

DV11

DV11 OVERLAY FOR ITEP
CZDVOB0

AH-8753B-MC

COPYRIGHT © 76-78

FICHE 1 OF 1

MAR 1978

digital

MADE IN USA

The microfiche card contains a vertical column of frames on the left side. Each frame appears to contain a small table or data entry. The text within these frames is extremely small and difficult to read, but they seem to be organized in a structured manner, possibly representing a list of items or a data table. The frames are arranged in a regular grid pattern.

IDENTIFICATION

PRODUCT CODE: AC-8752B-MC
PRODUCT NAME: CZDVOB0 DV11 OVERLAY FOR ITEP
PROGRAM DATE: FEB 1978
MAINTAINER: DIAGNOSTICS
AUTHORS: R A JONES
JOHN EGOLF
RON PLATIKUS, J. VALDES

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

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

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

COPYRIGHT (C) 1976, 1978, 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 DV11 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 DMBB.
 - 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 DM11BB.
 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 DM-11B ISR
SWR13=SETUP DN-11 ISR
SWR=000000=SETUP VARIABLE ISR
 2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
SETUP SEQUENCE IS: DN11 DM11-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 DM11-BB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.)
 - F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DM.
 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 (\$A QUICK BROWN FOX)
10=1 TEST MESSAGE #2 (\$B NUMERICS)
11=1 TEST MESSAGE #3 (\$C COMTEST/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. ↑U 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=RRRRRR
DATA SHOULD BE TTTTTT
DATA COMPARE ERROR; BAD DATA=BBB GOOD DATA=GGG

WHERE RRRRRR 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<001> 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 DV11 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 D. 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: \$OWI, IF "ONE WAY IN" MODE WAS SELECTED. \$OWO, IF "ONE WAY OUT" MODE WAS SELECTED. \$ILB, IF "INTERNAL LOOP BACK" MODE WAS SELECTED. \$XLB, IF "EXTERNAL LOOP BACK" WAS SELECTED.

9.31 \$OWI: 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 \$OWO: 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 \$OWO OR TYPES END PASS DEPENDING ON SWITCH SETTINGS.

9.33 \$ILB: 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. (\$ILB)

9.34 \$XLB: 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 DBR 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.

DV11 RESTRICTIONS

IF A DM11BB EXISTS IN THE SYSTEM WITH THE DV11 BEING
TESTED, BUT MODEM CONTROL IS NOT DESIRED AND THE DM11BB
WAS NOT INITIALIZED BY ITEP, THE PROGRAM WILL HANG IN THE
DV11 TRANSMITTER INITIALIZATION ROUTINE. TO CORRECT THIS
LOAD LOCATION "DMBB" WITH AN ADDRESS THAT WILL TIME OUT (NO
SLAVE SYNC RESPONSE). THE ADDRESS OF DMBB CAN BE FOUND
IN THE CROSS REFERENCE TABLE IN THE BACK OF THIS LISTING.

570

10.0 PARAMETERS FOR THE DV11

PARAM#1 IS USED TO DETERMINE THE LINE NUMBERS FOR XMIT AND RCV.
BITS 03:00 XMIT AND RCV LINE NUMBER--DEFAULT TO LN# 0

PARAM#2 CONTAINS SPECIFIC LINE INFORMATION
BITS 15:08 CONTAINS SYNC CODE--DEFAULT =26

BIT 1	=1	USE SYNC B
	=0	USE SYNC A (DEFAULT)

BIT 0	=1	FULL DUPLEX
	=0	HALF DUPLEX (DEFAULT)

PARAM#3 IS NOT USED

```

590
591
592
593
594      011000
595      011000 053104 000040
596      011004 175000
597      011006 000300
598      011010 000240
599      011012 000000
600      011014 013000
601      011016 177777
602      011020 000000
603      011022 000000
604      011024 000000
605      011026 000000
606      011030 000000
607      011032 000000
608      011034 000000
609      011036 011102
610      011040
611      011040      000
612      011041
613      011041      001
614      011042 000000
615      011044 177570
616      011046 177570
617
618
619
620
621      000000
622      100000
623      040000
624      020000
625      020000
626
627      011050 000000
628      011052 000000
629      011054 000000
630      011056 000000
631      011060 000000
632
633      011062 000000
634      011064 000000
635      011066 000000
636      011070 000000
637
638      011072 177560
639      011074 177562
640      011076 177564
641      011100 177566
642
643      000001

```

```

*****
:      DV11 INTERFACE SERVICE PARAMS
*****

```

```

DV11:  .ASCIZ  /DV /      ; ISR NAME
BA:    175000      ; BUS ADDRESS
RIV:   300      ; VECTOR ADDRESS
PRIOR: 240      ; PRIORITY
PARAM1: 0      ; PARAM #1
PARAM2: 13000   ; PARAM #2
PARAM3: 177777  ; PARAM #3
IRDA:  .WORD    0      ; INITIAL READ DATA ADDRESS
IXDA:  .WORD    0      ; INITIAL XMIT DATA ADDRESS
SETTLE: .WORD    0      ; LINE SETTLE DELAY FLAG
B2016: .WORD    0      ; ADDR OF BIN TO OCT TYPE ROUTINE
TIME:  .WORD    0      ; TIMER
      .WORD    0      ;
      .WORD    0      ;
TX. TERM: .WORD  START ; ADDR OF START OF PROGRAM
RX. TERM: .BYTE  000    ; TRANSMITTER TERMINATING CHAR.
FLAG:    .BYTE  001    ; RECEIVER TERMINATING CHAR.
SWR:     177570
DISPLAY: 177570

```

```

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

```

```

STAT=RD
XFLG=100000      ; XMIT COMPLETE FLAG
RFLG=40000      ; RCV COMPLETE FLAG
DSFLG=20000     ; DATA SET STATUS CHANGE FLAG
BIT13=20000     ; INHIBIT PRINTOUTS
SXCSR:  0      ; SAVED XMIT CSR
SRCSR:  0      ; SAVED RCV CSR
ERCSR:  0      ; RCV CSR SAVED ON ERROR
ERDBR:  0      ; RCV DATA REG SAVED ON ERROR
DSSTAT: 0      ; RCV CSR SAVED ON DS CHANGE
XCC:    0      ; XMIT CHAR COUNT
RCC:    0      ; RCV CHAR COUNT
RDA:    0      ; RCV DATA ADDR.
XDA:    0      ; XMIT DATA ADDR.
TKS:    177560
TKB:    177562
TPS:    177564
TPB:    177566
FULL.DUPLEX=000001

