX      ;WAS 'STX' RETURNED?
3036      BNE .+10      ;YES
3037      MODERR      ;'STX' WASN'T RECV'D FROM DEST.
3038      ERR1
3039      BR TAG1WA      ;EXIT ON ERROR
3040
3041
3042      ;SKIP OVER EOT HERE AND LOOK FOR AN 'X'.
3043      ;SINCE NO DATA SHOULD HAVE BEEN RETURNED, IT SHOULD BE 0.
3044      ;IF NON-ZERO ,THEN WE HAVE AN ERROR.
3045
3046
3047      TSTB RECBF0+2      ;WAS 'STX' THE ONLY DATA RECV'D
3048      BEQ .+6      ;YES
3049      MODERR      ;'STX' DIDN'T CLR DEST.
3050      ERR12
3051
3052      ;SEND AN 'EOT' TO CLR MODULE
3053
3054      TAG1WA: TSTB RECBF0+4      ;LOOK FOR THE 'X' HERE...
3055      BEQ SD10A      ;BRANCH IF NO ERRORS.
3056      MODERR
3057      ERR12
```

```
3058
3059
3060
3061
3062
3063 013252 104001 000006
3064 013256 032777 002000 166066
3065 013264 001166
3066 013266 104012
3067 013270 024316
3068 013272 104013
3069
3070 013274 012737 013274 020774
3071 013302 112737 000077 032134
3072 013310 112737 000077 017227
3073 013316 112737 000077 017277
3074 013324 104005
3075 013326 004737 017234
3076
3077 013332 104007
3078 013334 013340
3079 013336 000402
3080 013340 101
3081 013341 102
3082 013342 004
3083 013344 013344
3084 013344 104005
3085
3086 013346 004737 017220
3087 013352 104004
3088 013354 022712 041101
3089 013360 001403
3090 013362 104022
3091 013364 031041
3092 013366 000405
3093 013370 005737 016224
3094 013374 001002
3095
3096 013376 104022
3097 013400 031223
3098 013402 104012
3099 013404 030362
3100 013406 113737 013103 032134
3101 013414 113737 013103 017227
3102 013422 113737 013103 017277
3103
3104
3105
3106
3107
3108
3109
3110 013430 104001 000007
3111 013434 104035
3112
3113 013436 104012
```

```
*****
;THIS SUBTEST REQUESTS THE OPERATOR TO RE-SET THE MODULE ADDRESS TO '17'.
;IF DATA 'SW10' IS NOT SET THIS MANUAL INTERVENTION TEST IS SKIPPED.
*****
SD10A: SCOPE,6
      BIT #SW10,@SWR ;SW10 SET?
      BNE TAG1PC ;YES, TYPE TEST COMPLETE
      PRINT
      MES10
      TTYIN ;TEXT 'RE-SET MODULE ADDRESS TO '17'.
           ;WAIT FOR 'CR' TO CONTINUE

TAG1QA: MOV #.,RETURN ;RE-SET SCOPE LOOP ADDRESS POINTER
      MOVB #77,MODADR ;SET UP FOR ADDR. '17'
      MOVB #77,SRCADR
      MOVB #77,DSTADR
      RECVRO
      JSR PC,ADR DST ;ENABLE DL 0'S RECVR.
           ;ADDRESS DEST. MODULE

TAG1RA: LDPGMO
      .+4
      BR TAG1UA ;SEND SOME DATA
      .BYTE 'A' ;SEND DATA
      .BYTE 'B'
      .BYTE EOT ;TERMINATE
      .EVEN
TAG1UA: RECVRO ;CLR & RESET BUFFER
           ;FOR THE NEXT TEST.
           ;ADDRESS THE SOURCE

      JSR PC,ADR SRC
      DELAY
TAG1ZA: CMP #41101,(R2) ;WAS THE 'A & B' RETURNED?
      BEQ .+10 ;YES
      MODERR ;MODULE WASN'T ENABLED WITH ADDRESS '17'
      ERR11
      BR SD11A ;EXIT ON ERROR
      TST RECEOT ;WAS 'EOT' STRAPPED OUT?
      BNE .+6 ;NO.

      MODERR ;'EOT' WAS STRAPPED OUT
      ERR15
SD11A: PRINT ;TEXT 'RESET MODULE ADDRESS<CR>'
      MES88
      MOVB STDR7,MODADR ;RE-STUFF THE ORIGINAL ADDRESSES.
      MOVB STDR7,SRCADR
      MOVB STDR7,DSTADR

*****
; THIS SUBTEST CHECKS MODE 1 FOR TIMEOUT
*****

SCOPE,7
SETUP

PRINT ;TEXT SET CLOCK 3 ON CLOCK MODULE TO 100 MILLISEC
```

```
3114 013440 030152 MES84 ;TEXT SET SWITCH 1 OF P TO ON.
3115 013442 030236 MES85
3116 013444 104013 TTYIN
3117 013446 112737 000061 017231 MOVB #61,SOH1 ;SET UP MODE 1
3118 013454 004737 017220 JSR PC,ADRSRC
3119 013460 104007 LDPMO ;NOW CHECK THE TIME-OUT CLEAR.
3120 013462 013466 .+4
3121 013464 000403 BR TG1PA
3122 013466 022 .BYTE DC2
3123 013467 061 STDR12: .BYTE 61
3124 013470 023 .BYTE DC3
3125 013471 130 .BYTE 'X
3126 013472 101 .BYTE 'A
3127 013473 004 .BYTE EOT
3128
3129
3130 ;ADDRESS SOURCE USING MODE 3
3131
3132
3133
3134 ;ADDRESS NON-EXISTENT SOURCE (240=SPACE).
3135 ;VIA THIS PROGRAM: DC1,240,DC3
3136
3137 013474 112737 000240 017227 TG1PA: MOVB #240,SRCADR ;SET SPACE=ADDRESS TO BE ADDRESSED.
3138 013502 004737 017220 JSR PC,ADRSRC ;ADDRESS THE SOURCE MODULE.
3139 013506 012737 177763 032160 MOV #-15,COUNT
3140 013514 104005 RECVRO
3141
3142 ;WAIT FOR APPROXIMATELY 15 SECONDS...
3143
3144
3145 013516 004737 014472 JSR PC,CNTLOP
3146
3147 013522 104007 LDPMO
3148 013524 013530 .+4
3149 013526 000402 BR TG1PB
3150 013530 021 .BYTE DC1
3151 013531 061 STDR13: .BYTE 61
3152 013532 023 .BYTE DC3
3153 013533 000 .BYTE 0
3154 013534 105722 TG1PB: TSTB (R2)+ ;SKIP OVER THE EOT.
3155 013536 105722 TSTB (R2)+ ;LOOK AT THE BYTE.
3156
3157 013540 001403 BEQ TG1PC ;OK,NO DATA RETURNED.
3158 013542 104022 MODERR ;CLEAR LEFT GARBAGE IN MODULE FIFO.
3159 013544 031633 ERR24
3160
3161 ;NOW CHECK THE REMOTE CLEAR FUNCTION.
3162
3163 013546 104005 RECVRO
3164 013550 112737 000064 017231 TG1PC: MOVB #64,SOH1 ;LEAVE IN MODE 4.
3165 013556 113737 013103 017227 MOVB STDR7,SRCADR
3166 013564 004737 017220 JSR PC,ADRSRC ;ADDRESS THE SOURCE
3167
3168 ;DON'T DELAY THIS TIME.
3169
```



```

3170 013570 104007          LDPGM0          ;RETURNS FIRST EOT
3171
3172 013572 013576          .+4
3173 013574 000405          BR      TG1PE
3174 013576      022          .BYTE  DC2          ;ALERT DESTINATION.
3175 013577      061          STDR14: .BYTE  61
3176 013600      023          .BYTE  DC3
3177 013601      130          .BYTE  'X          ;SEND SOME DATA.
3178 013602      101          .BYTE  'A
3179 013603      005          .BYTE  ENQ          ;SEND ENQ TO DESTINATION.
3180                                     ;ENQ SHOULD CLEAR OUT THE DESTINATION.
3181 013604      000          .BYTE  0
3182                                     .EVEN
3183 013606 104020          DELAYL
3184
3185 013610 104007          TG1PE: LDPGM0
3186 013612 013616          .+4
3187 013614 000402          BR      TG1PF
3188 013616      021          .BYTE  DC1
3189 013617      061          STDR15: .BYTE  61          ;ALERT THE SOURCE.
3190 013620      023          .BYTE  DC3          ;2ND EOT RETURNED HERE
3191                                     .
3192 013621      000          .BYTE  0
3193                                     .EVEN
3194 013622 005722          TG1PF: TST      (R2)+          ;TWO EOT'S ARE EXPECTED BACK.
3195                                     TST      (R2)+          ;SKIP OVER THE EOT'S.
3196 013624 005722          TST      (R2)+          ;AND LOOK TO SEE IF ANY DATA WAS RETURNED.
3197 013626 005722          ;IF DATA CAME BACK, THEN REMOTE CLEAR
3198                                     ;DIDN'T WORK.
3199
3200
3201 013630 001402          BEQ      TG1PG          ;REMOTE CLEARED WORKED ?
3202 013632 104022          MODERR
3203 013634 031557          ERR23          ;NO, IT DIDN'T
3204                                     ;REMOTE CLEAR LEFT GARBAGE IN FIFO.
3205 013636 104001 000010          TG1PG: SCOPE,8.          ;YES, REMOTE CLEAR WORKED.
3206          ;*****
3207          ;TEST COMPLETE
3208          ;*****
3209
3210 013642 104026          TAG1PC: ADDRESS          ;SET UP NEW MODULE ADDRESS
3211 013644 113700 032134          TAG1PD: MOVB      MODADR,R0          ;RESET THE ADDRESS.
3212 013650 104012          PRINT
3213 013652 024236          MES7          ;TEXT 'TEST COMPLETE'
3214 013654 000137 012376          JMP      M7377B          ;RESTART TEST
3215          .SBTTL M7378A FOUNDATION MODULE TEST
3216

```

```
3217 ;*****
3218 ;M7378 FOUNDATION MODULE TEST
3219 ;*****
3220
3221
3222
3223 ;THIS TEST SETS THE SERIAL I/O UP AS A SOURCE AND THE FOUNDATION
3224 ;MODULE AS THE DESTINATION. A RANDOM(PSEUDO) BUFFER
3225 ;IS CREATED AND TRANSMITTED FROM SERIAL I/O TO THE FOUNDATION
3226 ;MODULE. THEN THE FOUNDATION MODULE IS ADDRESSED AS THE SOURCE
3227 ;AND THE SERIAL I/O IS ADDRESSED AS THE DESTINATION. BECAUSE
3228 ;OF THE 'WRAP-AROUND' CABLE, THE DATA IS RETURNED
3229 ;FROM FOUNDATION MODULE TO SERIAL I/O.
3230
3231 ;
3232
3233 ;IF THE SERIAL I/O IS BEING USED, A TOTAL OF 128 CHARACTERS
3234 ;RATHER THAN 64 CHARACTERS WILL BE RETURNED.
3235
3236 ;THE TEST THEN CHECKS TO MAKE SURE THAT ADDRESS 17
3237 ;WILL ALSO RETURN THE DATA.
3238
3239 013660 000000 FLAB7: .WORD 0 ;THIS LOC IS USED TO RESTORE
3240 ;THE CONTENTS OF ADDRESS
3241 ;WHEN LOOPING.
3242
3243 013662 000000 FOUNSW: .WORD 0
3244 013664 104012 ;M7378A: PRINT
3245 013666 027572 MES77 ;TEXT 'FOUNDATION
3246 ;MODULE TEST'.
3247 013670 005037 013662 FLO: CLR FOUNSW ;CLEAR OUT OUR SUBTEST SWITCH.
3248 013674 104035 FLOP: SETUP ;GET THE MODULE ADDRESS.
3249 013676 104026 ADDRESS ;PUT ADDRESS INTO R0.
3250
3251 013700 110037 013660 FLOPB: MOV B R0,FLAB7 ;SAVE THE ADDRESS IN FLAB7.
3252 013704 113700 013660 MOV B FLAB7,R0 ;MODIFY THE FOUNDATION ADDRESS
3253 ;IN THE PDM-70 PROGRAMS.
3254
3255 013710 004737 014660 JSR PC,FSTUF
3256 013714 113737 017202 014046 MOV B IADRS9,IADR11 ;SET UP SER I/O ADDR.
3257 013722 113737 017202 014032 MOV B IADRS9,IADR12
3258 013730 113737 017202 014051 MOV B IADRS9,IADR14
3259 013736 113737 017202 014036 MOV B IADRS9,IADR13
3260
3261 ;
3262 ;
3263 ;*****
3264 ;THIS SUBTEST XMITTS A RANDOM BUFFER TO THE FOUNDATION MODULE.
3265 ;*****
3266 ;
3267 ;NOTE THAT FOUNSW=0 HERE.
3268 ;
3269 013744 104001 000001 SCOPE,1
3270 013750 113737 013660 014042 MOV B FLAB7,FLAB1 ;MODIFY THE FOUNDATION ADDRESS IN PROG.
3271 013756 012746 000000 MOV #0, -(SP) ;ENABLE INTERRUPTS
3272 013762 012746 013770 MOV #1$, -(SP)
```



```

3273 013766 000002
3274 013770 104011
3275 013772 112737 000004 017767 1$: RTI
3276 014000 005037 017770 MOV# #EOT,TRNBFO+77 ;GENERATE RANDOM BUFFER
3277 014004 104005 CLR TRNBFO+100 ;TERMINATE AFTER 64 BYTES.
3278
3279 014006 112737 000060 017231 FOUNDL: MOV# #60,SOH1 ;MODE X
3280 014014 005737 032132 TST SIOSWH ;USING THE SERIAL I/O?
3281 014020 001417 BEQ FNORM ;NO, SO BRANCH TO NORMAL LOAD.
3282 014022 104007 LDPGMO ;ELSE USE PADDED PROGRAM.
3283 014024 014030 .+4
3284 014026 000421 BR FDATA ;XMIT THE DATA NEXT.
3285 014030 002 FPROG: .BYTE STX
3286 014031 021 .BYTE DC1
3287 014032 075 IADR12: .BYTE 75 ;SERIAL I/O SRC.
3288 014033 001 .BYTE SOH
3289 014034 061 .BYTE 61
3290 014035 022 .BYTE DC2
3291 014036 075 IADR13: .BYTE 75
3292 014037 075 FLAB2: .BYTE 75 ;FOUNDATION MODULE
3293 014040 023 .BYTE DC3
3294 014041 021 .BYTE DC1 ;ADDRESS FOUNDATION AS SRC.
3295 014042 075 FLAB1: .BYTE 75
3296 014043 001 .BYTE SOH
3297 014044 060 .BYTE 60
3298 014045 022 .BYTE DC2
3299
3300 ;ADDRESS THE SERIAL I/O AS DESTINATION.
3301
3302 014046 075 IADR11: .BYTE 75
3303 014047 023 .BYTE DC3
3304 014050 021 .BYTE DC1
3305 014051 075 IADR14: .BYTE 75
3306 014052 001 .BYTE SOH
3307 014053 061 .BYTE 61
3308 014054 023 .BYTE DC3
3309 014055 003 .BYTE ETX
3310 014056 000 .BYTE 0
3311 014060 014060 .EVEN
3312 014060 104007 FNORM: LDPGMO
3313 014062 014066 .+4
3314 014064 000402 BR FDATA
3315 014066 022 FLAB3A: .BYTE DC2
3316 014067 075 FLAB3: .BYTE 75 ;FOUNDATION MODULE.
3317 014070 023 .BYTE DC3 ;AS DESTINATION
3318 014071 000 .BYTE 0
3319 .EVEN
3320 014072 104007 FDATA: LDPGMO
3321 014074 017670 TRNBFO ;XMIT THE DATA.
3322 014076 005737 032132 TST SIOSWH
3323 014102 001005 BNE FTST ;;BRANCH IF USING SER I/O.
3324
3325 014104 104007 LDPGMO
3326 014106 014112 .+4
3327 014110 000402 BR FTST ;FOUNDATION AS SOURCE.
3328 014112 021 FLAB5A: .BYTE DC1

```

3329 014113 071
 3330 014114 023
 3331 014115 000

FLAB5: .BYTE 71
 .BYTE DC3
 .BYTE 0
 .EVEN

;DELAY AND CHECK TO MAKE SURE THAT AN EOT HAS BEEN RETURNED.

3338 014116 104020
 3339 014120 104004
 3340 014122 005737 016224
 3341 014126 001002
 3342 014130 104022
 3343 014132 030603

FTST: DELAYL
 DELAY
 TST RECEOT ;GIVE IT TIME TO RETURN.
 BNE FND1C ;LOOK FOR AN EOT.
 MODERR ;YES, EOT WAS RETURNED.
 ERR5 ;'EOT' NOT RETURNED.

;NOW CHECK THE DATA IN THE RECEIVER AND TRANSMITER BUFFERS.
 ;LOOK FOR MATCHES.

3351 014134 012701 017670
 3352 014140 122122
 3353 014142 001403
 3354 014144 104022
 3355 014146 030526
 3356 014150 000420

FND1C: MOV #TRNBFO,R1 ;XMITTED DATA.
 FND1B: CMPB (R1)+,(R2)+ ;DATA MATCH?
 BEQ FND1D ;YES.
 MODERR ;ELSE ERROR
 ERR3 ;XMITTED DATA NOT = RECVD DATA.
 BR FOUND2 ;NON-FATAL ERROR.

;NOW CHECK TO SEE IF WE SHOULD LOOK FOR 64 CHARACTERS OR 128
 ;CHARACTERS. IF WE ARE USING THE SERIAL I/O WE WILL
 ;HAVE 128 CHARACTERS RETURNED (INCLUDING TWO 'EOTS').

;NOTE THAT THE LOW BYTE OF TRNBFO+100
 ;SERVES AS A BUFFER TERMINATOR AND THAT THE
 ;HIGH BYTE SERVES AS A SWITCH. IF THE HIGH BYTE IS SET, THEN
 ;WE HAVE CHECKED ALL 128 CHARACTERS.

3370 014152 020127 017770
 3371 014156 001370
 3372 014160 005737 032132
 3373 014164 001412
 3374 014166 105737 017771
 3375 014172 001007
 3376 014174 105237 017771
 3377 014200 022737 000002 016224
 3378 014206 001374
 3379 014210 000751

FND1D: CMP R1,#TRNBFO+100 ;DONE?
 BNE FND1B ;NOT DONE YET.
 TST SIOSWH ;USING THE SERIAL I/O?
 BEQ FOUND2 ;NO, CK ONLY 64 CHARS
 TSTB TRNBFO+101 ;CHECKED 128 CHARS?
 BNE FOUND2 ;YES, EXIT.
 INCB TRNBFO+101 ;NO, CK NEXT 64 CHARS FROM FIFO.
 CMP #2,RECEOT ;EOT RECVD YET?
 BNE -6 ;NO, WAIT FOR IT.
 BR FND1C ;GO CHECK THE DATA FROM FIFO.

;IF THE 'FOUNSW' IS SET, THEN WE ARE EXECUTING
 ;THE 'ADDRESS 17' SUBTEST AND WE SHOULD SKIP OVER THE
 ;FOLLOWING SECTION. SET MEANS--1.

3381
 3382
 3383
 3384

```
3385
3386
3387 014212 005737 013662      FOUND2: TST      FOUNSW      ;LOOK AT THE SOFTWARE SWITCH.
3388 014216 003035              BGT      FOUN5      ;IF SWITCH=+1, THEN
3389                                BMI      FOUN3      ;WE ARE IN SUBTEST 3.
3390 014220 100423              ;SW=-1 MEANS WE HAVE
3391                                ;JUST FINISHED SUBTEST 2.
3392                                ;ELSE FALL THROUGH TO SUBTEST 2.
3393                                ;(SWITCH 0).
3394 014222 005737 032132      TST      SIOSWH
3395 014226 001067              BNE      FOUN6
3396 014230 000240              NOP
3397 014232 000240              NOP
```

```
3398
3399
3400
3401
3402
3403
3404 014234 104001 000002          SCOPE,2          ;****SUBTEST 2
3405
3406 014240 104012          PRINT
3407 014242 024316          MES10
3408 014244 104013          TTYIN
3409 014246 112700 000077          MOVB #77,R0
3410 014252 004737 014660          JSR PC,FSTUF
3411
3412
3413 014256 012737 177777 013662          MOV #-1,FOUNSW
3414 014264 000137 014014          JMP FOUNDL
3415
3416
3417
3418
```

;THIS SUBTEST USES ADDRESS '17' AND XMITS A RANDOM BUFFER.

;NOTE: FOUNSW=-1 HERE.
;TEXT RESET MODULE ADDRESS TO '17'.
;WAIT FOR CR
;REPLACE THE FOUNDATION ADDRESS WITH 17.
;SET THE SWITCH SO THAT WE WON'T ENTER THIS AGAIN.
;-1 MEANS WE ARE IN THIS TEST.
;SEND 2 CHARACTERS AND
;CHECK TO MAKE SURE THAT ADDRESS 17
;WILL RETURN THEM.

