DV11

DV11 OVERLAY FOR ITEP

CZDVOB0

AH-8753B-MC

FICHE 1 OF 1

MAR 1978

digital

MADE IN USA

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

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 SWICHES 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 DMBB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.
        F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DMB.

    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 PROCEEDURES
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. VERFIES 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 MANUALY 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)

TTTTTT 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 ASYNCRONOUS DEVICE, MODEMS AND THE
MAYNARD TEST STATION AND INITALIZE 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 IMMEDEATE ERROR MESSAGE FROM MAYNARD WILL BE RE-
CEIVED. THIS IS BECAUSE THE TEST STATION IS STILL LOOKING

FOR THE REST OF THE INTERUPTED 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 ELASPED TIME. IT IS INCREMENTED EVERY SECOND BY A CLOCK INTERRUPT ROUTINE IN ITEP.

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

9.3 THE OVERLAY THEN LOOPS IN ROUTINES: $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 INTERUPTS AND SAVES THE TRANSMITTER INTERUPT 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 INITALIZED BY ITEP,THE PROGRAM WILL HANG IN THE
DV11 TRANSMITTER INITALIZATION 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

```
                                ;*****************************************************************
                                ;       DV11 INTERFACE SERVICE PARAMS
                                ;*****************************************************************
                        011000          .=11000
011000  053104  000040        DV11:   .ASCIZ  /DV /           ;ISR NAME
011004  175000                BA:     175000                  ;BUS ADDRESS
011006  000300                RIV:    300                     ;VECTOR ADDRESS
011010  000240                PRIOR:  240                     ;PRIORITY
011012  000000                PARAM1: 0                       ;PARAM #1
011014  013000                PARAM2: 13000                   ;PARAM #2
011016  177777                PARAM3: 177777                  ;PARAM #3
011020  000000                IRDA:   .WORD   0               ;INITIAL READ DATA ADDRESS
011022  000000                IXDA:   .WORD   0               ;INITIAL XMIT DATA ADDRESS
011024  000000                SETTLE: .WORD   0               ;LINE SETTLE DELAY FLAG
011026  000000                        .WORD   0               ;
011030  000000                B2016:  .WORD   0               ;ADDR OF BIN TO OCT TYPE ROUTINE
011032  000000                TIME:   .WORD   0               ;TIMER
011034  000000                        .WORD   0               ;
011036  011102                        .WORD   START           ;ADDR OF START OF PROGRAM
011040                          TX.TERM:
011040     000                  .BYTE   000                     ;TRANSMITTER TERMINATING CHAR.
011041                          RX.TERM:
011041     001                  .BYTE   001                     ;RECEIVER TERMINATING CHAR.
011042  000000                FLAG:   .WORD   0
011044  177570                SWR:    177570
011046  177570                DISPLAY:177570

                                ;*****************************************************************
                                ;       CONSTANTS + WORKING STORAGE
                                ;*****************************************************************
                        000000          STAT=R0
                        100000          XFLG=100000                     ;XMIT COMPLETE FLAG
                        040000          RFLG=40000                      ;RCV COMPLETE FLAG
                        020000          DSFLG=20000                     ;DATA SET STATUS CHANGE FLAG
                        020000          BIT13=20000                     ;INHIBIT PRINTOUTS

011050  000000        SXCSR:  0               ;SAVED XMIT CSR
