

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZDVO-A-D

PRODUCT NAME: DV11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM

PROGRAM DATE: OCTOBER 1976

MAINTAINER: DIAGNOSTICS

AUTHORS: R A JONES
JOHN EGOLF
RON PLATUKIS

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) 1973, 1976, 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 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. TYPE IN 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. TYPE IN 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 91 IF THIS SETUP
WAS FOR DN11 OR DN11B8.
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=000000=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 B'S 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 MONITOR.)
 - 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 TIMEOUTS
SW12=1 INHIBIT ALL TIMEOUTS 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 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. ↑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 (CNTRL 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 (SW7=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 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<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 OVE UN 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: SOWI, IF "ONE WAY IN" MODE WAS SELECTED. SOWO, 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 SOWO: 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 SOWO 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 INTERJPTS AND SAVES THE TRANSMITTER INTERRUPT ENABLE
BIT AND ALL GENERAL REGISTERS. IT THEN RETURNS TO THE MONITOR
TO TYPE "END PA S". THE MONITOR CHECKS SW14 IF UP IT RETURNS
TO ENTER:, OTHERWISE IT RESTARTS THE PROGRAM.
- 9.5 ENTER: IS ENTEPED 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 OP-ATION.
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 ENTE-ED 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 XI 71: 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.

575

10.0 PARAMETERS FOR THE DV11

PARAM#1 IS USED TO DETERMINE THE LINE NUMBERS FOR XMIT AND RCV.
BITS 11:08 RCV LINE NUMBER---DEFAULT TO LN# 0
BITS 03:00 XMIT 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

```

596
597
598
599
600      011000      011000
601 011000 053104 000040
602 011004 160020
603 011006 000300
604 011010 000240
605 011012 000000
606 011014 013000
607 011016 177777
608 011020 000000
609 011022 000000
610 011024 000000
611 011026 000000
612 011030 000000
613 011032 000000
614 011034 000000
615 011036 011102
616 011040
617 011040      000
618 011041
619 011041      001
620 011042 000300
621 011044 177570
622 011046 177570
623
624
625
626
627      000000
628      100000
629      040000
630      020000
631      020000
632
633 011050 000000
634 011052 000000
635 011054 000000
636 011056 000000
637 011060 000000
638
639 011062 000000
640 011064 000000
641 011066 000000
642 011070 000000
643
644 011072 177560
645 011074 177562
646 011076 177564
647 011100 177566
648
649      000001

```

```

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

```

```

      =11000
DV11:  .ASCII /DV /
BA:    160020
RIV:   300
PRIOR: 240
PARAM1: 0
PARAM2: 13000
PARAM3: 177777
IRDA:  .WORD 0
IXDA:  .WORD 0
SETTLE: .WORD 0
B2016: .WORD 0
TIME:  .WORD 0
      .WORD START
TX.TERM: .BYTE 000
RX.TERM: .BYTE 001
FLAG:   .WORD 0
SWR:    177570
DISPLAY: 177570

```

```

;ISR NAME
;BUS ADDRESS
;VECTOR ADDRESS
;PRIORITY
;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 ;XMIT COMPLETE FLAG
RFLG=40000  ;RCV COMPLETE FLAG
DSFLG=20000 ;DATA SET STATUS CHANGE FLAG
BIT13=20000 ;INHIBIT PRINTOUTS

```

```

SXCSR: 0 ;SAVED XMIT CSR
SRCR: 0 ;SAVED RCV CSR
ERRR: 0 ;RCV CSR SAVED ON ERROR
ERDR: 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

```

```

650
651
652
653 011102 000240
654 011104 017700 177734
655 011110 042700 177400
656 011114 013702 011006
657 011120 012722 014016
658 011124 013722 011010
659 011130 012722 013436
660 011134 013722 011010
661 011140 013704 011004
662 011144 004737 015330
663 011150 004737 015402
664 011154 005214

```

```

:*****
:      DV11-X INTERFACE SERVICE ROUTINE
:*****
START:  NOP
        MOV     2SWR,  R0      ;SETUP MODE IN R0
        BIC     8177400, R0    ;STRIP JUNK
        MOV     RIV,   R2      ;SETUP
        MOV     8RISR, (R2)+   ;INTERRUPT
        MOV     PRIOR,(R2)+   ;VECTORS
        MOV     8XISR,(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

```

```

665
666
667
668
669
670
671
672
673 011156 005037 011032
674 011162 005037 013056
675 011166 005037 013062
676 011172 032700 000001
677 011176 001402
678 011200 000137 011354
679 011204 032700 000002
680 011210 001402
681 011212 000137 011246
682 011216 032700 000010
683 011222 001402
684 011224 000137 011452
685 011230 032700 000004
686 011234 001402
687 011236 000137 011702
688 011242 000000
689 011244 000776