```
3419 :*****
3420 :THIS SUBTEST USES THE WRONG ADDRESSES AND CHECK TO MAKE
3421 :SURE THAT THE MODULE IS NOT ENABLED.
3422 :*****
3423
3424 014270 005737 032132 FOUND3: TST SIOSWH ;SKIP THIS SUBTEST IF WE ARE USING SERIAL I/O.
3425 014274 001402 BEQ FND3A ;SIO NOT IN USE.
3426 014276 000137 013670 JMP FLO ;ELSE LOOP TO BEGINNING OF MODULE TEST.
3427
3428 ;ADDRESS THE MODULE WITH ADDRESSES 0-16.
3429 ;ASSUME PRESENT ADDRESS SELECTED TO BE 17.
3430
3431
3432 014302 104001 000003 FND3A: SCOPE,3 ;****SUBTEST 3
3433 014306 004737 005154 JSR PC,ADRSIT ;MULTIPLE
3434 ;ADDRESS TEST.
3435 ;(DESTINATION)
3436
3437
```

```
3438
3439
3440
3441
3442 014312 104001 000005
3443 014316 104012
3444 014320 027634
3445
3446
3447
3448 014322 027670
3449 014324 104013
3450
3451 014326 012737 014406 014404
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461 014334 005737 032132
3462 014340 001013
3463
3464
3465
3466
3467 014342 104007
3468 014344 014350
3469 014346 000402
3470 014350 022
3471 014351 077
3472 014352 023
3473 014353 004
3474
3475
3476 014354 104007
3477 014356 014362
3478 014360 000770
3479 014362 021
3480 014363 077
3481 014364 001
3482 014365 023
3483 014366 000
3484 014370
3485
3486
3487 014370 104007
3488
3489
3490 014372 014030
3491 014374 104006
3492 014376 000004
3493 014400 000137 014370

*****
ROUTINE TO CHECK CUSTOMER DEFINED
MODE FLIP FLOP (SUB-PROGRAM).
*****
FOUND5: SCOPE,5          ;****SUBTEST 5
        PRINT
        MES78            ;TEXT
                        ;PUT SCOPE PROBE
                        ;ON PIN 78(CR).
        MES79            ;'USE ^E TO EXIT
        TTYIN            ;WAIT FOR CR
:
        MOV      #FOUND6,EVECTOR
:OUTPUT THE FOLLOWING PROGRAM.
:THIS PROGRAM WILL LOOP ENDLESSLY
:UNTIL A '^E' IS INPUTTED VIA TTY.
:THE APPROXIMATE SIGNAL TO BE SCOPED WILL
:BE 1 MILLISEC @ 9600 BAUD.
:
:SW 14-SET TO SCOPE LOOP.
:SW 11 =SET TO ITERATE.
:
FND5:   TST      SIOSWH      ;USING THE SERIAL I/O?
        BNE      FND5B      ;YES, SO USE PADDED PROGRAM.
                        ;ELSE, USE THE FOLLOWING:

:LOAD THIS PROGRAM IF MODULE TEST
FND5A:  LDPMG0          ;LOAD THE PROGRAM
        .+4
        BR       FND5C      ;GO HERE WHEN DONE
        .BYTE    DC2
FLAB6:  .BYTE    77          ;FOUNDATION AS DESTIN.
        .BYTE    DC3
        .BYTE    EOT        ;SEND THE EOT

FND5C:  LDPMG0
        .+4
        BR       FND5A
        .BYTE    DC1
FLAB4:  .BYTE    77
        .BYTE    SOH
        .BYTE    DC3
        .BYTE    0
        .EVEN

:USE THIS PROGRAM IF SYSTEM TEST
FND5B:  LDPMG0          ;LOAD THE FOLLOWING PROGRAM.
                        ;(SERIAL I/O IN USE).
        FPROG
        LDCHRO          ;SEND AN EOT
        EOT
        JMP      FND5B
```


CZPMAC0 PDM70 DIAGNOSTIC TEST MACY11 30A(1052) 20-JAN-78 09:11 PAGE 80
CZPMAC.P11 19-JAN-78 14:50 M7378A FOUNDATION MODULE TEST

SEQ 0101

3494
3495 014404 000000 EVECTOR: .WORD 0 ;ADDRESS TO GET ME OUT OF INFINITE LOOPS.

3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507

014406 104001 000006
014412 104012
014414 024236
014416 000137 013670

```
*****  
:TEST COMPLETE  
:*****  
FOUND6: SCOPE,6  
PRINT  
MES7 ;TEXT 'TEST COMPLETE'  
JMP FLO ;LOOP THE TEST.  
  
.SBTTL SUBROUTINES
```


3508
 3509
 3510
 3511
 3512
 3513
 3514
 3515
 3516
 3517
 3518
 3519
 3520
 3521
 3522
 3523
 3524
 3525
 3526
 3527
 3528
 3529
 3530
 3531
 3532
 3533
 3534
 3535
 3536
 3537
 3538
 3539
 3540
 3541
 3542
 3543
 3544
 3545
 3546
 3547
 3548
 3549
 3550
 3551
 3552
 3553
 3554
 3555
 3556
 3557
 3558
 3559
 3560
 3561
 3562
 3563

014422 104007
 014424 014430
 014426 000420
 014430 101
 014431 130
 014432 004
 014433 000

 014434 104007
 014436 014442
 014440 000413
 014442 002
 014443 021
 014444 060
 014445 001
 014446 061
 014447 022
 014450 060
 014451 023
 014452 021
 014453 060
 014454 022
 014455 060
 014456 023
 014457 021
 014460 060
 014461 001
 014462 060
 014463 000
 014464 000
 014465 000
 014466 000
 014470
 014470 000207

```

*****
SUBROUTINE SENDAX
*****
SUBROUTINE TO LOAD AND SEND THE CHARACTERS
'A' AND 'X'
:
SENDAX: LDPGMO
      .+4
      BR      SNDAX1      ;GO HERE WHEN DONE.
BYTEA: .BYTE  'A'
BYTEX: .BYTE  'X'
      .BYTE  EOT
      .BYTE  0
      .EVEN

:*****SUBROUTINE SENDPG*****
SUBROUTINE TO SEND A PROGRAM.
:(USED FOR DEBUGGING PURPOSES.)

SENDPG: LDPGMO
      .+4
      BR      SNDAX1
      .BYTE  STX
      .BYTE  DC1
      .BYTE  60
      .BYTE  SOH
      .BYTE  61
      .BYTE  DC2
      .BYTE  60
      .BYTE  DC3
      .BYTE  DC1
      .BYTE  60
      .BYTE  DC2
      .BYTE  60
      .BYTE  DC3
      .BYTE  DC1
      .BYTE  60
      .BYTE  SOH
      .BYTE  60
      .BYTE  0
      .BYTE  0
      .BYTE  0
      .BYTE  0
      .EVEN
SNDAX1: RTS      PC      ;RETURN

:*****SUBROUTINE CNTLOP*****
SUBROUTINE TO PROVIDE AN 'X' SECOND WAIT.
:ENTERS WITH COUNT EQUAL TO THE COMPLEMENT OF THE NUMBER
:OF SECONDS DESIRED TO WAIT.

```

```

3564
3565 014472 104023          CNTLOP: NULL1          ;DELAY ONE SECOND.
3566 014474 005237 032160      INC      COUNT          ;UP THE DELAY COUNTER.
3567 014500 001374          BNE      CNTLOP      ;CONTINUE LOOPING UNTIL COUNTER IS ZERO.
3568 014502 000207          RTS      PC          ;RETURN WHEN DONE.
3569
3570
3571      ;*****
3572      ;ROUTINE TO ADDRESS A MODULE USING ALL OF THE WRONG ADDRESSES
3573      ;AND CHECK TO MAKE SURE THAT DATA ISN'T RETURNED.
3574      ;*****
3575
3576
3577      ;THIS ROUTINE IS DESIGNED FOR THE FOUNDATION MODULE
3578      ;BUT WILL WORK FOR OTHER MODULES.
3579 014504 112737 000060 017277 MATD:  MOVB      #60,DSTADR      ;SET UP 1ST ADDRESS
3580                                ;TO BE TESTED.
3581 014512 113700 017277      ADSLOP: MOVB      DSTADR,R0
3582 014516 004737 014660      JSR      PC,FSTUF          ;STUFF MODULE ADDRESS.
3583 014522 005027 016236      CLR      #RECBF0+2        ;CLEAR 1ST LOC.
3584 014526 123737 032134 017277 CMPB      MODADR,DSTADR      ;EQUAL TO SELECTED ADDRESS?
3585 014534 001434          BEQ      ADSNXT          ;YES, SELECT NEXT ADDR.
3586 014536 005737 032132      TST      SIOSWH          ;SERIAL I/O IN USE?
3587 014542 001403          BEQ      ADSLP1          ;NOPE
3588
3589 014544 104007          LDPGM0
3590 014546 014030          FPROG          ;USE PADDED SERIAL PROGRAM.
3591
3592 014550 000410          BR      ADSLP2
3593 014552 104007      ADSLP1: LDPGM0
3594 014554 014112          FLAB5A
3595      ;ADDRESS THE FOUNDATION MODULE AS A SOURCE (NON-SERIAL I/O).
3596 014556 104007          LDPGM0
3597 014560 014564          .+4
3598 014562 000403          BR      ADSLP2
3599 014564          021          .BYTE      DC1
3600 014565          060          FLAB17: .BYTE      60
3601 014566          001          .BYTE      SOH
3602 014567          060          .BYTE      60
3603 014570          023          .BYTE      DC3
3604          014572          .EVEN
3605
3606 014572 004737 014422      ADSLP2: JSR      PC,SENDAX      ;SEND 2 CHARS.
3607 014576 104005          RECVR0          ;ENABLE DL-11 RCVR.
3608 014600 104004          DELAY
3609
3610
3611      ;CHECK THE DATA TO SEE IF IT IS A,B,EOT.
3612      ;SINCE OTHER MODULES MAY INDEED BE IN THE SYSTEM, OTHER
3613      ;THAN THE FOUNDATION MOD, THEY COULD POSSIBLY XMIT DATA WHEN
3614      ;ADDRESSED.
3615
3616 014602 123722 000101          CMPB      'A',(R2)+          ;WAS AN 'A' RETURNED?
3617 014606 001007          BNE      ADSNXT          ;NOPE.
3618 014610 123722 000102          CMPB      'B',(R2)+          ;B?
3619 014614 001004          BNE      ADSNXT          ;NOT A B.

```

```
3620 014616 122722 000004      CMPB    #EOT,(R2)+      ;EOT?
3621 014622 001001              BNE      ADSNXT
3622
3623      ;ONLY THE STRING A,B,EOT CAN MAKE IT TO HERE.
3624
3625 014624 000407              BR       ADSER1
3626
3627
3628
3629 014626 105237 017277      ADSNXT: INCB    DSTADR      ;UPDATE MODULE ADDRESS.
3630 014632 122737 000077 017277      CMPB    #77,DSTADR  ;DONE?
3631 014640 001324              BNE      ADSLOP      ;NO.
3632 014642 000207              RTS      PC          ;YES.
3633 014644 113737 017277 031166      ADSER1: MOVB   DSTADR,ERR13A
3634 014652 104022              MODERR
3635 014654 031124              ERR13
3636
3637 014656 000763              BR       ADSNXT      ;MODULE ENABLED
                                           ;WITH ILLEGAL
                                           ;ADDRESS.
```

3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654

:ROUTINE TO STUFF THE ADDRESS IN R0 INTO THE PADDED SERIAL
:I/O PROGRAM AND UN-PADDED PROGRAM.
:*****

FSTUF: MOVB R0,FLAB1
 MOVB R0,FLAB2
 MOVB R0,FLAB3
 MOVB R0,FLAB4
 MOVB R0,FLAB5
 MOVB R0,FLAB6
 MOVB R0,FLAB7
 RTS PC ;RETURN.

```
3655
3656
3657      ;*****
3658      ;KEYBOARD SERVICE ROUTINE.  CHARACTERS ARE ACCEPTED FROM THE KEYBOARD,
3659      ;TESTED FOR DIFFERENT FUNCTIONS AND SAVED IN A BUFFER.
3660      ;*****
3661      014716 104002      XTTYIN: SAVREG      ;SAVE REGISTERS
3662      014720 005037 032136      CLR      REPTSW      ;C_L_R SOFTWARE SW.
3663      014724 005037 032214      CLR      CHRCNT      ;CHARACTER COUNTER
3664      014730 005037 032216      CLR      RUBSWH      ;RUBOUT SW.
3665      014734 012704 015330      MOV      #INBUF,R4      ;SET UP BUFFER POINTER
3666      014740 105777 164376      INPUTA: TSTB      @TKS      ;CHARACTER READY?
3667      014744 100375      BPL      INPUTA      ;NO, WAIT
3668      014746 117701 164372      MOVB      @TKB,R1      ;YES, SAVE IT
3669      014752 142701 000200      BICB      #200,R1      ;STRIPE OFF PARITY BIT
3670      014756 105701      TSTB      R1      ;WAS 'HERE IS' TYPED?
3671      014760 001757      BEQ      XTTYIN+2      ;YES, IGNORE IT
3672      014762 005737 032150      TST      SENDSW      ;INTERRUPTED FROM SEND ROUTINE
3673      014766 001407      SEQ      INPUTC      ;NO
3674      014770 005737 032122      TST      PRTSWH      ;INTERRUPT FROM PRINT?
3675      014774 001066      BNE      EXTITY      ;YES, IGNORE IT
3676      014776 110114      MOVB      R1,(R4)      ;NO, SAVE CHAR.
3677      015000 062716 000002      ADD      #2,(SP)      ;YES, RETURN CALL +4
3678      015004 000462      BR      EXTITY      ;EXIT
3679      015006 120127 000060      INPUTC: CMPB      R1,#60      ;SPECIAL CHARACTER
3680      015012 100426      BMI      SPCHR1      ;YES, TEST IT
3681      015014 122701 000132      CMPB      #132,R1      ;SPECIAL CHARACTER
3682      015020 100423      BMI      SPCHR1      ;YES, TEST IT
3683      015022 005737 032122      TST      PRTSWH      ;INTERRUPTED FROM PRINT ROUTINE?
3684      015026 001051      BNE      EXTITY      ;YES, IGNORE IT
3685      015030 005737 032216      INPUTB: TST      RUBSWH      ;RUBOUT SW. SET?
3686      015034 001404      BEQ      .+12      ;NO, NORMAL ECHO.
3687      015036 005037 032216      CLR      RUBSWH      ;YES, CLR IT.
3688      015042 104012      PRINT      SLASH      ;PRINT '\' TO TERMINATE RUBOUT MODE
3689      015044 032026      MOVB      R1,(R4)+      ;SAVE CHARACTER
3690      015046 110124      INC      CHRCNT
3691      015050 005237 032214      CMP      #66,CHRCNT      ;BUFFER FULL?
3692      015054 022737 000102 032214      BMI      TYPEQM      ;YES, TYPE '?'
3693      015062 100516      ECHO: TYPEIT      ;NO, ECHO CHAR.
3694      015064 104010      BR      INPUTA      ;WAIT FOR NEXT CHAR.
3695      015066 000724
```

```
3696
3697
3698
3699 015070 005737 032122      SPCHR1: TST      PRTSWH      ;INTERRUPTED FROM PRINT ROUTINE?
3700 015074 001036              BNE      CNTRLG      ;YES, CHECK FOR '^G'
3701 015076 122701 000177      CMPB     #177,R1      ;CHAR. = RUBOUT?
3702 015102 001016              BNE      SPCHR3      ;NO
3703 015104 005737 032214      TST      CHRCNT      ;YES, IS IT VALID?
3704 015110 001713              BEQ      INPUTA      ;NO, IGNORE IT
3705 015112 005337 032214      DEC      CHRCNT      ;YES, DECREMENT COUNTER
3706 015116 005737 032216      TST      RUBSWH      ;IN 'RUBOUT' MODE?
3707 015122 001002              BNE      .+6         ;YES, JUST ECHO BACK CHAR.
3708 015124 104012              PRINT
3709 015126 032026              SLASH
3710 015130 114401              MOVB     -(R4),R1      ;PRINT '\' TO INDICATE RUBOUT
3711 015132 005237 032216      INC      RUBSWH      ;GET LAST CHAR.
3712 015136 000752              BR       ECHO        ;SET 'RUBOUT' MODE
3713 015140 122701 000015      SPCHR3: CMPB     #15,R1      ;CHAR. = 'CR' .
3714 015144 001004              BNE      SPCHR5      ;NO
3715 015146 104012              PRINT
3716 015150 032032              CRLF
3717
3718 015152 104003              EXT'TY: GETREG      ;RESTORE REGISTERS
3719 015154 000002              RTI
3720 015156 122701 000040      SPCHR5: CMPB     #40,R1      ;EXIT
3721 015162 001740              BEQ      ECHO        ;CHAR. = SPACE?
3722 015164 122701 000054      CMPB     #54,R1      ;YES, ECHO BUT DON'T SAVE IT
3723 015170 001717              BEQ      INPUTB      ;CHAR = 'COMMA'?
3724 015172 104000              CNTRLG: PRCNTR      ;YES, SAVE IT
3725 015174 122701 000007      CMPB     #7,      R1      ;CONTROL-G?
3726 015200 001003              BNE      CNTRLC      ;NO
3727 015202 004737 023304      JSR      PC,      UPDAT1 ;CHECK FOR SOFTWARE SWR
3728 015206 000761              BR       EXT'TY      ;EXIT
3729 015210 122701 000003      CNTRLC: CMPB     #3,R1      ;CHAR. = '^C'
3730 015214 001002              BNE      CNTRLA      ;NO CHECK FOR '^A'
3731 015216 000137 001376      JMP      MONITR      ;RETURN TO MONITOR
3732 015222 122701 000001      CNTRLA: CMPB     #1,R1      ;CHAR. = '^A' ?
3733 015226 001004              BNE      CNTRLR      ;NO, CHECK FOR '^R'
3734 015230 012706 001000      MOV      #1000,SP      ;RESET STACK POINTER
3735 015234 000177 014666      JMP      @AVECTR      ;GO TO THE RESTART ADDRESS
3736 015240 122701 000022      CNTRLR: CMPB     #22,R1      ;CHAR. = '^R'
3737 015244 001006              BNE      CNTRLE      ;NO, TEST FOR '^E'
3738 015246 104012              PRINT
3739 015250 032032              CRLF
3740 015252 012706 001000      MOV      #1000,SP      ;RESET STACK POINTER
3741 015256 000177 014642      JMP      @RVECTR      ;GO TO RESTART ADDRESS
3742 015262 122701 000005      CNTRLE: CMPB     #5,R1      ;CHAR. = '^E'?
3743 015266 001003              BNE      CNTRLO      ;NO, TEST FOR '^O'
3744 015270 104005              RECVRO
3745 015272 000177 177106      JMP      @EVECTOR      ;CLEAR OUT THE BUFFER.
3746 015276 005737 032122      CNTRLO: TST      PRTSWH      ;CONTINUE ON TO NEXT SUBTEST.
3747 015302 001406              BEQ      TYPEQM      ;INTERRUPTED IN FROM PRINT ROUTINE?
3748 015304 122701 000017      CMPB     #17,R1      ;NO, ILLEGAL ENTRY
3749 015310 001320              BNE      EXT'TY      ;CHAR. = '^O'?
3750 015312 005137 032152      COM      OPRTSW      ;NO, IGNORE IT
3751 015316 000715              BR       EXT'TY      ;YES, SET/RESET PRINT INHIBIT SW.
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
```


3752 015320 104012
3753 015322 031702
3754 015324 000137 014720
3755 015330 000000
3756 015434
3757

TYPEQM: PRINT
QMARK
JMP XTTYIN+2
INBUF: 0
.=.+66.