```

```

644
645
646
647 011102 000240
648 011104 017700 177734
649 011110 042700 177400
650 011114 013702 011006
651 011120 012722 014016
652 011124 013722 011010
653 011130 012722 013436
654 011134 013722 011010
655 011140 013704 011004
656 011144 004737 015330
657 011150 004737 015402
658 011154 005214
659
660
661
662
663
664
665
666
667 011156 005037 011032
668 011162 005037 013056
669 011166 005037 013062
670 011172 032700 000C01
671 011176 001402
672 011200 000137 011354
673 011204 032700 000002
674 011210 001402
675 011212 000137 011246
676 011216 032700 000010
677 011222 001402
678 011224 000137 011452
679 011230 032700 000004
680 011234 001402
681 011236 000137 011702
682 011242 000000
683 011244 000776
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698 011246 104416
699 011250 004737 013620

```

```

:*****
: DV11-X INTERFACE SERVICE ROUTINE
:*****
START:  NOP
        MOV     2SWR, R0      ; SETUP MODE IN R0
        BIC     #177400, R0   ; STRIP JUNK
        MOV     RIV, R2      ; SETUP
        MOV     #RISR, (R2)+  ; INTERRUPT
        MOV     PRIOR, (R2)+ ; VECTORS
        MOV     #XISR, (R2)+
        MOV     PRIOR, (R2)+
        MOV     BA, R4       ; SETUP BUS ADDR INDEX
        JSR    PC, RAMCLR    ; CLEAR OUT RAM
        JSR    PC, SETUP    ; CALCULATE BYTE CNT AND SYNC
        INC    (R4)         ; START UCPU

```

```

:*****
: 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".
:*****

```

```

$OWI:   KBDIN
        JSR    PC, STARTR

```

```

700 011254 032700 040000 1$: BIT #RFLG,STAT
701 011260 001013 BNE 2$
702 011262 023727 011032 000100 CMP TIME,#100
703 011270 103771 BLO 1$
704 011272 011402 MOV @RCSR,R2
705 011274 016403 000000 MOV XCSR(R4),R3
706 011300 104001 HLT 1
707 011302 005037 011032 CLR TIME
708 011306 000762 BR 1$

```

```

709
710 011310 032777 000200 177526 2$: BIT #NODAT,@SWR
711 011316 001002 BNE 3$
712 011320 004737 012272 JSR PC,TESTD
713 011324 042700 040000 3$: BIC #RFLG,STAT
714 011330 032777 000020 177506 BIT #LOOP,@SWR
715 011336 001405 BEQ 4$
716 011340 012737 011352 013060 MOV #4$,BACK
717 011346 000137 012132 JMP EOP
718 011352 000735 4$: BR $OWI
719
720
721
722
723
724
725
726
727
728

```

```

:*****
: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."
:*****

```

```

729 011354 104416 $OWO: KBDIN
730 011356 004737 013064 JSR PC,STARTX
731 011362 005037 011032 CLR TIME
732 011366 032700 100000 1$: BIT #XFLG,STAT
733 011372 001013 BNE 2$
734 011374 023727 011032 000100 CMP TIME,#100
735 011402 103771 BLO 1$
736 011404 011402 MOV @RCSR,R2
737 011406 016403 000000 MOV XCSR(R4),R3
738 011412 104001 HLT 1
739 011414 005037 011032 CLR TIME
740 011420 000762 BR 1$
741 011422 042700 100000 2$: BIC #XFLG,STAT
742 011426 032777 000020 177410 BIT #LOOP,@SWR
743 011434 001405 BEQ 3$
744 011436 012737 011450 013060 MOV #3$,BACK
745 011444 000137 012132 JMP EOP
746 011450 000741 3$: BR $OWO
747
748
749

```



```

750
751
752
753
754
755
756
757
758
759
760
761 011452 104416
762 011454 004737 013620
763 011460 005037 011032
764 011464 032700 040000
765 011470 001013
766 011472 023727 011032 000100
767 011500 103771
768 011502 011402
769 011504 016403 000000
770 011510 104001
771 011512 005037 011032
772 011516 000762
773 011520 032777 000200 177316 2$:
774 011526 001002
775 011530 004737 012272
776 011534 042700 040000 3$:
777 011540 032777 000020 177276
778 011546 001405
779 011550 012737 011562 013060
780 011556 000137 012132
781 011562 032777 000400 177254 4$:
782 011570 001416
783 011572 013702 011020
784 011576 013703 011022
785 011602 010337 011070
786 011606 112223
787 011610 001376
788 011612 112743 000177
789 011616 005203
790 011620 112723 000177
791 011624 105023
792 011626 005037 011032 7$:
793 011632 004737 013064
794 011636 032700 100000 5$:
795 011642 001013
796 011644 023727 011032 000100
797 011652 103771
798 011654 011402
799 011656 016403 000000
800 011662 104001
801 011664 005037 011032
802 011670 000762
803 011672 042700 100000 6$:
804 011676 000137 011452

```

```

*****
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.
*****

```

```

$ILB: KBDIN
JSR PC, STARTR
CLR TIME
1$: BIT #RFLG, STAT
BNE 2$
CMP TIME, #100
BLO 1$
MOV @RCSR, R2
MOV XCSR(R4), R3
HLT 1
CLR TIME
BR 1$
2$: BIT #NODAT, @SWR
BNE 3$
JSR PC, TESTD
3$: BIC #RFLG, STAT
BIT #LOOP, @SWR
BEQ 4$
MOV #4$, BACK
4$: JMP EOP
BIT #400, @SWR ;USE EXTERNAL DATA?
BEQ 7$ ;BR IF NO
MOV IRDA, R2 ;SET POINTER
MOV IXDA, R3 ;SET POINTER
MOV R3, XDA ;SETUP XMIT DATA ADDR
MOVB (R2)+, (R3)+ ;MOVE INPUT TO OUTPUT
BNE -2 ;LOOP IF NOT ZERO CHAR
MOVB #177, -(R3) ;INSERT A FILL CHAR
INC R3 ;BUMP ADDRESS
MOVB #177, (R3)+ ;INSERT ANOTHER FILL
CLRB (R3)+ ;INSERT ZERO CHAR
7$: CLR TIME
JSR PC, STARTX
5$: BIT #XFLG, STAT
BNE 6$
CMP TIME, #100
BLO 5$
MOV @RCSR, R2
MOV XCSR(R4), R3
HLT 1
CLR TIME
BR 5$
6$: BIC #XFLG, STAT
JMP $ILB