```

```

:*****
:      ROUTINE USED TO GOTO
:      SUPERROUTINE DEPENDENT
:      ON MODE SELECTED.
:*****

```

```

GO:    CLR     TIME
        CLR     DELAY
        CLR     STOP
        BIT     8OWO,MODE
        BEQ    1$
        JMP    $OWO
1$:    BIT     8OWI,MODE
        BEQ    2$
        JMP    $OWI
2$:    BIT     8ILB,MODE
        BEQ    3$
        JMP    $ILB
3$:    BIT     8XLB,MODE
        BEQ    4$
        JMP    $XLB
4$:    HALT
        BR     .-2

```

```

690
691
692
693
694
695
696
697
698
699
700
701
702
703
704 011246 104416
705 011250 004737 013620

```

```

:*****
:      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

```

706	011254	032700	040000		15:	BIT	#RFLG, STAT
707	011260	001013				BNE	25
708	011262	023727	011032	000100		CMP	TIME, #100
709	011270	103771				BLO	15
710	011272	011402				MOV	@RCSR, R2
711	011274	016403	000000			MOV	XCSR(R4), R3
712	011300	104001				HLT	1
713	011302	005037	011032			CLR	TIME
714	011306	000762				BR	15
715							
716	011310	032777	000200	177526	25:	BIT	#NOOAT, @SWR
717	011316	001002				BNE	35
718	011320	004737	012272			JSR	PC, TESTD
719	011324	042700	040000		35:	BIC	#RFLG, STAT
720	011330	032777	000020	177506		BIT	#LOOP, @SWR
721	011336	001405				BEG	45
722	011340	012737	011352	013060		MOV	#45, BACK
723	011346	000137	012132			JMP	EOP
724	011352	000735			45:	BR	SOWI

```

*****
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."
*****

```

725							
726							
727							
728							
729							
730							
731							
732							
733							
734							
735	011354	104416			SOWO:	KBOIN	
736	011356	004737	013064			JSR	PC, STARTX
737	011362	005037	011032			CLR	TIME
738	011366	032700	100000		15:	BIT	#XFLG, STAT
739	011372	001013				BNE	25
740	011374	023727	011032	000100		CMP	TIME, #100
741	011402	103771				BLO	15
742	011404	011402				MOV	@RCSR, R2
743	011406	016403	000000			MOV	XCSR(R4), R3
744	011412	104001				HLT	1
745	011414	005037	011032			CLR	TIME
746	011420	000762				BR	15
747	011422	042700	100000		25:	BIC	#XFLG, STAT
748	011426	032777	000020	177410		BIT	#LOOP, @SWR
749	011434	001405				BEG	35
750	011436	012737	011450	013060		MOV	#35, BACK
751	011444	000137	012132			JMP	EOP
752	011450	000741			35:	BR	SOWO
753							
754							
755							


```

811
812
813
814
815
816
817
818
819
820
821
822
823
824 011702 104416
825 011704 032737 000001 011014
826 011712 001402
827 011714 004737 013620
828 011720 004737 013064
829 011724 005037 011032
830 011730 032700 100000
831 011734 001016
832 011736 032700 040000
833 011742 001024
834 011744 023727 011032 000100
835 011752 103766
836 011754 011402
837 011756 016403 000000
838 011762 104001
839 011764 005037 011032
840 011770 000757
841 011772 032737 000001 011014
842 012000 001356
843 012002 042700 100000
844 012006 004737 013620
845 012012 000746
846 012014 032737 000001 011014
847 012022 001420
848 012024 032700 100000
849 012030 001013
850 012032 023727 011032 000100
851 012040 103765
852 012042 011402
853 012044 016403 000000
854 012050 104001
855 012052 005037 011032
856 012056 000756
857 012060 042700 100000
858 012064 042700 040000
859 012070 005037 011032
860 012074 032777 000200 176742
861 012102 001002
862 012104 004737 012272
863 012110 032777 000020 176726
864 012116 001671
865 012120 012737 011702 013060
866 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.
*****

```

```

$XLB:  KBDIN
      BIT    #FULL DUPLEX, PARAM2
      BEQ    1$
      JSR    PC, STARTR
1$:    JSR    PC, STARTX
      CLR    TIME
2$:    BIT    #XFLG, STAT
      BNE    3$
      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$
5$:    BIC    #XFLG, STAT
6$:    BIC    #RFLG, STAT
      CLR    TIME
8$:    BIT    #NOOAT, #SWR
      BNE    5$
      JSR    PC, TESTD
5$:    BIT    #LOOP, #SWR
      BEQ    $XLB
      MOV    #XLB, BACK
      JMP    EOP