```
3758 ;SUBROUTINE TO CHECK FOR AND PRINT PDP-70 CONTROL CHAR.'S
3759 015434 122701 000021 PDMSET: CMPB #DC1,R1 ;YES, CHAR = 21?
3760 015440 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3761 015444 032037 MESDC1 ;TEXT 'DC1'
3762 015446 122701 000022 CMPB #DC2,R1 ;CHAR = 22?
3763 015452 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3764 015456 032044 MESDC2 ;TEXT 'DC2'
3765 015460 122701 000023 CMPB #DC3,R1 ;CHAR. = 23?
3766 015464 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3767 015470 032051 MESDC3 ;TEXT 'DC3'
3768 015472 122701 000024 CMPB #DC4,R1 ;CHAR. = 24?
3769 015476 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3770 015502 032056 MESDC4
3771 015504 122701 000002 CMPB #STX,R1
3772 015510 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3773 015514 032063 MESSTX
3774 015516 122701 000026 CMPB #SYN,R1
3775 015522 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3776 015526 032070 MESSYN
3777 015530 122701 000001 CMPB #SOH,R1
3778 015534 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3779 015540 032075 MESSOH
3780 015542 122701 000017 CMPB #SI,R1
3781 015546 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3782 015552 032102 MESSI
3783 015554 122701 000004 CMPB #EOT,R1
3784 015560 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3785 015564 032106 MESEOT
3786 015566 122701 000003 CMPB #ETX,R1
3787 015572 004737 015650 JSR PC,PDMPT ;PRINT PDM CNTRL CHAR.
3788 015576 032113 MESETX
3789 015600 132701 000140 BITB #140,R1 ;IS CHAR. PRINTABLE?
3790 015604 001417 BEQ PDMST1 ;NO, PRINT AS CONTROL CHAR.
3791 015606 104010 TYPEIT ;YES, TYPE IT
3792 015610 005737 032150 TST SENDSW
3793 015614 001006 BNE PDMST0
3794 015616 005237 032142 INC FORMT1
3795 015622 023727 032142 000110 CMP FORMT1,#72.
3796 015630 002406 BLT PDMST2
3797 015632 104012 PDMST0: PRINT
3798 015634 032032 CRLF
3799 015636 005037 032142 CLR FORMT1
3800 015642 000401 BR .+4
3801 015644 104000 PDMST1: PRCNTR ;PRINT AS CONTROL CHAR.
3802 015646 000207 PDMST2: RTS PC
3803 015650 001011 PDMPT: BNE PEXT2 ;CHAR. MATCH?
3804 015652 017637 000000 015662 MOV @ (SP),XPDMES ;YES, GET ADDRESS OF MESSAGE
3805 015660 104012 PRINT
3806 015662 000000 XPDMES: 0
3807 015664 005037 032142 CLR FORMT1 ;RE-SET 'CR/LF' FORMAT SW.
3808 015670 005726 POP1SP ;CLEAN UP STACK
3809 015672 000207 RTS PC ;EXIT
3810 015674 062716 000002 PEXT2: ADD #2,(SP) ;CHECK NEXT WORD
3811 015700 000207 RTS PC
```



```

3812
3813
3814
3815
3816
3817
3818 015702 104002
3819 015704 005037 016116
3820 015710 005037 016114
3821 015714 005037 016124
3822 015720 005037 016120
3823 015724 010137 016110
3824 015730 012204
3825 015732 005737 016124
3826 015736 001403
3827 015740 020437 016124
3828 015744 003402
3829 015746 010437 016124
3830 015752 005737 016120
3831 015756 001403
3832 015760 020437 016120
3833 015764 003002
3834 015766 010437 016120
3835 015772 060437 016114
3836 015776 005537 016116
3837 016002 005301
3838 016004 001351
3839 016006 004737 016016
3840 016012 104003
3841 016014 000002

;*****
;COMPUTE THE RESULT OF 'X' CONVERSIONS AS HIGH,LOW AND AVERAGE
;THE ROUTINE IS ENTERED WITH THE NUMBER OF CONVERSIONS TO BE TAKEN IN 'R1'
;AND WITH 'R2' CONTAINING THE ADDRESS OF THE DATA TO BE AVERAGED.
;*****

XAVRAGE: SAVREG
          CLR    HIDIVD
          CLR    LODIVD
          CLR    HIGH
          CLR    LOW
          MOV    R1,LODIVR
GETDAT:   MOV    (R2)+,R4
          TST    HIGH
          BEQ    .+10
          CMP    R4,HIGH
          BLE    TSTLO
          MOV    R4,HIGH
          TST    LOW
          BEQ    .+10
          CMP    R4,LOW
          BGT    .+6
          MOV    R4,LOW
          ADD    R4,LODIVD
          ADC    HIDIVD
          DEC    R1
          BNE    GETDAT
AVGDAT:   JSR    PC,DIVIDE
          GETREG
          RTI

;SAVE REGISTERS
;CLR HI-ORDER DIVIDEND
;CLR LO-ORDER DIVIDEND
;HIGH
; & LOW
;SET UP DIVISOR FOR DIVIDE
;GET VALUE

;IS NEW NO. GREATER THAN OLD NO.?
;NO, TEST IF LESS THAN
;YES, SAVE NEW HIGH

;NEW NO LESS THAN OLD NO.?
;NO
;YES, SAVE NEW LOW
;ADD VALUE TO LOW-ORDER DIVIDEND
;ADD CARRY TO HI-ORDER DIVIDEND
;DONE?
;NO
;PERFORM DIVIDE
;YES, RESTORE REG.'S
;EXIT

```

```
3842
3843
3844
3845
3846
3847
3848 016016 104002
3849 016020 013701 016110
3850 016024 013702 016112
3851 016030 013703 016114
3852 016034 013704 016116
3853 016040 005005
3854 016042 160103
3855 016044 005604
3856 016046 160204
3857 016050 005704
3858 016052 100402
3859 016054 005205
3860 016056 000771
3861 016060 060103
3862 016062 010337 016126
3863 016066 006201
3864 016070 001403
3865 016072 020103
3866 016074 101001
3867 016076 005205
3868 016100 010537 016122
3869 016104 104003
3870 016106 000207
3871 016110 000000
3872 016112 000000
3873 016114 000000
3874 016116 000000
3875 016120 000000
3876 016122 000000
3877 016124 000000
3878 016126 000000
3879

;*****
;DOUBLE PERCISION DIVIDE SUBROUTINE
;THIS ROUTINE IS ENTERED THIS WITH THE DIVISOR AND DIVIDENT PRE-LOADER
;INTO THE ROUTINE.
;*****

DIVIDE: SAVREG                ;SAVE REG.'S
      MOV LODIVR,R1          ;GET LOW ORDER DIVISOR
      MOV HIDIVR,R2          ;GET HIGH ORDER DIVISOR
      MOV LODIVD,R3          ;GET LOW ORDER DIVIDEND
      MOV HIDIVD,R4          ;GET HIGH ORDER DIVIDEND
      CLR R5                 ;USE 'R5' TO STORE QUOTIENT
DIVDIT: SUB R1,R3             ;SUBTRACT L-O DIVISOR FROM DIVIDEND
      SBC R4                 ;SUB CARRY FROM HI-ORDER DIVIDEND
      SUB R2,R4              ;SUBTRACT HI-ORDER DIVISOR
      TST R4                 ;SUBTRACTION SUCCESSFUL?
      BMI .+6               ;NO, EXIT
      INC R5                 ;YES, INCREMENT QUOTIENT
      BR DIVDIT             ;PERFORM NEXT SUBTRACTION
      ADD R1,R3             ;ADD BACK OVERFLOW
      MOV R3,REMAIN         ;SAVE AS REMAINDER
      ASR R1
      BEQ .+10
      CMP R1,R3             ;IS REMAINED > THAN HALF DIVISOR?
      BHI .+4               ;NO
      INC R5                 ;YES, ADD '1' TO QUOIENT
      MOV R5,QUOENT         ;SAVE QUOIENT
      GETREG                ;RESTORE REGISTER
      RTS PC                ;EXIT

LODIVR: 0
HIDIVR: 0
LODIVD: 0
HIDIVD: 0
LOW: 0
QUOENT: 0
HIGH: 0
REMAIN: 0
```

```
3880
3881
3882
3883
3884
3885
3886
3887 016130 012700 016234
3888 016134 010037 016232
3889 016140 005020
3890 016142 022700 016274
3891 016146 001374
3892 016150 005037 016220
3893 016154 005037 016224
3894 016160 005037 016234
3895 016164 005037 016222
3896 016170 005037 016226
3897 016174 005037 016230
3898 016200 005777 163154
3899 016204 052777 000100 163144
3900 016212 012702 016234
3901 016216 000002
3902 016220 000000
3903 016222 000000
3904 016224 000000
3905 016226 000000
3906 016230 000000
3907 016232 016234
3908 016234 000000
3909 016736 016736
3910 016736 000000
3911
```

```
*****
;DL11 RECEIVER INITIALIZATION ROUTINE.
;THIS ROUTINE SETS UP A RECE VER BUFFER WHERE DATA IS STORED AS IT COMES
;IN FROM THE DL11 RECEIVER.
*****
XRECRO: MOV    #RECBF0,R0
        MOV    R0,RECVPT
        CLR    (R0)+      ;CLR 1ST '20' LOCATIONS OF BUFFER
        CMP    #RECBF0+40,R0
        BNE    .-6
        CLR    PARITY
        CLR    RECEOT
        CLR    RECBF0
        CLR    RECD3
        CLR    RECSTX
        CLR    RECETX
        TST    @RBUF0      ;CLR RECV. FLAGS
        BIS    #100,@RCSR0 ;ENABLE THE INTERRUPT
        MOV    #RECBF0,R2  ;SET UP BUFFER POINTER
        RTI

PARITY: 0
RECD3: 0
RECEOT: 0
RECSTX: 0
RECETX: 0
RECVPT: RECBF0
RECBF0: 0
        .=-+500
RECEOT: 0
```

```
3912
3913
3914
3915
3916
3917
3918
3919 016740 010146
3920 016742 010246
3921 016744 013701 016232
3922 016750 017702 162404
3923 016754 110221
3924 016756 105011
3925 016760 012737 024070 017136
3926 016766 020127 016736
3927 016772 003054
3928 016774 005702
3929 016776 100013
3930 017000 012737 024161 017136
3931 017006 032702 040000
3932 017012 001044
3933 017014 032702 010000
3934 017020 001402
3935 017022 005237 016220
3936 017026 122702 000004
3937 017032 001003
3938 017034 005237 016224
3939 017040 000424
3940 017042 005737 032132
3941 017046 001021
3942 017050 122702 000023
3943 017054 001003
3944 017056 005237 016222
3945 017062 000413
3946 017064 122702 000002
3947 017070 001003
3948 017072 005237 016226
3949 017076 000405
3950 017100 122702 000003
3951 017104 001002
3952 017106 005237 016230
3953 017112 010137 016232
3954 017116 012602
3955 017120 012601
3956 017122 000002
3957 017124 005077 162226
3958 017130 005037 032150
3959 017134 104012
3960 017136 024070
3961 017140 000137 001376

;*****
;SBTTL DL11 RECEIVER SUBROUTINE.
;ROUTINE IS ENTERED ON DL11 RECEIVER INTERRUPTS WHERE THE CHARACTER IS
;READ & SAVED IN A BUFFER.
;*****

RECVER: MOV R1,-(SP) ;SAVE REG'S 'R1&R2' ON STACK
        MOV R2,-(SP)
        MOV RECVPT,R1 ;SET UP BUFFER POINTER
        MOV @RBUF0,R2 ;READ & SAVE CHAR.
        MOVB R2,(R1)+ ;SAVE CHAR. IN BUFFER
        CLRB (R1) ;TERMINATE BUFFER W/ NULL CHAR.
        MOV #MES2,ERRMES ;NO, SET UP 1ST ERROR MESSAGE
        CMP R1,#RECEDE ;RECEIVER BUFFER FULL?
        BGT RECERR ;YES PRINT BUFFER FULL MESSAGE
        TST R2 ;WAS RECVR. ERROR DETECTED?
        BPL RECVR1 ;NO
        MOV #MES4,ERRMES ;SETUP 2ND ERROR MESSAGE
        BIT #40000,R2 ;OVERRUN FLAG SET?
        BNE RECERR ;YES, PRINT OVERRUN ERROR MESSAGE
        BIT #10000,R2 ;PARITY BIT SET?
        BEQ .+6 ;NO, OK
        INC PARITY ;YES, SET PARITY ERROR FLAG
RECVR1: CMPB #EOT,R2 ;CHAR. -EOT?
        BNE .+10 ;NO
        INC RECEOT
        BR RECEXT
        TST SIOSWH ;USING SERIAL INPUT OPTION?
        BNE RECEXT ;YES, EXIT
        CMPB #DC3,R2 ;CHAR. =DC3?
        BNE .+10 ;NO
        INC RECD3 ;YES, SET FLAG
        BR RECEXT
        CMPB #STX,R2 ;CHAR. = STX?
        BNE .+10 ;NO
        INC RECSTX ;YES, SET FLAG.
        BR RECEXT
        CMPB #ETX,R2 ;CHAR. = ETX?
        BNE .+6 ;NO
        INC RECETX ;YES, SET FLAG
RECEXT: MOV R1,RECVPT
        MOV (SP)+,R2
        MOV (SP)+,R1
        RTI
RECERR: CLR @RCSR0 ;DISABLE FURTHER INTERRUPTS
        CLR SINDSW
ERRMES: MES2
        JMP MONITR ;MODIFIED DEPENDING ON TYPE OF ERROR
;RETURN TO MONITOR ON RECVR. ERRORS
```

```
3962
3963
3964
3965
3966
3967
3968
3969 017144 005737 032132
3970 017150 001463
3971 017152 104006
3972 017154 000002
3973 017156 005237 032154
3974 017162 000456
3975
3976
3977
3978
3979
3980
3981
3982 017164 005737 032132
3983 017170 001453
3984 017172 104007
3985 017174 017200
3986 017176 000403
3987 017200 002
3988 017201 021
3989 017202 075
3990 017203 001
3991 017204 061
3992 017205 000
3993
3994 017206 005237 032154
3995 017212 005237 032146
3996 017216 000440

;*****
;IF THE CONTROL MODULE IS BEING USED THIS ROUTINE PADS THE DATA
;BEING TRANSMITTED SO THAT THE DESTINATION PORTION OF THE
;SERIAL I/O MODULE GETS ADDRESSED.
;*****
XSOURC: TST      SIOSWH      ;SERIAL I/O INPUT?
        BEQ      XLDADD      ;NO, NORMAL LOAD
        LDCHRO
        STX
        INC      TERMSW
        BR       XLDADD

;*****
;IF THE CONTROL MODULE IS BEING USED THIS ROUTINE PADS THE DATA
;BEING TRANSMITTED SO THAT THE SOURCE PORTION OF THE SERIAL I/O
;MODULE GETS ADDRESSED.
;*****
XDSTIN: TST      SIOSWH      ;SERIAL I/O INPUT?
        BEQ      XLDADD      ;NO, NORMAL LOAD
        LDPGMO
        .+4
        BR       XDSTG1
        .BYTE    STX
        .BYTE    DC1
        .BYTE    75
        .BYTE    SOH
        .BYTE    61
        .BYTE    0
        IADRS9:
        XDSTG1: INC      TERMSW
        INC      DSTSWH
        BR       XLDADD
```

```
3997
3998
3999          ;*****
4000          ;SUBROUTINE TO ADDRESS ANY SOURCE MODULE
4001          ;*****
4002
4003 017220 104025  ADRSRC: SOURCE          ;ADDRESS AS SOURCE
4004 017222 017226      .+4
4005 017224 000207      RTS      PC
4006 017226      021      .BYTE  DC1          ;ALERT MODULE
4007 017227      060      SRCADR: .BYTE  60      ;ADDRESS MODIFIED BY USER
4008 017230      001      .BYTE  SOH
4009 017231      060      SOH1:  .BYTE  60      ;ADDRESS MODIFIED BY ME
4010 017232      023      .BYTE  DC3
4011 017233      000      .BYTE  0
4012      .EVEN
4013
4014          ;*****
4015          ;SUBROUTINE TO ADDRESS ANY DESTINATION MODULE
4016          ;*****
4017
4018 017234 005737 032132  ADRDST: TST      SIOSWH          ;USING SERIAL I/O?
4019 017240 001004      BNE      .+12          ;NO,
4020 017242 122737 000023 017300  CMPB     #DC3,DSTADR+1 ;YES, USING 'DC3'?
4021 017250 001404      BEQ      .+12          ;YES, LOAD 'SI'
4022 017252 112737 000023 017300  MOVB     #DC3,DSTADR+1 ;NO, LOAD DC3
4023 017260 000403      BR       .+10
4024 017262 112737 000017 017300  MOVB     #SI,DSTADR+1
4025 017270 104024      DESTIN          ;ADDRESS DESTINATION
4026 017272 017276      .+4
4027 017274 000207      RTS      PC
4028 017276      022      .BYTE  DC2          ;ALERT MODULE
4029 017277      060      DSTADR: .BYTE  60      ;ADDRESS MODIFIED BY USER
4030 017300      023      .BYTE  DC3
4031 017301      000      .BYTE  0
4032      .EVEN
4033
4034          ;*****
4035          ;SUBROUTINE TO TRANSMIT A SINGLE CHARACTER VIA THE DL11.
4036          ;*****
4037
4038 017302 005237 017664  XLDCHR: INC      SNGCHR          ;SET SOFTWARE FLAG
4039 017306 011637 017666      MOV      (SP),TRANPT      ;SET UP ADDRESS OF CHAR. TO BE TRANSMITTED
4040 017312 062716 000002  XLD1:  ADD      #2,(SP)          ;SET UP STACK TO EXIT
4041 017316 000404      BR       TRNSMT
4042
4043          ;*****
4044          ;SUBROUTINE TO SETUP AN ADDRESS FROM WHICH DATA IS TO BE TRANSMITTED VIA
4045          ;THE DL11.
4046          ;*****
4047
4048 017320 017637 000000 017666  XLDADD: MOV      @ (SP),TRANPT ;SETUP ADDRESS OF DATA TO BE TRANSFERRED
4049 017326 000771      BR       XLD1
```



```
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061 017330 104002
4062 017332 012746 000000
4063 017336 012746 017344
4064 017342 000002
4065 017344 013701 017666
4066 017350 032777 010000 161774
4067 017356 001406
4068 017360 005237 032214
4069 017364 000001
4070 017366 005737 032214
4071 017372 001374
4072 017374 032777 004000 161750
4073 017402 001401
4074 017404 105741
4075 017406 105711
4076 017410 001446
4077 017412 122711 000004
4078 017416 001443
4079 017420 122711 000023
4080 017424 001453
4081 017426 105711
4082 017430 001422
4083 017432 032777 010000 161712
4084 017440 001103
4085 017442 105777 161714
4086 017446 100375
4087 017450 111177 161710
4088 017454 005737 017664
4089 017460 001006
4090 017462 122711 000004
4091 017466 001403
4092 017470 122721 000003
4093 017474 001325
4094 017476 005037 017664
4095 017502 032777 001000 161642
4096 017510 001004
4097 017512 005737 032144
4098 017516 001001
4099 017520 104004
4100 017522 104003
4101 017524 000002

;*****
;SBTTL DL11 TRANSMITTER ROUTINE
;THIS ROUTINE IS ENTERED WITH THE ADDRESS OF THE CHARACTER OR CHARACTERS
;TO BE TRANSMITTED IN ADDRESS 'TRANPT'. CHARACTERS ARE TRANSMITTED UNTIL
;EITHER AND 'EOT', 'EXT' OR A NULL CHARACTER IS TRANSMITTED. IF 'SW11'
;IS SET, THE SAME CHARACTER IS TRANSMITTED EVERY TIME. IF 'SW12' IS SET,
;THE PROGRAM WAITS FOR A 'CR' TO BE TYPED BEFORE THE CHARACTER IS TRANS-
;MITTED. AS IT IS TRANSMITTED, IT IS ALSO PRINTED.
;*****

TRANSMT: SAVREG
          MOV     #0,      -(SP)      ;ENABLE INTERRUPTS
          MOV     #1$,    -(SP)
          RTI
          1$: MOV     TRANPT,R1        ;SET UP TRANSMITTER BUFFER POINTER.
          TRAN0: BIT     #SW12,@SWR    ;SINGLE STEP TRANSFER?
          BEQ     TRAN1              ;NO
          INC     CHRCNT              ;YES, SET TTY SOFTWARE FLAG
          JAIT                      ;WAIT FOR 'CR'
          TST     CHRCNT              ;WAS THE INTERRUPT FROM TTY?
          BNE     .-6                ;NO, WAIT AGAIN
          TRAN1: BIT     #SW11,@SWR    ;TRANSMIT SAME CHAR.?
          BEQ     .+4                ;NO
          TSTB    -(R1)              ;YES, BACK UP POINTER
          TSTB    (R1)               ;DONE?
          BEQ     TRAN4              ;YES, EXIT
          CMPB    #EOT,(R1)
          BEQ     TRAN4
          CMPB    #DC3,(R1)
          BEQ     TRAN5
          TRAN3: TSTB    (R1)          ;TERMINATOR CHAR.?
          BEQ     TRNEXT              ;YES, EXIT
          BIT     #SW12,@SWR          ;TRANSMITTING SINGLE STEP?
          BNE     TRAN6              ;YES, PRINT CHAR. TO BE TRANSMITTED
          TRAN7: TSTB    @XCSR0        ;WAIT FOR READY
          BPL     .-4
          MOVB    (R1),@XBUFO        ;TRANSMIT CHAR.
          TST     SNGCHR              ;SINGLE CHAR. TRANSFER?
          BNE     TRNEXT              ;YES, EXIT
          TRAN2: CMPB    #EOT,(R1)    ;TRANSMITTED LAST CHAR.?
          BEQ     TRNEXT              ;YES, EXIT
          CMPB    #ETX,(R1)+
          BNE     TRAN0              ;NO, TRANSMIT NEXT CHAR.
          TRNEXT: CLR     SNGCHR
          BIT     #SW09,@SWR          ;IS DATA 'SW9' SET?
          BNE     .+12                ;YES, INHIBIT DELAY
          TST     DLYSWH              ;ISSUE DELAY?
          BNE     .+4                ;NO, SKIP IT
          DELAY                      ;DELAY BEFORE EXITING
          GETREG                      ;RESTORE REG.'S
          RTI                        ;EXIT
```

```
4102 017526 005737 032154      TRAN4:  TST      TERMSW      ;ADDRESS SERIAL I/O?
4103 017532 001735              BEQ      TRAN3      ;NO
4104 017534 005037 032154      CLR      TERMSW
4105 017540 104007              TRAN4A:  LDPGMO      ;YES, ADD CODE TO ADDRESS SOURCE
4106 017542 017546              .+4
4107 017544 000754              BR      TRNEXT
4108 017546      021              .BYTE  DC1      ;ALERT SOURCE
4109 017547      075              IADRS7: .BYTE  75      ;MODIFIED BY USER
4110 017550      001              .BYTE  SOH
4111 017551      061              .BYTE  61
4112 017552      023              .BYTE  DC3      ;ENABLE IT
4113 017553      003              .BYTE  ETX
4114
4115 017554 005737 032154      TRAN5:  TST      TERMSW      ;SOURCE INPUT SW. SET:
4116 017560 001722              BEQ      TRAN3      ;NO, NORMAL TRANSMIT
4117 017562 005037 032154      CLR      TERMSW      ;YES, ADDRESS DESTINATION
4118 017566 005737 032146      TST      DSTSWH      ;CURRENTLY ADDR. A DST. MODULE?
4119 017572 001413              BEQ      TRAN5C      ;NO, SEND 'DC2' TO ALERT DST.
4120 017574 005737 032136      TST      REPTSW      ;YES, USING REMOTE DST.?
4121 017600 001404              BEQ      TRAN5B      ;NO
4122 017602 012737 017644 017636 TRAN5A: MOV      #TRAN5G,TRAN5E ;YES, DON'T ENABLE MY DST.
4123 017610 000407              BR      TRAN5D
4124 017612 012737 017643 017636 TRAN5B: MOV      #IADRS8,TRAN5E ;YES, SEND ONLY THE ADDR.
4125 017620 000403              BR      TRAN5D
4126 017622 012737 017642 017636 TRAN5C: MOV      #TRAN5F,TRAN5E ;SEND 'DC2'
4127 017630 005037 032146      TRAN5D: CLR      DSTSWH
4128 017634 104007              LDPGMO
4129 017636 017642              TRAN5E: .+4
4130 017640 000402              BR      .+6
4131 017642      022              TRAN5F: .BYTE  DC2      ;ALERT DEST.
4132 017643      075              IADRS8: .BYTE  75      ;MODIFIED BY USER
4133 017644      023              TRAN5G: .BYTE  DC3
4134 017645      000              .BYTE  0
4135 017646 000734              BR      TRAN4A
4136
4137 017650 104002              TRAN6:  SAVREG
4138 017652 111101              MOV      (R1),R1
4139 017654 004737 015434      JSR      PC,PDMSET
4140 017660 104003              GETREG
4141 017662 000667              BR      TRAN7
4142 017664 000000              SNGCHR: 0
4143 017666 017670              TRANPT: TRNBFO
4144 017670 000000              TRNBFO: 0
4145      020372              .+500
4146 020372 000000              TRNEND: 0
```