011052  000000        SRCSR:  0               ;SAVED RCV CSR
011054  000000        ERCSR:  0               ;RCV CSR SAVED ON ERROR
011056  000000        ERDBR:  0               ;RCV DATA REG SAVED ON ERROR
011060  000000        DSSTAT: 0               ;RCV CSR SAVED ON DS CHANGE

011062  000000        XCC:    0               ;XMIT CHAR COUNT
011064  000000        RCC:    0               ;RCV CHAR COUNT
011066  000000        RDA:    0               ;RCV DATA ADDR.
011070  000000        XDA:    0               ;XMIT DATA ADDR.

011072  177560        TKS:    177560
011074  177562        TKB:    177562
011076  177564        TPS:    177564
011100  177566        TPB:    177566

                        000001          FULL.DUPLEX=000001
```

```
                                ;*****************************************************************
                                ;       DV11-X INTERFACE SERVICE ROUTINE
                                ;*****************************************************************
011102  000240          START:  NOP
011104  017700  177734          MOV     @SWR,   R0      ;SETUP MODE IN R0
011110  042700  177400          BIC     #177400,R0      ;STRIP JUNK
011114  013702  011006          MOV     RIV,    R2      ;SETUP
011120  012722  014016          MOV     #RISR,  (R2)+   ;INTERRUPT
011124  013722  011010          MOV     PRIOR,  (R2)+   ;VECTORS
011130  012722  013436          MOV     #XISR,  (R2)+   ;
011134  013722  011010          MOV     PRIOR,  (R2)+   ;
011140  013704  011004          MOV     BA,     R4      ;SETUP BUS ADDR INDEX
011144  004737  015330          JSR     PC,RAMCLR       ;CLEAR OUT RAM
011150  004737  015402          JSR     PC,SETUP        ;CALCULATE BYTE CNT AND SYNCS
011154  005214                  INC     (R4)            ;START UCPU


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

011156  005037  011032  GO:     CLR     TIME
011162  005037  013056          CLR     DELAY
011166  005037  013062          CLR     STOP
011172  032700  000001          BIT     #OWO,MODE
011176  001402                  BEQ     1$
011200  000137  011354          JMP     $OWO
011204  032700  000002  1$:     BIT     #OWI,MODE
011210  001402                  BEQ     2$
011212  000137  011246          JMP     $OWI
011216  032700  000010  2$:     BIT     #ILB,MODE
011222  001402                  BEQ     3$
011224  000137  011452          JMP     $ILB
011230  032700  000004  3$:     BIT     #XLB,MODE
011234  001402                  BEQ     4$
011236  000137  011702          JMP     $XLB
011242  000000  4$:     HALT
011244  000776                  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".
                                ;*****************************************************************


011246  104416          $OWI:   KBDIN
011250  004737  013620          JSR     PC,STARTR
```

```
                                1$:     BIT     #RFLG,STAT
                                        BNE     2$
                                        CMP     TIME,#100
                                        BLO     1$
                                        MOV     @RCSR,R2
                                        MOV     XCSR(R4),R3
                                        HLT     1
                                        CLR     TIME
                                        BR      1$

                                2$:     BIT     #NODAT,@SWR
                                        BNE     3$
                                        JSR     PC,TESTD
                                3$:     BIC     #RFLG,STAT
                                        BIT     #LOOP,@SWR
                                        BEQ     4$
                                        MOV     #4$,BACK
                                        JMP     EOP
                                4$:     BR      $OWI


                ;*******************************************************************
                ;       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.
                ;*******************************************************************

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