```

```

867
868
869
870
871
872
873 012132
874 012132 104414 000340
875 012136 016437 000000 012270
876 012144 042737 15777 012270
877 012152 042764 020000 000000
878 012160 012766 012220 000002
879 012166 010037 013042
880 012172 010137 013044
881 012176 010237 013046
882 012202 010337 013050
883 012206 010437 013052
884 012212 010537 013054
885 012216 000207
886
887 012220
888 012220 013700 013042
889 012224 013701 013044
890 012230 013702 013046
891 012234 013703 013050
892 012240 013704 013052
893 012244 013705 013054
894 012250 012737 177777 013056
895 012256 053764 012270 000000
896 012264 000177 000570
897 012270 000000
898
899
900
901
902
903
904
905 012272 013746 011056
906 012276 001413
907 012300 032777 020000 176536
908 012306 001007
909 012310 104400 012502
910 012314 004077 176510
911 012320 005746
912 012322 104400 012563
913 012326 013701 011022
914 012332 013702 011020
915 012336 122122
916 012340 001776
917 012342 123741 011040
918 012346 001453
919 012350 122742 000002
920 012354 001005
921 012356 010237 012364
922 012362 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 #1<(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 ORIGINALLY SET; SET TX IE
JMP @BACK
QTPIE: 000000

```

```

*****
SUBROUTINE TO CHECK
RECEIVER DATA.
*****
TESTD: MOV ERCSR, -(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,@32016    ;PRINT CONTENTS OF RBUF
        TST -(SP)
        TYPE MSG1        ;<15><12>
TSTDAT: MOV IXOR, R1     ;SETUP XMIT DATA ADDR
        MOV IROA, 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 ZS
        MOV R2,18
        TYPE

```



```

963 ;*****
964 ; TRANSMITTER INIT ROUTINE
965 ;*****
966
967 013064 042700 100000 STARTX: BIC #XFLG,STAT ;CLEAR XMIT DONE FLAG
968 013070 005737 013056 TST DELAY
969 013074 001415 BEQ 2$
970 013076 005037 014706 CLR TEMP1
971 013102 012737 000007 014710 MOV #7,TEMP2
972 013110 005237 014706 1$: INC TEMP1
973 013114 001375 BNE 1$
974 013116 005337 014710 DEC TEMP2
975 013122 001372 BNE 1$
976 013124 005037 013056 CLR DELAY
977 013130 005037 011032 2$: CLR TIME
978 013134 113764 011012 000020 MOVB PARAM1,20(R4)
979 013142 113764 011012 000006 MOVB PARAM1,6(R4) ;SELECT LINE #
980 013150 112764 000000 000007 MOVB #TPCA,7(R4) ;
981 013156 012764 014720 000010 MOV #SYNC,10(R4) ;LOAD TPCA WITH SYNC ADDRESS
982 013164 112764 000001 000007 MOVB #TPBC,7(R4) ;
983 013172 012764 177772 000010 MOV #6,10(R4) ;LOAD TPBC WITH # OF SYNCs
984 013200 112764 000002 000007 MOVB #TACA,7(R4) ;
985 013206 013764 011022 000010 MOV IXDA,10(R4) ;LOAD TACA WITH MESSAGE ADDRESS
986 013214 112764 000003 000007 MOVB #TABC,7(R4) ;
987 013222 013764 014716 000010 MOV BCNT,10(R4) ;LOAD TABC WITH MESSAGE BYTE COUNT
988 013230 112764 000012 000007 MOVB #LPP,7(R4) ;
989 013236 012764 000100 000010 MOV #100,10(R4) ;SET DDCMP MODE XMIT
990 013244 032737 000002 011014 BIT #BIT1,PARAM2 ;USE SYNC A OR SYNC B
991 013252 001406 BEQ 12$ ;DEFAULT TO SYNC A
992 013254 052764 102000 000004 BIS #BIT10+BIT15,4(R4) ;SETUP FOR SYNC B
993 013262 005764 000004 13$: TST 4(R4) ;WAIT FOR CONTROL STROBE
994 013266 100775 BMI 13$
995 013270 052764 000003 000022 12$: BIS #BIT0+BIT1,22(R4) ;TERMINAL READY, LINE ENABLE
996 013276 005737 013062 TST STOP
997 013302 021005 BNE 6$
998 013304 104400 TYPE ,MSG4
999 013310 000000 HALT ;WAIT FOR CONNECTION TO BE MADE
1000 013312 005137 013062 COM STOP
1001 013316 032737 000001 011014 6$: BIT #FULL.DUPLEX,PARAM2 ;HALF OR FULL DUPLEX?
1002 013324 001006 BNE 8$ ;BRANCH IF FULL
1003 013326 032764 000100 000022 7$: BIT #100,22(R4) ;IS CARRIER ON
1004 013334 001374 BNE 7$ ;WAIT FOR CARRIER TO DIE
1005 013336 005037 011032 CLR TIME
1006 013342 052764 000004 000022 8$: BIS #BIT2,22(R4) ;SET RQTS
1007 013350 032764 000040 000022 9$: BIT #BIT5,22(R4) ;IS CTS UP YET
1008 013356 001016 BNE 11$ ;YES
1009 013360 023727 011032 000036 CMP TIME,#36
1010 013366 103770 BLO 9$
1011 013370 005002 CLR R2 ;DONT PRINT OUT
1012 013372 005003 CLR R3 ;
1013 013374 032777 010000 175442 BIT #SW12,#SWR ;INHIBIT PRINTOUT
1014 013402 001001 BNE 10$ ;YES
1015 013404 104002 HLT 2 ;TYPE WAITING TO TRANSMIT
1016 013406 005037 011032 10$: CLR TIME ;CLEAR TIMER
1017 013412 000756 BR 9$ ;WAIT FOR CTS
1018 013414 052714 030000 11$: BIS #30000,(R4) ;GOT CTS ON TRANSMIT