```
4147 ;*****
4148 ;ROUTINE TO REQUEST & SAVE MODULE ADDRESS TO BE USED FOR TESTING
4149 ;*****
4150
4151 XADRES: PRINT
4152 MES30 ;TEXT 'MODULE ADDR.?'
4153 ASEMBL ;WAIT & DECODE INPUT
4154 BISB #60,R0 ;CONVERT TO ASCII
4155 TST SIOSWH ;SERIAL INPUT?
4156 BEQ .+10 ;NO, ALLOW ANY ADDRESS
4157 CMPB IADRS0,R0 ;YES, CHECK AGAINST SERIAL I/O
4158 BEQ XADRES ;SAME, REQUEST IT AGAIN
4159 MOVB R0,MODADR
4160 MOVB R0,SRCADR ;SET UP SOURCE ADDR.
4161 MOVB R0,DSTADR ;SET UP PARAMETERS ADDR.
4162 RTI ;YES, EXIT
4163
4164 ;*****
4165 ;SUBROUTINE ENTERED ON AN ILLEGAL TRAP. THE ROUTINE REPORTS WHERE IT
4166 ;TRAPPED 'FROM' AND WHERE IT TRAP 'TO'.
4167 ;*****
4168
4169 ERTRAP: MOV (SP),TOPC ;SAVE LOCATION WHERE IT TRAPPED 'TO'
4170 POP2SP
4171 MOV (SP),FROMPC ;SAVE WHERE IT TRAPPED FROM.
4172 PRINT
4173 MES5 ;TEXT 'ILLEGAL TRAP TO'
4174 SUB #4,TOPC
4175 PRTCT
4176 TOPC ;TYPE 'PC' TRAPPED TO
4177 PRINT
4178 MES6 ;TEXT 'FROM'
4179 SUB #2,FROMPC
4180 PRTCT
4181 FROMPC ;TYPE WHERE IT TRAPPED FROM
4182 JMP MONITR ;RETURN TO MONITOR
4183
4184 ;*****
4185 ;SUBROUTINE TO REQUEST A/D CHANNEL FROM TELETYPE
4186 ;*****
4187
4188 XCHANEL:PRINT
4189 MES17 ;TEXT 'CH.?'
4190 TTYIN ;WAIT FOR INPUT
4191 CMPB #64,INBUF ;LEGAL CH.
4192 BLE XCHANEL ;NO, REQUEST NEW CH.
4193 MOVB INBUF,SOH1 ;YES, SETUP CH.
4194 RTI ;EXIT
```

```
4195
4196
4197
4198
4199
4200
4201
4202
4203 020540 042777 000100 160610 XERMES: BIC #100,@RCSR0 ;CLEAR RECVR. INTERRUPT ENABLES.
4204 020546 011637 032200 MOV (SP),KSTOR3 ;SAVE 'PC'
4205 020552 017637 000000 020616 MOV @ (SP),MESADR ;SAVE MESSAGE ADDRESS
4206 020560 062716 000002 ADD #2,(SP) ;SET UP STACK TO EXIT
4207 020564 032777 020000 160560 BIT #SW13,@SWR ;PRINT ERROR MESSAGE?
4208 020572 001012 BNE ERREXT ;NO, EXIT
4209 020574 104014 PRTOCT ;YES
4210 020576 032206 TSTNUM ;PRINT FAILING TEST NO.
4211 020600 104016 SPACE
4212 020602 162737 000002 032200 SUB #2,KSTOR3
4213 020610 104014 PRTOCT
4214 020612 032200 KSTOR3 ;PRINT 'MA' WHERE ERROR OCCURRED
4215 020614 104012 PRINT
4216 020616 000000 MESADR: 0 ;PRINT ERROR MESSAGE
4217
4218 020620 005777 160526 ERREXT: TST @SWR ;HALT ON ERROR
4219 020624 100403 BMI .+10 ;NO
4220 020626 004737 021626 JSR PC,TTYENB
4221 020632 000001 WAIT ;WAIT FOR 'CR' TO CONTINUE
4222 020634 000002 RTI
4223
4224
4225
4226
4227
4228 020636 104017 XSCOPE: TSTTKS ;CHECK FOR KEYBOARD FLAG
4229 020640 104005 RECVRO ;ENABLE DL11 RECEIVER
4230 020642 032777 040000 160502 BIT #40000,@SWR ;TEST SW-14 FOR SCOPE
4231 020650 001012 BNE SCOPEB ;YES, SCOPE
4232 020652 032777 004000 160472 BIT #4000,@SWR ;NO-TEST SW-11 FOR ITERATION
4233 020660 001015 BNE SCOPEG ;INHIBIT ITERATION
4234 020662 023737 020772 020770 CMP SCOPEF,ICOUNT ;COMPARE CURRENT COUNT TO MAX NUMBER
4235 020670 100011 BPL SCOPEG ;EXIT-DONE
4236 020672 005237 020772 INC SCOPEF ;INCREMENT COUNT
4237 020676 022606 SCOPEB: CMP (6)+,SP ;REPOSITION STACK
4238 020700 012646 MOV (6)+, -(SP) ;RESTORE PREVIOUS PROCESSOR STATUS
4239 020702 012746 020710 MOV #1$, -(SP)
4240 020706 000002 RTI
4241 020710 000177 000060 1$: JMP @RETURN ;REPEAT TEST
4242 020714 005037 020772 SCOPEG: CLR SCOPEF ;CLEAR COUNT
4243 020720 011601 MOV @SP,R1 ;SAVE TEST NO.
4244 020722 011137 032206 MOV (R1),TSTNUM
4245 020726 062716 000002 ADD #2,(SP)
4246 020732 017701 160414 MOV @SWR,R1 ;READ SW'S
4247 020736 042701 177700 BIC #177700,R1 ;CLR UNWANTED BITS
4248 020742 020137 032206 CMP R1,TSTNUM ;HALT ON THIS TEST
4249 020746 001005 BNE .+14 ;NO
4250 020750 104012 PRINT ;YES
```

```

4251 020752 024437 MES13 ;TEXT 'BREAK AT SCOPE X'
4252 020754 104014 PRTCT
4253 020756 032206 TSTNUM
4254 020760 104013 ITYIN ;WAIT FOR 'CR' TO CONTINUE
4255 020762 011637 020774 MOV @SP,RETURN ;SAVE SCOPE RETURN POINTER
4256 020766 000002 RTI ;RETURN INLINE-NEXT TEST
4257 020770 000000 ICOUNT: 0 ;ITERATION COUNT
4258 020772 000000 SCOPEF: 0 ;COUNT LOCATION FOR ITERATION LOOP
4259 020774 000000 RETURN: 0

```

```

4260
4261
4262
4263 *****
4264 ;RANDOM NUMBER SUBROUTINE
4265 ;THIS ROUTINE CREATES A RANDOM NUMBER, MASKS IT TO EIGHT BITS AND SAVES
4266 ;IT IN THE TRANSMITTER BUFFER AREA.
4267 *****
4268

```

```

4269 020776 012701 017670 XRANGN: MOV #TRNBFO,R1
4270 021002 063737 021136 021134 ADD RANB,RANA
4271 021010 063737 021140 021134 ADD RANC,RANA
4272 021016 006137 021134 RCL RANA
4273 021022 063737 021134 021136 ADD RANA,RANB
4274 021030 063737 021140 021136 ADD RANC,RANB
4275 021036 006137 021136 ROL RANB
4276 021042 063737 021134 021140 ADD RANA,RANC
4277 021050 063737 021136 021140 ADD RANB,RANC
4278 021056 006137 021140 ROL RANC
4279 021062 013711 021140 MOV RANC,(R1) ;SAVE NUMBER
4280 021066 042711 100200 BIC #100200,(R1) ;STRIPE NO. TO 7 BIT ASCII
4281 021072 032711 060000 BIT #60000,(R1) ;IS BIT 5 OR 6 HIGH BYTE SET
4282 021076 001002 BNE .+6 ;YES, LEAVE AS IS
4283 021100 052711 040000 BIS #40000,(R1) ;NO, FORCE BIT 6.
4284 021104 032711 000140 BIT #140,(R1) ;IS BIT 5 OR 6 OF LOW BYT SET
4285 021110 001002 BNE .+6 ;YES, LEAVE AS IS
4286 021112 052711 000040 BIS #40,(R1) ;NO, FORCE BIT '5'
4287 021116 005721 TST (R1)+
4288 021120 022701 020372 CMP #TRNEND,R1 ;DONE
4289 021124 001326 BNE XRANGN+4
4290 021126 005037 020372 CLR TRNEND ;TERMINATE BUFFER.
4291 021132 000002 RTI
4292 021134 072701 RANA: 072701
4293 021136 126543 RANB: 126543
4294 021140 101234 RANC: 101234
4295

```

```

4296
4297
4298
4299
4300
4301 021142 104012
4302 021144 024257
4303 021146 104015
4304 021150 005700
4305 021152 001006
4306 021154 005737 032174
4307 021160 001016
4308 021162 104012
4309 021164 031702
4310 021166 000766
4311 021170 010037 032174
4312 021174 062737 000002 032174
4313 021202 017737 010766 032206
4314 021210 062737 000002 032174
4315 021216 005037 020772
4316 021222 012737 021216 020774
4317 021230 000177 010740
4318
4319
4320
4321
4322
4323
4324
4325
4326 021234 104002
4327 021236 112701 000240
4328 021242 104010
4329 021244 005337 021262
4330 021250 003372
4331 021252 005037 021262
4332 021256 104003
4333 021260 000002
4334 021262 000000
4335
4336
4337
4338
4339
4340 021264 105777 160052
4341 021270 100001
4342 021272 104013
4343 021274 000002

;*****
;ROUTINE TO LOOP THRU A SINGLE LOGIC SUBTEST. ENTERED FROM THE 'MONITOR'
;VIA SELECTING TEST '?'.
;*****

SUBX: PRINT
      MES8 ;TEXT 'TEST ADDR.? '
      ASEMBL ;YES, GET ADDR. FROM TTY.
      TST R0 ;WAS AN ADDRESS ENTERED?
      BNE SUBX1 ;YES, LOAD IT
      TST KSTOR1 ;NO, WAS ONE PREVIOUSLY SET UP?
      BNE XLOOP ;YES, RUN OLD ADDRESS
      PRINT ;NO, ILLEGAL ENTRY
      QMARK
      BR SUBX+2 ;ASK FOR NEW ADDRESS

SUBX1: MOV R0,KSTOR1 ;SAVE ADDRESS
        ADD #2,KSTOR1 ;ADD '2' TO POINT TO INSTRUCTION AFTER SCOPE
        MOV @KSTOR1,TSTNUM ;LOAD TEST NO.
        ADD #2,KSTOR1

XLOOP: CLR SCOPEF ;KEEP COUNT AT ZERO
        MOV #XLOOP,RETURN ;LOAD SCOPE LOOP RETURN POINTER
        JMP @KSTOR1 ;JUMP TO TEST

;*****
;SUBROUTINE TO ISSUE N SPACES
;N IS ONE PLUS VALUE CONTAINED IN SPACEX
;SPACEX IS CLEARED WITHIN THE SUBROUTINE, SO THAT A CALL ON
;SPACE WITHOUT LOADING SPACEX ISSUES ONLY ONE SPACE
;*****

XSPACE: SAVREG ;SAVE REG'S
         MOVB #240,R1
         TYPEIT ;OUTPUT A SPACE
         DEC SPACEX ;DECREMENT COUNT
         BGT XSPACE+2 ;LOOP IF NOT DONE
         CLR SPACEX ;RESET COUNT TO ZERO
         GETREG ;RESTORE REG'S
         RTI ;RETURN

SPACEX: 0

;*****
;SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET
;*****

TKSFLG: TSTB @TKS ;FLAG SET?
         BPL .+4 ;NO, EXIT
         TTYIN ;YES, INQUIRE
         RTI

```

```
4344
4345
4346      ;*****
4347      ;SUBROUTINE TO SETUP PARAMETERS FOR THE MODULE PROGRAM TESTS.
4348      ;*****
4349 021276 011637 020774      XSETUP: MOV      (SP),RETURN      ;SET UP THE 'SCOPE' RETURN ADDRESS.
4350 021302 011637 032124      MOV      (SP),RVECTR
4351 021306 162737 000002 032124      SUB      #2,RVECTR      ;SET UP THE RESTART ADDRESS
4352 021314 013737 032124 032224      MOV      RVECTR,RESTR      ;AND THE 'C' POINTER
4353 021322 005037 020770      CLR      ICOUNT
4354 021326 012737 000001 032206      MOV      #1,TSTNUM      ;SET UP TEST '1'
4355 021334 104005      RECVRO      ;ENABLE DL11 RECEIVER
4356 021336 000002      RTI
4357
4358      ;*****
4359      ;SUBROUTINE TO PRINT CHARACTER IN 'R1'
4360      ;*****
4361
4362 021340 004737 021626      XTYPIT: JSR      PC,TTYENB      ;ENABLE INTERRUPTS
4363 021344 105777 157776      TSTB      @TPS      ;PRINTER READY
4364 021350 100375      BPL      .-4      ;NO
4365 021352 005737 032136      TST      REPTSW      ;REMOTE DST.?
4366 021356 001404      BEQ      XTYPE2      ;NO
4367 021360 110137 021366      MOVB      R1,XTYPE1      ;YES, SET UP TO TRANSMIT CHAR.
4368 021364 104006      LDCHRO
4369 021366 000004      XTYPE1: EOT
4370 021370 110177 157754      XTYPE2: MOVB      R1,@TPB      ;PRINT CHAR.
4371 021374 000002      RTI
4372
4373      ;*****
4374      ;SUBROUTINE TO PRINT THE CONTROL CHARACTER IN 'R1'.
4375      ;*****
4376
4377 021376 122701 000012      XPRCNT: CMPB      #12,R1      ;CHAR - LF?
4378 021402 001413      BEQ      XPRCT1      ;YES
4379 021404 122701 000015      CMPB      #15,R1      ;CHAR. = 'CR'?
4380 021410 001410      BEQ      XPRCT1
4381 021412 013746 032122      MOV      PRTSWH,-(SP)      ;SAVE SW. STATUS
4382 021416 104012      PRINT
4383 021420 032030      UPAROW
4384 021422 012637 032122      MOV      (SP)+,PRTSWH
4385 021426 052701 000100      BIS      #100,R1      ;MAKE CHAR. PRINTABLE
4386 021432 104010      XPRCT1: TYPEIT
4387 021434 042701 000100      BIC      #100,R1      ;RESTORE ORGINAL VALUE
4388 021440 000002      RTI
```

```
4389
4390
4391
4392
4393
4394 021442 110005
4395 021444 005037 032140
4396 021450 104012
4397 021452 027256
4398 021454 104013
4399 021456 122737 000131 015330
4400 021464 001003
4401 021466 104026
4402 021470 010037 032140
4403 021474 110537 017227
4404 021500 000207
4405
4406 021502 013737 032140 032136
4407 021510 005737 032136
4408 021514 001402
4409 021516 004737 017234
4410 021522 000207
4411
4412
4413 021524 005737 032136
4414 021530 001402
4415 021532 104006
4416 021534 000004
4417 021536 005037 032136
4418 021542 000207
4419
4420 021544 012737 000001 032144
4421 021552 000002
4422
4423
4424
4425
4426 021554 012737 000002 021262
4427 021562 000410
4428 021564 012746 000000
4429 021570 012746 021576
4430 021574 000002
4431 021576 012737 000011 021262
4432 021604 105777 157536
4433 021610 100375
4434 021612 005077 157532
4435 021616 005337 021262
4436 021622 001370
4437 021624 000002

;*****
;SUBROUTINE TO CHECK FOR AND SET UP A REMOTE DESTINATION MODULE.
;*****
REMOTE: MOVB R0,R5 ;TEMPORARILY SAVE MODULE ADDRESS
        CLR KSTOR0 ;CLR SOFTWARE SW.
        PRINT
        MES67 ;TEXT 'REMOTE DEST.?'
        TTYIN ;WAIT FOR INPUT
        CMPB #'Y',INBUF ;WAS YES TYPED?
        BNE .+10 ;NO
        ADDRESS ;YES, REQUEST IT'S ADDRESS
        MOV R0,KSTOR0 ;SAVE IT, THIS ALSO SETS SOFTWARE SW.
        MOVB R5,SRCADR ;SET UP A/D SOURCE ADDR.
        RTS PC ;RETURN

SETRMT: MOV KSTOR0,REPTSW ;SET UP THE REMOTE DESTINATION SW.
        TST REPTSW ;USING REMOTE DEST.?
        BEQ .+6 ;NO, EXIT
        JSR PC,ADDRDST ;YES, ADDRESS IT
        RTS PC

CLRMOT: TST REPTSW ;OUTPUTTING TO THE DEMOTE DST.?
        BEQ .+6 ;NO, EXIT
        LDCHRO ;YES, SEND 'EOT' TO CLR MODULE
        EOT
        CLR REPTSW
        RTS PC ;RETURN

XNODLY: MOV #1,DLYSWH ;SET THE TRANS. DELAY INHIBIT SW.
        RTI

;*****
;SUBROUTINE TO TRANSMIT A 'NULL' CHAR. TO THE PRINTER.
;*****
XNULL: MOV #2,SPACEX
        BR XNULL2
XNULL1: MOV #0, -(SP) ;ENABLE INTERRUPTS
        MOV #1$, -(SP)
        RTI
1$: MOV #11,SPACEX
XNULL2: TSTB @TPS
        BPL .-4
        CLR @TPB ;TRANSMIT A NULL CHAR.
        DEC SPACEX
        BNE XNULL2
        RTI
```



```
4438
4439
4440      ;*****
4441      ;SUBROUTINE ENABLE KEYBOARD INTERRUPTS.
4442      ;*****
4443 021626 012777 000100 157506 TTYENB: MOV    #100,@TKS    ;YES, ENABLE TTY INTERRUPTS
4444 021634 012746 000000      MOV    #0,    -(SP)    ;ENABLE INTERRUPTS
4445 021640 012746 021646      MOV    #1$,    -(SP)
4446 021644 000002      RTI
4447 021646 000207 1$:      RTS    PC
4448
4449      ;*****
4450      ;MESSAGE PRINT ROUTINE, ENTERED VIA EMT DISPATCH HANDLER.
4451      ;ROUTINE PICKS UP CONTENTS OF THE 'PC' AND USES THIS AS
4452      ;THE ADDRESS OF MESSAGE TO BE TYPED.
4453      ; IS NEXT MESSAGE SWITCH
4454      ;% IS CRLF SWITCH
4455      ;@ IS END OF MESSAGE SWITCH
4456      ;*****
4457
4458 021650 104002      XPRINT: SAVREG      ;SAVE REGISTERS ON STACK
4459 021652 005037 032152      CLR    OPRTSW
4460 021656 005237 032122      INC    PRTSWH
4461 021662 004737 021626      JSR    PC,TTYENB    ;ENABLE TTY INTERRUPTS
4462 021666 017602 000000      TYP3: MOV    @ (SP),R2    ;GET THE MESSAGE ADDRESS FROM STACK
4463 021672 062716 000002      ADD    #2,(SP)    ;SET UP STACK TO EXIT
4464 021676 112201      TYP3: MOVB    (R2)+,R1    ;GET CHAR.
4465 021700 005701      TST    R1    ;=NULL CHAR.?
4466 021702 001414      BEQ    PRTEXT    ;YES, EXIT
4467 021704 122701 000004      CMPB    #4,R1    ;TEST FOR 'EOT'
4468 021710 001003      BNE    .+10    ;NOT EOT
4469 021712 104012      PRINT    ;YES, PRINT 'EOT'
4470 021714 032106      MESEOT
4471 021716 000406      BR    PRTEXT    ;EXIT
4472 021720 122701 000137      CMPB    #137,R1    ;TEST FOR ' '
4473 021724 001760      BEQ    TYP3    ;YES PICK UP NEXT MESSAGE ADDRESS.
4474 021726 122701 000100      CMPB    #100,R1    ;TEST FOR '@'
4475 021732 001006      BNE    TYP1    ;BRANCH IF NO EQUAL
4476 021734 005037 032122      PRTEXT: CLR    PRTSWH
4477 021740 005037 032152      CLR    OPRTSW
4478 021744 104003      GETREG      ;RESTORE REGISTERS FROM STACK.
4479 021746 000002      RTI    ;OTHERWISE EXIT
4480 021750 005737 032152      TYP1: TST    OPRTSW    ;INHIBIT TYPEOUT?
4481 021754 001350      BNE    TYP2    ;YES, SCAN DATA
4482 021756 122701 000045      CMPB    #45,R1    ;TEST FOR '%'
4483 021762 001402      BEQ    TYPECL    ;IF - TYPE 'CR-LF'
4484 021764 104010      TYP2: TYPEIT    ;OUTPUT CHAR.
4485 021766 000743      BR    TYP2
4486 021770 012701 000015      TYPECL: MOV    #15,R1    ;TYPE 'CR'
4487 021774 104010      TYPEIT
4488 021776 104010      TYPEIT
4489 022000 012701 000012      MOV    #12,R1
4490 022004 104010      TYPEIT    ;INCREMENT BUFFER
4491 022006 000733      BR    TYP2
```