```

```

805
806
807
808
809
810
811
812
813
814
815
816
817
818 011702 104416
819 011704 032737 000001 011014
820 011712 001402
821 011714 004737 013620
822 011720 004737 013064
823 011724 005037 011032
824 011730 032700 100000
825 011734 0J1016
826 011736 032700 040000
827 011742 001024
828 011744 023727 011032 000100
829 011752 103766
830 011754 011402
831 011756 016403 000000
832 011762 104001
833 011764 005037 011032
834 011770 000757
835 011772 032737 000001 011014
836 012000 001356
837 012002 042700 100000
838 012006 004737 013620
839 012012 000746
840 012014 032737 000001 011014
841 012022 001420
842 012024 032700 100000
843 012030 001013
844 012032 023727 011032 000100
845 012040 103765
846 012042 011402
847 012044 016403 000000
848 012050 104001
849 012052 005037 011032
850 012056 000756
851 012060 042700 100000
852 012064 042700 040000
853 012070 005037 011032
854 012074 032777 000200 176742
855 012102 001002
856 012104 004737 012272
857 012110 032777 000020 176726
858 012116 001671
859 012120 012737 011702 013060
860 012126 000137 012132

```

```

*****
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.
*****

```

```

      3: KBDIN
      BIT #FULL.DUPLEX,PARAM2
      BEQ 1$
      JSR PC,STARTR
      1$: JSR PC,STARTX
      CLR TIME
      2$: BIT #XFLG,STAT
      BNE 3$
      7$: BIT #RFLG,STAT
      BNE 4$
      CMP TIME,#100
      BLO 2$
      MOV @RCSR,R2
      MOV XCSR(R4),R3
      HLT 1
      CLR TIME
      BR 2$
      3$: BIT #FULL.DUPLEX,PARAM2
      BNE 7$
      BIC #XFLG,STAT
      JSR PC,STARTR
      BR 2$
      4$: BIT #FULL.DUPLEX,PARAM2
      BEQ 8$
      BIT #XFLG,STAT
      BNE 6$
      CMP TIME,#100
      BLO 4$
      MOV @RCSR,R2
      MOV XCSR(R4),R3
      HLT 1
      CLR TIME
      BR 4$
      6$: BIC #XFLG,STAT
      8$: BIC #RFLG,STAT
      CLR TIME
      BIT #NODAT,@SWR
      BNE 5$
      JSR PC,TESTD
      5$: BIT #LOOP,@SWR
      BEQ $XLB
      MOV $XLB,BACK
      JMP EOP

```

```

861
862
863
864
865
866
867 012132
868 012132 104414 000340
869 012136 016437 000000 012270
870 012144 042737 157777 012270
871 012152 042764 020000 000000
872 012160 012766 012220 000002
873 012166 010037 013042
874 012172 010137 013044
875 012176 010237 013046
876 012202 010337 013050
877 012206 010437 013052
878 012212 010537 013054
879 012216 000207
880
881 012220
882 012220 013700 013042
883 012224 013701 013044
884 012230 013702 013046
885 012234 013703 013050
886 012240 013704 013052
887 012244 013705 013054
888 012250 012737 177777 013056
889 012256 053764 012270 000000
890 012264 000177 000570
891 012270 000000
892
893
894
895
896
897
898
899 012272 013746 011056
900 012276 001413
901 012300 032777 020000 176536
902 012306 001007
903 012310 104400 012502
904 012314 004077 176510
905 012320 005746
906 012322 104400 012563
907 012326 013701 011022
908 012332 013702 011020
909 012336 122122
910 012340 001776
911 012342 123741 011040
912 012346 001453
913 012350 122742 000002
914 012354 001005
915 012356 010237 012364
916 012362 104400

```

```

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

```

```

EOP: STPS,PTY7 ;SET PS PRIORITY TO 7
MOV XCSR(R4),QTPIE ;SAVE TX CSR
BIC #C<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
BIS QTPIE,XCSR(R4) ;IF ORIGINALLY 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 2$
MOV R2,1$
TYPE

```

```

917 012364 000000      1$: .WORD 0
918 012366 000443      BR TESTDX
919 012370      2$:
920 012370 105712      TSTB (R2)
921 012372 001441      BEQ TESTDX ;BR IF YES
922 012374 122721 000177  CMPB #177 (R1)+ ;IS IT FILL CHAR?
923 012400 001756      BEQ SCAN4 ;BR IF YES
924 012402 005301      DEC R1 ;BACKUP
925 012404 122722 000177  CMPB #177 (R2)+ ;IS IT FILL?
926 012410 001752      BEQ SCAN4 ;BR IF YES
927 012412 105742      TSTB -(R2) ;BACK UP POINTER
928 012414 123722 011015  CMPB PARAM2+1,(R2)+ ;WAS SYNC CHAR IN BUFFER?
929 012420 001746      BEQ SCAN4 ;BR IF CHAR WAS SYNC
930 012422 000240      SCANS: NOP ;DATA ERROR
931 012424 032777 020000 176412  BIT #BIT13,ASWR ;INHIBIT PRINTOUTS
932 012432 001016      BNE DERR ;BR IF YES
933 012434 104400 012566      TYPE MSG2 ;<15><12>RECEIVED DATA = <15><12>
934 012440 013737 011020 012450  MOV IRDA, RDAX ;SETUP DATA ADDRESS
935 012446 104400      TYPE ;PRINT RECEIVED DATA
936 012450 000000      RDAX: 0 ;RECEIVED DATA ADDR.
937 012452 104400 012613      TYPE MSG3 ;<15><12>DATA SHOULD BE<15><12>
938 012456 013737 011022 012466  MOV IXDA, .+10 ;SETUP ADDR.
939 012464 104400      TYPE ;PRINT GOOD DATA
940 012466 011022      IXDA
941 012470 111103      DERR: MOVB (R1),R3 ;SETUP XMIT DATA
942 012472 114202      MOVB -(R2),R2 ;SETUP RCV DATA
943 012474 104007      HLT+7 ;DATA ERROR HALT
944 012476 005726      TESTDX: TST (SP)+ ;POP STACK
945 012500 000207      RTS ;RETURN FROM SUB/ROUT
946
947 012502 005015 044124 051105  MSG0: .ASCIZ <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
(1) 012563 015 000012  MSG1: .ASCIZ <15><12>
(1) 012566 005015 042522 042503  MSG2: .ASCIZ <15><12>/RECEIVED DATA = /<15><12>
(1) 012613 015 042012 052101  MSG3: .ASCIZ <15><12>/DATA SHOULD BE/<15><12>
(1) 012636 005015 046120 040505  MSG4: .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
(1) 012705 015 053412 042510  MSG4: .ASCIZ <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>
(1) 012770 005015 046120 040505  MSG5: .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
(1)
948 013042 000000      SAVRO: 0
949 013044 000000      SAVR1: 0
950 013046 000000      SAVR2: 0
951 013050 000000      SAVR3: 0
952 013052 000000      SAVR4: 0
953 013054 000000      SAVR5: 0
954 013056 000000      DELAY: 0
955 013060 000000      BACK: 0
956 013062 000000      STOP: 0