```
                                ;*****************************************************************
                                ;       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 COMLPETE 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.
                                ;*****************************************************************

011452  104416                $ILB:   KBDIN
011454  004737  013620                JSR     PC,STARTR
011460  005037  011032                CLR     TIME
011464  032700  040000        1$:     BIT     #RFLG,STAT
011470  001013                        BNE     2$
011472  023727  011032  000100        CMP     TIME,#100
011500  103771                        BLO     1$
011502  011402                        MOV     @RCSR,R2
011504  016403  000000                MOV     XCSR(R4),R3
011510  104001                        HLT     1
011512  005037  011032                CLR     TIME
011516  000762                        BR      1$
011520  032777  000200  177316  2$:   BIT     #NODAT,@SWR
011526  001002                        BNE     3$
011530  004737  012272                JSR     PC,TESTD
011534  042700  040000        3$:     BIC     #RFLG,STAT
011540  032777  000020  177276        BIT     #LOOP,@SWR
011546  001405                        BEQ     4$
011550  012737  011562  013060        MOV     #4$,BACK
011556  000137  012132                JMP     EOP
011562  032777  000400  177254  4$:   BIT     #400,   @SWR    ;USE EXTERNAL DATA?
011570  001416                        BEQ     7$              ;BR IF NO
011572  013702  011020                MOV     IRDA,   R2      ;SET POINTER
011576  013703  011022                MOV     IXDA,   R3      ;SET POINTER
011602  010337  011070                MOV     R3,     XDA     ;SETUP XMIT DATA ADDR
011606  112223                        MOVB    (R2)+,  (R3)+   ;MOVE INPUT TO OUTPUT
011610  001376                        BNE     .-2             ;LOOP IF NOT ZERO CHAR
011612  112743  000177                MOVB    #177,   -(R3)   ;INSERT A FILL CHAR
011616  005203                        INC     R3              ;BUMP ADDRESS
011620  112723  000177                MOVB    #177,   (R3)+   ;INSERT ANOTHER FILL
011624  105023                        CLRB    (R3)+           ;INSERT ZERO CHAR
011626  005037  011032        7$:     CLR     TIME
011632  004737  013064                JSR     PC,STARTX
011636  032700  100000        5$:     BIT     #XFLG,STAT
011642  001013                        BNE     6$
011644  023727  011032  000100        CMP     TIME,#100
011652  103771                        BLO     5$
011654  011402                        MOV     @RCSR,R2
011656  016403  000000                MOV     XCSR(R4),R3
011662  104001                        HLT     1
011664  005037  011032                CLR     TIME
011670  000762                        BR      5$
011672  042700  100000        6$:     BIC     #XFLG,STAT
011676  000137  011452                JMP     $ILB
```

```
                                ;*****************************************************************
                                ;       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.
                                ;*****************************************************************