```

```

1019 013420 112764 000013 000007      MOVB    #LS,7(R4)      ;POINT TO LINE STATE REG.
1020 013426 052764 000004 000010      BIS     #4,10(R4)     ;SET XMIT GO
1021 013434 000207
1022
1023
1024
1025
1026
1027 013436 000240      XISR:   NOP
1028 013440 016437 000014 014706      MOV     14(R4),TEMP1  ;READ NSR
1029 013446 005737 014706      TST    TEMP1         ;VALID DATA
1030 013452 100046      BPL    4$            ;NO UNEXPECTED INTERRUPT
1031 013454 032737 000400 014706      BIT    #BIT8,TEMP1   ;IS XMIT DONE
1032 013462 001430      BEQ    1$            ;NO MUST BE ERROR
1033 013464 032737 001000 014706      BIT    #BIT9,TEMP1   ;PRINCIPAL OR ALTER?
1034 013472 001447      BEQ    3$            ;PRINCIPAL DONE-SYNCS OUT
1035 013474 052700 100000      BIS    #XFLG,STAT    ;SET XMIT DONE FLAG
1036 013500 032737 000001 011014      BIT    #FULL.DUPLEX,PARAM2 ;1/2 OR FULL DUPLEX
1037 013506 001003      BNE    6$            ;BRANCH IF FULL DUPLEX
1038 013510 042764 000004 000022      BIC    #BIT2,22(R4)  ;CLEAR R0TS
1039 013516 032777 000100 175320 6$:      BIT    #BIT6,25WR    ;MONITOR DATA?
1040 013524 001432      BEQ    3$            ;NO-EXIT
1041 013526 105777 175344      TSTB   2TPS         ;TTY READY
1042 013532 100027      BPL    3$            ;CAN'T WAIT-EXIT
1043 013534 112777 000124 175336      MOVB   #'T,2TPB     ;TYPE "T" FOR TRANSMIT
1044 013542 000423      BR     3$
1045 013544 005002      1$:      CLR    R2            ;NO RCV CSR
1046 013546 013703 014706      MOV    TEMP1,R3     ;TYPE OUT NSR
1047 013552 032703 007400      BIT    #BIT8+BIT9+BIT10+BIT11,R3 ;ERROR ON PRINCIPAL CAR
1048 013556 001002      BNE    2$            ;NO ON ALT
1049 013560 104012      HLT    12           ;TELL OPERATOR OF ERROR NXM PRIN CAR
1050 013562 000403      BR     5$            ;EXIT
1051 013564 104013      2$:      HLT    13           ;NXM ALT CAR
1052 013566 000401      BR     5$
1053 013570 104011      4$:      HLT    11           ;UNEXPECTED INTERRUPT
1054 013572 112764 000013 000007 5$:      MOVB   #LS,7(R4)    ;POINT TO SECONDARY LS REGISTER
1055 013600 042764 000060 000010      BIC    #BIT4+BIT5,10(R4) ;CLEAR ERROR BITS
1056 013606 000137 013064      JMP    STARTX       ;TRY AGAIN
1057 013612 005037 011032      3$:      CLR    TIME
1058 013616 000002
1059