```

```

957 ;*****
958 ; TRANSMITTER INIT ROUTINE
959 ;*****
960
961 013064 042700 100000 STARTX: BIC #XFLG,STAT ;CLEAR XMIT DONE FLAG
962 013070 005737 013056 TST DELAY
963 013074 001415 BEQ 2$
964 013076 005037 014706 CLR TEMP1
965 013102 012737 000007 014710 MOV #7,TEMP2
966 013110 005237 014706 1$: INC TEMP1
967 013114 001375 BNE 1$
968 013116 005337 014710 DEC TEMP2
969 013122 001372 BNE 1$
970 013124 005037 013056 CLR DELAY
971 013130 005037 011032 2$: CLR TIME
972 013134 013764 011012 000020 MOV PARAM1,20(R4)
973 013142 013764 011012 000006 MOV PARAM1,6(R4) ;SELECT LINE #
974 013150 112764 000000 000007 MOVB #TPCA,7(R4) ;LOAD TPCA WITH SYNC ADDRESS
975 013156 012764 014720 000010 MOV #SYNC,10(R4)
976 013164 112764 000001 000007 MOVB #TPBC,7(R4) ;LOAD TPBC WITH # OF SYNCs
977 013172 012764 177772 000010 MOV #-6,10(R4)
978 013200 112764 000002 000007 MOVB #TACA,7(R4) ;LOAD TACA WITH MESSAGE ADDRESS
979 013206 013764 011022 000010 MOV IXDA,10(R4)
980 013214 112764 000003 000007 MOVB #TABC,7(R4) ;LOAD TABC WITH MESSAGE BYTE COUNT
981 013222 013764 014716 000010 MOV BCNT,10(R4)
982 013230 112764 000012 000007 MOVB #LPP,7(R4)
983 013236 012764 000100 000010 MOV #100,10(R4) ;SET DDCMP MODE XMIT
984 013244 032737 000002 011014 BIT #BIT1,PARAM2 ;USE SYNC A OR SYNC B
985 013252 001406 BEQ 12$ ;DEFAULT TO SYNC A
986 013254 052764 102000 000004 BIS #BIT10+BIT15,4(R4) ;SETUP FOR SYNC B
987 013262 005764 000004 13$: TST 4(R4) ;WAIT FOR CONTROL STROBE
988 013266 100775 BMI 13$
989 013270 052764 000003 000022 12$: BIS #BIT0+BIT1,22(R4) ;TERMINAL READY, LINE ENABLE
990 013276 005737 013062 TST STOP
991 013302 001005 BNE 6$
992 013304 104400 012636 TYPE ,MSG4
993 013310 000000 HALT ;WAIT FOR CONNECTION TO BE MADE
994 013312 005137 013062 COM STOP
995 013316 032737 000001 011014 6$: BIT #FULL.DUPLEX,PARAM2 ;HALF OR FULL DUPLEX?
996 013324 001006 BNE 8$ ;BRANCH IF FULL
997 013326 032764 000100 000022 7$: BIT #100,22(R4) ;IS CARRIER ON
998 013334 001374 BNE 7$ ;WAIT FOR CARRIER TO DIE
999 013336 005037 011032 CLR TIME
1000 013342 052764 000004 000022 8$: BIS #BIT2,22(R4) ;SET RQTS
1001 013350 032764 000040 000022 9$: BIT #BIT5,22(R4) ;IS CTS UP YET
1002 013356 001016 BNE 11$ ;YES
1003 013360 023727 011032 000036 CMP TIME,#36
1004 013366 103770 BLO 9$
1005 013370 005002 CLR R2 ;DONT PRINT OUT
1006 013372 005003 CLR R3
1007 013374 032777 010000 175442 BIT #SW12,2SWR ;INHIBIT PRINTOUT
1008 013402 001001 BNE 10$ ;YES
1009 013404 104002 HLT 2 ;TYPE WAITING TO TRANSMIT
1010 013406 005037 011032 10$: CLR TIME ;CLEAR TIMER
1011 013412 000756 BR 9$ ;WAIT FOR CTS
1012 013414 052714 030000 11$: BIS #30000,(R4) ;GOT CTS ON TRANSMIT