```
4492
4493
4494
4495
4496
4497 022010 004737 021626      XOCTPR: JSR      PC,TTYENB      ;ENABLE TTY INTERRUPTS
4498 022014 104002              SAVREG      ;SAVE REGISTERS ON STACK
4499 022016 017601 000000      MOV      @ (SP),R1      ;THE ADDRESS OF WORD TO BE TYPED
4500 022022 062716 000002      ADD      #2,(SP)      ;SET UP STACK TO EXIT
4501 022026 012703 000006      MOV      #6,R3
4502 022032 012737 000376 022114  MOV      #376,MASK      ;MASK FOR FIRST BIT
4503 022040 000401              BR      .+4
4504 022042 006111      MOVEIT: ROL      (R1)
4505 022044 006111              ROL      (R1)
4506 022046 006111              ROL      (R1)
4507 022050 111102      MOV      (R1),R2
4508 022052 143702 022114      BIC      MASK,R2
4509 022056 052702 000260      BIS      #260,R2
4510 022062 132777 000200 157256  BIT      #200,@TPS
4511 022070 100374              BPL      .-6
4512 022072 110277 157252      MOV      R2,@TPB      ;PRINT CHAR.
4513 022076 012737 000370 022114  MOV      #370,MASK      ;MASK FOR NEXT '5' DIGITS
4514 022104 005303              DEC      R3
4515 022106 001355              BNE      MOVEIT
4516 022110 104003              GETREG
4517 022112 000002              RTI      ;RESTORE REGISTERS FROM STACK.
4518 022114 000376      MASK:      376
4519
4520
4521
4522
4523
4524 022116 012737 161000 032212  XDELAY: MOV      #161000,TEMP2      ;SET UP SHORT DELAY
4525 022124 000402              BR      .+6
4526 022126 005037 032212      XDLAYL: CLR      TEMP2      ;SET UP LONG DELAY
4527 022132 004737 021626      JSR      PC,TTYENB      ;ENABLE TTY INTERRUPTS
4528 022136 012737 177777 032210  MOV      #-1,TEMP1
4529 022144 005237 032212      XDLAY3: INC      TEMP2
4530 022150 001375              BNE      XDLAY3
4531 022152 005237 032210      INC      TEMP1
4532 022156 001372              BNE      XDLAY3
4533 022160 000002      XDLAY2: RTI
4534
```



```
4535 ;*****
4536 ;SUBROUTINE TO PRINT THE DATA IN THE DL11 RECEIVER & TRANSMITTER BUFFER.
4537 ;*****
4538
4539 022162 104012 PRTBF1: PRINT
4540 022164 032032 CRLF
4541 022166 005712 PRTBF2: TST (R2) ;BUFFER EMPTY?
4542 022170 001003 BNE .+10 ;NO, PRINT IT
4543 022172 104012 PRINT ;YES
4544 022174 024275 MES9 ;TEXT 'BUFFER EMPTY'
4545 022176 000411 BR PRT1B ;EXIT
4546 022200 004737 021626 JSR PC,TTYENB ;ENABLE INTR.'S.
4547 022204 005037 032142 CLR FORMAT1 ;'CR/LF' FORMAT SW.
4548 022210 112201 PRT1A: MOVB (R2)+,R1 ;GET CHARACTER
4549 022212 004737 015434 JSR PC,PDMSET ;PRINT CHAR.
4550 022216 105712 TSTB (R2) ;DONE?
4551 022220 001373 BNE PRT1A
4552 022222 000207 PRT1B: RTS PC ;RETURN
4553
4554 ;*****
4555 ;SUBROUTINE TO PRINT THE CONTENTS OF THE DL11 RECVR. BUFFER.
4556 ;*****
4557
4558 022224 012702 016234 RECBUF: MOV #RECBF0,R2 ;SET UP BUFFER POINTER
4559 022230 000402 BR TRNBUF+4
4560
4561 ;*****
4562 ;SUBROUTINE TO PRINT THE CONTENTS OF THE DL11 TRANSMITTER BUFFER
4563 ;*****
4564
4565 022232 012702 017670 TRNBUF: MOV #TRNBF0,R2 ;SET UP BUFFER POINTER
4566 022236 004737 022162 JSR PC,PRTBF1
4567 022242 000137 001376 JMP MONITR ;RETURN TO MONITOR
4568
4569 ;*****
4570 ;SUBROUTINE, ENTERED AS A SUBROUTINE, TO PRINT CONTENTS OF THE DL11
4571 ;RECEIVER BUFFER.
4572 ;*****
4573
4574 022246 104002 XPRTRB: SAVREG ;SAVE REG'S
4575 022250 012702 016234 MOV #RECBF0,R2 ;SETUP BUFFER POINTER
4576 022254 004737 022166 JSR PC,PRTBF2
4577 022260 104003 GETREG ;RESTORE REG.'S
4578 022262 000002 RTI
```

4579
4580
4581
4582
4583
4584
4585
4586 022264 104035
4587 022266 104012
4588 022270 031700
4589 022272 104036
4590 022274 005237 032150
4591 022300 104005
4592 022302 012702 017670
4593 022306 004737 021626
4594 022312 000001
4595 022314 000776
4596 022316 113701 015330
4597 022322 122701 000005
4598 022326 001003
4599 022330 104000
4600 022332 000137 001376
4601 022336 110112
4602 022340 112237 022354
4603 022344 105012
4604 022346 004737 015434
4605 022352 104006
4606 022354 000000
4607 022356 000755
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617 022360 104012
4618 022362 031700
4619 022364 104036
4620 022366 012746 000000
4621 022372 012746 022400
4622 022376 000002
4623 022400 104005
4624 022402 104007
4625 022404 017670
4626 022406 005737 032132
4627 022412 001375
4628 022414 017701 156732
4629 022420 005101
4630 022422 005201
4631 022424 001757
4632 022426 000775

```
.SBTTL SEND ROUTINE
*****
;THIS ROUTINE ACCEPTS CHARACTERS FROM THE TELETYPE AND TRANSMITS THEM
;TO THE DL11. THIS ROUTINE USES '^E' TO ESCAPE BACK TO THE MONITOR.
;CONTROL C (^C) IS ECHOED AND SENT AS AN 'EXT':
*****

SEND:  SETUP                      ;SETUP RESTART ADDRESS
        PRINT
        ASTRIC
        NODLAY
        INC SENDSW                ;INHIBIT TRANSMITTER DELAY
        RECVRO                     ;SET SOFTWARE SW.
        MOV #TRNBFO,R2            ;ENABLE DL 0'S RECVR
        JSR PC,TTYENB             ;SET UP BUFFER TO SAVE CHAR.S
        ;ENABLE TTY INTERRUPTS
SEND1: WAIT                        ;WAIT FOR KEYBOARD & RECEIVER INTERRUPTS
        BR -.2                    ;KEYBOARD INTERRUPTS RETURN .+2
        MOV INBUF,R1              ;GET CHAR.
        CMPB #5,R1               ;CHAR. = '^E' ?
        BNE .+10                 ;NO
        PRCNTR                    ;YES, TYPE IT
        JMP MONITR                ;EXIT
        MOVB R1,(R2)              ;SAVE CHAR.
        MOVB (R2)+,SEND2
        CLRB (R2)                 ;LOAD '0' TO TERMINATE BUFFER
        JSR PC,PDMSET             ;PRINT CHAR.
        LDCHRO                    ;TRANSMIT CHAR.

SEND2: 0
        BR SEND1
```

```
.SBTTL RUN ROUTINE
*****
;THIS ROUTINE IS USED TO LOAD AND RUN TRANSMIT THE USERS SEND
;IN PROGRAM. DATA SW.'S '0-15' CAN BE USED TO SET UP
;A LOOP DELAY. IF THIS SERIAL I/O OPTION INPUT IS BEING USED,
;THE USERS PROGRAM ISN'T LOOPED, IT IS JUST LOADED AND RUN.
*****

RUN:  PRINT
        ASTRIC
        NODLAY
        MOV #0, -(SP)            ;INHIBIT TRANS. DELAY
        MOV #1$, -(SP)          ;ENABLE INTERRUPTS
        RTI
1$:  RECVRO                       ;ENABLE DL RECVR
        LDPGMO                    ;LOAD THE USERS PROGRAM FROM
        TRNBFO                    ;THE TRANSMITTER BUFFER
        TST SIOSWH                ;SERIAL I/O INPUT?
        BNE .-4                  ;YES, STAY HERE
        MOV @SWR,R1              ;LOAD THE SW.'S TO SET DELAY
        COM R1
        INC R1
        BEQ RUN+4
        BR .-4
```

```
4633
4634
4635
4636
4637
4638
4639
4640 022430 104002
4641 022432 012704 015330
4642 022436 012703 022564
4643 022442 005037 022566
4644 022446 005005
4645 022450 005001
4646 022452 005002
4647 022454 005737 032214
4648 022460 003426
4649 022462 005337 032214
4650 022466 122714 000054
4651 022472 001421
4652 022474 121427 000060
4653 022500 002425
4654 022502 121427 000071
4655 022506 003022
4656 022510 142714 000360
4657 022514 112405
4658 022516 010102
4659 022520 006301
4660 022522 006301
4661 022524 006301
4662 022526 060201
4663 022530 060201
4664 022532 060501
4665 022534 000747
4666 022536 105724
4667 022540 010123
4668 022542 005737 032214
4669 022546 001337
4670 022550 104003
4671 022552 000002
4672 022554 104012
4673 022556 027274
4674 022560 000137 001376
4675 022564 000000
4676 022566 000000
4677 022570 000000
4678 022572 000000

.SBTTL SUBROUTINES
:*****
:SUBROUTINE WILL CONVERT 'N' BCD WORDS (SEPARATED VIA COMMA'S)
:WHICH WERE STORED IN A TABLE VIA 'TTYIN' TO OCTAL AND STORE THEM.
:*****

XBCDBIN: SAVREG          ;SAVE REG.'S
      MOV #INBUF,R4      ;SETUP ASCII STORAGE TABLE
      MOV #BCDTAB,R3     ;TABLE FOR STORAGE OF CONVERTED WORDS
      CLR BCDTAB+2
BCDBN1: CLR R5
      CLR R1             ;REG. TO STORE RUNNING TOTAL
      CLR R2             ;TEMP. STORAGE FOR 'R1'
BCDBN2: TST CHRCNT        ;END OF DATA?
      BLE BCDEND         ;YES, EXIT
      DEC CHRCNT         ;DECREMENT CHARACTER COUNTER
      CMPB #54,(R4)      ;IS CHARACTER = TO ','?
      BEQ BCDEND         ;YES, DECODE NEW WORD
      CMPB (R4),#60
      BLT BCDERR         ;TEST FOR LEGAL NO.
      CMPB (R4),#71
      BGT BCDERR
      BICB #360,(R4)     ;STRIPE NO. TO BCD
      MOVB (R4)+,R5      ;SAVE NO. IN R0.
      MOV R1,R2          ;SAVE CURRENT TOTAL
      ASL R1             ;NX2
      ASL R1             ;NX4
      ASL R1             ;NX8
      ADD R2,R1          ;NX9
      ADD R2,R1          ;NX10
      ADD R5,R1          ;N+NEW NO.
      BR BCDDBN2
BCDEND: TSTB (R4)+       ;UPDATE BUFFER
      MOV R1,(R3)+       ;SAVE CONVERTED VALUE & SETUP TO SAVE NEXT
      TST CHRCNT         ;FINISHED?
      BNE BCDDBN1        ;NO, CONVERT NEXT WORD
      GETREG
      RTI               ;YES, EXIT
BCDERR: PRINT            ;TEXT 'ILLEGAL DECIMAL NO.'
      MES68             ;RETURN TO THE MONITOR
      JMP MONITR         ;OCTAL STORAGE TABLE
BCDTAB: 0
        0
        0
        0
```

```

4679 ;*****
4680 ;PRINT DECIMAL VALUE IN R2
4681 ;*****
4682
4683 022574 004737 021626 XBINDEC:JSR PC,TTYENB
4684 022600 104002 SAVREG
4685 022602 012703 177774 MOV #-4,R3
4686 022606 012704 022712 MOV #DECPNT+2,R4
4687 022612 012737 000260 022706 MOV #260,ZERO
4688 022620 012701 177777 TYPT1: MOV #-1,R1
4689 022624 005201 TYPT2: INC R1
4690 022626 161402 SUB (R4),R2
4691 022630 100375 BPL TYPT2
4692 022632 062402 ADD (R4)+,R2
4693 022634 004737 022650 JSR PC,DECOUT
4694 022640 005203 INC R3
4695 022642 001366 BNE TYPT1
4696 022644 104003 GETREG
4697 022646 000002 RTI
4698 022650 005701 DECOUT: TST R1
4699 022652 001006 BNE DEC1
4700 022654 022703 177777 CMP #-1,R3
4701 022660 001403 BEQ DEC1
4702 022662 013701 022706 MOV ZERO,R1
4703 022666 000405 BR DEC2
4704 022670 012737 000260 022706 DEC1: MOV #260,ZERO
4705 022676 052701 000260 BIS #260,R1
4706 022702 104010 DEC2: TYPEIT
4707 022704 000207 RTS PC
4708 022706 000240 ZERO: 240
4709 022710 022712 DECPNT: .+2
4710 022712 001750 1000.
4711 022714 000144 100.
4712 022716 000012 10.
4713 022720 000001 1.
4714

```

```
4715
4716
4717 ;*****
4718 ;POWER FAIL HANDLER
4719 ;*****
4720 PWRFAL: MOV R0,-(SP)
4721 MOV R1,-(SP)
4722 MOV R2,-(SP)
4723 MOV R3,-(SP)
4724 MOV R4,-(SP)
4725 MOV R5,-(SP)
4726 MOV 24,-(SP)
4727 MOV SP,TEMP1
4728 MOV #PWRUP,@#24
4729 HALT
4730 PWRUP: MOV #340,-(SP) ;INHIBIT INTERRUPTS
4731 MOV #1$,-(SP)
4732 RTI
4733 1$: MOV TEMP1,SP
4734 MOV (SP)+,@#24
4735 MOV (SP)+,R5
4736 MOV (SP)+,R4
4737 MOV (SP)+,R3
4738 MOV (SP)+,R2
4739 MOV (SP)+,R1
4740 MOV (SP)+,R0
4741 NULL ;POWER UP DELAY
4742 PRINT
4743 MES28
4744 JMP MONITR
4745
4746 ;*****
4747 ;SUBROUTINE TO SAVE 'R1-R5' ON STACK
4748 ;*****
4749
4750 XSAVRG: MOV (SP)+,SAVEPC
4751 MOV (SP)+,SAVPSW
4752 MOV (SP)+,SAV2PC
4753 MOV (SP)+,SAV2SW
4754 MOV R1,-(SP)
4755 MOV R2,-(SP)
4756 MOV R3,-(SP)
4757 MOV R4,-(SP)
4758 MOV R5,-(SP)
4759 MOV SAV2SW,-(SP)
4760 MOV SAV2PC,-(SP)
4761 MOV SAVPSW,-(SP)
4762 MOV SAVEPC,-(SP)
4763 RTI
```

```
4764
4765
4766
4767
4768 023102 012637 032164
4769 023106 012637 032166
4770 023112 012637 032170
4771 023116 012637 032172
4772 023122 012605
4773 023124 012604
4774 023126 012603
4775 023130 012602
4776 023132 012601
4777 023134 013746 032172
4778 023140 013746 032170
4779 023144 013746 032166
4780 023150 013746 03216
4781 023154 000002
4782
4783
4784
4785
4786
4787 023156 104013
4788 023160 005000
4789 023162 005737 032214
4790 023166 001001
4791 023170 000002
4792 023172 012701 015330
4793 023176 004737 023232
4794 023202 010400
4795 023204 000002
4796
4797 023206 105721
4798 023210 006204
4799 023212 006204
4800 023214 006204
4801 023216 005337 032214
4802 023222 001767
4803 023224 010400
4804 023226 062716 000002
4805 023232 005004
4806 023234 122711 000054
4807 023240 001762
4808 023242 142711 000370
4809 023246 152104
4810 023250 005337 032214
4811 023254 003003
4812 023256 005237 032214
4813 023262 000207
4814 023264 006304
4815 023266 006304
4816 023270 006304
4817 023272 000760
4818
4819
```

```
*****
;SUBROUTINE TO RESTORE 'R1-R5' FROM THE STACK
*****
XGETRG: MOV      (SP)+,SAVEPC
        MOV      (SP)+,SAVPSW
        MOV      (SP)+,SAV2PC
        MOV      (SP)+,SAV2SW
        MOV      (SP)+,R5
        MOV      (SP)+,R4
        MOV      (SP)+,R3
        MOV      (SP)+,R2
        MOV      (SP)+,R1
        MOV      SAV2SW,-(SP)
        MOV      SAV2PC,-(SP)
        MOV      SAVPSW,-(SP)
        MOV      SAVEPC,-(SP)
        RTI
*****
;SUBROUTINE TO WAIT FOR AND ASSEMBLE CHARACTERS INPUT
;FROM THE KEYBOARD INTO OCTAL NUMBERS.
*****
XASEMB: TTYIN                      ;GET CHAR.'S FROM KEYBOARD
        CLR      R0
        TST      CHRCNT             ;ANY CHARACTERS ENTERED
        BNE      .+4               ;YES
        RTI                        ;NO, EXIT
        MOV      #INBUF,R1         ;SET UP CHAR. BUFFER POINTER
        JSR      PC,STRIPN         ;STRIPE NO.
XASEM1: MOV      R4,R0              ;RETURNS HERE IF ONLY '1' NO.
        RTI
WORD2:  TSTB      (R1)+             ;ADVANCE POINTER PAST COMMA
        ASR      R4
        ASR      R4
        ASR      R4
        DEC      CHRCNT             ;DEC. CHAR. CNTR.
        BEQ      XASEM1            ;COMMA LAST CHAR.?
        MOV      R4,R0             ;NO, SAVE 1ST NO.
        ADD      #2,(SP)           ;SET UP STACK TO EXIT
STRIPN: CLR      R4
        CMPB      #54,(R1)         ;CHAR. = COMMA?
        BEQ      WORD2            ;YES, SAVE 1ST NO.
        BICB      #370,(R1)        ;NO, STRIPE NO. TO OCTAL
        BISB      (R1)+,R4
        DEC      CHRCNT            ;FINISHED?
        BGT      .+10             ;NO
        INC      CHRCNT            ;CHARACTERS WERE ENTERED
        RTS      PC               ;YES, EXIT
        ASL      R4
        ASL      R4
        ASL      R4
        BR       STRIPN+2
*****
;SUBROUTINE TO OUTPUT CURRENT SWR VALUE IF USING SOFTWARE
```



```
4820      ;SWR AND TO ASSEMBLE CHARACTERS INPUT FROM THE KEYBOARD
4821      ;TO FORM THE NEW SWR VALUE.
4822      ;*****
4823 023274 022737 000176 001352 UPDATE: CMP      #SWSWR, SWR      ;USING SOFTWARE SWR?
4824 023302 001021      BNE      A              ;NO-BRANCH
4825 023304 022737 000176 001352 UPDAT1: CMP      #SWSWR, SWR      ;USING SOFTWARE SWR?
4826 023312 001014      BNE      1$            ;NO-BRANCH
4827 023314 104012      PRINT                     ;YES-PRINT 'SWR='
4828 023316 030415      MES89
4829 023320 104014      PRTOUT                     ;PRINT VALUE
4830 023322 000176      SWSWR
4831 023324 104012      PRINT                     ;PRINT 'NEW SWR='
4832 023326 030424      MES90
4833 023330 104015      ASEMBL                     ;WAIT AND DECODE
4834 023332 005737 032214      TST      CHRCNT      ;WAS A NEW VALUE ENTERRED?
4835 023336 001402      BEQ      1$              ;NO-SAVE OLD VALUE
4836 023340 010037 000176      MOV      %0,      SWSWR ;YES-USE NEW VALUE
4837 023344 000207      1$:      RTS      PC              ;EXIT
4838 023346 104013      A:      TTYIN
4839 023350 000207      RTS      PC              ;WAIT FOR CR
                                         ;EXIT
```

```
4840 :*****
4841 :SBTTL MESSAGES
4842 :*****
4843 023352 000 .BYTE
4844 023353 045 041445 050132 TITLE: .ASCII ;%%CZPMAC0 PDM-70 DIAGNOSTIC TEST @;
4845 023360 040515 030103 050040
4846 023366 046504 033455 020060
4847 023374 044504 043501 047516
4848 023402 052123 041511 052040
4849 023410 051505 020124 100
4850
4851 023415 045 054524 042570 HEADER: .ASCII ;%TYPE IN THE FOLLOWING TO RUN THE DESIRED TEST:%;
4852 023422 044440 020116 044124
4853 023430 020105 047506 046114
4854 023436 053517 047111 020107
4855 023444 047524 051040 047125
4856 023452 052040 042510 042040
4857 023460 051505 051111 042105
4858 023466 052040 051505 035124
4859 023474 045
4860
4861 023475 115 031467 030070 TSTLST: .ASCII ;M7380A, M7381A, M7381E, M7382A, BCDIO, M7383A, M7383C, M7383R, M7383G,
4862 023502 026101 046440 031467
4863 023510 030470 026101 046440
4864 023516 031467 030470 026105
4865 023524 046440 031467 031070
4866 023532 026101 041040 042103
4867 023540 047511 020054 033515
4868 023546 034063 040463 020054
4869 023554 033515 034063 041463
4870 023562 020054 033515 034063
4871 023570 051063 020054 033515
4872 023576 034063 043463 020054
4873 023604 046445 031467 032070 .ASCII ;%M7384A, M7384E, M7385A, M7385I, M7385T, M7386A, M7387A, M7388A, ;
4874 023612 026101 046440 031467
4875 023620 032070 026105 046440
4876 023626 031467 032470 026101
4877 023634 046440 031467 032470
4878 023642 026111 046440 031467
4879 023650 032470 026124 046440
4880 023656 031467 033070 026101
4881 023664 046440 031467 033470
4882 023672 026101 046440 031467
4883 023700 034070 026101 040
4884 023705 115 031467 034070 .ASCII ;M7388F,%M7377A,M7378A, SUBX, RECBUF, TRNBUF, SEND, RUN,@;
4885 023712 026106 046445 031467
4886 023720 033467 026101 033515
4887 023726 033463 040470 020054
4888 023734 052523 054102 020054
4889 023742 042522 041103 043125
4890 023750 020054 051124 041116
4891 023756 043125 020054 042523
4892 023764 042116 020054 052522
4893 023772 026116 100
4894
4895
```



