

DMC11

ITEP OVERLAY
MD-11-DZDMO-A

EP-DZDMO-A-DL-A

AUG 1977

COPYRIGHT © 1977

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains a grid of frames. The leftmost column contains frames with text labels, including 'GENERAL INFORMATION', 'OPERATIONAL PROCEDURES', 'MAINTENANCE', 'SAFETY', 'IDENTIFICATION', 'REVISIONS', 'REFERENCES', 'GLOSSARY', 'INDEX', and 'APPENDICES'. The remaining frames in each row contain technical data, likely tables or diagrams, related to the corresponding label. The data is presented in a structured, tabular format with multiple columns and rows of text.

B01

EOFIDENJACSEQ 00010000 770720 PDP10 411 HDR1DZDMORSEQ 00010000 770720
DMC11PDP10B OVERLAY0001 MACY11-30(1046) 11-JUL-77 12:39 PAGE 2
DZDMO.P11 18-MAY-77 11:28

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDMO-A-D
PRODUCT NAME: DMC11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM
PROGRAM DATE: MAY 1977
MAINTAINER: DIAGNOSTICS
AUTHORS: R A JONES
JOHN EGOLF
FAY BASHAW

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

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

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

COPYRIGHT (C) 1977 BY DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT.

THIS PROGRAM IS DESIGNED AS A MAINTENANCE AID FOR FIELD SERVICE PERSONEL. IT WILL VERIFY THE PROPER OPERATION OF A COMPLETE COMMUNICATION LINK FROM ONE PDP-11 SYSTEM TO ANOTHER OR TO A COMMUNICATION TEST CENTER.

THIS PROGRAM MUST BE USED IN CONJUNCTION WITH THE INTERPROCESSOR TEST PROGRAM(DZITP) ON A PDP-11 SYSTEM WITH A DL-11 INTERFACE.

2.0 REQUIREMENTS.

2.1 EQUIPMENT

- A. PDP-11 SYSTEM WITH 4K OF CORE.
- B. A DMC11 COMMUNICATION INTERFACE.

2.2 STORAGE.

4K OF CORE

3.0 LOADING PROCEDURE

THIS PROGRAM IS IN ABSOLUTE FORMAT.
THE ABS LOADER MUST BE USED TO LOAD THE PROGRAM.

4.0 OPERATING PROCEDURES.

- A. TWO METHODS OF ENTERING PARAMETERS ARE PROVIDED
 - 1. LOAD ADDRESS 200 AND START TO ENTER PARAMS FROM CONSOLE TTY, PROCEED TO SECTION B.
 - 2. LOAD ADDRESS 200 AND SET SWITCH REGISTER BIT 15 BEFORE STARTING TO ENTER PARAMS FROM CONSOLE SWITCHES, PROCEED TO SECTION C.

*THE PROGRAM MAY BE RESTARTED AT LOC 204 (ONCE PARAMETERS HAVE ALREADY BEEN SELECTED)

- B. CONSOLE DIALOGUE PARAMETER INPUT (CURRENT VALUES FOR PARAMETERS ARE FOUND IN OVERLAY)

- 1. THE PROGRAM WILL TYPEOUT THE NAME OF THE VARIABLE OVERLAY.
 - A. IF YOU WISH TO SETUP JUST THE INDICATED OVERLAY, TYPE A CARAGE RETURN
 - B. IF YOU WISH TO SETUP A DN11, TYPE IN DN.
 - C. IF YOU WISH TO SETUP A DM11BB, TYPE IN DMB.

IF DN OR DMB WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS VECTOR ETC. REFERED TO IN STEPS 2 THRU 7, PERTAIN TO THE DN11 OR DM11BB.

- 2. THE PROGRAM WILL TYPE THE DEFAULT BUS ADDRESS OF THE INTERFACE UNDER TEST.
 - A. TYPE A CAR. RETURN TO USE DEFAULT BUS ADDRESS
 - B. TYPEIN ACTUAL BUS ADDRESS
- 3. THE PROGRAM WILL TYPE OUT THE DEFAULT VECTOR ADDRESS
 - A. TYPE A CAR. RETURN TO USE DEFAULT ADDRESS
 - B. TYPEIN ACTUAL VECTOR ADDRESS
- 4. THE PROGRAM WILL TYPE OUT THE DEFAULT INTERFACE PRIORITY
 - NOTE: 200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.
 - A. TYPE A CAR. RETURN TO USE DEFAULT VALUE

- B. TYPEIN ACTUAL VALUE
5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
IF REQUIRED BY THE ISR. (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
 - A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
 - B. TYPEIN ACTUAL VALUE
 6. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#2
IF REQUIRED BY THE ISR.
 - A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
 - B. ENTER ACTUAL VALUE
 7. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#3
IF REQUIRED BY THE OVERLAY.
 - A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
THE DN-11 WILL USE PARAM #3 AS THE # TO DIAL.
IF USING A MODEM WITHOUT AUTOMATIC HANDSHAKING,
THE NUMBER MUST TERMINATE WITH A
"END-OF-NUMBER" CHARACTER (:).
 - B. ENTER ACTUAL VALUE.
 8. THE PROGRAM WILL RETURN TO STEP B1 IF THIS SETUP
WAS FOR DN11 OR DN11BB.
 9. THE PROGRAM WILL REQUEST THAT SWITCH REGISTER BE SET.
 - A. SETUP SWITCH REGISTER AS SPECIFIED IN STEP D.
AND TYPE A CAR. RETURN.

NOTE: IF ANY OF THE ABOVE ITEMS 2 THRU 7 WERE CHANGED BY ENTERING
NEW VALUES, THE NEW VALUE BECOMES THE DEFAULT VALUE FOR SUBSEQUENT
RESTARTS OF THE PROGRAM.

- C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
1. THE PROGRAM HALTS FOR ISR (INTERFACE SERVICE ROUTINE) SPECIFICATION
 SWR14=SETUP DN-11B ISR
 SWR13=SETUP DN-11 ISR
 SWR=00000=SETUP VARIABLE ISR
 2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
 SETUP SEQUENCE IS: DN11, DN11-BB THEN VARIABLE OVERLAY. (EACH ENTRY SET SWITCHES THEN HIT CONTINUE)
 - A. HALT FOR BUS ADDRESS OF INTERFACE
 - B. HALT FOR VECTOR ADDRESS OF INTERFACE
 - C. HALT FOR PRIORITY OF INTERFACE
 - D. HALT FOR INTERFACE PARAM #1 (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
 - E. HALT FOR INTERFACE PARAM #2 (DN11 AND DNBB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MON)
 - F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DNBB.
 3. HALT FOR OPERATIONAL SWITCH SETTINGS. (SEE STEP D.)
 - A. PRESS CONTINUE TO START TESTING

BEFORE ATTEMPTING TO RUN THIS PROGRAM, THE OPERATOR MUST ACCERTAIN THE COMPLETE COMMUNICATION LOOP AND PROCEDURES TO BE USED, INCLUDING THE TYPE OF MODEMS, THE TYPE OF INTERFACE BEING USED AT THE OTHER CPU AND THE MODES OF OPERATION, DATA AND PARAMETERS TO BE USED AT EACH CPU.

THIS WILL REQUIRED VOCAL COMMUNICATION WITH THE OPERATOR AT THE OTHER CPU UNLESS ITS CONFIGURATION AND OPERATION ARE FIXED AS A TEST CENTER.

AFTER DETERMINING THAT THE EQUIPMENTS ARE COMPATIBLE AND AGREEING ON THE MODE AND VARIABLE PARAMETERS TO BE USED, THE SYSTEM WHICH IS TO RECEIVE DATA FIRST SHOULD BE LOADED AND STARTED. IF THE MODEM BEING USED ON THIS SYSTEM HAS AN AUTOMATIC ANSWER FEATURE, IT SHOULD BE ENABLED.

THE SYSTEM WHICH IS TO TRANSMIT FIRST SHOULD THEN BE LOADED AND STARTED AND THE CONNECTION ESTABLISHED EITHER MANUALLY OR AUTOMATICALLY (VIA DN-11).

D. OPERATIONAL SWITCH SETTINGS.

SW15=1 HALT ON ERROR
SW14=1 SINGLE PASS
SW14 HAS NO EFFECT IF SW04=0
SW13=1 INHIBIT ERROR TYPEOUTS
SW12=1 INHIBIT ALL TYPEOUTS EXCEPT ERRORS
IF SW12=0 AND SW04=1 END PASS IS TYPED
AND TRANSMITTED/RECEIVED DATA IS TYPED.
SW11=1 USE PREVIOUSLY SPECIFIED DATA
SW10=1 DATA SELECT (WITH SW09)
SW09=1 DATA SELECT (WITH SW10)
00=1 GET DATA FROM OPERATOR
01=1 TEST MESSAGE #1 (SA QUICK BROWN FOX)
10=1 TEST MESSAGE #2 (SB NUMERICS)
11=1 TEST MESSAGE #3 (SC CONTEST/QUICK BROWN FOX/NUMERICS)
SW08=1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)
SW07=1 DO NOT TEST RECEIVED DATA
SW06=1 MONITOR TRANSMITTED DATA ON CONSOLE TTY.*
SW05=1 MONITOR RECEIVED DATA ON CONSOLE TTY.*
* IN MANY CASES, NOT ALL DATA WILL APPEAR ON THE CONSOLE
TTY. THIS IS ESPECIALLY TRUE WHEN THE COMM INTERFACE IS
RUNNING AT A FASTER BAUD THAN THE CONSOLE, BUT EVEN AT EQUAL
OR SLOWER BAUDS, ALL CHARACTERS MAY NOT APPEAR ON THE CONSOLE.
SW04=1 RETURN TO MONITOR FOR END PASS
WHEN SW04=0 PROGRAM LOOPS IN THE OVERLAY NEVER RETURNING TO THE MONITOR.
SW03=1 INTERNAL LOOPBACK MODE
SW02=1 EXTERNAL LOOPBACK MODE
SW01=1 ONE-WAY-IN MODE
SW00=1 ONE-WAY-OUT MODE

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3. ↑ TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

IF OPERATOR SPECIFIED DATA WAS INDICATED, THE PROGRAM WILL TYPE A REQUEST FOR THE DATA. DATA MAY BE ENTERED AS ASCII CHARACTERS OR OCTAL CODE. TYPE IN THE DATA TERMINATED WITH A CR. OCTAL CODE MAY BE ENTERED BY TYPING AN ↑(UP ARROW) FOLLOWED BY THE OCTAL CODE (IN THE RANGE 000 TO 377) SEPERATED BY SPACES AND TERMINATED BY ↑(UP ARROW).
I.E. ABCD↑ 000 123 377↑ EFG (CAR.RETURN)

A TYPICAL SWITCH SETTING FOR HALF-DUPLEX=003150 THIS SETTING USES INTERNAL LOOPBACK MODE, LOOPS IN OVERLAY, MONITORS TRANSMITTED AND RECEIVED DATA ON THE CONSOLE TTY, AND TESTS RECEIVED DATA USING TEST MESSAGE #3.

A TYPICAL SWITCH SETTING FOR FULL-DUPLEX=003144 THIS SETTING IS THE SAME AS ABOVE EXCEPT IT USES THE EXTERNAL LOOPBACK MODE.

ALL STANDARD MESSAGES(TEST MESSAGES 1-3) ARE PRECEDED BY 2 FILL CHARACTERS(177), AND ARE FOLLOWED BY A CR(015), LF(012), RECEIVE TERMINATING CHARACTER(001), 4 FILLS(177), AND A TRANSMIT TERMINATING CHARACTER(000). DURING TRANSMISSION, WHEN A 000 CHARACTER IS SEEN THE TRANSMISSION IS STOPPED.DURING RECEPTION, WHEN A 001 CHARACTER IS RECEIVED, THE RECEIVER IS SHUT OFF. IF THE MESSAGE WAS INPUTED BY THE OPERATER, THE TERMINATING CHARACTERS ARE ADDED.

TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY <001>)
2. VERIFIES THE DATA AGAINST THE DATA SELECTED BY SW09 AND SW10 (SW7=0)
3. TRANSMIT THE DATA SELECTED BY SW09 AND SW10 (SW8=0) OR
TRANSMIT THE RECEIVED DATA (SW8=1)
4. RETURNS TO MONITOR FOR "END PASS" (SW4=1) OR
GO TO STEP 1. (SW4=0)

EXTERNAL LOOPBACK MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAIT FOR CLEAR TO SEND
3. TRANSMITS THE SELECTED DATA
4. RESETS REQUEST TO SEND
5. WAIT FOR MESSAGE TO BE RECEIVED
6. VERIFIES THE DATA (SW07=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR
GO TO STEP 1 (SW04=0)

ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA (SW07=0)
3. RETURNS TO MONITOR FOR "END PASS" (SW04=1) OR
GO TO STEP 1 (SW04=0)

ONE-WAY-OUT MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAITS FOR CLEAR TO SEND
3. TRANSMITS SELECTED DATA
4. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR
GO TO STEP 1 (SW04=0)

- E. THE OVERLAY IS THEN ENTERED AND A CONNECTION ESTABLISHED EITHER
MANUALLY OR AUTOMATICALLY.