```

```

1013 013420 112764 000013 000007      MOVB  #LS,7(R4)      ;POINT TO LINE STATE REG.
1014 013426 052764 000004 000010      BIS   #4,10(R4)     ;SET XMIT GO
1015 013434 000207      RTS   PC
1016
1017
1018
1019
1020
1021 013436 000240      XISR:  NOP
1022 013440 016437 000014 014706      MOV   14(R4),TEMP1  ;READ NSR
1023 013446 005737 014706      TST   TEMP1         ;VALID DATA
1024 013452 100046      BPL   4$            ;NO UNEXPECTED INTERRUPT
1025 013454 032737 000400 014706      BIT   #BIT8,TEMP1   ;IS XMIT DONE
1026 013462 001430      BEQ   1$            ;NO MUST BE ERROR
1027 013464 032737 001000 014706      BIT   #BIT9,TEMP1   ;PRINCIPAL OR ALTER?
1028 013472 001447      BEQ   3$            ;PRINCIPAL DONE-SYNCS OUT
1029 013474 052700 100000      BIS   #XFLG,STAT    ;SET XMIT DONE FLAG
1030 013500 032737 000001 011014      BIT   #FULL-DUPLEX,PARAM2 ;1/2 OR FULL DUPLEX
1031 013506 001003      BNE   6$            ;BRANCH IF FULL DUPLEX
1032 013510 042764 000004 000022      BIC   #BIT2,22(R4)  ;CLEAR RQTS
1033 013516 032777 000100 175320 6$:      BIT   #BIT6,ASWR    ;MONITOR DATA?
1034 013524 001432      BEQ   3$            ;NO-EXIT
1035 013526 105777 175344      TSTB  #TPS          ;TTY READY
1036 013532 100027      BPL   3$            ;CAN'T WAIT-EXIT
1037 013534 112777 000124 175336      MOVB  #'T,#TPB     ;TYPE "T" FOR TRANSMIT
1038 013542 000423      BR    3$
1039 013544 005002      CLR   R2           ;NO RCV CSR
1040 013546 013703 014706      MOV   TEMP1,R3     ;TYPE OUT NSR
1041 013552 032703 007400      BIT   #BIT8+BIT9+BIT10+BIT11,R3 ;ERROR ON PRINCIPAL CAR
1042 013556 001002      BNE   2$            ;NO ON ALT
1043 013560 104012      HLT   12           ;TELL OPERATOR OF ERROR NXM PRIN CAR
1044 013562 000403      BR    5$           ;EXIT
1045 013564 104013      HLT   13           ;NXM ALT CAR
1046 013566 000401      BR    5$
1047 013570 104011      HLT   11           ;UNEXPECTED INTERRUPT
1048 013572 112764 000013 000007 5$:      MOVB  #LS,7(R4)     ;POINT TO SECONDARY LS REGISTER
1049 013600 042764 000060 000010      BIC   #BIT4+BITS,10(R4) ;CLEAR ERROR BITS
1050 013606 000137 013064      JMP   STARTX       ;TRY AGAIN
1051 013612 005037 011032 3$:      CLR   TIME
1052 013616 000002      RTI
1053

```

1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109

013620 013764 011012 000020
013626 013764 011012 000006
013634 042700 040000
013640 112764 000004 000007
013646 013764 011020 000010
013654 112764 000005 000007
013662 012764 177000 000010
013670 112764 000011 000007
013676 012764 014726 000010
013704 112764 000012 000007
013712 012764 000002 000010
013720 112764 000015 000007
013726 105064 000010

013732 052764 000002 000022
013740 005737 013062
013744 001013
013746 104400 012770
013752 005137 013062
013756 032737 000002 011014
013764 001403
013766 052764 002000 000004
013774 052764 120000 000004
014002 005764 000004
014006 100775
014010 052714 000100
014014 000207

RECEIVE INIT ROUTINE

```
STARTR: MOV PARAM1,20(R4)
MOV PARAM1,6(R4)
BIC #RFLG,STAT ;CLEAR RCV DONE FLAG
MOVB #RCA,7(R4) ;POINT TO RCV CURRENT ADDRESS REG
MOV IRDA,10(R4) ;LOAD IT WITH RCV BUFF ADD
MOVB #RBC,7(R4) ;POINT TO RCV BYTE COUNT REG
MOV #-1000,10(R4) ;LOAD IT
MOVB #RCTBA,7(R4) ;POINT TO RCV CONTROL TABLE REG
MOV #CORTAB,10(R4) ;LOAD IT
MOVB #LPP,7(R4) ;POINT TO LINE PROTOCOL REG
MOV #2,10(R4) ;SET STRIP SYNC
MOVB #RMB,7(R4) ;POINT TO RCV MODE REG
CLRB 10(R4) ;MODE 0

BIS #DTR,22(R4) ;SET DATA TERMINAL READY
TST STOP ;SEE IF FIRST TIME
BNE 1$
TYPE MSGS ;TYPE MESSAGE
COM STOP
BIT #BIT1,PARAM2 ;SYNC A OR SYNC B
BEQ 1$ ;DEFAULT TO SYNC A
BIS #BIT10,4(R4) ;SET SYNC B
1$: BIS #BIT15+BIT13,4(R4) ;SET RCV ENABLE+CONTROL STROBE
2$: TST 4(R4) ;LOOP ON CONTROL
BMI 2$ ;STROBE TO SETTLE
BIS #BIT6,(R4) ;SET INT ENABLE
RTS PC ;EXIT
```

RCV SERVICE ROUTINE

```
RISR: NOP ;SPARE
MOV 2(R4),TEMP1 ;SAVE RIC REGISTER
BIT #170000,TEMP1 ;CHECK FOR SPECIAL CHARACTER INTR
BNE 1$ ;NO-BRANCH
CMPB TEMP1,RX.TERM ;WAS IT TERM CHARACTER
BNE 1$ ;NO-BRANCH
BIT #BITS,ASWR ;MONITOR RCV DATA
BEQ 2$ ;NO
TSTB @TPS
BPL 2$
MOV #R,@TPB ;TYPE "R" FOR RCV
2$: BIS #RFLG,STAT ;SET RCV DONE FLAG
BIS #BIT8,(R4) ;SET RCV INT SRV COMPLETE
NOP
NOP
BIC #BIT6,(R4) ;RESET RCV INT. ENABLE
MOV #BIT15,4(R4) ;TURN OFF RECV.
5$: TST 4(R4) ;WAIT FOR CONTROL STROB
BMI 5$ ;TO CLEAR
```