```

```

1060
1061
1062
1063
1064
1065 013620 113764 011013 000020 STARTR: MOVB PARAM1+1,20(R4)
1066 013626 113764 011013 000006 MOVB PARAM1+1,6(R4)
1067 013634 042700 040000 BIC #RFLG,STAT ;CLEAR RCV DONE FLAG
1068 013640 112764 000004 000007 MOVB #RCA,7(R4) ;POINT TO RCV CURRENT ADDRESS REG
1069 013646 013764 011020 000010 MOV IRDA,10(R4) ;LOAD IT WITH RCV BUFF ADD
1070 013654 112764 000005 000007 MOVB #RBC,7(R4) ;POINT TO RCV BYTE COUNT REG
1071 013652 012764 177000 000010 MOV #1000,10(R4) ;LOAD IT
1072 013670 112764 000011 000007 MOVB #RCTBA,7(R4) ;POINT TO RCV CONTROL TABLE REG
1073 013676 012764 014726 000010 MOV #CORTAB,10(R4) ;LOAD IT
1074 013704 112764 000012 000007 MOVB #LPP,7(R4) ;POINT TO LINE PROTOCOL REG
1075 013712 012764 000002 000010 MOV #2,10(R4) ;SET STRIP SYNC
1076 013720 112764 000015 000007 MOVB #RMB,7(R4) ;POINT TO RCV MODE REG
1077 013726 105064 000010 CLRB 10(R4) ;MODE 0
1078
1079 013732 052764 000002 000022 BIS #DTR,22(R4) ;SET DATA TERMINAL READY
1080 013740 005737 013062 TST STOP ;SEE IF FIRST TIME
1081 013744 001013 BNE 1$
1082 013746 104400 012770 TYPE ,MSG5 ;TYPE MESSAGE
1083 013752 005137 013062 COM STOP
1084 013756 032737 000002 011014 BIT #BIT1,PARAM2 ;SYNC A OR SYNC B
1085 013764 001403 BEQ 1$ ;DEFAULT TO SYNC A
1086 013766 052764 002000 000004 BIS #BIT10,4(R4) ;SET SYNC B
1087 013774 052764 120000 000004 1$: BIS #BIT15+BIT13,4(R4) ;SET RCV ENABLE+CONTROL STROBE
1088 014002 005764 000004 2$: TST 4(R4) ;LOOP ON CONTROL
1089 014006 100775 BMI 2$ ;STROBE TO SETTLE
1090 014010 052714 000100 BIS #BIT6,(R4) ;SET INT ENABLE
1091 014014 000207 RTS PC ;EXIT
1092
1093
1094
1095
1096
1097 014016 000240 RISR: NOP ;SPARE
1098 014020 016437 000002 014706 MOV 2(R4),TEMP1 ;SAVE RIC REGISTER
1099 014026 032737 170000 014706 BIT #170000,TEMP1 ;CHECK FOR SPECIAL CHARACTER INTR
1100 014034 001043 BNE 1$ ;NO-BRANCH
1101 014036 123737 014706 011041 CMPB TEMP1,RX.TERM ;WAS IT TERM CHARACTER
1102 014044 001037 BNE 1$ ;NO-BRANCH
1103 014046 032777 000040 174770 BIT #BITS,2$WR ;MONITOR RCV DATA
1104 014054 001406 BEQ 2$ ;NO
1105 014056 105777 175014 TSTB 2TPS
1106 014062 100003 BPL 2$
1107 014064 112777 000122 175006 2$: MOVB #R,2TPB ;TYPE "R" FOR RCV
1108 014072 052700 040000 BIS #RFLG,STAT ;SET RCV DONE FLAG
1109 014076 052714 000400 BIS #BIT8,(R4) ;SET RCV INT SRV COMPLETE
1110 014102 000240 NOP
1111 014104 000240 NOP
1112 014106 042714 000100 BIC #BIT6,(R4) ;RESET RCV INT. ENABLE
1113 014112 012764 100000 000004 MOV #BIT15,4(R4) ;TURN OFF RCV.
1114 014120 005764 000004 5$: TST 4(R4) ;WAIT FOR CONTROL STROB
1115 014124 100775 BMI 5$ ;TO CLEAR

```

1116	014126	112764	000013	000007		MOV	#LS,7(R4)	:POINT TO LS REG
1117	014134	012764	000002	000010		MOV	#BIT1,10(R4)	:SET RCV RESYNC
1118	014142	000002				RTI		:EXIT
1119	014144	005003			1S:	CLR	R3	
1120	014146	013702	014706			MOV	TEMP1,R2	
1121	014152	004737	014206			JSR	PC,8S	
1122	014156	104400	014621			TYPE	,FATAL	
1123	014162	104400				TYPE		
1124	014164	000000			4S:	000000		
1125	014166	104000				HLT	0	
1126	014170	023737	000006	014706		CMP	6,TEMP1	
1127	014176	002335				BGE	2S	
1128	014200	104400	014655			TYPE	,NOREC	
1129	014204	000000				HALT		
1130	014206	005046			8S:	CLR	-(SP)	:CLEAR LOCATION ON STACK
1131	014210	116416	000003			MOVB	3(R4),(SP)	:GET HIGH - BYTE OF RIC
1132	014214	042716	000017			BIC	#17,(SP)	:CLEAR LINE NUMBER
1133	014220	006016				ROR	(SP)	:ROTATE UNTIL (INT CODE)X2
1134	014222	006016				ROR	(SP)	:IN LOW BYTE
1135	014224	111637	014706			MOVB	(SP),TEMP1	:SAVE FOR LATTER
1136	014230	062716	014242			ADD	#ERRTAB,(SP)	:GET OFFSET
1137	014234	012637	014164			MOV	(SP)+,4S	:MAKE ADDRESS OF MSG
1138	014240	000207				RTS	PC	:EXIT
1139								


```

1140
1141
1142
1143
1144
1145 014242 014302
1146 014244 014327
1147 014246 014353
1148 014250 014373
1149 014252 014431
1150 014254 014577
1151 014256 014577
1152 014250 014577
1153 014252 014431
1154 014264 014577
1155 014266 014452
1156 014270 014577
1157 014272 014577
1158 014274 014467
1159 014276 014515
1160 014300 014540
1161
1162 014302 005015 054122 052056
1163 014310 051105 020115 047516
1164 014316 020124 047125 050511
1165 014324 042525 000
1166 014327 015 041412 040510
1167 014334 020122 040520 044522
1168 014342 054524 042440 051122
1169 014350 051117 000
1170 014353 015 047412 042526
1171 014360 051122 047125 042440
1172 014366 051122 051117 000
1173 014373 015 041412 040510
1174 014400 020122 040520 044522
1175 014406 054524 042440 051122
1176 014414 051117 025440 047440
1177 014422 042526 051122 047125
1178 014430 000
1179 014431 015 051012 053103
1180 014436 041040 052131 020105
1181 014444 047103 036524 000060
1182 014452 005015 054116 020115
1183 014460 047111 051040 040503
1184 014466 000
1185 014467 015 047012 046530
1186 014474 044440 020116 047503
1187 014502 052116 047522 020114
1188 014510 054502 042524 000
1189 014515 015 046412 046505
1190 014522 050040 051101 052111
1191 014530 020131 051105 047522
1192 014536 000122
1193 014540 005015 040520 044522
1194 014546 054524 042440 051122
1195 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>/NOM IN RCA/
EM15: .ASCIZ <15><12>/NOM IN CONTROL BYTE/
EM16: .ASCIZ <15><12>/MEM PARITY ERROR/
EM17: .ASCIZ <15><12>/PARITY ERROR IN CONTROL BYTE/