IF ONE-WAY-IN OR INTERNAL LOOPBACK MODES ARE SELECTED.
THE OVERLAY WILL SET DATA TERMINAL READY AND WAIT FOR DATA.

IF ONE-WAY-OUT OR EXTERNAL LOOPBACK MODES WERE SELECTED.
THE OVERLAY WILL SET DATA TERMINAL READY AND REQUEST TO SEND.
THE OVERLAY WILL THEN WAIT FOR CLEAR TO SEND BEFORE ATTEMPTING TO
TRANSMIT DATA.

THE PROGRAM WILL PRINTOUT A "WAITING FOR CLEAR TO SEND"
MESSAGE AND THE CONTENTS OF THE XMIT CSR EVERY 60 SECS.
UNTIL CLEAR TO SEND IS ASSERTED.

F. IF SW04=0 THE OVERLAY WILL CONTINUE TO TRANSMIT/RECEIVE DATA.

IF SW04=1 THE OVERLAY WILL RETURN TO THE MONITOR AND TYPE "END PASS".

IF BOTH SW04=1 AND SW14=1, THE PROGRAM WILL REQUEST NEW INTERFACE PARAMS AFTER ONE PASS OF THE SELECTED TEST MODE.

TEST EXECUTION MAY BE INTERRUPTED BY TYPING THE FOLLOWING CHARACTERS ON THE CONSOLE TTY.
LINE FEED = RESTART PROGRAM AT LOCATION 200.
QUESTION MARK = PRINTOUT FIRST 8 WORDS OF INPUT BUFFER.(ASCII)

THEN TYPE EITHER:

#WXXXXXX TO PRINTOUT THE 8 WORDS AT LOC XXXXXX.

#BXXXXXX TO PRINTOUT THE 16 BYTES AFTER LOC XXXXXX.

#C TO CONTINUE

PROGRAM MUST BE RESTARTED AT 200 AFTER PRINTING.
CARRIAGE RETURN = RESTART AT REQUEST FOR NEW OPERATIONAL SWITCHES.

5.0 PROGRAM AND/OR OPERATOR ACTION

IF THE OPERATOR WISHES TO MANUALLY EXAMINE THE TRANSMIT OR RECEIVE BUFFERS, DO THE FOLLOWING: TO FIND THE STARTING ADDRESS OF THE RECEIVE BUFFER, LOAD ADDRESS 11020 AND EXAMINE. TO FIND THE STARTING ADDRESS OF THE TRANSMIT BUFFER, LOAD ADDRESS 11022 AND EXAMINE.

5.1 NORMAL HALTS SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

THE ONLY ERROR REPORT FROM THE CONTROL PROGRAM OCCURS IF THE INTERFACE SPECIFIED IS NOT LOADED.

IF DATA IS RECEIVED AND SWITCH 7 (NO DATA COMPARE) IS RESET, THE DATA WILL BE COMPARED AGAINST THE PRESELECTED DATA AFTER A LINE FEED CHARACTER IS RECEIVED. IF THERE IS A MISMATCH, THE FOLLOWING ERROR REPORT IS PRINTED:

RECEIVED DATA=RRRRR
DATA SHOULD BE TTTTT
DATA COMPARE ERROR; BAD DATA=BBB GOOD DATA=GGG

WHERE RRRRR IS THE RECEIVE BUFFER (UP TO 512 CHARACTERS)

TTTTT IS THE TRANSMIT BUFFER (UP TO 512 CHARACTERS)
 BBB IS THE BAD DATA CHARACTER
 GGG IS THE GOOD DATA CHARACTER

IF THE INTERFACE DETECTS A DATA ERROR, THE FOLLOWING
 WILL BE PRINTED BEFORE THE DATA IS COMPARED:

THERE WAS A RECEIVER ERROR. RECEIVER DATA REGISTER =XXXXXX

WHERE XXXXXX IS THE CONTENTS OF THE RECEIVER DATA REGISTER
 THE LOW BYTE IS THE DATA, AND THE HIGH BYTE IS THE ERROR BITS.

IF A RECEIVE TERMINATING CHARACTER<OO1> IS NOT DETECTED
 WITHIN 512 CHARACTERS A "BUFFER FULL" PRINTOUT WILL OCCUR.

7.0 RESTRICTIONS

THE OPERATION OF THIS PROGRAM REQUIRES COORDINATION BETWEEN
 THE OPERATOR AND THE OPERATOR OF ANOTHER PDP-11 SYSTEM
 UNLESS ONE OF THE SYSTEMS IS ALWAYS OPERATING IN A FIXED
 MODE. THE FOLLOWING TABLE LISTS THE VALID COMBINATIONS:

CPU #1	CPU #2
ONE-WAY-OUT	ONE-WAY-IN
ONE-WAY-IN	ONE-WAY-OUT
EXTERNAL-LOOPBACK	INTERNAL-LOOPBACK
INTERNAL-LOOPBACK	EXTERNAL-LOOPBACK
EXTERNAL-LOOPBACK	EXTERNAL-LOOPBACK (FULL DUPLEX)

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
 RESTRICTION APPLY:

IF RUNNING IN FULL DUPLEX MODE BOTH SYSTEMS
 MUST BE IN EXTERNAL LOOP BACK MODE.

BOTH SYSTEMS SHOULD BE RUNNING IDENTICAL ROUTINES.

EXAMPLE:
 SWITCHES 14,13,7,4 SHOULD BE THE SAME
 ON BOTH CPU'S

IF PROGRAM IS WAITING IN A SCAN ROUTINE AND TYPES OUT
 A "WAITING MESSAGE", IF AN INCOMING MESSAGE STARTS DURING
 THE TYPE OUT, IT WILL BE LOST BECAUSE THE TYPEOUT PRIORITY
 IS AT LEVEL 7. THIS WILL RESULT IN OVERRUN OR SILO OVER-
 RUN ERRORS, DEPENDING ON THE DEVICE. TO AVOID THIS SITUATION
 RUN WITH SWITCH 13 UP. IF OVERRUN DOES OCCURE DURING A
 TYPEOUT THE PROGRAM SHOULD BE RESTARTED.

IF USING AN ASYNCHRONOUS DEVICE, MODEMS AND THE
 MAYNARD TEST STATION AND INITIALIZE DOES NOT CLEAR THE
 CONNECTION (EXAMPLE THE DJ11) IF THE PROGRAM IS RESTARTED
 IN THE MIDDLE OF A MESSAGE AT LOC 204 OR BY HITTING CR
 AN IMMEDIATE ERROR MESSAGE FROM MAYNARD WILL BE RE-
 CEIVED. THIS IS BECAUSE THE TEST STATION IS STILL LOOKING

FOR THE REST OF THE INTERRUPTED MESSAGE. TO AVOID THIS ERROR, RESTART PROGRAM ONLY AT THE END OF THE MESSAGE CURRENTLY BEING TRANSMITTED.

8.0 MISCELLANEOUS

ITEP WAS CHECKED OUT USING THE FOLLOWING BELL TELEPHONE MODEMS.
201A (HALF-DUPLEX SYNCHRONOUS 2000 BAUD)
202C (HALF-DUPLEX ASYNCHRONOUS 1200 BAUD)
103A (FULL-DUPLEX ASYNCHRONOUS 110 BAUD)

9.0 PROGRAM DESCRIPTION

9.1 THE DMC11 INTERFACE SERVICE PARAMS ARE SETUP, AS SPECIFIED BY THE OPERATOR, BY THE ITEP CONTROL PROGRAM.

TIME: PROVIDES A MEANS OF MEASURING ELAPSED TIME. IT IS INCREMENTED EVERY SECOND BY A CLOCK INTERRUPT ROUTINE IN ITEP.

9.2 WHEN THE OVERLAY IS FIRST ENTERED BY ITEP AT LOCATION START:, THE CONTENTS OF THE SWITCH REGISTER ARE STORED IN REGISTER 0. THE MODE AND DATA SELECTIONS ARE FIXED AT THIS TIME AND CANNOT BE ALTERED WITHOUT RETURNING TO THE CONTROL PROGRAM. THE INTERRUPT VECTORS AND VARIABLES ARE THEN SETUP. THE SELECTED ROUTINE DETERMINED BY THE MODE IS THEN ENTERED

9.3 THE OVERLAY THEN LOOPS IN ROUTINES: SOWI, IF "ONE WAY IN" MODE WAS SELECTED. SOMO, IF "ONE WAY OUT" MODE WAS SELECTED. SILB, IF "INTERNAL LOOP BACK" MODE WAS SELECTED. SXLB, IF "EXTERNAL LOOP BACK" WAS SELECTED.

9.31 SOWI: IN THIS ROUTINE THE RECEIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR THE RECEIVER TO FINISH. IF NOTHING IS RECEIVED FOR 60 SECS A "WAITING" MESSAGE IS TYPED. WHEN THE RECEIVER IS DONE, THE PROGRAM CHECKS DATA IF SWITCHES PERMIT, AND TYPES END PASS DEPENDING ON SWITCH SETTINGS.

9.32 SOMO: THE TRANSMITTER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR TRANSMITTER TO FINISH, A "WAITING" MESSAGE IS TYPED EVERY 60 SECS IF THERE IS NO ACTION. WHEN THE TRANSMITTER IS DONE, THE PROGRAM EITHER LOOPS BACK TO SOMO OR TYPES END PASS DEPENDING ON SWITCH SETTINGS.

9.33 SILB: THE RECEIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR RECEIVER TO FINISH, A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF NO ACTION. WHEN RECEIVER IS DONE PROGRAM CHECKS DATA IF SWITCH SETTINGS PERMIT, AND END PASS IS TYPED IF SWITCH SETTINGS PERMIT. THEN THE TRANSMITTER IS INITIALIZED, A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF NO ACTION. WHEN TRANSMITTER IS DONE PROGRAM RETURNS TO START OF ROUTINE. (SILB)

9.34 SXLB: IF IN HALF DUPLEX THE TRANSMITTER IS INITIALIZED, A "WAITING MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO ACTION WHEN THE TRANSMITTER IS DONE THE RECEIVER IS INITIALIZED

A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO ACTION. WHEN THE RECEIVER IS DONE, DATA IS CHECKED IF SWITCH SETTINGS PERMIT AND END PASS IS TYPED IF SWITCHES ALLOW. THE PROGRAM NOW REPEATS CYCLE STARTING AT \$XLB.
IF IN FULL DUPLEX THE RECEIVER AND TRANSMITTER ARE INITIALIZED. A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO ACTION. WHEN BOTH THE RECEIVER AND TRANSMITTER ARE DONE, DATA IS CHECKED, END PASS IS TYPED AND PROGRAM LOOPS TO \$XLB DEPENDING ON THE SWITCH SETTINGS.

- 9.4 THE RETURN TO MONITOR ROUTINE FOR END PASS AT EOP:
LOCKS OUT INTERRUPTS AND SAVES THE TRANSMITTER INTERRUPT ENABLE BIT AND ALL GENERAL REGISTERS. IT THEN RETURNS TO THE MONITOR TO TYPE "END PASS". THE MONITOR CHECKS SW14 IF UP IT RETURNS TO ENTER:, OTHERWISE IT RESTARTS THE PROGRAM.
- 9.5 ENTER: IS ENTERED FROM THE MONITOR AFTER TYPEING "END PASS", IT RESTORES THE GENERAL REGISTERS AND THE TRANSMITTER CSR AS SAVED IN EOP. THE DELAY FLAG IS SET AND PROGRAM RETURNS TO THE SCAN ROUTINE(OWO,OWI,ILB,XLB) WHERE IT CAME FROM.
- 9.6 THE INITIALIZE TRANSMIT SUBROUTINE AT STARTX:
SETS UP THE INTERFACE AND POINTERS NECESSARY TO INITIATE A TRANSMIT OPERATION.
AFTER SETTING "DATA TERMINAL READY" AND "REQUEST TO SEND" A CHECK IS MADE ON PARAM2 TO DETERMINE IF HALF DUPLEX OPERATION WAS SELECTED BY THE OPERATOR. IF IT WAS, THE SUBROUTINE WAITS FOR CLEAR TO SEND.
A 'WAITING FOR CLEAR TO SEND' PRINTOUT OCCURS EVERY 30 SECONDS UNTIL CLEAR TO SEND IS ASSERTED.
- 9.7 THE INITIALIZE RECEIVED SUBROUTINE AT STARTR:
SETS UP THE INTERFACE AND POINTERS NECESSARY TO RECEIVE A MESSAGE.
- 9.8 THE TRANSMIT INTERRUPT SERVICE ROUTINE,
AT XISR:, IS ENTERED VIA TRANSMIT INTERRUPTS FROM THE INTERFACE.
A TEST IS MADE TO SEE IF THE LAST CHARACTER TRANSMITTED WAS A NULL (ALL ZEROS) CHARACTER. IF IT WAS: THE TRANSMIT LOGIC IN THE INTERFACE IS RESET AND THE TRANSMIT COMPLETE FLAG IS SET.
AT XISR1: THE NEXT CHARACTER IS TRANSMITTED AND PRINTED ON THE TTY IF THE MONITOR TRANSMIT SWITCH IS SET.
- 9.9 THE RECEIVE INTERRUPT SERVICE ROUTINE
AT RISR: IS ENTERED VIA RECEIVER INTERRUPTS FROM THE INTERFACE.
THE RECEIVED CHARACTER IS STORED IN THE INPUT BUFFER AND PRINTED ON THE TTY IF THE MONITOR RECEIVER SWITCH IS SET.
IF THE INPUT BUFFER IS FULL, A 'BUFFER FULL' PRINTOUT WILL OCCUR. THIS INDICATES THAT A LINE FEED CHARACTER WAS NOT RECOGNIZED IN THE RECEIVED DATA (WITHIN 1000 CHARACTERS).