1110	014126	112764	000013	000007	MOVB	#LS,7(R4)	;POINT TO LS REG
1111	014134	012764	000002	000010	MOV	#EIT1,10(R4)	;SET RCV RESYNC
1112	014142	000002			RTI		;EXIT
1113	014144	005003			1\$: CLR	R3	
1114	014146	013702	014706		MOV	TEMP1,R2	
1115	014152	004737	014206		JSR	PC,8\$	
1116	014156	104400	014621		TYPE	,FATAL	
1117	014162	104400			TYPE		
1118	014164	000000			4\$: 000000		
1119	014166	104000			HLT	0	
1120	014170	023737	000006	014706	CMP	6,TEMP1	
1121	014176	002335			BGE	2\$	
1122	014200	104400	014655		TYPE	,NOREC	
1123	014204	000000			HALT		
1124	014206	005046			8\$: CLR	-(SP)	;CLEAR LOCATION ON STACK
1125	014210	116416	000003		MOVB	3(R4),(SP)	;GET HIGH - BYTE OF RIC
1126	014214	042716	000017		BIC	#17,(SP)	;CLEAR LINE NUMBER
1127	014220	006016			ROR	(SP)	;ROTATE UNTIL (INT CODE)X2
1128	014222	006016			ROR	(SP)	;IN LOW BYTE
1129	014224	111637	014706		MOVB	(SP),TEMP1	;SAVE FOR LATTER
1130	014230	062716	014242		ADD	#ERRTAB,(SP)	;GET OFFSET
1131	014234	012637	014164		MOV	(SP)+,4\$;MAKE ADDRESS OF MSG
1132	014240	000207			RTS	PC	;EXIT
1133							


```

1134
1135
1136
1137
1138
1139 014242 014302
1140 014244 014327
1141 014246 014353
1142 014250 014373
1143 014252 014431
1144 014254 014577
1145 014256 014577
1146 014260 014577
1147 014262 014431
1148 014264 014577
1149 014266 014452
1150 014270 014577
1151 014272 014577
1152 014274 014467
1153 014276 014515
1154 014300 014540
1155
1156 014302 005015 054122 052056
1157 014310 051105 020115 047516
1158 014316 020124 047125 050511
1159 014324 042525 000
1160 014327 015 041412 040510
1161 014334 020122 040520 044522
1162 014342 054524 042440 051122
1163 014350 051117 000
1164 014353 015 047412 042526
1165 014360 051122 047125 042440
1166 014366 051122 051117 000
1167 014373 015 041412 040510
1168 014400 020122 040520 044522
1169 014406 054524 042440 051122
1170 014414 051117 025440 047440
1171 014422 042526 051122 047125
1172 014430 000
1173 014431 015 051012 053103
1174 014436 041040 052131 020105
1175 014444 047103 036524 000060
1176 014452 005015 054116 020115
1177 014460 047111 051040 040503
1178 014466 000
1179 014467 015 047012 046530
1180 014474 044440 020116 047503
1181 014502 052116 047522 020114
1182 014510 054502 042524 000
1183 014515 015 046412 046505
1184 014522 050040 051101 052111
1185 014530 020131 051105 047522
1186 014536 000122
1187 014540 005015 040520 044522
1188 014546 054524 042440 051122
1189 014554 051117 044440 020116

```

```

*****
: ERROR MESSAGE TABLES
*****

```

```

ERRTAB: EMO
        EM1
        EM2
        EM3
        EM4
        UNDF
        UNDF
        UNDF
        EM4
        UNDF
        EM12
        UNDF
        UNDF
        EM15
        EM16
        EM17

```

```

EMO: .ASCIZ <15><12>/RX.TERM NOT UNIQUE/
EM1: .ASCIZ <15><12>/CHAR PARITY ERROR/
EM2: .ASCIZ <15><12>/OVERRUN ERROR/
EM3: .ASCIZ <15><12>/CHAR PARITY ERROR + OVERRUN/
EM4: .ASCIZ <15><12>/RCV BYTE CNT=0/
EM12: .ASCIZ <15><12>/NXM IN RCA/
EM15: .ASCIZ <15><12>/NXM IN CONTROL BYTE/
EM16: .ASCIZ <15><12>/MEM PARITY ERROR/
EM17: .ASCIZ <15><12>/PARITY ERROR IN CONTROL BYTE/

```

1190	014562	047503	052116	047522
1191	014570	020114	054502	042524
1192	014576	000		
1193	014577	015	052412	042116
1194	014604	043105	047111	042105
1195	014612	042440	051122	051117
1196	014620	000		
1197	014621	015	042412	051122
1198	014626	051117	051040	041511
1199	014634	030440	035065	031061
1200	014642	044440	042116	041511
1201	014650	052101	051505	000
1202	014655	015	043012	052101
1203	014662	046101	047040	047117
1204	014670	051055	041505	053117
1205	014676	044055	046101	020524
1206	014704	000		
1207				
1208		014706		
1209	014706	000000		
1210	014710	000000		
1211	014712	000000		
1212	014714	000000		
1213	014716	000000		
1214	014720	000000		
1215	014722	000000		
1216	014724	000000		
1217				

```

UNDF: .ASCIZ <15><12>/UNDEFINED ERROR/
FATAL: .ASCIZ <15><12>/ERROR RIC 15:12 INDICATES/
NOREC: .ASCIZ <15><12>/FATAL NON-RECOV-HALT!/

.EVEN
TEMP1: 0
TEMP2: 0
TEMP3: 0
TEMP4: 0
BCNT: 0
SYNC: 0

```


1274	015012	000	.BYTE	0
1275	015013	000	.BYTE	0
1276	015014	000	.BYTE	0
1277	015015	000	.BYTE	0
1278	015016	000	.BYTE	0
1279	015017	000	.BYTE	0
1280	015020	000	.BYTE	0
1281	015021	000	.BYTE	0
1282	015022	000	.BYTE	0
1283	015023	000	.BYTE	0
1284	015024	000	.BYTE	0
1285	015025	000	.BYTE	0
1286	015026	000	.BYTE	0
1287	015027	000	.BYTE	0
1288	015030	000	.BYTE	0
1289	015031	000	.BYTE	0
1290	015032	000	.BYTE	0
1291	015033	000	.BYTE	0
1292	015034	000	.BYTE	0
1293	015035	000	.BYTE	0
1294	015036	000	.BYTE	0
1295	015037	000	.BYTE	0
1296	015040	000	.BYTE	0
1297	015041	000	.BYTE	0
1298	015042	000	.BYTE	0
1299	015043	000	.BYTE	0
1300	015044	000	.BYTE	0
1301	015045	000	.BYTE	0
1302	015046	000	.BYTE	0
1303	015047	000	.BYTE	0
1304	015050	000	.BYTE	0
1305	015051	000	.BYTE	0
1306	015052	000	.BYTE	0
1307	015053	000	.BYTE	0
1308	015054	000	.BYTE	0
1309	015055	000	.BYTE	0
1310	015056	000	.BYTE	0
1311	015057	000	.BYTE	0
1312	015060	000	.BYTE	0
1313	015061	000	.BYTE	0
1314	015062	000	.BYTE	0
1315	015063	000	.BYTE	0
1316	015064	000	.BYTE	0
1317	015065	000	.BYTE	0
1318	015066	000	.BYTE	0
1319	015067	000	.BYTE	0
1320	015070	000	.BYTE	0
1321	015071	000	.BYTE	0
1322	015072	000	.BYTE	0
1323	015073	000	.BYTE	0
1324	015074	000	.BYTE	0
1325	015075	000	.BYTE	0
1326	015076	000	.BYTE	0
1327	015077	000	.BYTE	0
1328	015100	000	.BYTE	0
1329	015101	000	.BYTE	0