```

1196	014562	047503	052116	047522
1197	014570	020114	054502	042524
1198	014576	000		
1199	014577	015	052412	042116
1200	014604	043105	047111	042105
1201	014612	042440	051122	051117
1202	014620	000		
1203	014621	015	042412	051122
1204	014626	051117	051040	041511
1205	014634	030440	035065	031061
1206	014642	044440	042116	041511
1207	014650	052101	051505	000
1208	014655	015	043012	052101
1209	014662	046101	047040	047117
1210	014670	051055	041505	053117
1211	014676	044055	046101	020524
1212	014704	000		
1213				
1214		014706		
1215	014706	000000		
1216	014710	000000		
1217	014712	000000		
1218	014714	000000		
1219	014716	000000		
1220	014720	000000		
1221	014722	000000		
1222	014724	000000		
1223				

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:
 TEMP2:
 TEMP3:
 TEMP4:
 BCNT:
 SYNC:

0
0
0
0
0
0
0
0

1280	015012	000	.BYTE	0
1281	015013	000	.BYTE	0
1282	015014	000	.BYTE	0
1283	015015	000	.BYTE	0
1284	015016	000	.BYTE	0
1285	015017	000	.BYTE	0
1286	015020	000	.BYTE	0
1287	015021	000	.BYTE	0
1288	015022	000	.BYTE	0
1289	015023	000	.BYTE	0
1290	015024	000	.BYTE	0
1291	015025	000	.BYTE	0
1292	015026	000	.BYTE	0
1293	015027	000	.BYTE	0
1294	015030	000	.BYTE	0
1295	015031	000	.BYTE	0
1296	015032	000	.BYTE	0
1297	015033	000	.BYTE	0
1298	015034	000	.BYTE	0
1299	015035	000	.BYTE	0
1300	015036	000	.BYTE	0
1301	015037	000	.BYTE	0
1302	015040	000	.BYTE	0
1303	015041	000	.BYTE	0
1304	015042	000	.BYTE	0
1305	015043	000	.BYTE	0
1306	015044	000	.BYTE	0
1307	015045	000	.BYTE	0
1308	015046	000	.BYTE	0
1309	015047	000	.BYTE	0
1310	015050	000	.BYTE	0
1311	015051	000	.BYTE	0
1312	015052	000	.BYTE	0
1313	015053	000	.BYTE	0
1314	015054	000	.BYTE	0
1315	015055	000	.BYTE	0
1316	015056	000	.BYTE	0
1317	015057	000	.BYTE	0
1318	015060	000	.BYTE	0
1319	015061	000	.BYTE	0
1320	015062	000	.BYTE	0
1321	015063	000	.BYTE	0
1322	015064	000	.BYTE	0
1323	015065	000	.BYTE	0
1324	015066	000	.BYTE	0
1325	015067	000	.BYTE	0
1326	015070	000	.BYTE	0
1327	015071	000	.BYTE	0
1328	015072	000	.BYTE	0
1329	015073	000	.BYTE	0
1330	015074	000	.BYTE	0
1331	015075	000	.BYTE	0
1332	015076	000	.BYTE	0
1333	015077	000	.BYTE	0
1334	015100	000	.BYTE	0
1335	015101	000	.BYTE	0