IF THE RECEIVED CHARACTER IS A LINE FEED,
THE RECEIVED LOGIC IS RESET AND THE
RECEIVE COMPLETE FLAG IS SET.
IF A 'RECEIVE ERROR' IS DETECTED AT RISR:, THE
CSR AND DCR WILL BE SAVED AND PRINTED OUT
AFTER THE COMPLETE MESSAGE HAS BEEN RECEIVED.

9.10 THE DATA TEST SUBROUTINE AT TESTD: IS
ENTERED AFTER A COMPLETE MESSAGE HAS BEEN
RECEIVED.

IF A 'RECEIVE ERROR' HAD BEEN DETECTED,
THE CONTENTS OF THE 'RECEIVE BUFFER' AT THE
TIME THE ERROR OCCURRED WILL BE PRINTED.
THE DATA IS COMPARED UNTIL A 'ALL ZEROS'
CHARACTER IS RECOGNIZED. 'FILL' (ALL ONES)
CHARACTERS ARE IGNORED. IF A MISMATCH
IS DETECTED, THE COMPLETE CONTENTS OF THE
INPUT BUFFER AND GOOD DATA IS PRINTED.

10.0 PARAMETERS FOR THE DMC11

DZDMOA PROVIDES THREE TESTS FOR THE DMC-11, SELECTABLE BY
THE PARAMETER LOCATIONS PROVIDED IN ITEP. THE THREE
TESTS ARE:

10.1 1.) LINK TEST

NORMAL ITEP OPERATION, THE ONLY RESTRICTION
IS IT MUST BE DMC TO DMC. IT IS NORMAL TO GET
SOFT ERRORS DURING THE LINK TEST. THE PARAMETERS
FOR THE LINK TEST ARE AS FOLLOWS:

PARAM#1 IS NOT USED (0)

PARAM#2 FULL/HALF DUPLEX SELECTION
BIT0 = 0 HALF DUPLEX
 = 1 FULL DUPLEX (DEFAULT)

PARAM#3 IS NOT USED (177777)

10.2 2.) SECONDARY MODE TEST

THIS TEST CHECKS THE SECONDARY STATION DELAY.
IF RUNNING THIS TEST, EXTERNAL LOOP BACK IS
THE ONLY LEGAL MODE OF OPERATION. ALSO BOTH
DMC-11'S MUST HAVE HALF-DUPLEX SELECTED AND THE
SECONDARY MODE BIT SET IN THE PARAMETER WORD.
ADDITIONALLY ONE DMC IS SET TO BE THE SECONDARY
STATION AND THE OTHER THE PRIMARY STATION.
AGAIN IT IS NORMAL TO GET SOFT ERRORS DURING
THE SECONDARY MODE TEST AS IN THE LINK TEST.
THE PARAMETERS FOR THE SECONDARY MODE TEST
ARE AS FOLLOWS:

PARAM#1 IS NOT USED (0)

PARAM#2 SECONDARY MODE TEST SELECTION
 BIT0 = 0 HALF DUPLEX(MUST BE 0 FOR THIS TEST)
 BIT1 = 1 SECONDARY MODE TEST(MUST BE 1)
 BIT2 = 0 PRIMARY STATION
 = 1 SECONDARY STATION

PARAM#3 IS NOT USED (177777)

10.3 3.) BOOTSTRAP TEST

THIS WILL TEST THE ABILITY OF A DMC TO BOOT ANOTHER DMC USING MOP MESSAGES. THIS TEST REQUIRES A M9301-YJ AT ONE STATION. THE STATION WITH THE M9301-YJ IS THE BOOT STATION. THE OTHER IS THE ORIGINATING STATION. THERE ARE TWO VARIATIONS OF THIS TEST: 1) AUTOMATIC MODE IN WHICH THE ORIGINATING STATION SENDS THE BOOT MESSAGE TO THE BOOT STATION DMC WHICH CAUSES THE DMC TO BOOT THE M9301-YJ. 2) MANUAL MODE, IN THIS MODE THE BOOT STATION M9301 IS MANUALLY BOOTED. IN ADDITION TO THE PARAMETERS THE SWITCH REGISTER MUST BE SET AS FOLLOWS: SW09=1 SW02=1. MAKE ALL NECESSARY MODEM CONNECTIONS AND CALLS BEFORE STARTING THE BOOTSTRAP TEST. THE PARAMETERS FOR THE BOOTSTRAP TEST ARE AS FOLLOWS:

PARAM#1 BOOTSTRAP TEST SELECTION
 BIT8 = 1 BOOT STRAP TEST
 BITS 1089 MODE & STATION SELECT
 = 00 ORIGINATING STATION-AUTOMATIC MODE
 = 01 BOOT STATION-AUTOMATIC MODE
 = 10 ORIGINATING STATION-MANUAL MODE
 = 11 BOOT STATION-MANUAL MODE
 BITS 0-7 SWITCH PAC SETTING OF BOOT STATION DMC-11 DDCMP LINE# (USED AS PASSWORD IN MOP MESSAGE)

PARAM#2 IS NOT USED (0)

PARAM#3 IS NOT USED (177777)

11.0 DMC11 RESTRICTIONS

11.1 THE DMC11 IS A DMA DEVICE AND THEREFORE THE TRANSMITTED AND RECEIVED DATA CAN NOT BE MONITORED ON A PER CHARACTER BASIS BY THE CONSOLE TTY. BECAUSE OF THIS, SW05 AND SW06 HAVE NO EFFECT.

11.2 DMC ITEP IS MEANT TO BE AN ON LINE LINK TEST FOR TWO DMC11S. DMC ITEP WILL NOT WORK WITH ANY OTHER DEVICE

RUNNING ITEP EXCEPT ANOTHER DMC11.

- 11.3 BECAUSE THE DMC11 SUPPORTS DDCHP OPERATION IN THE FIRMWARE, THE PDP-11 PROGRAM (ITEP) IS UNABLE TO CONTROL OR KNOW EXACTLY WHAT IS BEING TRANSMITTED AT ANY GIVEN TIME. ALL DATA MESSAGES ARE ENCLOSED IN A DDCHP ENVELOPE AND THERE MAY ALSO BE CONTROL MESSAGES (AKS NAKS ETC) BEING TRANSMITTED. BECAUSE OF THIS PLEASE BEWARE IF YOU ARE SCOPING DATA.

```

659
660
661
662
663      011000      011000
664      011000      046504      000103
665      011004      160010
666      011006      000300
667      011010      000240
668      011012      000000
669      011014      000001
670      011016      177777
671      011020      000000
672      011022      000000
673      011024      000000
674      011026      000000
675      011030      000000
676      011032      000000
677      011034      000000
678      011036      011106
679      011040
680      011040      000
681      011041
682      011041      001
683      011042      000000
684      011044      177570
685      011046      177570
686
687
688
689
690      000000
691      100000
692      040000
693      020000
694      020000
695
696      011050      000000
697      011052      000000
698      011054      000000
699      011056      000000
700      011060      000000
701
702      011062      000000      01400
703      011064      000000      01500
704      011066      000000
705      011070      000000
706      011072      000000
707      011074      000000
708
709      011076      177560
710      011100      177562
711      011102      177564
712      011104      177566
713
714      000001

```

```

*****
: DMC11 INTERFACE SERVICE PARAMS
*****

```

```

DMC11:  =11000
BA:     .ASCIZ  /DMC/
RIV:    300
PRIOR:  240
PARAM1: 0
PARAM2: 1
PARAM3: 177777
IRDA:   .WORD  0
IXDA:   .WORD  0
SETTLE: .WORD  0
B2016:  .WORD  0
TIME:   .WORD  0
TX. TERM: .WORD  START
RX. TERM: .BYTE  000
FLAG:    .BYTE  001
SWR:     .WORD  0
DISPLAY: 177570

```

```

:ISR NAME
:BUS ADDRESS
:VECTOR ADDRESS
:PARAM #1
:PARAM #2
:PARAM #3
:INITIAL READ DATA ADDRESS
:INITIAL XMIT DATA ADDRESS
:LINE SETTLE DELAY FLAG
:ADDR OF BIN TO OCT TYPE ROUTINE
:TIMER
:ADDR OF START OF PROGRAM
:TRANSMITTER TERMINATING CHAR.
:RECEIVER TERMINATING CHAR.

```

```

*****
: CONSTANTS + WORKING STORAGE
*****

```

```

STAT=R0
XFLG=100000
RFLG=40000
DSFLG=20000
BIT13=20000
SXCSR: 0
SACSR: 0
ERCSR: 0
ERDBR: 0
DSSTAT: 0
TXWC: 0
RXWC: 0
XCC: 0
RCC: 0
RDA: 0
XDA: 0
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
FULL.DUPLEX=000001

```

```

: XMIT COMPLETE FLAG
: RCV COMPLETE FLAG
: DATA SET STATUS CHANGE FLAG
: INHIBIT PRINTOUTS
: SAVED XMIT CSR
: SAVED RCV CSR
: RCV CSR SAVED ON ERROR
: RCV DATA REG SAVED ON ERROR
: RCV CSR SAVED ON DS CHANGE
: XMIT CHAR COUNT
: RCV CHAR COUNT
: RCV DATA ADDR.
: XMIT DATA ADDR.

```



```

715
716
717
718 011106 000240
719 011110 017700 177730
720 011114 042700 177400
721 011120 013702 011006
722 011124 012722 014032
723 011130 013722 011010
724 011134 012722 013544
725 011140 013722 011010
726 011144 013704 011004
727 011150 013702 011006          01700
728 011154 012712 013544          01800
729 011160 012762 014032 000004 01900
730 011166 032737 000400 011012 02000
731 011174 001405          02100
732 011176 012712 015126          02200
733 011202 012762 015312 000004 02300
734 011210 012714 040000          02400
735 011214 013702 011022          02500
736 011220 005003          02600
737 011222 005203          02700
738 011224 123722 011040          02800
739 011230 001374          02900
740 011232 010337 011062          03000
741 011236 062703 000010          03100
742 011242 010337 011064          03200
743 011246 005714          03300
744 011250 100376          03400
745 011252 005037 016744          03500
746 011256 005037 016754          03600
747 011262 005037 016752          03700
748 011266 005714          03800
749 011270 100376          03900
750 011272 052764 000100 000002 04000
751 011300 052714 000143          04100
752 011304 005037 011054          04200
753 011310 005037 011056          04300
754 011314 005037 016766          04400
755 011320 005037 016770          04500
756 011324 005037 016772          04600
757 011330 005037 016774          04700
758 011334 105037 017001          04800
759 011340 005037 017002          04900
760 011344 005037 017004          05000
761 011350 005037 017006          05100
762 011354 105037 017010          05200
763 011360 005037 016750          05300
764 011364 005037 016742          05400
765 011370 032737 000400 011012 05500
766 011376 001402          05600
767 011400 000137 014254          05700
768 011404          05800
769          05900
770

```

```

*****
: DMC11-X INTERFACE SERVICE ROUTINE
*****
START: NOP
MOV 2SWR, R0 ; SETUP MODE IN R0
BIC 2177400, R0 ; STRIP JUNK
MOV RIV, R2 ; SETUP
MOV 2RISR, (R2)+ ; INTERRUPT
MOV 2PRIOR, (R2)+ ; VECTORS
MOV 2XISR, (R2)+
MOV 2PRIOR, (R2)+
MOV BA, R4 ; SETUP BUS ADDR INDEX
MOV RIV, R2 ; ADJUST VECTORS FOR
MOV 2XISR, (R2) ; INPUT SERVICE ROUTINE
MOV 2RISR, 4(R2) ; OUTPUT SERVICE ROUTINE
BIT 2BITB, PARAM1 ; BOOT MODE?
BEQ 3$, ; BR IF NO
MOV 2IISR, (R2) ; LOAD BOOT VECTORS
MOV 2OISR, 4(R2)
MOV 2BIT14, (R4) ; MASTER CLEAR DMC11
3$: MOV IXDA, R2 ; CALCULATE WORD COUNTS
CLR R3 ; CLEAR COUNT
1$: INC R3 ; INC COUNT
CMPB TX.TERM, (R2)+ ; LAST CHARACTER?
BNE 1$, ; BR IF NO
MOV R3, TXWC ; STORE XMIT COUNT
ADD 210, R3 ; ADD 10 TO IT
MOV R3, RXWC ; STORE REC COUNT
2$: TST (R4) ; WAIT FOR RUN
BPL 2$, ; BR IF RUN NOT SET
CLR BASEFG ; CLEAR BASE LOAD FLAG
CLR RFLAG ; CLEAR RECEIVE FLAG
CLR XFLAG ; CLEAR XMIT FLAG
4$: TST (R4) ; RUN SET?
BPL 4$, ; BR IF NO
BIS 2100, 2(R4) ; SET OIE
BIS 2143, (R4) ; ASK FOR BASE TRANSFER
CLR ERCSR ; CLEAR ERROR LOCATIONS
CLR ERDBR
CLR ERRCNT ; CLEAR SOFT ERROR STORAGE
CLR ERRCNT+2 ; LOCATIONS
CLR ERRCNT+4
CLR ERRCNT+6
CLRBASE BASE+3 ; CLEAR BASE ERROR COUNT LOCATIONS
CLRBASE BASE+4
CLRBASE BASE+6
CLRBASE BASE+10
CLRBASE BASE+12
CLR TEMP2
CLR RESUME ; CLEAR RESUME FLAG
BIT 2BITB, PARAM1 ; BOOT MODE?
BEQ 5$, ; BR IF NO
JMP BOOT ; GO TO START OF BOOT CODE
5$:

```

```

771
772
773
774
775
776
777 011404 005037 011032
778 011410 005037 013376
779 011414 005037 013402
780 011420 032700 000001
781 011424 001402
782 011426 000137 011606
783 011432 032700 000002
784 011436 001402
785 011440 000137 011474
786 011444 032700 000010
787 011450 001402
788 011452 000137 011710
789 011456 032700 000004
790 011462 001402
791 011464 000137 012144
792 011470 000000
793 011472 000776
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808 011474 104416
809 011476 004737 013474
810 011502 032700 040000
811 011506 001013
812 011510 023727 011032 000100
813 011516 103771
814 011520 011402
815 011522 016403 000002
816 011526 104001
817 011530 005037 011032
818 011534 000762
819
820 011536 032777 000200 177300
821 011544 001002
822 011546 004737 012622
823 011552 042700 040000
824 011556 004737 016176
825 011562 032777 000020 177254
826 011570 001405

```

```

*****
ROUTINE USED TO GOTO
SUBROUTINE DEPENDENT
ON MODE SELECTED.
*****

```

```

GO: CLR TIME
CLR DELAY
CLR STOP
BIT #OWO, MODE
BEQ 1$
JMP $OWO
1$: BIT #OWI, MODE
BEQ 2$
JMP $OWI
2$: BIT #ILB, MODE
BEQ 3$
JMP $ILB
3$: BIT #XLB, MODE
BEQ 4$
JMP $XLB
4$: HALT
BR .-2

```

```

*****
ROUTINE USED IF "ONE WAY IN" MODE WAS SELECTED.
NOTE THAT WHEN IN THIS MODE HALF DUPLEX IS THE
ONLY MODE AVAILABLE.
"ONE WAY IN" MEANS THAT ONLY THE RECEIVER IS
ENABLED. THE TRANSMITTER IS NEVER "TURNED ON".
*****

```

```

SOWI: KBOIN
JSR PC, STARTR
1$: BIT #RFLG, STAT
BNE 2$
CMP TIME, #100
BLO 1$
MOV @RCSR, R2
MOV XCSR(R4), R3
HLT 1
CLR TIME
BR 1$
2$: BIT #NODAT, @SWR
BNE 3$
JSR PC, TESTD
3$: BIC #RFLG, STAT
JSR PC, CKBASE
BIT #LOOP, @SWR
BEQ 4$

```

;CHECK DMC SOFT ERROR COUNTERS

827 011572 012737 011604 013400
 828 011600 000137 012462
 829 011604 000733
 830
 831
 832
 833
 834
 835
 836
 837
 838
 839
 840 011606 104416
 841 011610 004737 013404
 842 011614 005037 011032
 843 011620 032700 100000
 844 011624 001013
 845 011626 023727 011032 000100
 846 011634 103771
 847 011636 011402
 848 011640 016403 000002
 849 011644 104001
 850 011646 005037 011032
 851 011652 000762
 852 011654 042700 100000
 853 011660 004737 016176
 854 011664 032777 000020 177152
 855 011672 001405
 856 011674 012737 011706 013400
 857 011702 000137 012462
 858 011706 000737
 859
 860
 861

MOV #4\$,BACK
 JMP EOP
 BR SOWI
 4\$:

 ROUTINE USED IF "ONE WAY OUT" WAS SELECTED.
 NOTE THAT WHEN IN THIS MODE HALF DUPLEX IS THE ONLY
 MODE AVAILABLE.
 "ONE WAY OUT" MEANS THAT ONLY THE TRANSMITTER IS
 ENABLED. THE RECEIVER IS NEVER "TURNED ON."

SOWO: KBDIN
 JSR PC,STARTX
 CLR TIME
 1\$: BIT #XFLG,STAT
 BNE 2\$
 CMP TIME,#100
 BLO 1\$
 MOV @RCSR,R2
 MOV XCSR(R4),R3
 HLT 1
 CLR TIME
 BR 1\$
 2\$: BIC #XFLG,STAT
 JSR PC,CKBASE
 BIT #LOOP,@SWR
 BEQ 3\$
 MOV #3\$,BACK
 JMP EOP
 BR SOWO
 3\$:

;CHECK DMC SOFT ERROR COUNTERS

```

862
863
864
865
866
867
868
869
870
871
872
873 011710 104416
874 011712 004737 013474
875 011716 005037 011032
876 011722 032700 040000
877 011726 001013
878 011730 023727 011032 000100
879 011736 103771
880 011740 011402
881 011742 016403 000002
882 011746 104001
883 011750 005037 011032
884 011754 000762
885 011756 032777 000200 177060
886 011764 001002
887 011766 004737 012622
888 011772 042700 040000
889 011776 004737 016176
890 012002 032777 000020 177034
891 012010 001405
892 012012 012737 012024 013400
893 012020 000137 012462
894 012024 032777 000400 177012
895 012032 001416
896 012034 013702 011020
897 012040 013703 011022
898 012044 010337 011074
899 012050 112223
900 012052 001376
901 012054 112743 000177
902 012060 005203
903 012062 112723 000177
904 012066 105023
905 012070 005037 011032
906 012074 004737 013404
907 012100 032700 100000
908 012104 001013
909 012106 023727 011032 000100
910 012114 103771
911 012116 011402
912 012120 016403 000002
913 012124 104001
914 012126 005037 011032
915 012132 000762
916 012134 042700 100000
917 012140 000137 011710

```

```

*****
ROUTINE USED IF INTERNAL LOOP BACK WAS SELECTED.
NOTE THAT WHEN IN THIS MODE; HALF DUPLEX IS THE
ONLY MODE AVAILABLE.
INTERNAL LOOP BACK MEANS THAT THE RECEIVER IS "TURNED ON"
AND A COMPLETE MESSAGE IS RECEIVED. IF DATA IS TO BE CHECKED
IT IS; IF "END PASS" IS DESIRED; IT IS GIVEN.
THEN THE TRANSMITTER IS ENABLED. AFTER THE WHOLE MESSAGE
IS TRANSMITTED; THE CYCLE IS REPETED AS ABOVE.
*****
SILB:  KBDIN
        JSR    PC,STARTR
        CLR    TIME
1$:     BIT    #RFLG,STAT
        BNE    2$
        CMP    TIME,#100
        BLO   1$
        MOV    @RCSR,R2
        MOV    XCSR(R4),R3
        HLT    1
        CLR    TIME
2$:     BR    1$
        BIT    #NODAT,@SWR
        BNE    3$
        JSR    PC,TESTD
3$:     BIC    #RFLG,STAT
        JSR    PC,CKBASE
        BIT    #LOOP,@SWR
        BEQ    4$
        MOV    #4$ BACK
        JMP    EOP
4$:     BIT    #400,@SWR
        BEQ    7$
        MOV    IRDA,R2
        MOV    IXDA,R3
        MOV    R3,XDA
        MOV    (R2)+,(R3)+
        BNE    -2
        MOV    #177,-(R3)
        INC    R3
        MOV    #177,(R3)+
        CLRB  (R3)+
7$:     CLR    TIME
5$:     JSR    PC,STARTX
        BIT    #XFLG,STAT
        BNE    6$
        CMP    TIME,#100
        BLO   5$
        MOV    @RCSR,R2
        MOV    XCSR(R4),R3
        HLT    1
        CLR    TIME
6$:     BR    5$
        BIC    #XFLG,STAT
        JMP    SILB
;CCKE DMC SOFT ERROR COUNTERS
;USE EXTERNAL DATA?
;BR IF NO
;SET POINTER
;SET POINTER
;SETUP XMIT DATA ADDR
;MOVE INPUT TO OUTPUT
;LOOP IF NOT ZERO CHAR
;INSERT A FILL CHAR
;BUMP ADDRESS
;INSERT ANOTHER FILL
;INSERT ZERO CHAR

```

```

918
919
920
921
922
923
924
925
926
927
928
929
930
931 012144 104416
932 012146 032737 000002 011014
933 012154 001004
934 012156 032737 000001 011014
935 012164 001417
936 012166 004737 013474
937 012172 032737 000004 011014
938 012200 001411
939 012202 005737 016750
940 012206 001403
941 012210 032700 040000
942 012214 001774
943 012216 012737 177777 016750
944 012224 004737 013404
945 012230 005037 011032
946 012234 032700 100000
947 012240 001016
948 012242 032700 040000
949 012246 001030
950 012250 023727 011032 000100
951 012256 103766
952 012260 011402
953 012262 016403 000002
954 012266 104001
955 012270 005037 011032
956 012274 000757
957 012276 032737 000002 011014
958 012304 001356
959 012306 032737 000001 011014
960 012314 001352
961 012316 042700 100000
962 012322 004737 013474
963 012326 000742
964 012330 032737 000002 011014
965 012336 001004
966 012340 032737 000001 011014
967 012346 001420
968 012350 032700 100000
969 012354 001013
970 012356 023727 011032 000100
971 012364 103761
972 012366 011402
973 012370 016403 000002

```

```

*****
ROUTINE USED IF "EXTERNAL LOOP BACK" WAS SELECTED.
EITHER HALF OR FULL DUPLEX MAY BE SELECTED IN THIS MODE.
"EXTERNAL LOOP BACK" MEANS THAT THE TRANSMITTER IS FIRST
TURNED ON (IF HALF DUPLEX) AND THE WHOLE MESSAGE IS TRANSMITTED;
THEN THE RECEIVER IS ENABLED. AFTER THE WHOLE MESSAGE IS RECEIVED
DATA WILL THEN BE CHECKED IF DESIRED AND END PASS WILL
BE GIVEN IF DESIRED. THEN THE CYCLE IS REPEATED
AS ABOVE. IF RUNNING IN FULL DUPLEX THE PROGRAM
WAITS FOR BOTH THE RECEIVER AND TRANSMITTER TO
FINISH THEN RESTARTS THE RECEIVER AND TRANSMITTER.
*****

```

```

$XLB:  KBDIN
      BIT      #BIT1,PARAM2      ;SECONDARY MODE?
      BNE      9$                ;BR IF YES
      BIT      #FULL.DUPLEX,PARAM2
      BEQ      1$
9$:    JSR      PC,STARTR
      BIT      #BIT2,PARAM2      ;SECONDARY STATION?
      BEQ      1$                ;BR IF NO
      TST      TEMP2             ;FIRST TIME HERE?
      BEQ      11$              ;BR IF YES
      BIT      #RFLG,STAT        ;WAIT FOR RECEIVE BEFORE
      BEQ      -6                ;TRANSMITTING
      MOV      #-1,TEMP2        ;SET FIRST TIME FLAG
11$:   JSR      PC,STARTX
1$:    CLR      TIME
2$:    BIT      #XFLG,STAT
      BNE      3$
7$:    BIT      #RFLG,STAT
      BNE      4$
      CMP      TIME,#100
      BLO
      MOV      @RCSR,R2
      MOV      XCSR(R4),R3
      HLT      1
      CLR      TIME
      BR      2$
3$:    BIT      #BIT1,PARAM2      ;SECONDARY MODE?
      BNE      7$                ;BR IF YES
      BIT      #FULL.DUPLEX,PARAM2
      BNE      7$
      BIC      #XFLG,STAT
      JSR      PC,STARTR
      BR      2$
4$:    BIT      #BIT1,PARAM2      ;SECONDARY MODE?
      BNE      10$              ;BR IF YES
      BIT      #FULL.DUPLEX,PARAM2
      BEQ      8$
10$:   BIT      #XFLG,STAT
      BNE      6$
      CMP      TIME,#100
      BLO
      MOV      @RCSR,R2
      MOV      XCSR(R4),R3

```

DMC11 ITEP OVERLAY MACY11 30(1046) 11-JUL-77 12:39 PAGE 22
DZDMC.P11 18-MAY-77 11:28

974	012374	104001		
975	012376	005037	011032	
976	012402	000752		
977	012404	042700	100000	
978	012410	042700	040000	
979	012414	005037	011032	
980	012420	032777	000200	176416
981	012426	001002		
982	012430	004737	012622	
983	012434	004737	016176	
984	012440	032777	000020	176376
985	012446	001636		
986	012450	012737	012144	013400
987	012456	000137	012462	

6S:
8S:

5S:

HLT	I
CLR	TIME
BR	4S
BIC	#XFLG, STAT
BIC	#RFLG, STAT
CLR	TIME
BIT	#NODAT, JSWR
BNE	5S
JSR	PC, TESTD
JSR	PC, CKBASE
BIT	#LOOP, JSWR
BEQ	\$XLB
MOV	#XLB, BACK
JMP	EOP

;CHECK DMC SOFT ERROR COUNTERS

```

988
989
990
991
992
993
994 012462
995 012462 104414 000340
996 012466 016437 000002 012620
997 012474 042737 177677 012620
998 012502 042764 000100 000002
999 012510 012766 012550 000002
1000 012516 010037 013362
1001 012522 010137 013364
1002 012526 010237 013366
1003 012532 010337 013370
1004 012536 010437 013372
1005 012542 010537 013374
1006 012546 000207
1007
1008 012550
1009 012550 013700 013362
1010 012554 013701 013364
1011 012560 013702 013366
1012 012564 013703 013370
1013 012570 013704 013372
1014 012574 013705 013374
1015 012600 012737 177777 013376
1016 012606 053764 012620 000002
1017 012614 000177 000560
1018 012620 000000
1019
1020
1021
1022
1023
1024
1025
1026 012622 013746 011056
1027 012626 001413
1028 012630 032777 020000 176206
1029 012636 001007
1030 012640 104400 013022
1031 012644 004077 176160
1032 012650 005746
1033 012652 104400 013103
1034 012656 013701 011022
1035 012662 013702 011020
1036 012666 122122
1037 012670 001776
1038 012672 123741 011040
1039 012676 001447
1040 012700 122742 000002
1041 012704 001005
1042 012706 010237 012714
1043 012712 104400

```

```

*****
ROUTINE TO RETURN
TO MONITOR FOR
END PASS.
*****

```

```

EOP: STPS,PRTY7 ;SET PS PRIORITY TO 7
MOV XCSR(R4),QTPIE ;SAVE TX CSR
BIC #1C<TIE>,QTPIE ;CLEAR ALL BUT TX IE.
BIC #TIE,XCSR(R4) ;CLEAR TX IE (EVEN IF IT WASN'T SET)
MOV #ENTER,2(SP) ;SET FOR RETURN IF SW 14=1
MOV R0,SAVR0 ;SAVE REGISTER 0
MOV R1,SAVR1 ;SAVE REGISTER 1
MOV R2,SAVR2 ;SAVE REGISTER 2
MOV R3,SAVR3 ;SAVE REGISTER 3
MOV R4,SAVR4 ;SAVE REGISTER 4
MOV R5,SAVR5 ;SAVE REGISTER 5
RTS PC ;RETURN TO CONTROL PROGRAM

```

```

ENTER: MOV SAVR0,R0 ;RESTORE R0
MOV SAVR1,R1 ;RESTORE R1
MOV SAVR2,R2 ;RESTORE R2
MOV SAVR3,R3 ;RESTORE R3
MOV SAVR4,R4 ;RESTORE R4
MOV SAVR5,R5 ;RESTORE R5
MOV #-1,DELAY ;-1 DELAY
BIS QTPIE,XCSR(R4) ;IF ORGINALLY SET; SET TX IE
JMP @BACK
QTPIE: 000000

```

```

*****
SUBROUTINE TO CHECK
RECEIVER DATA.
*****
TESTD: MOV ERDBR,-(SP) ;WAS THERE A RECEIVE ERROR?
BEQ TSTDAT ;BR IF NO
BIT #BIT13,@SWR ;INHIBIT PRINTOUTS?
BNE TSTDAT ;BR IF YES
TYPE MSG0 ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
JSR R0,@B2016 ;PRINT CONTENTS OF RBUF
TST -(SP)
TYPE MSG1 ;<15><12>
TSTDAT: MOV IXDA,R1 ;SETUP XMIT DATA ADDR
MOV IRDA,R2 ;SETUP RCV DATA ADDR
SCAN4: CMPB (R1)+,(R2)+ ;DATA OK?
BEQ SCAN4 ;BR IF OK
CMPB TX.TERM,-(R1) ;IS IT END OF DATA
BEQ TESTDX ;BR IF YES
CMPB #002,-(R2)
BNE Z$
MOV R2,1$
TYPE

```

```

1044 012714 000000
1045 012716 000437
1046 012720
1047 012720 105712
1048 012722 001435
1049 012724 122721 000177
1050 012730 001756
1051 012732 005301
1052 012734 122722 000177
1053 012740 001752
1054 012742 000240
1055 012744 032777 020000 176072
1056 012752 001016
1057 012754 104400 013106
1058 012760 013737 011020 012770
1059 012766 104400
1060 012770 000000
1061 012772 104400 013133
1062 012776 013737 011022 013006
1063 013004 104400
1064 013006 011022
1065 013010 111103
1066 013012 114202
1067 013014 104007
1068 013016 005726
1069 013020 000207
1070
1071 013022 005015 044124 051105
(1) 013103 015 000012
(1) 013106 005015 042522 042503
(1) 013133 015 042012 052101
(1) 013156 005015 046120 040505
(1) 013225 015 053412 042510
(1) 013310 005015 046120 040505
(1)
(1) 013362 000000
1072 013364 000000
1073 013366 000000
1074 013370 000000
1075 013372 000000
1076 013374 000000
1077 013376 000000
1078 013400 000000
1079 013402 000000
1080

```

```

1S:  WORD 0
BR  TESTDX
2S:  TSTB (R2)
BEQ  TESTDX ;BR IF YES
CMPB #177 (R1)+ ;IS IT FILL CHAR?
BEQ  SCAN4 ;BR IF YES
DEC  R1 ;BACKUP
CMPB #177 (R2)+ ;IS IT FILL?
BEQ  SCAN4 ;BR IF YES
SCANS: NOP ;DATA ERROR
BIT  #BIT13,JSWR ;INHIBIT PRINTOUTS
BNE  DERR ;BR IF YES
TYPE MSG2 ;<15><12>RECEIVED DATA = <15><12>
MOV  IRDA, RDAX ;SETUP DATA ADDRESS
TYPE ;PRINT RECEIVED DATA
RDAX: 0 ;RECEIVED DATA ADDR
TYPE MSG3 ;<15><12>DATA SHOULD BE<15><12>
MOV  IXDA, .+10 ;SETUP ADDR.
TYPE ;PRINT GOOD DATA
IXDA
DERR: MOVB (R1),R3 ;SETUP XMIT DATA
MOVB -(R2),R2 ;SETUP RCV DATA
HLT+7 ;DATA ERROR HALT
TESTDX: TST (SP)+ ;POP STACK
RTS PC ;RETURN FROM SUB/ROUT
MSG0: .ASCIZ <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
MSG1: .ASCIZ <15><12>
MSG2: .ASCIZ <15><12>/RECEIVED DATA = /<15><12>
MSG3: .ASCIZ <15><12>/DATA SHOULD BE/<15><12>
MSG4: .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
.ASCIZ <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>
MSG5: .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
.EVEN
SAVR0: 0
SAVR1: 0
SAVR2: 0
SAVR3: 0
SAVR4: 0
SAVR5: 0
DELAY: 0
BACK: 0
STOP: 0

```



```

1081
1082
1083
1084
1085 013404 005737 013402 06300
1086 013410 001005 06400
1087 013412 104400 013156 06500
1088 013416 000000 06600
1089 013420 005137 013402 06700
1090 013424 005737 016744 06800
1091 013430 001775 06900
1092 013432 005037 016744 07000
1093 013436 005037 016746 07100
1094 013442 012737 000025 016750 07200
1095 013450 062737 000001 016746 07300
1096 013456 001374 07400
1097 013460 005337 016750 07500
1098 013464 001371 07600
1099 013466 152714 000140 07700
1100 013472 000207 07800
1101 08000 07900
1102
1103
1104
1105 08200
1106 013474 005737 013402 08300
1107 013500 001004 08400
1108 013502 104400 013310 08500
1109 013506 005137 013402 08600
1110 013512 005737 016754 08700
1111 013516 001007 08800
1112 013520 005737 016744 08900
1113 013524 001772 09000
1114 013526 005037 016744 09100
1115 013532 152714 000144 09200
1116 013536 005037 016754 09300
1117 013542 000207 09400
1118 09500
1119
1120
1121
1122 09700
1123 013544 032714 000002 09800
1124 013550 001426 09900
1125 013552 032714 000004 10000
1126 013556 001403 10100
1127 013560 004737 014020 10200
1128 013564 000002 10300
1129 013566 012764 016776 000004 10400
1130 013574 005064 000006 10500
1131 013600 005737 016742 10600
1132 013604 001403 10700
1133 013606 012764 010000 000006 10800
1134 013614 004737 014020 10900
1135 013620 152714 000141 11000
1136 013624 000002 11100

```

```

*****
: TRANSMIT INIT ROUTINE
*****

```

```

STARTX: TST      STOP      ;FIRST TIME HERE?
          BNE     1$        ;BR IF NOT
          TYPE   ,MSG4     ;TYPE CONNECT MESS
          HALT
          COM     STOP      ;SET FIRST TIME FLAG
1$:      TST     BASEFG    ;BASE AND CNTL IN ALL DONE?
          BEQ     1$        ;BR IF NO
          CLR     BASEFG    ;CLEAR FLAG
          CLR     TEMP1     ;GET SET TO DELAY
          MOV     #25,TEMP2 ;
          ADD     #1,TEMP1  ;INC DELAY
          BNE     -6        ;
          DEC     TEMP2     ;DEC DELAY COUNTER
          BNE     -14       ;BR IF NOT DONE
          BISB   #140,(R4)  ;ASK FOR XMIT BUFFER
          RTS     PC        ;RETURN

```

```

*****
: RECEIVE INIT ROUTINE
*****

```

```

STARTR: TST      STOP      ;FIRST TIME HERE?
          BNE     1$        ;BR IF NOT
          TYPE   ,MSG5     ;TYPE CONNECT MESS
          COM     STOP      ;SET FIRST TIME FLAG
1$:      TST     RFLAG     ;HAS AN REC BUFFER ALREADY BEEN GIVEN
          BNE     2$        ;BR IF YES
          TST     BASEFG    ;BASE AND CNTL IN ALL DONE?
          BEQ     1$        ;BR IF NO
          CLR     BASEFG    ;CLEAR FLAG
          BISB   #144,(R4)  ;ASK FOR REC BUFFER
2$:      CLR     RFLAG     ;CLEAR FLAG
          RTS     PC        ;RETURN

```

```

*****
: INPUT SERVICE ROUTINE
*****

```

```

XISR:   BIT     #BIT1,(R4) ;BASE REQUEST?
          BEQ     1$        ;BR IF NO
          BIT     #BIT2,(R4) ;IS IT REALLY A SHUT DOWN?
          BEQ     9$        ;BR IF NO
          JSR    PC,4$      ;YES, CLEAR RQI
          RTI
9$:     MOV     #BASE,4(R4) ;LOAD BASE ADDRESS
          CLR     6(R4)     ;CLEAR SEL 6
          TST     RESUME    ;RESUME FLAG SET?
          BEQ     8$        ;BR IF NOT
          MOV     #BIT12,6(R4) ;OTHERWISE SET RESUME BIT
8$:     JSR    PC,4$        ;CLEAR RQI
          BISB   #141,(R4)  ;ASK FOR CNTL I
          RTI

```

1137	013626	032714	000001		11200	15:	BIT	#BIT0,(R4)	:CNTL I REQUEST?
1138	013632	001430			11300		BEQ	25	:BR IF NO
1139	013634	032737	000001	011014	11400		BIT	#FULL.DUPLEX,PARAM2	:FULL OR HALF?
1140	013642	001403			11500		BEQ	55	:BR IF HALF
1141	013644	005064	000006		11600		CLR	6(R4)	:SELECT FULL DUPLEX
1142	013650	000413			11700		BR	65	:CONTINUE
1143	013652	032737	000004	011014	11800	55:	BIT	#BIT2,PARAM2	:SECONDARY STATION?
1144	013660	001404			11900		BEQ	75	:BR IF NO
1145	013662	012764	006000	000006	12000		MOV	#BIT10!BIT11,6(R4)	:SET SEC MODE AND HALF DUPLEX
1146	013670	000403			12100		BR	65	:CONTINUE
1147	013672	012764	002000	000006	12200	75:	MOV	#BIT10,6(R4)	:SELECT HALF DUPLEX
1148	013700	004737	014020		12300	65:	JSR	PC,45	:CLEAR RQI
1149	013704	012737	177777	016744	12400		MOV	#-1,BASEFG	:SET BASE LOADED FLAG
1150	013712	000002			12500		RTI		:RETURN
1151	013714	032714	000004		12600	25:	BIT	#BIT2,(R4)	:RECEIVE BUFFER REQUEST?
1152	013720	001416			12700		BEQ	35	:BR IF NO
1153	013722	042700	040000		12800		BIC	#RFLG,STAT	:CLEAR RECEIVE FLAG BIT
1154	013726	013764	011020	000004	12900		MOV	IRDA,4(R4)	:LOAD REC BUFFER ADDRESS
1155	013734	013764	011064	000006	13000		MOV	RDMC,6(R4)	:LOAD REC BYTE COUNT
1156	013742	004737	014020		13100		JSR	PC,45	:CLEAR RQI
1157	013746	012737	177777	016744	13200		MOV	#-1,BASEFG	:SET FLAG
1158	013754	000002			13300		RTI		:RETURN
1159	013756	042700	100000		13400	35:	BIC	#XFLG,STAT	:CLEAR XMIT FLAG BIT
1160	013762	012737	177777	016752	13500		MOV	#-1,XFLAG	:SET XMIT FLAG(FOR TIMEOUT ERROR)
1161	013770	013764	011022	000004	13600		MOV	IXDA,4(R4)	:LOAD XMIT BUFFER ADDRESS
1162	013776	013764	011062	000006	13700		MOV	TXMC,6(R4)	:LOAD XMIT BYTE COUNT
1163	014004	004737	014020		13800		JSR	PC,45	:CLEAR RQI
1164	014010	012737	177777	016744	13900		MOV	#-1,BASEFG	:SET FLAG
1165	014016	000002			14000		RTI		:RETURN
1166	014020	142714	000040		14100	45:	BICB	#40,(R4)	:CLEAR RQI
1167	014024	105714			14200		TSTB	(R4)	:WAIT FOR RQI TO DROP
1168	014026	100776			14300		BMI	-2	:BR IF STILL SET
1169	014030	000207			14400		RTS	PC	:RETURN
1170					14500				

```

;*****
;      OUTPUT SERVICE ROUTINE
;*****