1330	015102	000	.BYTE	0
1331	015103	000	.BYTE	0
1332	015104	000	.BYTE	0
1333	015105	000	.BYTE	0
1334	015106	000	.BYTE	0
1335	015107	000	.BYTE	0
1336	015110	000	.BYTE	0
1337	015111	000	.BYTE	0
1338	015112	000	.BYTE	0
1339	015113	000	.BYTE	0
1340	015114	000	.BYTE	0
1341	015115	000	.BYTE	0
1342	015116	000	.BYTE	0
1343	015117	000	.BYTE	0
1344	015120	000	.BYTE	0
1345	015121	000	.BYTE	0
1346	015122	000	.BYTE	0
1347	015123	000	.BYTE	0
1348	015124	000	.BYTE	0
1349	015125	000	.BYTE	0
1350	015126	000	.BYTE	0
1351	015127	000	.BYTE	0
1352	015130	000	.BYTE	0
1353	015131	000	.BYTE	0
1354	015132	000	.BYTE	0
1355	015133	000	.BYTE	0
1356	015134	000	.BYTE	0
1357	015135	000	.BYTE	0
1358	015136	000	.BYTE	0
1359	015137	000	.BYTE	0
1360	015140	000	.BYTE	0
1361	015141	000	.BYTE	0
1362	015142	000	.BYTE	0
1363	015143	000	.BYTE	0
1364	015144	000	.BYTE	0
1365	015145	000	.BYTE	0
1366	015146	000	.BYTE	0
1367	015147	000	.BYTE	0
1368	015150	000	.BYTE	0
1369	015151	000	.BYTE	0
1370	015152	000	.BYTE	0
1371	015153	000	.BYTE	0
1372	015154	000	.BYTE	0
1373	015155	000	.BYTE	0
1374	015156	000	.BYTE	0
1375	015157	000	.BYTE	0
1376	015160	000	.BYTE	0
1377	015161	000	.BYTE	0
1378	015162	000	.BYTE	0
1379	015163	000	.BYTE	0
1380	015164	000	.BYTE	0
1381	015165	000	.BYTE	0
1382	015166	000	.BYTE	0
1383	015167	000	.BYTE	0
1384	015170	000	.BYTE	0
1385	015171	000	.BYTE	0

1386	015172	000	.BYTE	0
1387	015173	000	.BYTE	0
1388	015174	000	.BYTE	0
1389	015175	000	.BYTE	0
1390	015176	000	.BYTE	0
1391	015177	000	.BYTE	0
1392	015200	000	.BYTE	0
1393	015201	000	.BYTE	0
1394	015202	000	.BYTE	0
1395	015203	000	.BYTE	0
1396	015204	000	.BYTE	0
1397	015205	000	.BYTE	0
1398	015206	000	.BYTE	0
1399	015207	000	.BYTE	0
1400	015210	000	.BYTE	0
1401	015211	000	.BYTE	0
1402	015212	000	.BYTE	0
1403	015213	000	.BYTE	0
1404	015214	000	.BYTE	0
1405	015215	000	.BYTE	0
1406	015216	000	.BYTE	0
1407	015217	000	.BYTE	0
1408	015220	000	.BYTE	0
1409	015221	000	.BYTE	0
1410	015222	000	.BYTE	0
1411	015223	000	.BYTE	0
1412	015224	000	.BYTE	0
1413	015225	000	.BYTE	0
1414	015226	000	.BYTE	0
1415	015227	000	.BYTE	0
1416	015230	000	.BYTE	0
1417	015231	000	.BYTE	0
1418	015232	000	.BYTE	0
1419	015233	000	.BYTE	0
1420	015234	000	.BYTE	0
1421	015235	000	.BYTE	0
1422	015236	000	.BYTE	0
1423	015237	000	.BYTE	0
1424	015240	000	.BYTE	0
1425	015241	000	.BYTE	0
1426	015242	000	.BYTE	0
1427	015243	000	.BYTE	0
1428	015244	000	.BYTE	0
1429	015245	000	.BYTE	0
1430	015246	000	.BYTE	0
1431	015247	000	.BYTE	0
1432	015250	000	.BYTE	0
1433	015251	000	.BYTE	0
1434	015252	000	.BYTE	0
1435	015253	000	.BYTE	0
1436	015254	000	.BYTE	0
1437	015255	000	.BYTE	0
1438	015256	000	.BYTE	0
1439	015257	000	.BYTE	0
1440	015260	000	.BYTE	0
1441	015261	000	.BYTE	0

1442	015262	000	.BYTE	0
1443	015263	000	.BYTE	0
1444	015264	000	.BYTE	0
1445	015265	000	.BYTE	0
1446	015266	000	.BYTE	0
1447	015267	000	.BYTE	0
1448	015270	000	.BYTE	0
1449	015271	000	.BYTE	0
1450	015272	000	.BYTE	0
1451	015273	000	.BYTE	0
1452	015274	000	.BYTE	0
1453	015275	000	.BYTE	0
1454	015276	000	.BYTE	0
1455	015277	000	.BYTE	0
1456	015300	000	.BYTE	0
1457	015301	000	.BYTE	0
1458	015302	000	.BYTE	0
1459	015303	000	.BYTE	0
1460	015304	000	.BYTE	0
1461	015305	000	.BYTE	0
1462	015306	000	.BYTE	0
1463	015307	000	.BYTE	0
1464	015310	000	.BYTE	0
1465	015311	000	.BYTE	0
1466	015312	000	.BYTE	0
1467	015313	000	.BYTE	0
1468	015314	000	.BYTE	0
1469	015315	000	.BYTE	0
1470	015316	000	.BYTE	0
1471	015317	000	.BYTE	0
1472	015320	000	.BYTE	0
1473	015321	000	.BYTE	0
1474	015322	000	.BYTE	0
1475	015323	000	.BYTE	0
1476	015324	000	.BYTE	0
1477	015325	000	.BYTE	0
1478	015326	000	.BYTE	0
1479	015327	000	.BYTE	0

```

*****
: DV11 RAM CLEAR ROUTINE
*****