1336	015102	000	.BYTE	0
1337	015103	000	.BYTE	0
1338	015104	000	.BYTE	0
1339	015105	000	.BYTE	0
1340	015106	000	.BYTE	0
1341	015107	000	.BYTE	0
1342	015110	000	.BYTE	0
1343	015111	000	.BYTE	0
1344	015112	000	.BYTE	0
1345	015113	000	.BYTE	0
1346	015114	000	.BYTE	0
1347	015115	000	.BYTE	0
1348	015116	000	.BYTE	0
1349	015117	000	.BYTE	0
1350	015120	000	.BYTE	0
1351	015121	000	.BYTE	0
1352	015122	000	.BYTE	0
1353	015123	000	.BYTE	0
1354	015124	000	.BYTE	0
1355	015125	000	.BYTE	0
1356	015126	000	.BYTE	0
1357	015127	000	.BYTE	0
1358	015130	000	.BYTE	0
1359	015131	000	.BYTE	0
1360	015132	000	.BYTE	0
1361	015133	000	.BYTE	0
1362	015134	000	.BYTE	0
1363	015135	000	.BYTE	0
1364	015136	000	.BYTE	0
1365	015137	000	.BYTE	0
1366	015140	000	.BYTE	0
1367	015141	000	.BYTE	0
1368	015142	000	.BYTE	0
1369	015143	000	.BYTE	0
1370	015144	000	.BYTE	0
1371	015145	000	.BYTE	0
1372	015146	000	.BYTE	0
1373	015147	000	.BYTE	0
1374	015150	000	.BYTE	0
1375	015151	000	.BYTE	0
1376	015152	000	.BYTE	0
1377	015153	000	.BYTE	0
1378	015154	000	.BYTE	0
1379	015155	000	.BYTE	0
1380	015156	000	.BYTE	0
1381	015157	000	.BYTE	0
1382	015160	000	.BYTE	0
1383	015161	000	.BYTE	0
1384	015162	000	.BYTE	0
1385	015163	000	.BYTE	0
1386	015164	000	.BYTE	0
1387	015165	000	.BYTE	0
1388	015166	000	.BYTE	0
1389	015167	000	.BYTE	0
1390	015170	000	.BYTE	0
1391	015171	000	.BYTE	0

1448	015262	000	.BYTE	000
1449	015263	000	.BYTE	000
1450	015264	000	.BYTE	000
1451	015265	000	.BYTE	000
1452	015266	000	.BYTE	000
1453	015267	000	.BYTE	000
1454	015270	000	.BYTE	000
1455	015271	000	.BYTE	000
1456	015272	000	.BYTE	000
1457	015273	000	.BYTE	000
1458	015274	000	.BYTE	000
1459	015275	000	.BYTE	000
1460	015276	000	.BYTE	000
1461	015277	000	.BYTE	000
1462	015300	000	.BYTE	000
1463	015301	000	.BYTE	000
1464	015302	000	.BYTE	000
1465	015303	000	.BYTE	000
1466	015304	000	.BYTE	000
1467	015305	000	.BYTE	000
1468	015306	000	.BYTE	000
1469	015307	000	.BYTE	000
1470	015310	000	.BYTE	000
1471	015311	000	.BYTE	000
1472	015312	000	.BYTE	000
1473	015313	000	.BYTE	000
1474	015314	000	.BYTE	000
1475	015315	000	.BYTE	000
1476	015316	000	.BYTE	000
1477	015317	000	.BYTE	000
1478	015320	000	.BYTE	000
1479	015321	000	.BYTE	000
1480	015322	000	.BYTE	000
1481	015323	000	.BYTE	000
1482	015324	000	.BYTE	000
1483	015325	000	.BYTE	000
1484	015326	000	.BYTE	000
1485	015327	000	.BYTE	000

```

;*****
; DV11 RAM CLEAR ROUTINE
;*****

```

1490									
1491	015330	012714	004000	RAMCLR:	MOV	#4000, (R4)		:	CLEAR PRIMARY REGS
1492	015334	010246			MOV	R2, -(SP)		:	SAVE R2
1493	015336	010346			MOV	R3, -(SP)		:	SAVE R3
1494	015340	012703	000017		MOV	#17, R3		:	SET UP LINE # COUNT IN R3
1495	015344	012702	000017	15:	MOV	#17, R2		:	SET UP REGISTER # COUNT IN R2
1496	015350	110264	000007	25:	MOV	R2, 7(R4)			
1497	015354	110364	000006		MOV	R3, 6(R4)		:	SET UP SRS REGISTER
1498	015360	005064	000010		CLR	10(R4)		:	CLEAR IT
1499	015364	005302			DEC	R2		:	FIRST CLEAR ALL REGS. FOR LN #
1500	015366	100370			BPL	R2			
1501	015370	005303			DEC	R3		:	NOW CLEAR GO TO NEXT LN #
1502	015372	100364			BPL	R3			
1503	015374	012603			MOV	(SP)+, R3		:	RESTORE R3