```

1174					14700				
1175	014032	032764	000001	000002	14800	RISR:	BIT	#BIT0,2(R4)	:ERROR?
1176	014040	001463			14900		BEQ	15	:BR IF NO
1177	014042	005737	016742		15000		TST	RESUME	:RESUME FLAG SET?
1178	014046	001413			15100		BEQ	55	:BR IF NOT
1179	014050	022764	001000	000006	15200		CMP	#BIT9,6(R4)	:IS PROC. ERROR BIT SET?
1180	014056	001007			15300		BNE	55	:BR IF NOT
1181	014060	142764	000207	000002	15400		BICB	#207,2(R4)	:CLEAR DONE
1182	014066	012737	177777	016744	15500		MOV	#-1,BASEFG	:SET BASEFG
1183	014074	000002			15600		RTI		:RETURN
1184	014076	022764	000004	000006	15700	55:	CMP	#4,6(R4)	:OVERUN ERROR?
1185	014104	001003			15800		BNE	35	:BR IF NO
1186	014106	152714	000144		15900		BISB	#144,(R4)	:REQUEUE XMIT BUFFER
1187	014112	000432			16000		BR	45	
1188	014114	016437	000004	011054	16100	35:	MOV	4(R4),ERCSR	:SAVE SEL4
1189	014122	016437	000006	011056	16200		MOV	6(R4),ERDBR	:SAVE SEL6 (ERROR BITS)
1190	014130	104400	016366		16300		TYPE	DMCER	:ERROR MESSAGE
1191	014134	013746	011054		16400		MOV	ERCSR,-(SP)	:PUSH SEL4 ON STACK FOR TYPEOUT
1192	014140	004037	015770		16500		JSR	RD,\$B20CT	:TYPE IT OUT

1193	014144	006			16600
1194	014145	007			16700
1195	014146	104400	016562		16800
1196	014147	013746	011056		16900
1197	014156	004037	015770		17000
1198	014162	006			17100
1199	014163	001			17200
1200	014164	005777	174654		17300
1201	014170	100001			17400
1202	014172	000000			17500
1203	014174	005037	011056		17600
1204	014200	142764	000207	000002	17700
1205	014206	000002			17800
1206	014210	032764	000004	000002	17900
1207	014216	001406			18000
1208	014220	052700	040000		18100
1209	014224	142764	000207	000002	18200
1210	014232	000002			18300
1211	014234	052700	100000		18400
1212	014240	005037	016752		18500
1213	014244	142764	000207	000002	18600
1214	014252	000002			18700
1215					18800
1216					18900
1217					
1218					
1219					
1220					19100
1221	014254	032737	003000	011012	19200
1222	014262	001413			19300
1223	014264	032737	002000	011012	19400
1224	014272	001522			19500
1225	014274	032737	001000	011012	19600
1226	014302	001540			19700
1227	014304	104400	016566		19800
1228	014310	000777			19900
1229					20000
1230					
1231					
1232					
1233					20200
1234	014312	012701	015603		20300
1235	014316	113721	011012		20400
1236	014322	113721	011012		20500
1237	014326	113721	011012		20600
1238	014332	113721	011012		20700
1239	014336	005737	016744		20800
1240	014342	001763			20900
1241	01344	005037	016744		21000
1242	014350	012737	002400	016764	21100
1243	014356	052714	000141		21200
1244	014362	005737	016744		21300
1245	014366	001775			21400
1246	014370	005037	016744		21500
1247	014374	005037	016754		21600
1248	014400	052714	000144		21700

```

: BYTE 6
: BYTE 1
TYPE SPACE3 : INSERT 3 SPACES
MOV ERDR, -(SP) : PUSH SEL6 ON STACK FOR TYPEOUT
JSR RD, SB2OCT : TYPE IT OUT
: BYTE 6
: BYTE 1
TST JSR : CHECK BIT 15
BPL .+4 : SKIP HALT IF = 0
HALT : HALT IF SMR15 = 1
CLR ERDR : CLR ERDR LOCATION
4S: BICB #207, 2(R4) : CLEAR DONE
RTI : RETURN
1S: BIT #BIT2, 2(R4) : REC DONE?
BEQ 2S : BR IF NO
BIS #RFLG, STAT : SET REC DONE FLAG
BICB #207, 2(R4) : CLEAR DONE
RTI : RETURN
2S: BIS #XFLG, STAT : SET XMIT DONE FLAG
CLR XFLAG : CLEAR XFLAG
BICB #207, 2(R4) : CLEAR DONE
RTI : RETURN

: *****
: ENTER HERE IF BOOT MODE WAS SELECTED
: *****
BOOT: BIT #BIT10:BIT9, PARAM1 : DETERMINE WHICH BOOT MODE
BEQ AUTOG : BR IF AUTO MODE, ORIGINATING STATION
BIT #BIT10, PARAM1
BEQ AUTOBO : BR IF AUTO MODE, BOOT STATION
BIT #BIT9, PARAM1
BEQ MANORG : BR IF MANUAL MODE, ORIGINATING STATION
TYPE , BOOMSG : MANUAL MODE, BOOT STATION
BR . : WAIT FOR MANUAL BOOT

: *****
: AUTOMATIC MODE ORIGINATING STATION
: *****
AUTORG: MOV #MOP1+1, R1 : LOAD MOP1 MESSAGE WITH
MOVb PARAM1, (R1)+ : PASSWORD FOUR TIMES
MOVb PARAM1, (R1)+
MOVb PARAM1, (R1)+
MOVb PARAM1, (R1)+
TST BASEFG : IS BASE REQUEST DONE?
BEQ AUTOG : BR IF NO
CLR BASEFG : CLEAR LOCK FLAG
MOV #2400, SEL6 : MAINT. MODE BIT(MOP)
1S: BIS #141, (R4) : ASK FOR CNTL IN
TST BASEFG : IS CNTL IN DONE?
BEQ 1S : BR IF NO
CLR BASEFG : CLEAR LOCK FLAG
CLR RFLAG : CLEAR RECEIVE FLAG
BIS #144, (R4) : ASK FOR REC BA/CC

```

DMC11 ITEP OVERLAY
DZDMO.P11 18-MAY-77

MACY11 30(1046) 11-JUL-77 12:39 PAGE 28
11:28

1249	014404	005737	016744	21800
1250	014410	001775		21900
1251	014412	005037	016744	22000
1252	014416	005037	016752	22100
1253	014422	012737	015602	22200
1254	014430	012737	000005	22300
1255	014436	052714	000140	22400
1256	014442	012737	000005	22500
1257	014450	005037	013376	22600
1258	014454	005737	016754	22700
1259	014460	001012		22800
1260	014462	005337	013376	22900
1261	014466	001372		23000
1262	014470	005337	016746	23100
1263	014474	001367		23200
1264	014476	005737	016752	23300
1265	014502	001340		23400
1266	014504	000000		23500
1267				23600
1268				23700
1269				23800
1270	014506	012701	015610	23900
1271	014512	013702	011020	24000
1272	014516	005003		24100
1273	014520	122122		24200
1274	014522	001317		24300
1275	014524	005203		24400
1276	014526	022703	000004	24500
1277	014532	001372		24600
1278	014534	004737	015506	24700
1279				24800
1280				
1281				
1282				
1283				25000
1284	014540	005737	016744	25100
1285	014544	001775		25200
1286	014546	005037	016744	25300
1287	014552	005037	016760	25400
1288	014556	005037	016754	25500
1289	014562	052714	000144	25600
1290	014566	005737	016760	25700
1291	014572	001362		25800
1292	014574	005737	016754	25900
1293	014600	001357		26000
1294	014602	000771		26100
1295				26200
1296				
1297				
1298				
1299				26400
1300	014604	005037	016760	26500
1301	014610	005037	016756	26600
1302	014614	005737	016744	26700
1303	014620	001775		26800
1304	014622	005037	016744	26900

```

25:  TST  BASEFG      ; IS REQUEST DONE?
      BEQ  25        ; BR IF NO
      CLR  BASEFG    ; CLEAR LOCK FLAG
      CLR  XFLAG     ; CLEAR XMIT FLAG
      MOV  #MOP1,SEL4 ; SET FOR DMC LOAD
      MOV  #5,SEL6   ; SET FOR DMC LOAD
      BIS  #140,(R4) ; ASK FOR XMIT BA/CC
      MOV  #5,TEMP1  ; SET UP DELAY COUNT
      CLR  DELAY
35:  TST  RFLAG      ; RECEIVED ANYTHING?
      BNE  45        ; BR IF YES
      DEC  DELAY     ; DEC DELAY COUNTER
      BNE  35        ; WAIT TO RECEIVE
      DEC  TEMP1     ; DEC SECOND COUNT
      BNE  35        ; BR IF NOT DONE DELAY
      TST  XFLAG     ; WAS XMIT COMPLETED?
      BNE  25        ; BR IF YES, SEND MOP1 AGAIN
      HALT          ; ERROR, MOP1 WAS NEVER SENT OUT

; CHECK TO SEE IF RECEIVED MESSAGE IS MOP2
45:  MOV  #MOP2,R1   ; STARTING ADDRESS OF MOP2
      MOV  IRDA,R2   ; RECEIVE BUFFER ADDRESS
      CLR  R3        ; CLEAR COUNT
55:  CMPB (R1)+,(R2)+ ; COMPARE DATA
      BNE  15        ; IF NOT MOP2 TRY AGAIN
      INC  R3        ; DATA OK, BUMP COUNTER
      CMP  #4,R3    ; DONE YET?
      BNE  55        ; BR IF NO
      JSR  PC,MP3   ; IT WAS MOP2, SO SEND MOP3

*****
; AUTOMATIC MODE BOOT STATION
*****
AUTBOO: TST  BASEFG      ; BASE COMPLETED?
        BEQ  AUTBOO    ; BR IF NO
        CLR  BASEFG    ; CLEAR LOCK FLAG
        CLR  MFLAG     ; CLEAR MAINT FLAG
        CLR  RFLAG     ; CLEAR REC FLAG
        BIS  #144,(R4) ; ASK FOR REC BA/CC
15:    TST  MFLAG     ; ARE WE IN MAINT MODE?
        BNE  AUTBOO   ; BR IF YES
25:    TST  RFLAG     ; DID WE RECEIVE ANYTHING?
        BNE  AUTBOO   ; YES REQUEUE RECEIVE BUFFER
        BR   15       ; KEEP ON TRUCKIN'

*****
; MANUAL MODE ORGINATING STATION
*****
MANORG: CLR  MFLAG     ; CLEAR MAINT FLAG
        CLR  SFLAG     ; CLEAR DDCMP START RECEIVED ERROR FLAG
75:    TST  BASEFG    ; BASE LOADED?
        BEQ  75        ; BR IF NO
        CLR  BASEFG    ; RESET FLAG