```

1484				
1485	015330	012714	004000	
1486	015334	010246		
1487	015336	010346		
1488	015340	012703	000017	
1489	015344	012702	000017	
1490	015350	110264	000007	
1491	015354	110364	000006	
1492	015360	005064	000010	
1493	015364	005302		
1494	015366	100370		
1495	015370	005303		
1496	015372	100364		
1497	015374	012603		

```

RAMCLR: MOV      #4000,(R4)      ;CLEAR PRIMARY REGS
        MOV      R2,-(SP)      ;SAVE R2
        MOV      R3,-(SP)      ;SAVE R3
1$:     MOV      #17,R3        ;SET UP LINE # COUNT IN R3
2$:     MOV      #17,R2        ;SET UP REGISTER # COUNT IN R2
        MOVVB   R2,7(R4)      ;SET UP SRS REGISTER
        MOVVB   R3,6(R4)
        CLR     10(R4)        ;CLEAR IT
        DEC     R2            ;FIRST CLEAR ALL REGS. FOR LN #
        BPL    2$
        DEC     R3
        BPL    1$
        MOV     (SP)+,R3      ;RESTORE R3

```

1498	015376	012602		MOV	(SP)+,R2	;RESTORE R2
1499	015400	000207		RTS	PC	;EXIT
1500						
1501	015402	010146		SETUP: MOV	R1,-(SP)	;SAVE R1
1502	015404	010046		MOV	RO,-(SP)	;SAVE RO
1503	015406	013700	011022	MOV	IXDA,RO	
1504	015412	005001		CLR	R1	
1505	015414	123720	011040	3\$: CMPB	TX.TERM,(RO)+	;FIGURE OUT BYTE COUNT-
1506	015420	001402		BEB	4\$;OF MESSAGE TO BE-
1507	015422	005201		INC	R1	;TRANSMITTED
1508	015424	000773		BR	3\$	
1509	015426	010137	014716	4\$: MOV	R1,BCNT	
1510	015432	005437	014716	NEG	BCNT	
1511	015436	012700	014720	MOV	#SYNC,RO	;SET UP CORE LABEL OF
1512	015442	012701	000006	MOV	#6,R1	;SYNC CHARACTERS FOUND
1513	015446	113720	011015	5\$: MOVB	PARAM2+1,(RO)+	;IN HIGH-BYTE OF PARAM2
1514	015452	005301		DEC	R1	
1515	015454	001374		BNE	5\$	
1516	015456	012600		MOV	(SP)+,RO	;RESTORE RO
1517	015460	012601		MOV	(SP)+,R1	;RESTORE R1
1518	015462	000207		RTS	PC	;EXIT
1519		000001		.END		

BA	011004	596#	655						
BACK	013060	716*	744*	779*	859*	890	954#		
BCNT	014716	981	1213#	1509*	1510*				
BIT0	= 000001	590#	989						
BIT1	= 000002	590#	984	989	1078	1111			
BIT10	= 002000	590#	986	1041	1080				
BIT11	= 004000	590#	1041						
BIT12	= 010000	590#							
BIT13	= 020000	590#	625#	901	931	1081			
BIT14	= 040000	590#							
BIT15	= 100000	590#	986	1081	1107				
BIT2	= 000004	590#	1000	1032					
BIT3	= 000010	590#							
BIT4	= 000020	590#	1049						
BIT5	= 000040	590#	1001	1049	1097				
BIT6	= 000100	590#	1033	1084	1106				
BIT7	= 000200	590#							
BIT8	= 000400	590#	1025	1041	1103				
BIT9	= 001000	590#	1027	1041					
B2016	011030	606#	904						
CORTAB	014726	1067	1222#						
DELAY	013056	668*	888*	953#	962	970*			
DERR	012470	932	941#						
DISPLA	011046	616#							
DSFLG	= 020000	590#	624#						
DSSTAT	011060	631#							
DTR	= 000002	590#	1073						
DV11	011000	595#							
EM0	014302	1139	1156#						
EM1	014327	1140	1160#						
EM12	014452	1149	1176#						
EM15	014467	1152	1179#						
EM16	014515	1153	1183#						
EM17	014540	1154	1187#						
EM2	014353	1141	1164#						
EM3	014373	1142	1167#						
EM4	014431	1143	1147	1173#					
ENTER	012220	872	881#						
EOP	012132	717	745	780	860	867#			
ERCSR	011054	629#							
ERDBR	011056	630#	899						
ERRTAB	014242	1130	1139#						
FATAL	014621	1116	1197#						
FLAG	011042	614#							
FULL.D=	000001	643#	819	835	840	995	1030		
GO	011156	667#							
ILB	= 000010	590#	676						
IRDA	011020	602#	783	908	934	1063			
IXDA	011022	603#	784	907	938	940	979	1503	
KBDIN	= 104416	590#	698	729	761	818			
LOOP	= 000020	590#	714	742	777	857			
LP	= 000016	590#							
LPP	= 000012	590#	982	1068					
LS	= 000013	590#	1013	1048	1110				
MSG0	012502	903	947#						
MSG1	012563	906	947#						

BOX	1#	591	618	644	957	1017	1055	1087	1135	1218	1481		
DCPARM	1#												
DHDOC1	1#												
DHPARM	1#												
DJPARM	1#												
DLPARM	1#												
DPPARM	1#												
DQDOC1	1#												
DQPARM	1#												
DUPARM	1#												
DUPPAR	1#												
DVDOC1	1#	555											
DVPARM	1#	571											
DZPARM	1#												
HELLO	1#												
HLT	590#	706	738	770	800	832	848	943	1009	1043	1045	1047	1119
SEQUAT	1#	590											
SINTF	1#	590											
SITEP	1#	660											
SSERV	1#	633											

. ABS. 015464 000

ERRORS DETECTED: 0

CZDVOB,CZDVOB/SOL/CRF=CZDVOB.MAC,CZDVOB.P11
RUN-TIME: 3 5 .3 SECONDS
RUN-TIME RATIO: 85/9=9.4
CORE USED: 16K (31 PAGES)