1504	015376	012602		MOV	(SP)+,R2	;RESTORE R2
1505	015400	000207		RTS	PC	;EXIT
1506						
1507	015402	010146		SETUP: MOV	R1,-(SP)	;SAVE R1
1508	015404	010046		MOV	R0,-(SP)	;SAVE R0
1509	015406	013700	011022	MOV	IXDA,R0	
1510	015412	005001		CLR	R1	
1511	015414	123720	011040	3\$: CMPB	TX.TERM,(R0)+	;FIGURE OUT BYTE COUNT-
1512	015420	001402		BEQ	4\$;OF MESSAGE TO BE-
1513	015422	005201		INC	R1	;TRANSMITTED
1514	015424	000773		BR	3\$	
1515	015426	010137	014716	4\$: MOV	R1,BCNT	
1516	015432	005437	014716	NEG	BCNT	
1517	015436	012700	014720	MOV	#SYNC,R0	;SET UP CORE LABEL OF
1518	015442	012701	000006	MOV	#6,R1	;SYNC CHARACTERS FOUND
1519	015446	113720	011015	5\$: MOVB	PARAM2+1,(R0)+	;IN HIGH-BYTE OF PARAM2
1520	015452	005301		DEC	R1	
1521	015454	001374		BNE	5\$	
1522	015456	012600		MOV	(SP)+,R0	;RESTORE R0
1523	015460	012601		MOV	(SP)+,R1	;RESTORE R1
1524	015462	000207		RTS	PC	;EXIT
1525		000001		.END		

BA	011004	602#	661					
BACK	013060	722#	750*	785*	865*	896	960#	
BCNT	014716	987	1219#	1515*	1516*			
BIT0	= 000001	596#	995					
BIT1	= 000002	596#	990	995	1084	1117		
BIT10	= 002000	596#	992	1047	1086			
BIT11	= 004000	596#	1047					
BIT12	= 010000	596#						
BIT13	= 020000	596#	631#	907	937	1087		
BIT14	= 040000	596#						
BIT15	= 100000	596#	992	1087	1113			
BIT2	= 000004	596#	1006	1038				
BIT3	= 020010	596#						
BIT4	= 030020	596#	1055					
BIT5	= 030040	596#	1037	1055	1103			
BIT6	= 000100	596#	1039	1090	1112			
BIT7	= 000200	596#						
BIT8	= 030400	596#	1031	1047	1109			
BIT9	= 001000	596#	1033	1047				
B2016	011030	612#	910					
CORTAB	014726	1073	1228#					
DELAY	013056	674#	894*	959#	968	976#		
DEFR	012470	938	947#					
DISPLA	011046	622#						
DSFLG	= 020000	596#	630#					
DSSTAT	011050	637#						
DTR	= 000002	596#	1079					
DV11	011000	601#						
EM0	014302	1145	1162#					
EM1	014327	1146	1166#					
EM12	014452	1155	1182#					
EM15	014467	1159	1185#					
EM16	014515	1159	1189#					
EM17	014540	1160	1193#					
EM2	014353	1147	1170#					
EM3	014373	1148	1173#					
EM4	014431	1149	1153	1179#				
ENTER	012220	878	837#					
EOP	012132	723	751	786	866	873#		
ERCSR	011054	635#						
ERCSR	011056	636#	905					
ERPTAB	014242	1136	1145#					
FATAL	014621	1122	1203#					
FLAG	011042	620#						
FULL.D=	000001	649#	825	841	846	1001	1036	
GO	011156	673#						
ILB	= 000010	596#	682					
IRDA	011020	609#	789	914	940	1069		
IXDA	011022	609#	790	913	944	946	985	1509
KBSIN	= 104416	596#	704	735	767	824		
LOOP	= 000020	596#	720	748	783	863		
LP	= 000016	596#						
LPP	= 030012	596#	958	1074				
LS	= 000013	596#	1019	1054	1116			
MSG0	012502	909	953#					
MSG1	012563	912	953#					

BOX	10	597	624	650	963	1023	1061	1093	1141	1224	1407		
DCPARN	10												
DHDOC1	10												
DHPARN	10												
DJPARN	10												
DLPARN	10												
DPPARN	10												
DDDOC1	10												
DQPARN	10												
DUPARN	10												
DUPPAR	10												
DVDOC1	10	560											
DVPARN	10	576											
DZPARN	10												
HELLO	10												
HLT	5960	712	744	776	806	838	854	949	1015	1049	1051	1053	1125
SEQUAT	10	596											
\$INTF	10	596											
\$ITEP	10	666											
\$SERV	10	639											

. ABS. 015464 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZDVOA.SEQ/SOL/CRF/NL:TOC=ITEP1.MAC,DZDVOA.P11
RUN-TIME: 11 15 .8 SECONDS
RUN-TIME RATIO: 75/28=2.6
CORE USED: 16K (31 PAGES)