011702  104416                      ;:      KBDIN
011704  032737  000001  011[illegible]          BIT     #FULL.DUPLEX,PARAM2
011712  001402                              BEQ     1$
011714  004737  013620                      JSR     PC,STARTR
011720  004737  013064          1$:         JSR     PC,STARTX
011724  005037  011032                      CLR     TIME
011730  032700  100000          2$:         BIT     #XFLG,STAT
011734  001016                              BNE     3$
011736  032700  040000          7$:         BIT     #RFLG,STAT
011742  001024                              BNE     4$
011744  023727  011032  000100              CMP     TIME,#100
011752  103766                              BLO     2$
011754  011402                              MOV     @RCSR,R2
011756  016403  000000                      MOV     XCSR(R4),R3
011762  104001                              HLT     1
011764  005037  011032                      CLR     TIME
011770  000757                              BR      2$
011772  032737  000001  011014  3$:         BIT     #FULL.DUPLEX,PARAM2
012000  001356                              BNE     7$
012002  042700  100000                      BIC     #XFLG,STAT
012006  004737  013620                      JSR     PC,STARTR
012012  000746                              BR      2$
012014  032737  000001  011014  4$:         BIT     #FULL.DUPLEX,PARAM2
012022  001420                              BEQ     8$
012024  032700  100000                      BIT     #XFLG,STAT
012030  001013                              BNE     6$
012032  023727  011032  000100              CMP     TIME,#100
012040  103765                              BLO     4$
012042  011402                              MOV     @RCSR,R2
012044  016403  000000                      MOV     XCSR(R4),R3
012050  104001                              HLT     1
012052  005037  011032                      CLR     TIME
012056  000756                              BR      4$
012060  042700  100000          6$:         BIC     #XFLG,STAT
012064  042700  040000          8$:         BIC     #RFLG,STAT
012070  005037  011032                      CLR     TIME
012074  032777  000200  176742              BIT     #NODAT,@SWR
012102  001002                              BNE     5$
012104  004737  012272                      JSR     PC,TESTD
012110  032777  000020  176726  5$:         BIT     #LOOP,@SWR
012116  001671                              BEQ     $XLB
012120  012737  011702  013060              MOV     #$XLB,BACK
012126  000137  012132                      JMP     EOP
```

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

012132                                  EOP:
012132  104414  000340                          STPS,PRTY7              ;SET PS PRIORITY TO 7
012136  016437  000000  012270                  MOV     XCSR(R4),QTPIE  ;SAVE TX CSR
012144  042737  157777  012270                  BIC     #^C<TIE>,QTPIE  ;CLEAR ALL BUT TX IE.
012152  042764  020000  000000                  BIC     #TIE,XCSR(R4)   ;CLEAR TX IE (EVEN IF IT WASN'T SET)
012160  012766  012220  000002                  MOV     #ENTER,2(SP)    ;SET FOR RETURN IF SW 14=1
012166  010037  013042                          MOV     R0,SAVR0        ;SAVE REGISTER 0
012172  010137  013044                          MOV     R1,SAVR1        ;SAVE REGISTER 1
012176  010237  013046                          MOV     R2,SAVR2        ;SAVE REGISTER 2
012202  010337  013050                          MOV     R3,SAVR3        ;SAVE REGISTER 3
012206  010437  013052                          MOV     R4,SAVR4        ;SAVE REGISTER 4
012212  010537  013054                          MOV     R5,SAVR5        ;SAVE REGISTER 5
012216  000207                                  RTS     PC              ;RETURN TO CONTROL PROGRAM

012220                                  ENTER:
012220  013700  013042                          MOV     SAVR0,R0        ;RESTORE R0
012224  013701  013044                          MOV     SAVR1,R1        ;RESTORE R1
012230  013702  013046                          MOV     SAVR2,R2        ;RESTORE R2
012234  013703  013050                          MOV     SAVR3,R3        ;RESTORE R3
012240  013704  013052                          MOV     SAVR4,R4        ;RESTORE R4
012244  013705  013054                          MOV     SAVR5,R5        ;RESTORE R5
012250  012737  177777  013056                  MOV     #-1,DELAY
012256  053764  012270  000000                  BIS     QTPIE,XCSR(R4)  ;IF ORGINALLY SET; SET TX IE
012264  000177  000570                          JMP     @BACK
012270  000000                          QTPIE:  000000


                                        ;*********************************
                                        ;       SUBROUTINE TO CHECK
                                        ;       RECEIVER DATA.
                                        ;*********************************
012272  013746  011056                  TESTD:  MOV     ERDBR,  -(SP)   ;WAS THERE A RECEIVE ERROR?
012276  001413                                  BEQ     TSTDAT          ;BR IF NO
012300  032777  020000  176536                  BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS?