```

DMC11 ITEP OVERLAY MACY11 30(1046) 11-JUL-77 12:39 PAGE 29
DZDMO.P11 18-MAY-77 11:28

1305	014626	005037	016764	27000
1306	014632	052714	000141	27100
1307	014636	005737	016744	27200
1308	014642	001775		27300
1309	014644	005037	016744	27400
1310	014650	005037	016754	27500
1311	014654	052714	000144	27600
1312	014660	005737	016756	27700
1313	014664	001437		27800
1314	014666	012714	040000	27900
1315	014672	005714		28000
1316	014674	100376		28100
1317	014676	005037	016756	28200
1318	014702	005037	016744	28300
1319	014706	052764	000100	28400
1320	014714	052714	000143	28500
1321	014720	005737	016744	28600
1322	014724	001775		28700
1323	014726	005037	016744	28800
1324	014732	005037	016764	28900
1325	014736	052714	000141	29000
1326	014742	005737	016744	29100
1327	014746	001775		29200
1328	014750	005037	016744	29300
1329	014754	005037	016754	29400
1330	014760	052714	000144	29500
1331	014764	005737	016760	29600
1332	014770	001733		29700
1333	014772	012714	040000	29800
1334	014776	005714		29900
1335	015000	100376		30000
1336	015002	005037	016744	30100
1337	015006	052764	000100	30200
1338	015014	052714	000143	30300
1339	015020	005737	016744	30400
1340	015024	001775		30500
1341	015026	005037	016744	30600
1342	015032	012737	002400	30700
1343	015040	052714	000141	30800
1344	015044	005737	016744	30900
1345	015050	001775		31000
1346	015052	005037	016744	31100
1347	015056	005037	016754	31200
1348	015062	052714	000144	31300
1349	015066	005737	016754	31400
1350	015072	001775		31500
1351				31600
1352				31700
1353				31800
1354	015074	012701	015610	31900
1355	015100	013702	011020	32000
1356	015104	005003		32100
1357	015106	122122		32200
1358	015110	001343		32300
1359	015112	005203		32400
1360	015114	022703	000004	32500

```

      CLR      SEL6      ;LOAD SEL6 FOR FULL-DUPLEX
      BIS      #141,(R4) ;ASK FOR CNTLI
      TST      BASEFG    ;CNTLI DONE?
8S:    BEQ      8S        ;BR IF NOT
      CLR      BASEFG    ;RESET FLAG
      CLR      RFLAG     ;CLEAR RECEIVE FLAG
      BIS      #144,(R4) ;ASK FOR REC BA/CC
      TST      SFLAG     ;DOCMP START ERROR?
      BEQ      4S        ;BR IF NO
      MOV      #BIT14,(R4);INITIALIZE DMC
2S:    TST      (R4)      ;RUN SET?
      BPL      2S        ;BR IF NO
      CLR      SFLAG     ;CLEAR FLAG
      CLR      BASEFG    ;CLEAR LOCK FLAG
      BIS      #100,2(R4) ;SET INT ENABLE
      BIS      #143,(R4) ;ASK FOR BASE
3S:    TST      BASEFG    ;BASE DONE?
      BEQ      3S        ;BR IF NO
      CLR      BASEFG    ;CLEAR FLAG
      CLR      SEL6     ;SET UP FOR FULL-DUPLEX
      BIS      #141,(R4) ;ASK FOR CNTLI
9S:    TST      BASEFG    ;CNTLI FINISHED?
      BEQ      9S        ;BR IF NOT
      CLR      BASEFG    ;CLEAR FLAG
      CLR      RFLAG     ;CLEAR RECEIVER FLAG
      BIS      #144,(R4) ;ASK FOR REC BA/CC
      TST      MFLAG     ;ARE WE IN MAINT MODE?
      BEQ      1S        ;BR IF NO
      MOV      #BIT14,(R4);INITIALIZE DMC
10S:   TST      (R4)      ;RUN SET?
      BPL      10S       ;BR IF NO
      CLR      BASEFG    ;CLEAR LOCK FLAG
      BIS      #100,2(R4) ;SET INT ENABLE
      BIS      #143,(R4) ;ASK FOR BASE
11S:   TST      BASEFG    ;BASE DONE?
      BEQ      11S       ;BR IF NO
      CLR      BASEFG    ;CLEAR FLAG
      MOV      #2400,SEL6;MAINT. MODE (MOP)
      BIS      #141,(R4) ;ASK FOR CNTLI
12S:   TST      BASEFG    ;CNTLI FINISHED?
      BEQ      12S       ;BR IF NOT
      CLR      BASEFG    ;CLEAR FLAG
      CLR      RFLAG     ;CLEAR RECEIVER FLAG
      BIS      #144,(R4) ;ASK FOR REC BA/CC
5S:    TST      RFLAG     ;HAVE WE RECEIVED ANYTHING?
      BEQ      5S        ;BR IF NO

      ;CHECK TO SEE IF RECEIVED MESSAGE IS MOP2
      MOV      #MOP2,R1  ;MOP2 STARTING ADDRESS
      MOV      IRDA,R2   ;RECEIVE BUFFER ADDRESS
      CLR      R3        ;CLEAR COUNT
6S:    CMPB    (R1)+,(R2)+;COMPARE DATA
      BNE     11S        ;IT ISN'T MOP2, TRY AGAIN
      INC     R3         ;DATA OK, BUMP COUNT
      CMP     #4,R3      ;DONE YET?

```

```

1361 015120 001372          32600
1362 015122 004737 015506 32700
1363          32800
1364
1365
1366
1367          33000
1368 015126 032714 000002 33100
1369 015132 001413          33200
1370 015134 012764 016776 000004 33300
1371 015142 005064 000006 33400
1372 015146 004737 015300 33500
1373 015152 012737 177777 016744 33600
1374 015160 000002          33700
1375 015162 032714 000001 33800
1376 015166 001411          33900
1377 015170 013764 016764 000006 34000
1378 015176 004737 015300 34100
1379 015202 012737 177777 016744 34200
1380 015210 000002          34300
1381 015212 032714 000004 34400
1382 015216 001414          34500
1383 015220 013764 011020 000004 34600
1384 015226 012764 000010 000006 34700
1385 015234 004737 015300 34800
1386 015240 012737 177777 016744 34900
1387 015246 000002          35000
1388 015250 013764 016762 000004 35100
1389 015256 013764 016764 000006 35200
1390 015264 004737 015300 35300
1391 015270 012737 177777 016744 35400
1392 015276 000002          35500
1393 015300 142714 000040 35600
1394 015304 105714          35700
1395 015306 100776          35800
1396 015310 000207          35900
1397          36000
1398
1399
1400
1401          36200
1402 015312 032764 000001 000002 36300
1403 015320 001453          36400
1404 015322 022764 000010 000006 36500
1405 015330 001004          36600
1406 015332 012737 177777 016760 36700
1407 015340 000456          36800
1408 015342 022764 000200 000006 36900
1409 015350 001004          37000
1410 015352 012737 177777 016756 37100
1411 015360 000446          37200
1412 015362 016437 000004 011054 37300
1413 015370 016437 000006 011056 37400
1414 015376 104400 016366 37500
1415 015402 013746 011054 37600
1416 015406 004037 015770 37700

```

```

BNE 6S ;BR IF NO
JSR PC,MP3 ;IT WAS MOP2, SO SEND OUT MOP3

;*****
; INPUT INTERRUPT SERVICE ROUTINE (BOOT MODE)
;*****

IISR: BIT #BIT1,(R4) ;IS IT A BASE REQUEST?
      BEQ 1S ;NO
      MOV #BASE,4(R4) ;YES, LOAD BASE ADDRESS
      CLR 6(R4) ;CLEAR SEL6
      JSR PC,4S ;CLEAR RQI
      MOV #-1,BASEFG ;SET FLAG
      RTI ;RETURN

1S: BIT #BIT0,(R4) ;IS IT CNTL IN?
     BEQ 2S ;BR IF NO
     MOV SEL6,6(R4) ;LOAD SEL6 FOR CNTLI
     JSR PC,4S ;CLEAR RQI
     MOV #-1,BASEFG ;SET FLAG
     RTI ;RETURN

2S: BIT #BIT2,(R4) ;IS IT RECEIVE REQUEST?
     BEQ 3S ;NO
     MOV IRDA,4(R4) ;YES, LOAD REC BA
     MOV #10,6(R4) ;CC
     JSR PC,4S ;CLEAR RQI
     MOV #-1,BASEFG ;SET FLAG
     RTI ;RETURN

3S: MOV SEL4,4(R4) ;XMIT REQUEST, LOAD XMIT BA
     MOV SEL6,6(R4) ;XMIT CC
     JSR PC,4S ;CLEAR RQI
     MOV #-1,BASEFG ;SET FLAG
     RTI ;RETURN

4S: BICB #40,(R4) ;CLEAR RQI
     TSTB (R4) ;RQI CLEAR?
     BMI -2 ;NO
     RTS PC ;RETURN

;*****
; OUTPUT INTERRUPT SERVICE ROUTINE (BOOT MODE)
;*****

OISR: BIT #BIT0,2(R4) ;ERROR?
      BEQ 3S ;NO
      CMP #10,6(R4) ;YES, MAINT MODE ENTERED?
      BNE 1S ;NO
      MOV #-1,MFLAG ;YES, SET MFLAG
      BR 5S ;RETURN

1S: CMP #200,6(R4) ;DDCMP START RECEIVED ERROR?
     BNE 2S ;NO
     MOV #-1,SFLAG ;YES, SET SFLAG
     BR 5S ;RETURN

2S: MOV 4(R4),ERCSR ;SAVE SEL4
     MOV 6(R4),ERDBR ;SAVE SEL6 (ERROR BITS)
     TYPE ,DMCER ;ERROR MESSAGE
     MOV ERCSR,-(SP) ;PUSH SEL4 ON STACK FOR TYPEOUT
     JSR RO,$B20CT ;TYPE IT OUT

```

```

1417 015412 006 37800
1418 015413 001 37900
1419 015414 104400 016562 38000
1420 015420 013746 011056 38100
1421 015424 004037 015770 38200
1422 015430 006 38300
1423 015431 001 38400
1424 015432 005777 173406 38500
1425 015436 100001 38600
1426 015440 000000 38700
1427 015442 005037 011056 38800
1428 015446 000413 38900
1429 015450 032764 000004 000002 39000
1430 015456 001404 39100
1431 015460 012737 177777 016754 39200
1432 015466 000403 39300
1433 015470 012737 177777 016752 39400
1434 015476 142764 000207 000002 39500
1435 015504 000002 39600
1436 39700
1437
1438
1439
1440 39900
1441 015506 005737 016744 40000
1442 015512 001775 40100
1443 015514 005037 016744 40200
1444 015520 005037 016752 40300
1445 015524 012737 015614 016762 40400
1446 015532 012737 000154 016764 40500
1447 015540 052714 000140 40600
1448 015544 005037 013376 40700
1449 015550 005737 016752 40800
1450 015554 001004 40900
1451 015556 005337 013376 41000
1452 015562 001372 41100
1453 015564 000000 41200
1454 015566 104400 016635 41300
1455 015572 005726 41400
1456 015574 000000 41500
1457 015576 000137 011106 41600
1458 41700
1459 015602 006 000 000 41800
1460 015605 000 000 41900
1461 015610 010 014 001 42000
1462 015613 000 42100
1463 015614 000 000 006 42100
1464 015617 000 006 000 42100
1465 42200
1466 42300
1467 42400
1468 015622 005037 000006 42400
1469 015626 000005 42500
1470 015630 012706 001000 42600
1471 015634 012701 177560 42700
1472 015640 010700 42800

```

```

        .BYTE 6
        .BYTE 1
        TYPE SPACE3 ;INSERT 3 SPACES
        MOV ERDBR, -(SP) ;SPUSH SEL6 ON STACK FOR TYPEOUT
        JSR RO, $B20CT ;TYPE IT OUT
        .BYTE 6
        .BYTE 1
        TST $SWR ;CHECK BIT 15
        BPL .+4 ;SKIP HALT IF = 0
        HALT ;HALT IF SWR15 = 1
        CLR ERDBR ;CLR ERDBR LOCATION
        BR 5$ ;RETURN
3$: BIT #BIT2, 2(R4) ;RECEIVE DONE?
    BEQ 4$ ;BR IF NO
    MOV #-1, RFLAG ;SET RECEIVE FLAG
    BR 5$ ;RETURN
4$: MOV #-1, XFLAG ;XMIT DONE, SET XMIT FLAG
5$: BICB #207, 2(R4) ;CLEAR DONE
    RTI ;RETURN

;*****
; SUBROUTINE TO SEND MOP3 MESSAGE
;*****
MOP3: TST BASEFG ;IS IT OK TO REQUEST
      BEQ MP3 ;BR IF NO
      CLR BASEFG ;CLEAR LOCK FLAG
      CLR XFLAG ;CLEAR XMIT FLAG
      MOV #MOP3, SEL4 ;MOP3 ADDRESS
      MOV #MOP3ED-MOP3, SEL6 ;MOP3 COUNT
      BIS #140, (R4) ;ASK FOR XMIT BA/CC
      CLR DELAY ;START DELAY COUNT
1$: TST XFLAG ;XMIT DONE?
    BNE 2$ ;BR IF YES
    DEC DELAY ;DEC DELAY COUNT
    BNE 1$ ;BR IF NO DONE
    HALT ;ERROR, MOP3 SEND NOT DONE
2$: TYPE ORGOK ;OK MOP3 SEND DONE
    TST (SP)+ ;POP STACK (ENTERED BY JSR)
    HALT ;ALL DONE HIT CONT TO DO IT AGAIN
    JMP START

MOP1: .BYTE 6,0,0,0,0
      .EVEN
MOP2: .BYTE 10,12.,1,0
MOP3: .BYTE 0,0,6,0,6,0

;IMAGE OF PROGRAM TO BE DOWN LINE LOADED
CLR $#6 ;SET UP TIMEOUT VECTOR TO HALT
RESET ;CLEAR ALL!!
MOV #1000, SP ;SET UP STACK
MOV #177560, R1 ;SET TTY CSR
MOV PC, RO ;MAKE ADDRESS PIC