```
4896
4897
4898 023775      045 051525 047111 MES0: .ASCII ;%USING SERIAL I/O INTERFACE OPTION? a;
4899 024002 020107 042523 044522
4900 024010 046101 044440 047457
4901 024016 044440 052116 051105
4902 024024 040506 042503 047440
4903 024032 052120 047511 037516
4904 024040 040040
4905
4906
4907 024042 047503 052116 047522 MES1: .ASCII ;CONTROL MODULE TEST.%a;
4908 024050 020114 047515 052504
4909 024056 042514 052040 051505
4910 024064 027124 040045
4911
4912
4913 024070 042045 030514 020061 MES2: .ASCII ;%DL11 RECV. BUFFER OVERFLOW.%a;
4914 024076 042522 053103 027122
4915 024104 041040 043125 042506
4916 024112 020122 053117 051105
4917 024120 046106 053517 022456
4918 024126      100
4919
4920 024127      123 051105 040511 MES3: .ASCII ;SERIAL I/O ADDRESS TEST.%a;
4921 024134 020114 027511 020117
4922 024142 042101 051104 051505
4923 024150 020123 042524 052123
4924 024156 022456      100
4925 024161      104 030514 020061 MES4: .ASCII ;DL11 OVERRUN ERROR.a;
4926 024166 053117 051105 052522
4927 024174 020116 051105 047522
4928 024202 027122      100
4929
4930 024205      045 046111 042514 MES5: .ASCII ;%ILLEGAL TRAP TO a;
4931 024212 040507 020114 051124
4932 024220 050101 052040 020117
4933 024226      100
4934
4935 024227      040 051106 046517 MES6: .ASCII ; FROM a;
4936 024234 040040
4937
4938 024236 052045 051505 020124 MES7: .ASCII ;%TEST COMPLETE.%a;
4939 024244 047503 050115 042514
4940 024252 042524 022456      100
4941
4942 024257      045 042524 052123 MES8: .ASCII ;%TEST ADDR.? a;
4943 024264 040440 042104 027122
4944 024272 020077      100
4945
4946 024275      102 043125 042506 MES9: .ASCII ;BUFFER IS EMPTY.a;
4947 024302 020122 051511 042440
4948 024310 050115 054524 040056
4949 024316 051045 026505 042523 MES10: .ASCII ;%RE-SET MODULE ADDR. TO '17'(OCTAL). a;
4950 024324 020124 047515 052504
4951 024332 042514 040440 042104
```

4952	024340	027122	052040	020117	
4953	024346	030447	023467	047450	
4954	024354	052103	046101	027051	
4955	024362	040040			
4956	024364	027501	020104	042101	MES11: .ASCII ;A/D ADDRESSING TEST.a;
4957	024372	051104	051505	044523	
4958	024400	043516	052040	051505	
4959	024406	027124	100		
4960	024411	101	042057	041440	MES12: .ASCII ;A/D CALIBRATION TEST.a;
4961	024416	046101	041111	040522	
4962	024424	044524	047117	052040	
4963	024432	051505	027124	100	
4964	024437	045	051102	040505	MES13: .ASCII ;%BREAK AT SCOPE a;
4965	024444	020113	052101	051440	
4966	024452	047503	042520	040040	
4967	024460	044445	051516	051105	MES14: .ASCII ;%INSERT D JUMPER TO INHIBIT 'EOT' AND _;
4968	024466	020124	020104	052512	
4969	024474	050115	051105	052040	
4970	024502	020117	047111	044510	
4971	024510	044502	020124	042447	
4972	024516	052117	020047	047101	
4973	024524	020104	137		
4974	024527	045	042522	047515	MES14A: .ASCII ;%REMOVE JUMPER, CLR MODULE AND ENTERa;
4975	024534	042526	045040	046525	
4976	024542	042520	026122	041440	
4977	024550	051114	046440	042117	
4978	024556	046125	020105	047101	
4979	024564	020104	047105	042524	
4980	024572	040122			
4981	024574	027501	020104	042522	MES15: .ASCII ;A/D REPEATIBILITY TEST.a;
4982	024602	042520	052101	041111	
4983	024610	046111	052111	020131	
4984	024616	042524	052123	040056	
4985	024624	053045	043123	020077	MES16: .ASCII ;%V3F? a;
4986	024632	100			
4987	024633	103	027110	020077	MES17: .ASCII ;CH.? a;
4988	024640	100			
4989					
4990	024641	107	044501	020116	MES18: .ASCII ;GAIN ACCURACY TEST.%a;
4991	024646	041501	052503	040522	
4992	024654	054503	052040	051505	
4993	024662	027124	040045		
4994					
4995	024666	051445	050125	046120	MES19: .ASCII ;%SUPPLY +1.990V WITH a;
4996	024674	020131	030453	034456	
4997	024702	030071	020126	044527	
4998	024710	044124	040040		
4999					
5000	024714	040507	047111	023440	MES20: .ASCII ;GAIN 'LOW'.a;
5001	024722	047514	023527	040056	
5002					
5003	024730	053523	052111	044103	MES21: .ASCII ;SWITCH VOLTAGE NEGATIVE.a;
5004	024736	053040	046117	040524	
5005	024744	042507	047040	043505	
5006	024752	052101	053111	027105	
5007	024760	100			

```

5008
5009 02475 107 044501 020116 MES22: .ASCII ;GAIN 'MEDIUM'.a;
5010 024766 046447 042105 052511
5011 024774 023515 040056
5012
5013 025000 040507 047111 023440 MES23: .ASCII ;GAIN 'HIGH'.a;
5014 025006 044510 044107 040047
5015
5016 025014 051445 050125 046120 MES24: .ASCII ;%SUPPLY +.1990V WITH _;
5017 025022 020131 027053 034461
5018 025030 030071 020126 044527
5019 025036 044124 057440
5020
5021 025042 051445 050125 046120 MES24A: .ASCII ;%SUPPLY +0.01990V WITH _;
5022 025050 020131 030053 030056
5023 025056 034461 030071 020126
5024 025064 044527 044124 057440
5025 025072 051445 050125 046120 MES25: .ASCII ;%SUPPLY +0.000V.a;
5026 025100 020131 030053 030056
5027 025106 030060 027126 100
5028
5029 025113 045 047111 047503 MES26: .ASCII ;%INCORRECT GAIN.a;
5030 025120 051122 041505 020124
5031 025126 040507 047111 040041
5032
5033 025134 020045 046040 053517 MES27: .ASCII ;% LOW AVG HIGH.a;
5034 025142 020040 020040 053101
5035 025150 020107 020040 044510
5036 025156 044107 100
5037 025161 045 042522 047503 MES28: .ASCII ;%RECOVERED FROM POWER FAILURE - BY GOLLY!.a;
5038 025166 042526 042522 020104
5039 025174 051106 046517 050040
5040 025202 053517 051105 043040
5041 025210 044501 052514 042522
5042 025216 026440 041040 020131
5043 025224 047507 046114 020531
5044 025232 100
5045
5046 025233 102 042103 044440 MES29: .ASCII ;BCD INPUT ADDRESS TEST.a;
5047 025240 050116 052125 040440
5048 025246 042104 042522 051523
5049 025254 052040 051505 027124
5050 025262 100
5051
5052 025263 045 047515 052504 MES30: .ASCII ;%MODULE ADDR.? a;
5053 025270 042514 040440 042104
5054 025276 027122 020077 100
5055 025303 123 052105 041440 MES31: .ASCII ;SET CUST. SW.'S _;
5056 025310 051525 027124 051440
5057 025316 027127 051447 057440
5058 025324 046101 020114 047117 MES31A: .ASCII ;ALL ON WITH INPUTS HI.a;
5059 025332 053440 052111 020110
5060 025340 047111 052520 051524
5061 025346 044040 027111 100
5062 025353 101 046114 047440 MES31B: .ASCII ;ALL OFF WITH INPUTS HI.a;
5063 025360 043106 053440 052111
  
```

5064	025366	020110	047111	052520	
5065	025374	051524	044040	027111	
5066	025402	100			
5067	025403	124	020117	046101	MES31C: .ASCII ;TO ALTERNATE ON & OFF'S.a;
5068	025410	042524	047122	052101	
5069	025416	020105	047117	023040	
5070	025424	047440	043106	051447	
5071	025432	040056			
5072	025434	046101	020114	047117	MES31D: .ASCII ;ALL ON WITH INPUTS LO.a;
5073	025442	053440	052111	020110	
5074	025450	047111	052520	051524	
5075	025456	046040	027117	100	
5076	025463	101	046114	047440	MES31E: .ASCII ;ALL OFF.a;
5077	025470	043106	040056		
5078	025474	041502	020104	047111	MES32: .ASCII ;BCD INPUT EXERCISER TEST.a;
5079	025502	052520	020124	054105	
5080	025510	051105	044503	042523	
5081	025516	020122	042524	052123	
5082	025524	040056			
5083	025526	041502	020104	052517	MES33: .ASCII ;BCD OUTPUT ADDRESS TEST.a;
5084	025534	050124	052125	040440	
5085	025542	042104	042522	051523	
5086	025550	052040	051505	027124	
5087	025556	100			
5088	025557	045	054105	046501	MES34: .ASCII ;%EXAMINE OUTPUT LINES FOR _;
5089	025564	047111	020105	052517	
5090	025572	050124	052125	046040	
5091	025600	047111	051505	043040	
5092	025606	051117	057440		
5093	025612	046101	020114	047514	MES35: .ASCII ;ALL LOGIC 1'S.a;
5094	025620	044507	020103	023461	
5095	025626	027123	100		
5096	025631	122	053105	051105	MES37: .ASCII ;REVERSED _;
5097	025636	042523	020104	137	
5098	025643	045	041523	050117	MES38: .ASCII ;%SCOPE FOR 'OUTPUT DONE H&L' (TYPE ^R TO RESTART).a;
5099	025650	020105	047506	020122	
5100	025656	047447	052125	052520	
5101	025664	020124	047504	042516	
5102	025672	044040	046046	020047	
5103	025700	052050	050131	020105	
5104	025706	051136	052040	020117	
5105	025714	042522	052123	051101	
5106	025722	024524	040056		
5107	025726	045445	054505	047502	MES39: .ASCII ;%KEYBOARD/DISPLAY MODULE ADDRESS TEST.a;
5108	025734	051101	027504	044504	
5109	025742	050123	040514	020131	
5110	025750	047515	052504	042514	
5111	025756	040440	042104	042522	
5112	025764	051523	052040	051505	
5113	025772	027124	100		
5114	025775	061	051447	040440	MES40: .ASCII ;1'S AND LEAVE THE INPUTS OPEN.a;
5115	026002	042116	046040	040505	
5116	026010	042526	052040	042510	
5117	026016	044440	050116	052125	
5118	026024	020123	050117	047105	
5119	026032	040056			

5120	026034	023461	020123	047101	MES40A: .ASCII ;1'S AND GND ALL INPUTS.a;
5121	026042	020104	047107	020104	
5122	026050	046101	020114	047111	
5123	026056	052520	051524	040056	
5124	026064	046101	042524	047122	MES40B: .ASCII ;ALTERNATE 1'S & 0'S.a;
5125	026072	052101	020105	023461	
5126	026100	020123	020046	023460	
5127	026106	027123	100		
5128	026111	060	051447	040056	MES41: .ASCII ;0'S.a;
5129	026116	047111	027124	047440	MES42: .ASCII ;INT. OR EXT. SYNC.? a;
5130	026124	020122	054105	027124	
5131	026132	051440	047131	027103	
5132	026140	020077	100		
5133	026143	102	042103	044440	MES43: .ASCII ;BCD I/O TEST.a;
5134	026150	047457	052040	051505	
5135	026156	027124	100		
5136	026161	103	040510	040522	MES44: .ASCII ;CHARACTER I/O ADDRESS TEST_;
5137	026166	052103	051105	044440	
5138	026174	047457	040440	042104	
5139	026202	042522	051523	052040	
5140	026210	051505	057524	040	
5141	026215	104	040457	040440	MES45: .ASCII ;D/A ADDRESSING TEST.a;
5142	026222	042104	042522	051523	
5143	026230	047111	020107	042524	
5144	026236	052123	040056		
5145	026242	051445	047503	042520	MES46: .ASCII ;%SCOPE FOR 'PROG L' & 'FLOP L' LO.a;
5146	026250	043040	051117	023440	
5147	026256	051120	043517	046040	
5148	026264	020047	020046	043047	
5149	026272	047514	020120	023514	
5150	026300	046040	027117	100	
5151	026305	123	047503	042520	MES47: .ASCII ;SCOPE FOR 'PROG L' HI & 'FLOP L' LO.a;
5152	026312	043040	051117	023440	
5153	026320	051120	043517	046040	
5154	026326	020047	044510	023040	
5155	026334	023440	046106	050117	
5156	026342	046040	020047	047514	
5157	026350	040056			
5158	026352	041523	050117	020105	MES48: .ASCII ;SCOPE FOR 'FLOP L' HI.a;
5159	026360	047506	020122	043047	
5160	026366	047514	020120	023514	
5161	026374	044040	027111	100	
5162	026401	123	047503	042520	MES49: .ASCII ;SCOPE FOR 'FLOP L' LO.a;
5163	026406	043040	051117	023440	
5164	026414	046106	050117	046040	
5165	026422	020047	047514	040056	
5166	026430	044103	041505	020113	MES50: .ASCII ;CHECK CH. '0' OUTPUT FOR _;
5167	026436	044103	020056	030047	
5168	026444	020047	052517	050124	
5169	026452	052125	043040	051117	
5170	026460	057440			
5171	026462	044103	041505	020113	MES51: .ASCII ;CHECK CH. '1' OUTPUT FOR _;
5172	026470	044103	020056	030447	
5173	026476	020047	052517	050124	
5174	026504	052125	043040	051117	
5175	026512	057440			

5176	026514	027060	030060	040126	MES52:	.ASCII	:0.00V@;
5177	026522	027061	030461	040126	MES53:	.ASCII	:1.11V@;
5178	026530	027062	031062	040126	MES54:	.ASCII	:2.22V@;
5179	026536	027064	032064	040126	MES55:	.ASCII	:4.44V@;
5180	026544	027070	034070	040126	MES56:	.ASCII	:8.88V@;
5181	026552	027504	020101	054105	MES57:	.ASCII	:D/A EXERCISER TEST.@;
5182	026560	051105	044503	042523			
5183	026566	020122	042524	052123			
5184	026574	040056					
5185	026576	027504	020101	040526	MES58:	.ASCII	:D/A VALUES(X,Y)? @;
5186	026604	052514	051505	054050			
5187	026612	054454	037451	040040			
5188	026620	051445	052105	040440	MES59:	.ASCII	:%SET ALL DATA SW.'S TO '0'.@;
5189	026626	046114	042040	052101			
5190	026634	020101	053523	023456			
5191	026642	020123	047524	023440			
5192	026650	023460	040056				
5193	026654	052523	050120	054514	MES60:	.ASCII	:SUPPLY AN EXTERNAL SYNC.@;
5194	026662	040440	020116	054105			
5195	026670	042524	047122	046101			
5196	026676	051440	047131	027103			
5197	026704	100					
5198	026705	123	047503	042520	MES61:	.ASCII	:SCOPE FOR THE SIGNAL 'DATA READY, AND CHECK %;
5199	026712	043040	051117	052040			
5200	026720	042510	051440	043511			
5201	026726	040516	020114	042047			
5202	026734	052101	020101	042522			
5203	026742	042101	026131	040440			
5204	026750	042116	041440	042510			
5205	026756	045503	022440				
5206	026762	044103	023456	020123		.ASCII	:CH.'S '0 & 1' OUTPUTS FOR 5 USEC RISE TIMES.@;
5207	026770	030047	023040	030440			
5208	026776	020047	052517	050124			
5209	027004	052125	020123	047506			
5210	027012	020122	020065	051525			
5211	027020	041505	051040	051511			
5212	027026	020105	044524	042515			
5213	027034	027123	100				
5214	027037	124	046124	044440	MES62:	.ASCII	:TTL I/O TEST%@;
5215	027044	047457	052040	051505			
5216	027052	022524	100				
5217	027055	104	030514	020061	MES63:	.ASCII	:DL11 ADRS., VEC.? @;
5218	027062	042101	051522	026056			
5219	027070	053040	041505	037456			
5220	027076	040040					