012306  001007                                  BNE     TSTDAT          ;BR IF YES
012310  104400  012502                          TYPE    ,MSG0           ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
012314  004077  176510                          JSR     R0,@B2016       ;PRINT CONTENTS OF RBUF
012320  005746                                  TST     -(SP)
012322  104400  012563                          TYPE    ,MSG1           ;<15><12>
012326  013701  011022          TSTDAT: MOV     IXDA,   R1              ;SETUP XMIT DATA ADDR
012332  013702  011020                  MOV     IRDA,   R2              ;SETUP RCV DATA ADDR
012336  122122                  SCAN4:  CMPB    (R1)+,  (R2)+           ;DATA OK ?
012340  001776                          BEQ     SCAN4                   ;BR IF OK
012342  123741  011040                  CMPB    TX.TERM,-(R1)           ;IS IT END OF DATA
012346  001453                          BEQ     TESTDX                  ;BR IF YES
012350  122742  000002                  CMPB    #002,-(R2)
012354  001005                          BNE     2$
012356  010237  012364                  MOV     R2,1$
012362  104400                          TYPE
```

```
                                    1$:     .WORD   0
                                            BR      TESTDX
                                    2$:
                                            TSTB    (R2)            ;
                                            BEQ     TESTDX          ;BR IF YES
                                            CMPB    #177,   (R1)+   ;IS IT FILL CHAR?
                                            BEQ     SCAN4           ;BR IF YES
                                            DEC     R1              ;BACKUP
                                            CMPB    #177,   (R2)+   ;IS IT FILL?
                                            BEQ     SCAN4           ;BR IF YES
                                            TSTB    -(R2)           ;BACK UP POINTER
                                            CMPB    PARAM2+1,(R2)+  ;WAS SYNC CHAR IN BUFFER?
                                            BEQ     SCAN4           ;BR IF CHAR WAS SYNC
                                    SCAN5:  NOP                     ;DATA ERROR
                                            BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS
                                            BNE     DERR            ;BR IF YES
                                            TYPE    ,MSG2           ;<15><12>RECEIVED DATA = <15><12>
                                            MOV     IRDA,   RDAX    ;SETUP DATA ADRESS
                                            TYPE                    ;PRINT RECEIVED DATA
                                    RDAX:   0                       ;RECEIVED DATA ADDR.
                                            TYPE    ,MSG3           ;<15><12>DATA SHOULD BE<15><12>
                                            MOV     IXDA,   .+10    ;SETUP ADDR.
                                            TYPE                    ;PRINT GOOD DATA
                                            IXDA
                                    DERR:   MOVB    (R1)+,R3 ;SETUP XMIT DATA
                                            MOVB    -(R2),R2        ;SETUP RCV DATA
                                            HLT+7                   ;DATA ERROR HALT
                                    TESTDX: TST     (SP)+           ;POP STACK
                                            RTS     PC              ;RETURN FROM SUB/ROUT

                                    MSC0:   .ASCIZ  <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
(1)                                 MSG1:   .ASCIZ  <15><12>
(1)                                 MSG2:   .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
(1)                                 MSG3:   .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
(1)                                 MSG4:   .ASCII  <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
(1)                                         .ASCIZ  <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>
(1)                                 MSG5:   .ASCIZ  <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
(1)                                         .EVEN
(1)                                 SAVR0:  0
                                    SAVR1:  0
                                    SAVR2:  0
                                    SAVR3:  0
                                    SAVR4:  0
                                    SAVR5:  0
                                    DELAY:  0
                                    BACK:   0
                                    STOP:   0
```

```
                                        ;*****************************************************************
                                        ;       TRANSMITTER INIT ROUTINE
                                        ;*****************************************************************

013064  042700  100000          STARTX: BIC     #XFLG,STAT      ;CLEAR XMIT DONE FLAG
013070  005737  013056                  TST     DELAY
013074  001415                          BEQ     2$