```

DMC11 ITEP OVERLAY MACY11 30(1046) 11-JUL-77 12:39 PAGE 32
DZDMO.P11 18-MAY-77 11:28

1473	015642	062700	000034	42900
1474	015646	105761	000004	43000
1475	015652	100375		43100
1476	015654	112061	000006	43200
1477	015660	001372		43300
1478	015662	012737	000026 000024	43400
1479	015670	005037	000026	43500
1480	015674	000777		43600
1481	015676	006412	047502 052117	43700
1482	015704	046440	051505 040523	
1483	015712	042507	053440 051501	
1484	015720	051040	041505 044505	
1485	015726	042526	020104 052523	
1486	015734	041503	051505 043123	
1487	015742	046125	054514 026440	
1488	015750	042440	042116 047440	
1489	015756	020106	042524 052123	
1490	015764	020441	000	
1491	015767	006		43800
1492	015770			43900
1493				44000
1494				44100
1495				44200
1496				44300
1497				44400
1498				44500
1499				44600
1500				44700
1501				44800
1502				44900
1503				45000
1504				45100
1505				45200
1506				45300
1507				45400
1508				45500
1509				45600
1510				45700
1511				45800
1512				45900
1513				46000
1514				46100
1515				46200
1516				46300
1517	015770	112037	016175	46400
1518	015774	112037	016173	46500
1519	016000	000406		46600
1520	016002	112737	000001 016173	46700
1521	016010	112737	000006 016175	46800
1522	016016	112737	000005 016172	46900
1523	016024	010346		47000
1524	016026	010446		47100
1525	016030	010546		47200
1526	016032	113704	016175	47300
1527	016036	005404		47400
1528	016040	062704	000006	47500

```

ADD      #<MSG-.>,R0      ; ADDRESS OF MESSAGE
TSTB    4(R1)             ; READY SET?
BPL     1$                ; BR IF NO
MOV     (R0)+,6(R1)      ; TYPE A CHARACTER
BNE     1$                ; KEEP TYPING IF NOT ZERO
MOV     #26,2#24         ; SET UP POWER FAIL VECTOR
CLR     2#26             ; MAKE SURE T BIT CLEAR
BR      BR               ; BR
MSG:    .ASCIZ <12><15>/BOOT MESSAGE WAS RECEIVED SUCCESSFULLY - END OF TEST!!/

MOP3ED: .BYTE 6

*****
; BINARY TO OCTAL (ASCII) AND TYPE
; $B2OCT---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
; CALL:
;       MOV     NUM, -(SP)      ; NUMBER TO BE TYPED
;       JSR     RO, $B2OCT     ; CALL FOR TYPEOUT
;       .BYTE  N              ; N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
;       .BYTE  M              ; M=1 OR 0
;                               ; 1=TYPE LEADING ZEROS
;                               ; 0=SUPPRESS LEADING ZEROS

; $B201----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST $B5OCT OR
; CALL:
;       MOV     NUM, -(SP)
;       JSR     RO, $B201

; $B2016---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
; CALL:
;       MOV     NUM, -(SP)
;       JSR     RO, $B2016

; PICKUP THE NUMBER OF DIGITS TO TYPE
; GET THE ZERO FILL SWITCH
$B2OCT: MOV     (R0)+, $OMODE+1
        MOV     (R0)+, $OFILL
        BR      $B201
; SET THE ZERO FILL SWITCH
; SET FOR SIX(6) DIGITS
; SET THE ITERATION COUNT
$B2016: MOV     #1, $OFILL
        MOV     #6, $OMODE+1
        MOV     #5, $OCNT
        MOV     R3, -(SP)
        MOV     R4, -(SP)
        MOV     R5, -(SP)
        MOV     $OMODE+1, R4
        NEG     R4
        ADD     #6, R4
; SUBTRACT IT FOR MAX. ALLOWED

```


1529	016044	110437	016174	47600	MOVW	R4, \$OMODE	: SAVE IT FOR USE
1530	016050	113704	016173	47700	MOVW	\$OFILL, R4	: GET THE ZERO FILL SWITCH
1531	016054	016605	000010	47800	MOV	10(SP), R5	: PICKUP THE INPUT NUMBER
1532	016060	005003		47900	CLR	R3	: CLEAR THE OUTPUT WORD
1533	016062	006105		48000	1\$: ROL	R5	: ROTATE MSB INTO "C"
1534	016064	000404		48100	BR	3\$: GO DO MSB
1535	016066	006105		48200	2\$: ROL	R5	: FORM THIS DIGIT
1536	016070	006105		48300	ROL	R5	
1537	016072	006105		48400	ROL	R5	
1538	016074	010503		48500	MOV	R5, R3	
1539	016076	006103		48600	3\$: ROL	R3	: GET LSB OF THIS DIGIT
1540	016100	105337	016174	48700	DECB	\$OMODE	: TYPE THIS DIGIT?
1541	016104	100016		48800	BPL	7\$: BR IF NO
1542	016106	042703	177770	48900	BIC	#177770, R3	: GET RID OF JUNK
1543	016112	001002		49000	BNE	4\$: TEST FOR 0
1544	016114	005704		49100	TST	R4	: SUPPRESS THIS 0?
1545	016116	001403		49200	BEQ	5\$: BR IF YES
1546	016120	005204		49300	4\$: INC	R4	: DON'T SUPPRESS ANYMORE 0'S
1547	016122	052703	000060	49400	BIS	#'0, R3	: MAKE THIS DIGIT ASCII
1548	016126	052703	000040	49500	5\$: BIS	#', R3	: MAKE ASCII IF NOT ALREADY
1549	016132	110337	016170	49600	MOVW	R3, 8\$: SAVE FOR TYPING
1550	016136	104400	016170	49700	TYPE	8\$: GO TYPE THIS DIGIT
1551	016142	105337	016172	49800	7\$: DECB	\$OCNT	: COUNT BY 1
1552	016146	003347		49900	BGT	2\$: BR IF MORE TO DO
1553	016150	002402		50000	BLT	6\$: BR IF DONE
1554	016152	005204		50100	INC	R4	: INSURE LAST DIGIT ISN'T A BLANK
1555	016154	000744		50200	BR	2\$: GO DO THE LAST DIGIT
1556	016156	012605		50300	6\$: MOV	(SP)+, R5	: RESTORE R5
1557	016160	012604		50400	MOV	(SP)+, R4	: RESTORE R4
1558	016162	012603		50500	MOV	(SP)+, R3	: RESTORE R3
1559	016164	012616		50600	MOV	(SP)+, (SP)	: SET THE STACK FOR RETURNING
1560	016166	000200		50700	RTS	R0	: RETURN
1561	016170	000		50800	8\$: .BYTE	0	: STORAGE FOR ASCII DIGIT
1562	016171	000		50900	.BYTE	0	: TERMINATOR FOR TYPE ROUTINE
1563	016172	000		51000	\$OCNT: .BYTE	0	: OCTAL DIGIT COUNTER
1564	016173	000		51100	\$OFILL: .BYTE	0	: ZERO FILL SWITCH
1565	016174	000000		51200	\$OMODE: 0		: NUMBER OF DIGITS TO TYPE
1566				51300			
1567				51400			
1568	016176	012737	177777	51500	CKBASE: MOV	#-1, RESUME	: SET RESUME FLAG
1569	016204	005037	016744	51600	CLR	BASEFG	: CLEAR BASEFG
1570	016210	052714	000146	51700	BIS	#146, (R4)	: SHUT DOWN DMC TO UPDATE BASE TABLE
1571	016214	005737	016744	51800	1\$: TST	BASEFG	: SHUT DOWN DONE?
1572	016220	001775		51900	BEQ	1\$: BR IF NO
1573	016222	012714	040000	52000	MOV	#BIT14, (R4)	: MASTER CLEAR DMC
1574	016226	005714		52100	2\$: TST	(R4)	: RUN SET?
1575	016230	100376		52200	BPL	2\$: BR IF NO
1576	016232	005037	016744	52300	CLR	BASEFG	: CLEAR BASEFG
1577	016236	052714	000143	52400	BIS	#143, (R4)	: ASK FOR BASE REQUEST
1578	016242	005737	016744	52500	3\$: TST	BASEFG	: BASE LOADED?
1579	016246	001775		52600	BEQ	3\$: BR IF NO
1580	016250	012764	000100	52700	MOV	#100, 2(R4)	: SET INTERRUPT ENABLE
1581	016256	005037	016742	52800	CLR	RESUME	: CLEAR RESUME FLAG
1582	016262	012702	000003	52900	MOV	#3, R2	: LOAD BASE OFFSET TO ERROR COUNTS
1583	016266	005001		53000	CLR	R1	: R1 IS OFFSET INTO SOFTWARE TABLE
1584	016270	126261	016776	53100	5\$: CMPB	BASE(R2), ERRCNT(R1)	: ANY ERRORS THIS PASS?

1585	016276	001005		53200	BNE	6\$:BR IF YES
1586	016300	122122		53300	CMPB	(R1)+(R2)+		:INC INDEXS
1587	016302	022702	000013	53400	CMP	#13,R2		:DONE?
1588	016306	001370		53500	BNE	5\$:BR IF NO
1589	016310	000207		53600	RTS	PC		:RETURN
1590	016312	104400	016437	53700	6\$:	TYPE	SOFT	:TYPE SOFT ERROR MESSAGE
1591	016316	012702	000003	53800	MOV	#3,R2		:LOAD BASE OFFSET TO ERROR COUNTS
1592	016322	005001		53900	CLR	R1		:SOFTWARE TABLE OFFSET
1593	016324	116261	016776	54000	7\$:	MOVB	BASE(R2),ERRCNT(R1)	:SAVE ERROR COUNTS
1594	016332	116246	016776	54100	MOVB	BASE(R2),-(SP)		:PUSH ON STACK FOR TYPEOUT
1595	016336	042716	177400	54200	BIC	#1C<37>(SP)		:CLEAR HI-BYTE
1596	016342	004037	015770	54300	JSR	RO,\$B20CT		:TYPE IT OUT
1597	016346	003		54400	.BYTE	3		
1599	016347	001		54500	.BYTE	1		
1599	016350	104400	016562	54600	TYPE	SPACE3		:INSERT 3 SPACES
1600	016354	122221		54700	CMPB	(R2)+(R1)+		:INC INDEXS
1601	016356	022702	000013	54800	CMP	#13,R2		:DONE?
1602	016362	001360		54900	BNE	7\$:BR IF NO
1603	016364	000207		55000	RTS	PC		:RETURN
1604				55100				
1605				55200				
1606				55300	.NLIST	BEX		
	016366	005015	041412	55400	DMCER:	.ASCII	<15><12><12>/CONTROL OUT ERROR/	
	016412	005015	020040	55500		.ASCII	<15><12>/ SEL4 SEL6/	
	016433	015	020012	55600		.ASCIZ	<15><12>/ /	
	016437	015	005012	55700	SOFT:	.ASCII	<15><12><12>/SOFT ERROR - DDCMP ERROR COUNTS ARE NON ZERO/	
	016516	005015	020040	55800		.ASCII	<15><12>/	BASE+3 THRU BASE+12/<15><12><0>
	016562	020040	000040	55900	SPACE3:	.ASCIZ	/ /	
	016566	005015	040515	56000	BOOMSG:	.ASCIZ	<15><12>/MANUALLY BOOT DMC NOW (VIA M9301-YJ)/	
	016635	015	047412	56100	ORGOK:	.ASCIZ	<15><12>/ORIGINATING STATION HAS COMPLETED BOOT SUCCESSFULLY - E	
		016742		56200	.EVEN			
				56300	.LIST	BEX		
1607				56400				
1608	016742	000000		56500	RESUME:	0		
1609	016744	000000		56600	BASEFG:	0		:BASE LOAD FLAG
1610	016746	000000		56700	TEMP1:	0		
1611	016750	000000		56800	TEMP2:	0		
1612	016752	000000		56900	XFLAG:	0		
1613	016754	000000		57000	RFLAG:	0		
1614	016756	000000		57100	SFLAG:	0		
1615	016760	000000		57200	MFLAG:	0		
1616	016762	000000		57300	SEL4:	0		
1617	016764	000000		57400	SEL6:	0		
1618				57500				
1619	016766	000004		57600	ERRCNT:	.BLKW 4		
1620				57700				
1621				57800				
1622	016776	017376		57900	BASE:	.=.+256.		
1623		000001		58000	.END			

BOX	18	660	687	715	1081	1102	1119	1171	1217	1230	1280	1296	1364	1398	1437
DCPARM	18														
DHDOC1	18														
DHPARM	18														
DJPARM	18														
DLPARM	18														
DMPARM	18	555													
DPPARM	18														
DQDOC1	18														
DQPARM	18														
DUPARM	18														
DUPPAR	18														
DVDOC1	18														
DVPARM	18														
DZPARM	18														
HELLO	18														
HLT	6598	816	849	882	913	954	974	1067							
SEQUAT	18	659													
SINTF	18	659													
SITEP	18	770													
SSERV	18	704													

. ABS. 017376 000

ERRORS DETECTED: 0

DZDMO,DZDMO/SOL/CRF+ITEP1,DZDMO
 RUN-TIME: 4 5 .3 SECONDS
 RUN-TIME RATIO: 104/10=9.9
 CORE USED: 17K (33 PAGES)