5221	027100	047111	042523	052122	MES64:	.ASCII	:INSERT THE M7387 READ-IN MODULE & INITIALIZE SYSTEM. %@;
5222	027106	052040	042510	046440			
5223	027114	031467	033470	051040			
5224	027122	040505	026504	047111			
5225	027130	046440	042117	046125			
5226	027136	020105	020046	047111			
5227	027144	052111	040511	044514			
5228	027152	042532	051440	051531			
5229	027160	042524	027115	040045			
5230	027166	050042	047522	020115	MES65:	.ASCII	: 'PROM OK' REMOVE THE M7387. %@;
5231	027174	045517	020042	042522			

5232	027202	047515	042526	052040			
5233	027210	042510	046440	031467			
5234	027216	033470	022456	100			
5235	027223	123	051105	040511	MES66:	.ASCII	;SERIAL I/O INTERFACE TEST%a;
5236	027230	020114	027511	020117			
5237	027236	047111	042524	043122			
5238	027244	041501	020105	042524			
5239	027252	052123	040045				
5240	027256	042522	047515	042524	MES67:	.ASCII	;REMOTE DST.? a;
5241	027264	042040	052123	037456			
5242	027272	040040					
5243							
5244	027274	046111	042514	040507	MES68:	.ASCII	;ILLEGAL DECIMAL NO. ??a;
5245	027302	020114	042504	044503			
5246	027310	040515	020114	047516			
5247	027316	020056	037477	100			
5248							
5249	027323	050	047111	044055	MES69:	.ASCII	;(IN-HOUSE).a;
5250	027330	052517	042523	027051			
5251	027336	100					
5252							
5253	027337	050	044506	046105	MES70:	.ASCII	;(FIELD).a;
5254	027344	024504	040056				
5255							
5256	027350	047111	052520	020124	MES71:	.ASCII	;INPUT DATA, TERMINATE TEST W/EOT.%a;
5257	027356	040504	040524	020054			
5258	027364	042524	046522	047111			
5259	027372	052101	020105	042524			
5260	027400	052123	053440	042457			
5261	027406	052117	022456	100			
5262							
5263	027413	105	040530	044515	MES72:	.ASCII	;EXAMINE 'FIFO' DATA.a;
5264	027420	042516	023440	044506			
5265	027426	047506	020047	040504			
5266	027434	040524	040056				
5267	027440	047105	042524	044522	MES73:	.ASCII	;ENTERING THE DISPLAY TEST. ;
5268	027446	043516	052040	042510			
5269	027454	042040	051511	046120			
5270	027462	054501	052040	051505			
5271	027470	026124	040				
5272	027473	122	026505	047111	MES73A:	.ASCII	;RE-INITIALIZE PDM70.a;
5273	027500	052111	040511	044514			
5274	027506	042532	050040	046504			
5275	027514	030067	040056				
5276	027520	041505	047510	052040	MES74:	.ASCII	;ECHO TEST, _;
5277	027526	051505	026124	057440			
5278	027534	052123	051117	043501	MES75:	.ASCII	;STORAGE TEST _;
5279	027542	020105	042524	052123			
5280	027550	057440					
5281	027552	042522	042523	020124	MES76:	.ASCII	;RESET ADDRESS %a;
5282	027560	042101	051104	051505			
5283	027566	020123	040045				
5284	027572	033515	033463	040470	MES77:	.ASCII	;M7378A FOUNDATION MODULE TEST. %a;
5285	027600	020040	047506	047125			
5286	027606	040504	044524	047117			
5287	027614	046440	042117	046125			

CZ
CZ[illegible]

5288 027622 020105 042524 052123
5289 027630 020056 040045
5290 027634 052520 020124 041523 MES78: .ASCII ;PUT SCOPE PROBE ON PIN 78 ._:
5291 027642 050117 020105 051120
5292 027650 041117 020105 047117
5293 027656 050040 047111 033440
5294 027664 020070 057456
5295 027670 051525 020105 042536 MES79: .ASCII ;USE ^E TO EXIT FROM SUBTEST. @;
5296 027676 052040 020117 054105
5297 027704 052111 043040 047522
5298 027712 020115 052523 052102
5299 027720 051505 027124 040040
5300 027726 033515 033463 040467 MES80: .ASCII ;M7377A REMOTE SERIAL I/O TEST. %@;
5301 027734 051040 046505 052117
5302 027742 020105 042523 044522
5303 027750 046101 0444-0 047457
5304 027756 052040 051505 027124
5305 027764 022440 100
5306 027767 040 042523 042514 MES81: .ASCII ; SELECT 012 (LINEFEED) SWITCH 'V'(VARIABLE TERMINATOR).<CR>@;
5307 027774 052103 030040 031061
5308 030002 024040 044514 042516
5309 030010 042506 042105 020051
5310 030016 053523 052111 044103
5311 030024 023440 023526 053050
5312 030032 051101 040511 046102
5313 030040 020105 042524 046522
5314 030046 047111 052101 051117
5315 030054 027051 041474 037122
5316 030062 100
5317 030063 110 052111 041440 MES82: .ASCII ;HIT CARRIAGE RETURN TO CONTINUE.%%@;
5318 030070 051101 044522 043501
5319 030076 020105 042522 052524
5320 030104 047122 052040 020117
5321 030112 047503 052116 047111
5322 030120 042525 022456 040045
5323 030126 042522 042523 020124 MES83: .ASCII ;RESET MODE SWITCH.%@;
5324 030134 047515 042504 051440
5325 030142 044527 041524 027110
5326 030150 040045
5327 030152 042523 020124 046103 MES84: .ASCII ;SET CLOCK 3 ON CLOCK MODULE TO 100 MILLI SECONDS.%%_:
5328 030160 041517 020113 020063
5329 030166 047117 041440 047514
5330 030174 045503 046440 042117
5331 030202 046125 020105 047524
5332 030210 030440 030060 046440
5333 030216 046111 044514 051440
5334 030224 041505 047117 051504
5335 030232 022456 057445
5336 030236 022445 051440 052105 MES85: .ASCII ;%% SET SWITCH 1 OF SWITCH P TO ON. %@;
5337 030244 051440 044527 041524
5338 030252 020110 030440 047440
5339 030260 020106 053523 052111
5340 030266 044103 050040 052040
5341 030274 020117 047117 020056
5342 030302 040045
5343 030304 022445 051440 051124 MES86: .ASCII ;%% STRAP OUT EOT<CR>%@;

5344	030312	050101	047440	052125	
5345	030320	042440	052117	041474	
5346	030326	037122	040045		
5347	030332	022445	042522	047515	MES87: .ASCII ;%%REMOVE EOT JUMPER<CR>a;
5348	030340	042526	042440	052117	
5349	030346	045040	046525	042520	
5350	030354	036122	051103	040076	
5351	030362	042522	042523	020124	MES88: .ASCII ;RESET MODULE ADDRESS <CR>a;
5352	030370	047515	052504	042514	
5353	030376	040440	042104	042522	
5354	030404	051523	036040	051103	
5355	030412	022476	100		
5356	030415	015	051412	051127	MES89: .ASCIIZ<15><12>/SWR=
5357	030422	000075			
5358	030424	005015	047012	053505	MES90: .ASCIIZ<15><12><12>/NEW SWR=
5359	030432	051440	051127	000075	
5360	030440	023440	052123	023530	ERR1: .ASCII ; 'STX' WASN'T RETURNED.a;
5361	030446	053440	051501	023516	
5362	030454	020124	042522	052524	
5363	030462	047122	042105	022456	
5364	030470	100			
5365					
5366	030471	040	044504	047104	ERR2: .ASCII ; DIDN'T ENTER ADDRESS MODE.a;
5367	030476	052047	042440	052116	
5368	030504	051105	040440	042104	
5369	030512	042522	051523	046440	
5370	030520	042117	027105	040045	
5371					
5372	030526	042040	052101	020101	ERR3: .ASCII ; DATA ERROR.a;
5373	030534	051105	047522	027122	
5374	030542	040045			
5375	030544	040440	042104	042522	ERR4: .ASCII ; ADDRESS ERROR IN 2ND PROGRAM.a;
5376	030552	051523	042440	051122	
5377	030560	051117	044440	020116	
5378	030566	047062	020104	051120	
5379	030574	043517	040522	022515	
5380	030602	100			
5381					
5382	030603	040	042447	052117	ERR5: .ASCII ; 'EOT' WASN'T RETURNED.a;
5383	030610	020047	040527	047123	
5384	030616	052047	051040	052105	
5385	030624	051125	042516	027104	
5386	030632	040045			
5387					
5388	030634	030440	052123	050040	ERR6: .ASCII ; 1ST PROGRAM WASN'T RECIRCULATED.a;
5389	030642	047522	051107	046501	
5390	030650	053440	051501	023516	
5391	030656	020124	042522	044503	
5392	030664	041522	046125	052101	
5393	030672	042105	022456	100	
5394					
5395	030677	040	040504	040524	ERR7: .ASCII ; DATA PARITY ERROR.a;
5396	030704	050040	051101	052111	
5397	030712	020131	051105	047522	
5398	030720	027122	040045		
5399					

CZ
CZ

OB
OB
OB
OF
OP
OR
OR
PA
PD
PD
PD
PD
PD
PE
PO
PO
PO
PR
PR
PR

PR
PR
PR
PR
PR
PR
PR
PR
PR
PS
PU
PU
PW
PW
QM
QU
RA
RA
RA
RE
RE

5400	030724	044440	046114	043505	ERR8: .ASCII ; ILLEGAL DATA XFER%a;
5401	030732	046101	042040	052101	
5402	030740	020101	043130	051105	
5403	030746	040045			
5404	030750	023440	054523	023516	ERR9: .ASCII ; 'SYN' DELAY 'X' TOO SHORT.%a;
5405	030756	042040	046105	054501	
5406	030764	023440	023530	052040	
5407	030772	047517	051440	047510	
5408	031000	052122	022456	100	
5409					
5410	031005	040	051447	047131	ERR10: .ASCII ; 'SYN' DELAY 'X' TOO LONG.%a;
5411	031012	020047	042504	040514	
5412	031020	020131	054047	020047	
5413	031026	047524	020117	047514	
5414	031034	043516	022456	100	
5415					
5416	031041	040	044504	047104	ERR11: .ASCII ; DIDN'T ENTER DATA MODE.%a;
5417	031046	052047	042440	052116	
5418	031054	051105	042040	052101	
5419	031062	020101	047515	042504	
5420	031070	022456	100		
5421	031073	040	051447	054124	ERR12: .ASCII ; 'STX' DIDN'T CLR DEST.%a;
5422	031100	020047	044504	047104	
5423	031106	052047	041440	051114	
5424	031114	042040	051505	027124	
5425	031122	040045			
5426	031124	040440	046440	042117	ERR13: .ASCII ; A MODULE WAS ENABLED WITH ADDR. ';
5427	031132	046125	020105	040527	
5428	031140	020123	047105	041101	
5429	031146	042514	020104	044527	
5430	031154	044124	040440	042104	
5431	031162	027122	023440		
5432	031166	023440	100		ERR13A: .ASCII ; 'a;
5433	031171	040	052105	020130	ERR14: .ASCII ; ETX DIDN'T CLR SOURCE, %a;
5434	031176	044504	047104	052047	
5435	031204	041440	051114	051440	
5436	031212	052517	041522	026105	
5437	031220	022440	100		
5438	031223	040	047505	020124	ERR15: .ASCII ; EOT WASN'T STRAPPED OUT. %a;
5439	031230	040527	047123	052047	
5440	031236	051440	051124	050101	
5441	031244	042520	020104	052517	
5442	031252	027124	022440	100	
5443	031257	040	052105	020130	ERR16: .ASCII ; ETX WASN'T RETURNED.a;
5444	031264	040527	047123	052047	
5445	031272	051040	052105	051125	
5446	031300	042516	027104	100	
5447					
5448	031305	040	042523	044522	ERR17: .ASCII ; SERIAL I/O_;
5449	031312	046101	044440	047457	
5450	031320	137			
5451					
5452	031321	040	047062	020104	ERR18: .ASCII ; 2ND PROGRAM DIDN'T ENTER DATA MODEa;
5453	031326	051120	043517	040522	
5454	031334	020115	044504	047104	
5455	031342	052047	042440	052116	

5456	031350	051105	042040	052101	
5457	031356	020101	047515	042504	
5458	031364	100			
5459					
5460	031365	040	047103	051124	ERR19: .ASCII ; CNTRL MODULE DIDN'T ENTER DATA MODE.@;
5461	031372	020114	047515	052504	
5462	031400	042514	042040	042111	
5463	031406	023516	020124	047105	
5464	031414	042524	020122	040504	
5465	031422	040524	046440	042117	
5466	031430	027105	100		
5467	031433	040	047516	042040	ERR20: .ASCII ; NO DATA RETURNED WITH EXT. SYNC.@;
5468	031440	052101	020101	042522	
5469	031446	052524	047122	042105	
5470	031454	053440	052111	020110	
5471	031462	054105	027124	051440	
5472	031470	047131	027103	100	
5473	031475	045	046111	042514	ERR21: .ASCII ;%ILLEGAL EXTERNAL CONVERSION.@;
5474	031502	040507	020114	054105	
5475	031510	042524	047122	046101	
5476	031516	041440	047117	042526	
5477	031524	051522	047511	027116	
5478	031532	100			
5479	031533	040	040504	040524	ERR22: .ASCII ; DATA FORMAT ERROR.@;
5480	031540	043040	051117	040515	
5481	031546	020124	051105	047522	
5482	031554	027122	100		
5483	031557	122	046505	052117	ERR23: .ASCII ;REMOTE CLEAR LEFT GARBAGE IN MODULE FIFO%%@;
5484	031564	020105	041440	042514	
5485	031572	051101	046040	043105	
5486	031600	020124	040507	041122	
5487	031606	043501	020105	047111	
5488	031614	046440	042117	046125	
5489	031622	020105	044506	047506	
5490	031630	022445	100		
5491	031633	103	042514	051101	ERR24: .ASCII ;CLEAR LEFT GARBAGE IN MODULE FIFO.%%@;
5492	031640	046040	043105	020124	
5493	031646	040507	041122	043501	
5494	031654	020105	047111	046440	
5495	031662	042117	046125	020105	
5496	031670	044506	047506	022456	
5497	031676	040045			
5498	031700	040052			ASTRIC: .ASCII ;*@;
5499					
5500	031702	022477	040056		QMARK: .ASCII ;?%.@;
5501					
5502	031706	022445	040056		DOT: .ASCII ;%%.@;
5503					
5504	031712	026445	100		DASH: .ASCII ;%-@;
5505					
5506	031715	045	025440	025453	SCALE: .ASCII ;% ++++++++ ++++++++ (SCALE=_;
5507	031722	025453	025453	025453	
5508	031730	025440	025453	025453	
5509	031736	025453	025453	024040	
5510	031744	041523	046101	036505	
5511	031752	137			

5512 031753 061 053115 137 X1MV: .ASCII ;1MV;
5513 031757 061 030060 053125 X100UV: .ASCII ;100UV_
5514 031764 137
5515 031765 061 052460 057526 X10UV: .ASCII ;10UV;
5516 031772 042057 053111 022451 XDIV: .ASCII ;/DIV)% a;
5517 032000 020045 100
5518
5519 032003 045 051117 047514 XLOW: .ASCII ;%ORLOW a;
5520 032010 020127 040040
5521 032014 047445 044122 043511 XHIGH: .ASCII ;%ORHIGH a;
5522 032022 020110 100
5523 032025 100 END: .ASCII ;a;
5524
5525 032026 040134 SLASH: .ASCII ;\a;
5526
5527 032030 040136 UPAROW: .ASCII ;^a;
5528
5529 032032 040045 CRLF: .ASCII ;%a;
5530
5531 032034 022445 100 CRLF2: .ASCII ;%%a;
5532
5533 032037 104 030503 040045 MESDC1: .ASCII ;DC1%a;
5534
5535 032044 041504 022462 100 MESDC2: .ASCII ;DC2%a;
5536
5537 032051 104 031503 040045 MESDC3: .ASCII ;DC3%a;
5538
5539 032056 041504 022464 100 MESDC4: .ASCII ;DC4%a;
5540
5541 032063 123 054124 040045 MESSTX: .ASCII ;STX%a;
5542
5543 032070 054523 022516 100 MESSYN: .ASCII ;SYN%a;
5544
5545 032075 123 044117 040045 MESSOH: .ASCII ;SOH%a;
5546
5547 032102 044523 040045 MESSI: .ASCII ;SI%a;
5548
5549 032106 047505 022524 100 MESEOT: .ASCII ;EOT%a;
5550
5551 032113 105 054124 040045 MESETX: .ASCII ;ETX%a;