013076  005037  014706                  CLR     TEMP1
013102  012737  000007  014710          MOV     #7,TEMP2
013110  005237  014706          1$:     INC     TEMP1
013114  001375                          BNE     1$
013116  005337  014710                  DEC     TEMP2
013122  001372                          BNE     1$
013124  005037  013056                  CLR     DELAY
013130  005037  011032          2$:     CLR     TIME
013134  013764  011012  000020          MOV     PARAM1,20(R4)
013142  013764  011012  000006          MOV     PARAM1,6(R4)    ;SELECT LINE #
013150  112764  000000  000007          MOVB    #TPCA,7(R4)     ;
013156  012764  014720  000010          MOV     #SYNC,10(R4)    ;LOAD TPCA WITH SYNC ADDRESS
013164  112764  000001  000007          MOVB    #TPBC,7(R4)
013172  012764  177772  000010          MOV     #-6,10(R4)      ;LOAD TPBC WITH # OF SYNCS
013200  112764  000002  000007          MOVB    #TACA,7(R4)
013206  013764  011022  000010          MOV     IXDA,10(R4)     ;LOAD TACA WITH MESSAGE ADDRESS
013214  112764  000003  000007          MOVB    #TABC,7(R4)
013222  013764  014716  000010          MOV     BCNT,10(R4)     ;LOAD TABC WITH MESSAGE BYTE COUNT
013230  112764  000012  000007          MOVB    #LPP,7(R4)
013236  012764  000100  000010          MOV     #100,10(R4)     ;SET DDCMP MODE XMIT
013244  032737  000002  011014          BIT     #BIT1,PARAM2    ;USE SYNC A OR SYNC B
013252  001406                          BEQ     12$             ;DEFAULT TO SYNC A
013254  052764  102000  000004          BIS     #BIT10+BIT15,4(R4)      ;SETUP FOR SYNC B
013262  005764  000004          13$:    TST     4(R4)           ;WAIT FC  CONTROL STROBE
013266  100775                          BMI     13$
013270  052764  000003  000022  12$:    BIS     #BIT0+BIT1,22(R4)  ;TERMINAL READY, LINE ENABLE
013276  005737  013062                  TST     STOP
013302  001005                          BNE     6$
013304  104400  012636                  TYPE    ,MSG4
013310  000000                          HALT                    ;WAIT FOR CONNECTION TO BE MADE
013312  005137  013062                  COM     STOP
013316  032737  000001  011014  6$:     BIT     #FULL.DUPLEX,PARAM2     ;HALF OR FULL DUPLEX?
013324  001006                          BNE     8$              ;BRANCH IF FULL
013326  032764  000100  000022  7$:     BIT     #100,22(R4)     ;IS CHARIER ON
013334  001374                          BNE     7$              ;WAIT FOR CARRIER TO DIE
013336  005037  011032                  CLR     TIME
013342  052764  000004  000022  8$:     BIS     #BIT2,22(R4)    ;SET RQTS
013350  032764  000040  000022  9$:     BIT     #BIT5,22(R4)    ;IS CTS UP YET
013356  001016                          BNE     11$             ;YES
013360  023727  011032  000036          CMP     TIME,#36
013366  103770                          BLO     9$
013370  005002                          CLR     R2              ;DONT PRINT OUT
013372  005003                          CLR     R3              ;
013374  032777  010000  175442          BIT     #SW12,@SWR      ;INHIBIT PRINTOUT
013402  001001                          BNE     10$             ;YES
013404  104002                          HLT     2               ;TYPE WAITING TO TRANSMIT
013406  005037  011032          10$:    CLR     TIME            ;CLEAR TIMER
013412  000756                          BR      9$              ;WAIT FOR CTS
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                                  ;       XMIT SERVICE ROUTINE
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$:     BIT     #BIT6,@SWR      ;MONITOR DATA?
1034  013524  001432                          BEQ     3$              ;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                  1$:     CLR     R2              ;NO RCV CSR
1040  013546  013703  014706                  MOV     TEMP1,R3        ;TYPE OUT NSR
1041  013552  032703  007400                  BIT     #BIT8+BIT9+BIT10+BIT11,R3  ;ERROR ON PRINCIPAL CAR
1042  013556  001002                          BNE     2$              ;NO ON ALT
1043  013560  104012                          HLT     12              ;TELL OPERATOR OF ERROR NXM PRIN CAR
1044  013562  000403                          BR      5$              ;EXIT
1045  013564  104013                  2$:     HLT     13              ;NXM ALT CAR
1046  013566  000401                          BR      5$
1047  013570  104011                  4$:     HLT     11              ;UNEXPECTED INTERRUPT
1048  013572  112764  000013  000007  5$:     MOVB    #LS,7(R4)       ;POINT TO SECONDARY LS REGISTER
1049  013600  042764  000060  000010          BIC     #BIT4+BIT5,10(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                                    ;       RECEIVE INIT ROUTINE
1057                                    ;*****************************************************************
1058
1059  013620  013764  011012  000020  STARTR: MOV     PARAM1,20(R4)
1060  013626  013764  011012  000006          MOV     PARAM1,6(R4)
1061  013634  042700  040000                 BIC     #RFLG,STAT      ;CLEAR RCV DONE FLAG
1062  013640  112764  000004  000007         MOVB    #RCA,7(R4)      ;POINT TO RCV CURRENT ADDRESS REG