```
5552
5553 ;*****
5554 ;SBTTL SOFTWARE 'SWITCH' ADDRESSES
5555 ;*****
5556 ;EVEN
5557 032120 000000 MTRSWH: 0
5558 032122 000000 PRTSWH: 0
5559 032124 001506 RVECTR: MONTR1 ;CONTAINS THE 'CNTRL R' RESTART ADDRESS
5560 032126 001506 AVECTR: MONTR1 ;CONTAINS THE 'CNTRL A' RESTART ADDRESS
5561 032130 003717 OFFSET: 1999. ;A/D OFFSET
5562 032132 000000 SIOSWH: 0 ;SERIAL I/O SWITCH, SET IF SERIAL INPUT USED
5563 032134 000000 MODADR: 0 ;STORAGE OF CURRENT MODULE ADDRESS
5564 032136 000000 REPTSW: 0
5565 032140 000000 KSTOR0: 0
5566 032142 000000 FORMT1: 0
5567 032144 000000 DLYSWH: 0
5568 032146 000000 DSTSWH: 0
5569 032150 000000 SENDSW: 0
5570 032152 000000 OPRTSW: 0
5571 032154 000000 TERMSW: 0
5572 032156 000000 TOPC: 0
5573 032160 000000 COUNT: .WORD 0 ;TEMPORARY COUNTER(REMOTE SER I/O).
5574 032162 000000 FROMPC: 0
5575 032164 000000 SAVEPC: 0
5576 032166 000000 SAVPSW: 0
5577 032170 000000 SAV2PC: 0
5578 032172 000000 SAV2SW: 0
5579 032174 000000 KSTOR1: 0
5580 032176 000000 KSTOR2: 0
5581 032200 000000 KSTOR3: 0
5582 032202 000000 KSTOR4: 0
5583 032204 000000 KSTOR5: 0
5584 032206 000000 TSTNUM: 0
5585 032210 000000 TEMP1: 0
5586 032212 000000 TEMP2: 0
5587 032214 000000 CHRCNT: 0
5588 032216 000000 RUBSWH: 0
5589 032220 000000 PRGSWH: 0
5590 032222 000000 LOPSWH: 0
5591 032224 000000 RESTR1: 0
5592 032226 000000 ORLOW: 0
5593 032230 000000 MINUS9: 0
5594 032300 000000 ORHIGH: 0 ;.+46
5595 032302 000000
5596
5597 032302 000000 AVGTAB: 0
5598 032614 ;.+200.
5599 001376 ;END MONTR
```

A	023346	4824	4838#														
ADCHX1	005211	1293*	1294	1302#	1310	1314*	1315										
ADCNVT-	104027	137#	1392	1434	1636	1644											
ADCTO	006710	1685	1687#	1713													
ADCT1	006716	1690#															
ADCT2	006736	1695#	1711														
ADCT3	007030	1705	1708	1712#													
ADDRES-	104026	136#	270	800	1027	1052	1166	1237	1380	1410	1568	1738	1993	2029			
		2317	2425	2482	2623	2749	3210	3249	4401								
ADDRDST	017234	1062	1089	1115	1144	1192	1749	1769	1780	1966	2075	2155	2287	2571			
		2708	2810	2912	3075	4018#	4409										
ADRSIT	005154	892	1288	1291#	2193	2722	2961	3433									
ADRSRC	017220	815	851	865	878	913	942	967	984	1003	1039	1208	1248	1274			
		1335	1688	2045	2061	2161	2298	2495	2634	2650	2685	2769	2819	2917			
		3086	3118	3138	3166	4003#											
ADSER1	014644	3625	3633#														
ADSLOP	014512	3581#	3631														
ADSLP1	014552	3587	3593#														
ADSLP2	014572	3592	3598	3606#													
ADSNxT	014626	3585	3617	3619	3621	3629#	3637										
ADTO	005022	1238#	1364														
ADT1	005024	1245#															
ADT2	005110	1255	1260	1271#													
ADT3	005142	1286#															
ADT4	005252	1289	1324#														
ADT5	005402	1327	1340	1360#													
ASEMBL	104015	127#	285	4153	4303	4833											
ASTRIC	031700	1506	4588	4618	5498#												
AVECTR	032126	334*	335*	3735	5560#												
AVERAG=	104031	139#	1435	1451	1645												
AVGDAT	016006	3839#															
AVGTAB	032302	1437	1456	1546	5597#												
BCDBIN=	104030	138#	1417	1698													
BCDEN1	022446	4644#	4669														
BCDEN2	022454	4647#	4665														
BCDEND	022536	4648	4651	4666#													
BCDERR	022554	4653	4655	4672#													
BCDIO	004606	357	1164#														
BCDIO1	004652	1184#	1228														
BCDTAB	022564	1418	1420	1699	4642	4643*	4675#										
BCDTO	003456	801#	1016														
BCDT1	003472	812#															
BCDT10	004132	947	958#														
BCDT11	004206	905	978#														
BCDT12	004244	986	997#														
BCDT13	004302	1005	1012#														
BCDT2	003610	820	825	830	835	848#											
BCDT3	003640	862#															
BCDT4	003672	875#															
BCDT5	003722	890#															
BCDT6	003732	902#															
BCDT7	004024	918	929#														
BINDEC-	104033	141#	1535	1539	1729												
BYTEA	014430	3519#															
BYTEX	014431	3520#															
CALBT1	005440	1385#															

[illegible]

DATA4	010000	1798*	1812*	1826*	1840*	1854*	1868*	1882*	1896*	1909*	1922*	1942*	1948*	1973#
DATA5	010002	1943*	1949*	1965*	1974#	2017*								
DATA6	010004	1944*	1950*	1975#										
DATSI3	007206	1795#												
DATST4	007242	1809#												
DATST5	007276	1823#												
DATST6	007332	1837#												
DATST7	007366	1851#												
DATS10	007422	1865#												
DATS11	007456	1879#												
DATS12	007512	1893#												
DATS13	007546	1906#												
DATS14	007602	1919#												
DATS15	007636	1936#												
DAT1	007102	1747#												
DAT15A	007652	1941#	1952											
DAT2	007136	1767#												
DC1	= 000021	222#	403	411	580	650	660	1301	2204	2244	2452	2521	2986	3010
		3029	3150	3188	3286	3294	3304	3328	3479	3535	3542	3547	3599	3759
		3980	4006	4108										
DC2	- 000022	223#	407	415	584	654	2208	2248	2344	2456	2523	2972	3013	3122
		3174	3290	3298	3315	3470	3539	3544	3762	4028	4131			
DC3	- 000023	224#	409	419	587	659	664	1305	2210	2250	2458	2525	2974	2988
		3012	3016	3031	3124	3152	3176	3190	3293	3303	3308	3317	3330	3472
		3482	3541	3546	3603	3765	3942	4010	4020	4022	4030	4079	4112	4133
DC4	- 000024	225#	410	3768										
DECODE	001712	310#	347											
DECOD1	001746	313	318#											
DECOUT	022650	4693	4698#											
DECPNT	022710	4686	4709#											
DEC1	022670	4699	4701	4704#										
DEC2	022702	4703	4706#											
DELAY	= 104004	118#	592	720	1209	1350	2581	2582	2771	3087	3339	3608	4099	
DELAYL	= 104020	130#	2816	2817	2818	3183	3338							
DESTIN	- 104024	134#	4025											
DIVDIT	016042	3854#	3860											
DIVIDE	016016	1425	3839	3848#										
DLYSWH	032144	251*	390*	1938*	2486*	4097	4420*	5567#						
DOT	031706	305	5502#											
DSTADR	017277	2285*	3073*	3102*	3579*	3581	3584	3629*	3630	3633	4020	4022*	4024*	4029#
		4161*												
DSTSWH	032146	2448*	2462*	2517*	2529*	3995*	4118	4127*	5568#					
ECHO	015064	3694#	3712	3721										
EMTOK	001220	157	159#											
EMTSRV	001200	102	153#											
EMTTAB	001240	161	166#											
END	032025	5523#												
END2	002234	420#	497											
ENQ	000005	232#	3179											
EOT	000004	231#	483	485	517	547	548	838	852	866	879	948	968	1075
		1101	1128	1147	1204	1261	1276	1756	1774	1785	1965	2011	2047	2113
		2130	2153	2267	2294	2389	2405	2470	2496	2537	2580	2635	2664	2706
		2772	2808	2981	3082	3127	3275	3473	3492	3521	3620	3783	3936	4077
		4090	4369	4416										
ERREXT	020620	4208	4218#											
ERRMES	017136	3925*	3930*	3960#										

ERR1	030440	622	2256	3038	5360#														
ERR10	031005	645*	684	5410#															
ERR11	031041	819	946	1254	2090	2303	2364	3091	5416#										
ERR12	031073	2262	3050	3057	5421#														
ERR13	031124	1312	3635	5426#															
ERR13A	031166	1310*	3633*	5432#															
ERR14	031171	2225	3002	5433#															
ERR15	031223	2308	3097	5438#															
ERR16	031257	2219	2996	5443#															
ERR17	031305	5448#																	
ERR18	031321	508	5452#																
ERR19	031365	447	5460#																
ERR2	030471	433	596	5366#															
ERR20	031433	988	1007	1354	5467#														
ERR21	031475	1339	5473#																
ERR22	031533	5479#																	
ERR23	031557	3203	5483#																
ERR24	031633	3159	5491#																
ERR3	030526	464	605	728	824	829	834	841	869	882	917	951	971	1215					
		1264	1279	2101	2169	2375	2830	2931	3355	5372#									
ERR4	030544	495	5375#																
ERR5	030603	488	551	855	1259	2050	2127	2402	2499	2638	2775	3343	5382#						
ERR6	030634	525	537	559	568	5388#													
ERR7	030677	469	2095	2369	5395#														
ERR8	030724	2066	2079	2133	2353	2408	2654	5400#											
ERR9	030750	644*	677	5404#															
ERTRAP	020440	97	4169#																
ETX -	000003	226#	420	588	665	2211	2216	2543	2975	2993	3309	3786	3950	4092					
		4113																	
EVECTO	014404	3451*	3495#	3745															
EXTTY	015152	3675	3678	3684	3718#	3728	3749	3751											
FDA*A	014072	3284	3314	3320#															
FLAB1	014042	3270*	3295#	3646*															
FLAB17	014565	3600#																	
FLAB2	014037	3292#	3647*																
FLAB3	014067	3316#	3648*																
FLAB3A	014066	3315#																	
FLAB4	014363	3480#	3649*																
FLAB5	014113	3329#	3650*																
FLAB5A	014112	3328#	3594																
FLAB6	014351	3471#	3651*																
FLAB7	013660	3239#	3251*	3252	3270	3652*													
FLO	013670	3247#	3426	3505															
FLOP	013674	3248#																	
FLOPB	013704	3252#																	
FLUSH	002050	328	338#	343															
FND1B	014140	3352#	3371																
FND1C	014134	3341	3351#	3379															
FND1D	014152	3353	3370#																
FND3A	014302	3425	3432#																
FND5	014334	3461#																	
FND5A	014342	3467#	3478																
FND5B	014370	3462	3487#	3493															
FND5C	014354	3469	3476#																
FNORM	014060	3281	3312#																
FORMT1	032142	3794*	3795	3799*	3807*	4547*	5566#												

FOUNDL	014014	3280#	3414															
FOUND2	014212	3356	3373	3375	3387#													
FOUND3	014270	3390	3424#															
FOUND5	014312	3388	3442#															
FOUND6	014406	3395	3451	3501#														
FOUNSW	013662	3243#	3247*	3387	3413*													
FPROG	014030	3285#	3490	3590														
FROMPC	032162	4171*	4179*	4181	5574#													
FSTUF	014660	3255	3410	3582	3646#													
FTST	014116	3323	3327	3338#														
GANEXT	006662	1648	1651	1654	1673#													
GETDAT	015730	3824#	3838															
GETREG=	104003	117#	1670	1714	1730	3718	3840	3869	4100	4140	4332	4478	4516	4577				
		4670	4696															
HEADER	023415	302	4851#															
HIDIVD	016116	1422*	3819*	3836*	3852	3874#												
HIDIVR	016112	1421*	3850	3872#														
HIGH	016124	3821*	3825	3827	3829*	3877#												
IADRS0	002214	271*	404#	581	585	4157												
IADRS1	002220	272*	408#															
IADRS2	002224	273*	412#															
IADRS3	002230	274*	416#															
IADRS4	003120	275*	651#															
IADRS5	003124	276*	655#															
IADRS6	003132	277*	661#															
IADRS7	017547	278*	4109#															
IADRS8	017643	279*	4124	4132#														
IADRS9	017202	280*	3256	3257	3258	3259	3989#											
IADR10	011167	281*	2345#															
IADR11	014046	3256*	3302#															
IADR12	014032	3257*	3287#															
IADR13	014036	3259*	3291#															
IADR14	014051	3258*	3305#															
ICOUNT	020770	4234	4257#	4353*														
INBUF	015330	268	310*	312	320	1170	1695*	1696*	2000	3665	3755#	4191	4193	4399				
		4596	4641	4792														
INPUTA	014740	3666#	3667	3695	3704													
INPUTB	015030	3685#	3723															
INPUTC	015006	3673	3679#															
KEYAD1	011623	2483*	2522#															
KEYAD2	011625	2484*	2524#															
KEYTO	011524	2485#	2595															
KEYT1	011532	2493#																
KEYT2	011554	2497	2507#	2538	2552													
KEYT3	011734	2544	2562#															
KEYT4	012036	2591#																
KSTOR0	032140	4395*	4402*	4406	5565#													
KSTOR1	032174	1420*	1423	1429	1448	4306	4311*	4312*	4313	4314*	4317	5579#						
KSTOR2	032176	1444*	1465	5580#														
KSTOR3	032200	1446*	1469	1474	4204*	4212*	4214	5581#										
KSTOR4	032202	1428*	1447	1463	5582#													
KSTOR5	032204	1429*	1430*	1453	5583#													
LDCHRO	104006	120#	442	482	503	516	532	546	563	617	1146	1750	1755	2112				
		2213	2266	2388	2468	2535	2573	2575	2579	3491	3971	4368	4415	4605				
LDPGMO-	104007	121#	399	458	590	646	1064	1090	1117	1193	1770	1781	1968	2085				
		2114	2157	2201	2229	2240	2289	2341	2359	2390	2709	2812	2914	2969				

		2977	2983	3007	3025	3077	3119	3147	3170	3185	3282	3312	3320	3325
		3467	3476	3487	3516	3531	3589	3593	3596	3984	4105	4128	4624	
		1424*	3820*	3835*	3851	3873#								
		1423*	3823*	3849	3871#									
		388*	426	429*	736*	792*	1570*	1659	1661*	2034*	2760*	5590#		
		1436	1668	3822*	3830	3832	3834*	3875#						
		4502*	4508	4513*	4518#									
		327#	330											
		3579#												
		4205*	4216#											
		3761	5533#											
		3764	5535#											
		3767	5537#											
		3770	5539#											
		3785	4470	5549#										
		3788	5551#											
		3782	5547#											
		3779	5545#											
		3773	5541#											
		3776	5543#											
		266	4898#											
		386	4907#											
		932	1329	2279	3067	3407	4949#							
		1236	4956#											
		*379	4960#											
		4251	4964#											
		2278	4967#											
		2316	4974#											
		1409	4981#											
		1415	4985#											
		4189	4987#											
		1567	4990#											
		1576	4995#											
		3925	3960	4913#										
		1578	1595	5000#										
		1586	1603	5003#										
		5009#												
		5013#												
		1594	5016#											
		5021#												
		1611	5025#											
		1640	5029#											
		1663	5033#											
		4743	5037#											
		799	5046#											
		2028	4920#											
		4152	5052#											
		802	907	934	961	5055#								
		803	5058#											
		5062#												
		962	5067#											
		908	5072#											
		935	5076#											
		1026	5078#											
		1051	5083#											
		1079	1105	1131	5088#									

[illegible]

MES84	030152	3114	5327#													
MES85	030236	3115	5336#													
MES86	030304	5343#														
MES87	030332	5347#														
MES88	030362	3099	5351#													
MES89	030415	4828	5356#													
MES9	024275	4544	4946#													
MES90	030424	4832	5358#													
MINUS9	032230	1476*	1487	5593#												
MODADR	032134	821	937*	1294	2283*	2320	3071*	3100*	3211	3584	4159*	5563#				
MODERR-	104022	132#	432	446	463	468	487	494	507	524	536	550	558	567		
		595	604	621	676	683	727	818	823	828	833	840	854	868		
		881	916	945	950	970	987	1006	1214	1253	1258	1263	1278	1311		
		1338	1353	2049	2065	2078	2089	2094	2100	2126	2132	2168	2218	2224		
		2255	2261	2302	2307	2352	2363	2368	2374	2401	2407	2498	2637	2653		
		2774	2829	2930	2995	3001	3037	3049	3056	3090	3096	3158	3202	3342		
		3354	3634													
MONITR	001376	110	238#	2418	3731	3961	4182	4567	4600	4674	4744	5599				
MONTR1	001506	264#	303	5559	5560											
MONTR2	001630	289#	297													
MONTR3	001660	287	295	298#												
MONTR4	001674	301#														
MONTR5	001700	255	303#													
MONT1A	001612	269	283#													
MOVE IT	022042	4504#	4515													
MTRSWH	032120	254	256*	5557#												
M377A1	012442	2759#														
M381E1	004332	1031#														
M381E2	004336	1033#	1043	1045												
M385A1	010134	2030#	2321													
M7377A	012370	371	2747#													
M7377B	012376	2750#	3214													
M7378A	013664	372	3244#													
M7380A	002160	353	385#	793												
M7381A	003450	354	798#													
M7381E	004316	355	1025#													
M7382A	004412	356	1050#													
M7383A	005014	358	1235#													
M7383C	005416	359	1378#													
M7383G	006356	361	1566#	1620												
M7383R	005514	360	1408#													
M7384A	007072	362	1736#													
M7384E	010006	363	1991#													
M7385A	010126	364	2027#	2609												
M7385I	011150	365	2331#													
M7385T	011360	366	2423#													
M7386A	011506	367	2480#													
M7387A	003254	368	711#													
M7388A	012050	369	2606#													
M7388F	012062	370	2620#	2732												
NMATCH	002070	315	340	345#												
NODLAY=	104036	144#	633	1384	1413	2509	2567	4589	4619							
NOP	- 000240	46#														
NULL	- 104021	131#	248	250	4741											
NULL1 =	104023	133#	668	680	3565											
OBCDT1	004422	1060#														

OBCDT2	004462	1087#																
OBCDT3	004522	1113#																
OBCDT4	004564	1140#																
OFFSET	032130	1703	1724	5561#														
OPRTSW	032152	3750*	4459*	4477*	4480	5570#												
ORHIGH	032300	1461	1467*	1492	1508	1530	1538	5595#										
ORLOW	032226	1459	1471*	1529	1534	5592#												
PARITY	016220	466	2092	2366	3892*	3902#	3935*											
PDMPT	015650	3760	3763	3766	3769	3772	3775	3778	3781	3784	3787	3803#						
PDMSET	015434	2542	2663	3759#	4139	4549	4604											
PDMST0	015632	3793	3797#															
PDMST1	015644	3790	3801#															
PDMST2	015646	3796	3802#															
PEXT2	015674	3803	3810#															
POP1SP=	005726	43#	332	3808														
POP2SP=	022626	45#	246	4170														
POSTIT	007040	1548	1722#															
PRCNTR=	104000	114#	3724	3801	4599													
PRGM1	002213	400	403#	425	521	555												
PRGM2	002222	410#	435	484	527	561												
PRGSWH	032220	5589#																
PRINT =	104012	124#	257	265	283	301	304	345	385	712	733	790	798	801				
		906	931	933	960	980	999	1014	1025	1031	1050	1078	1104	1130				
		1142	1164	1167	1226	1235	1328	1341	1362	1378	1385	1408	1414	1498				
		1505	1524	1532	1536	1540	1566	1575	1577	1585	1593	1602	1610	1618				
		1639	1662	1664	1736	1752	1757	1776	1787	1800	1814	1828	1842	1856				
		1870	1884	1898	1911	1924	1939	1991	1995	2027	2277	2315	2318	2331				
		2416	2423	2480	2549	2564	2593	2606	2620	2657	2676	2711	2730	2747				
		2895	3066	3098	3113	3212	3244	3406	3443	3503	3688	3708	3715	3738				
		3752	3797	3805	3959	4151	4172	4177	4188	4215	4250	4301	4308	4382				
		4396	4469	4539	4543	4587	4617	4672	4742	4827	4831							
PROMD	003427	731	780#															
PROMS	003360	724	741#															
PRTBF1	022162	4539#	4566															
PRTBF2	022166	4541#	4576															
PRTEXT	021734	4466	4471	4476#														
PRTOCT=	104014	126#	4175	4180	4209	4213	4252	4829										
PRTRBF=	104037	145#	1044	1397	2686													
PRTSWH	032122	252*	3674	3683	3699	3746	438	4384*	4460*	4476*	5558#							
PRT1A	022210	4548#	4551															
PRT1B	022222	4545	4552#															
PSW	001340	200#																
PUSH1S=	005746	42#																
PUSH2S=	024646	44#																
PWRFAL	022722	99	4720#															
PWRUP	022756	4728	4730#															
QMARK	031702	346	3753	4309	5500#													
QUOENT	016122	1428	1442	1452	1646	3868*	3876#											
RANA	021134	4270*	4271*	4272*	4273	4276	4292#											
RANB	021136	4270	4273*	4274*	4275*	4277	4293#											
RANC	021140	4271	4274	4276*	4277*	4278*	4279	4294#										
RANDOM=	104011	123#	457	577	2082	2152	2340	2356	2807	2904	3274							
RBUFO	001360	211#	3898	3922														
RSRO	001356	210#	288	3899*	3957*	4203*												
RECBFO	016234	836	866	919	948	968	985	1004	1211	1251	1261	1276	1514	1637				
		1692	1693	2087	2096	2259	3047	3054	3583*	3887	3890	3894*	3900	3907				

		3908#	4558	4575										
RECBUF	022224	374	4558#											
REDC3	016222	3895*	3903#	3944*										
RECECD	016736	3910#	3926											
RECEOT	016224	674	681	816	1040	1256	1690	2124	2162	2178	2305	2399	2820	2839
		2918	2942	3093	3340	3377	3893*	3904#	3938*					
RECERR	017124	3927	3932	3957#										
RECETX	016230	3897*	3906#	3952*										
RECEXT	017112	3939	3941	3945	3949	3953#								
RECSTX	016226	2253	3035	3896*	3905#	3948*								
RECVET	016740	298	3919#											
RECVPT	016232	3888*	3907#	3921	3953*									
RECVRO=	104005	119#	638	714	911	940	965	1038	1207	1296	1333	1687	2084	2239
		2286	2296	2358	2444	2516	2570	2968	3024	3074	3084	3140	3163	3277
		3607	3744	4229	4355	4591	4623							
RECVR1	017026	3929	3936#											
RECYCL	001754	320#	326	342										
REMAIN	016126	1426	3862*	3878#										
REMOTE	021442	1381	1411	4394#										
REPTSW	032136	3662*	4120	4365	4406*	4407	4413	4417*	5564#					
REPTO	005636	1432#	1543											
REPTOA	005532	1414#	1419	1427										
REPT1	005646	1434#												
REPT10	006236	1517*	1520*	1523*	1526#									
REPT11	006246	1529#												
REPT12	006304	1531	1540#											
REPT13	006320	1528	1545#											
REPT14	006330	1547#	1554	1669										
REPT2	005736	1451#	1455											
REPT3	005762	1450	1458#											
REPT4	006010	1465#	1479											
REPT5	006054	1468	1472	1477#										
REPT6	006066	1487#	1493	1495										
REPT6A	006102	1492#												
REPT6B	006110	1494#	1509											
REPT7	006116	1490	1498#											
REPT8	006122	1500#	1511											
REPT8A	006144	1508#	1513											
REPT9	006162	1496	1514#											
RESTR1	032224	314	317	333*	4352*	5591#								
RETURN	020774	910*	939*	964*	1331*	2083*	2282*	2357*	2568*	2707*	3070*	4241	4255*	4259#
		4316*	4349*											
RINTO	001366	214#	298*											
RLVLO	001370	215#	300*											
RPT12A	006314	1543#												
RUBSWH	032216	3664*	3685	3687*	3706	3711*	5588#							
RUN	022360	377	4617#	4631										
RVECTR	032124	303*	3741	4350*	4351*	4352	5559#							
SAVEPC	032164	4750*	4762	4768*	4780	5575#								
SAVPSW	032166	4751*	4761	4769*	4779	5576#								
SAVREG=	104002	116#	1666	1682	1722	3661	3818	3848	4061	4137	4326	4458	4498	4574
		4640	4684											
SAV2PC	032170	4752*	4760	4770*	4778	5577#								
SAV2SW	032172	4753*	4759	4771*	4777	5578#								
SCALE	031715	1525	5506#											
SCOPE =	104001	115#	456	481	515	545	576	616	632	709	789	848	862	875

		890	902	929	958	978	997	1012	1087	1113	1140	1271	1286	1324
		1360	1767	1795	1809	1823	1837	1851	1865	1879	1893	1906	1919	1936
		2059	2074	2111	2148	2187	2200	2274	2314	2386	2414	2507	2562	2591
		2647	2674	2696	2720	2728	2767	2779	2898	2955	2967	3063	3110	3205
		3269	3404	3432	3442	3501								
SCOPEB	020676	4231	4237#											
SCOPEF	020772	4234	4236*	4242*	4258#	4315*								
SCOPEG	020714	4233	4235	4242#										
SD10	010764	2176	2192	2274#										
SD10A	013252	2960	2997	3003	3055	3063#								
SD11	011122	2304	2314#											
SD11A	013402	3092	3098#											
SD2	010220	2059#												
SD3	010246	2064	2074#											
SD4	010364	2111#												
SD4A	012502	2773	2779#											
SD5	010434	2052	2148#											
SD5A	012662	2844	2895#											
SD6	010574	2170	2174	2187#										
SD6A	013034	2932	2938	2945#										
SD6B	013036	2940	2955#											
SD7	010624	2200#												
SD7A	013066	2967#												
SEND	022264	376	4586#											
SENDAX	014422	3516#	3606											
SENDPG	014434	3531#												
SENDSW	032150	253*	2679*	2683*	3672	3792	3958*	4590*	5569#					
SEND1	022312	4594#	4607											
SEND2	022354	4602*	4606#											
SETRMT	021502	1395	1480	4406#										
SETUP =	104035	143#	389	711	805	1028	1053	1175	1238	1382	1412	1569	1739	1994
		2035	2333	2428	2485	2624	2759	3111	3248	4586				
SI -	000017	230#	586	656	2346	3780	4024							
SIOSWH	032132	264*	282*	423	490	519	553	599	1291	2051	2173	2191	2465	2532
		2547	2834	2843	2937	2959	3280	3322	3372	3394	3424	3461	3586	3940
		3969	3982	4018	4155	4626	5562#							
SLASH	032026	3689	3709	5525#										
SNDAX1	014470	3518	3533	3556#										
SNGCHR	017664	4038*	4088	4094*	4142#									
SOH -	000001	229#	405	413	582	652	662	1303	2206	2246	2454	3288	3296	3306
		3481	3537	3549	3601	3777	3990	4008	4110					
SOH1	017231	814*	850*	864*	877*	912*	941*	966*	983*	1002*	1036*	1037*	1172*	1174*
		1247*	1273*	1334*	1391*	1704	2044*	2060*	2160*	2297*	2633*	2649*	2684*	2768*
		2815*	2899*	3117*	3164*	3279*	4009#	4193*						
SOURCE-	104025	135#	1297	2449	2518	4003								
SPACE =	104016	128#	1502	1553	4211									
SPACEX	021262	1500*	1552*	4329*	4331*	4334#	4426*	4431*	4435*					
SPCHR1	015070	3680	3682	3699#										
SPCHR3	015140	3702	3713#											
SPCHR5	015156	3714	3720#											
SRCADR	017227	938*	1332*	2284*	3072*	3101*	3137*	3165*	4007#	4160*	4403*			
STADR7	010637	2030*	2205#											
STADR8	010643	2031*	2209#											
STADR9	010723	2032*	2245#											
STAD10	010727	2033*	2249#											
STDR10	013170	2753*	3015#											

[illegible]

TAG1L	010476	2160#						
TAG1P	011134	2276	2318#					
TAG1PC	013642	3065	3210#					
TAG1PD	013644	2845	3211#					
TAG1Q	011010	2282#						
TAG1QA	013274	3070#						
TAG1R	011046	2289#						
TAG1RA	013332	3077#						
TAG1S	010712	2231	2239#					
TAG1SA	013156	3007#						
TAG1SB	013176	3009	3024#					
TAG1T	010732	2242	2253#					
TAG1TA	013212	3027	3035#					
TAG1U	011060	2291	2296#					
TAG1UA	013344	3079	3084#					
TAG1W	010760	2220	2226	2257	2266#			
TAG1WA	013240	3039	3054#					
TAG1Z	011074	2300#						
TAG1ZA	013354	3088#						
TAG2A	005042	1250#						
TAG2B	005126	1276#						
TAG2C	005170	1294#	1316					
TAG2F	005234	1295	1309	1314#				
TAG2G	005216	1299	1308#					
TAG2H	005250	1292	1317#					
TAG3A	004452	1066	1078#					
TAG3B	004512	1092	1104#					
TAG3C	004552	1119	1130#					
TAG3D	004574	1144#	1148					
TAG4A	004700	1192#	1222					
TAG4B	004724	1195	1207#					
TAG4C	004744	1212#	1218					
TAG4D	004764	1220#	1225					
TAG4E	005004	1216	1225#					
TAG4F	010016	1995#	1999	2006				
TAG4G	010074	2013#	2019					
TAG6A	011630	2520	2529#	2546	2548			
TAG6B	011654	2533	2534*	2536#				
TAG6C	011702	2541	2543#					
TAG6D	012002	2569*	2575#	2578	2583*	2584		
TAG6E	011764	2570#	2585					
TAG7A	011450	2451	2462#	2472				
TAG7B	011474	2466	2467*	2469#				
TAG8A	012166	2655	2660#	2665				
TAG88A	012270	2700#	2705					
TEMP1	032210	4528*	4531*	4727*	4733	5585#		
TEMP2	032212	4524*	4526*	4529*	5586#			
TERMSW	032154	3973*	3994*	4102	4104*	4115	4117*	5571#
TEST1	011156	2340#						
TEST2	011274	2354	2365	2370	2376	2386#		
TEST3	011344	2414#						
TG1H	012540	2812#						
TG1KA	013106	2971	2977#					
TG1L	012544	2815#						
TG1LA	013116	2979	2983#					
TG1PA	013474	3121	3137#					

[illegible]

[illegible]

I 13
CZPMAC0 PDM70 DIAGNOSTIC TEST MACY11 30A(1052) 20-JAN-78 09:11 PAGE 145
CZPMAC.P11 19-JAN-78 14:50 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0164

CTX	112#	456	481	515	545	576	616	632	709	789
SIO	112#									
TA	112#									
TS	111#	456	481	515	545	576	616	632	709	789

. ABS. 032614 000

ERRORS DETECTED: 0

CZPMAC.BIN,CZPMAC.LST/CRF/SOL/NL:TOC=CZPMAC.P11
RUN-TIME: 5 13 2 SECONDS
RUN-TIME RATIO: 113/21=5.3
CORE USED: 13K (25 PAGES)