1063  013646  013764  011020  000010         MOV     IRDA,10(R4)     ;LOAD IT WITH RCV BUFF ADD
1064  013654  112764  000005  000007         MOVB    #RBC,7(R4)      ;POINT TO RCV BYTE COUNT REG
1065  013662  012764  177000  000010         MOV     #-1000,10(R4)   ;LOAD IT
1066  013670  112764  000011  000007         MOVB    #RCTBA,7(R4)    ;POINT TO RCV CONTROL TABLE REG
1067  013676  012764  014726  000010         MOV     #CORTAB,10(R4)  ;LOAD IT
1068  013704  112764  000012  000007         MOVB    #LPP,7(R4)      ;POINT TO LINE PROTOCOL REG
1069  013712  012764  000002  000010         MOV     #2,10(R4)       ;SET STRIP SYNC
1070  013720  112764  000015  000007         MOVB    #RMB,7(R4)      ;POINT TO RCV MODE REG
1071  013726  105064  000010                 CLRB    10(R4)          ;MODE 0
1072
1073  013732  052764  000002  000022         BIS     #DTR,22(R4)     ;SET DATA TERMINAL READY
1074  013740  005737  013062                 TST     STOP            ;SEE IF FIRST TIME
1075  013744  001013                         BNE     1$
1076  013746  104400  012770                 TYPE    ,MSG5           ;TYPE MESSAGE
1077  013752  005137  013062                 COM     STOP
1078  013756  032737  000002  011014         BIT     #BIT1,PARAM2    ;SYNC A OR SYNC B
1079  013764  001403                         BEQ     1$              ;DEFAULT TO SYNC A
1080  013766  052764  002000  000004         BIS     #BIT10,4(R4)    ;SET SYNC B
1081  013774  052764  120000  000004  1$:    BIS     #BIT15+BIT13,4(R4)  ;SET RCV ENABLE+CONTROL STROBE
1082  014002  005764  000004          2$:    TST     4(R4)           ;LOOP ON CONTROL
1083  014006  100775                         BMI     2$              ;STROBE TO SETTLE
1084  014010  052714  000100                 BIS     #BIT6,(R4)      ;SET INT ENABLE
1085  014014  000207                         RTS     PC              ;EXIT
1086
1087                                    ;*****************************************************************
1088                                    ;       RCV SERVICE ROUTINE
1089                                    ;*****************************************************************
1090
1091  014016  000240          RISR:   NOP                             ;SPARE
1092  014020  016437  000002  014706         MOV     2(R4),TEMP1     ;SAVE RIC REGISTER
1093  014026  032737  170000  014706         BIT     #170000,TEMP1   ;CHECK FOR SPECIAL CHARACTER INTR
1094  014034  001043                         BNE     1$              ;NO-BRANCH
1095  014036  123737  014706  011041         CMPB    TEMP1,RX.TERM   ;WAS IT TERM CHARACTER
1096  014044  001037                         BNE     1$              ;NO-BRANCH
1097  014046  032777  000040  174770         BIT     #BIT5,@SWR      ;MONITOR RCV DATA
1098  014054  001406                         BEQ     2$              ;NO
1099  014056  105777  175014                 TSTB    @TPS
1100  014062  100003                         BPL     2$
1101  014064  112777  000122  175006         MOVB    #'R,@TPB        ;TYPE "R" FOR RCV
1102  014072  052700  040000          2$:    BIS     #RFLG,STAT      ;SET RCV DONE FLAG
1103  014076  052714  000400                 BIS     #BIT8,(R4)      ;SET RCV INT SRV COMPLETE
1104  014102  000240                         NOP
1105  014104  000240                         NOP
1106  014106  042714  000100                 BIC     #BIT6,(R4)      ;RESET RCV INT. ENABLE
1107  014112  012764  100000  000004         MOV     #BIT15,4(R4)    ;TURN OFF RECV.
1108  014120  005764  000004          5$:    TST     4(R4)           ;WAIT FOR CONTROL STROB
1109  014124  100775                         BMI     5$              ;TO CLEAR
```

```
1110  014126  112764  000013  000007          MOVB    #LS,7(R4)       ;POINT TO LS REG
1111  014134  012764  000002  000010          MOV     #FIT1,10(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                                    ;       ERROR MESSAGE TABLES
1137                                    ;*****************************************************************
1138
1139  014242  014302                    ERRTAB: EM0
1140  014244  014327                            EM1
1141  014246  014353                            EM2
1142  014250  014373                            EM3
1143  014252  014431                            EM4
1144  014254  014577                            UNDF
1145  014256  014577                            UNDF
1146  014260  014577                            UNDF
1147  014262  014431                            EM4
1148  014264  014577                            UNDF
1149  014266  014452                            EM12
1150  014270  014577                            UNDF
1151  014272  014577                            UNDF
1152  014274  014467                            EM15
1153  014276  014515                            EM16
1154  014300  014540                            EM17
1155
1156  014302  005015  054122  052056  EM0:     .ASCIZ  <15><12>/RX.TERM NOT UNIQUE/
1157  014310  051105  020115  047516
1158  014316  020124  047125  050511
1159  014324  042525     000
1160  014327     015  041412  040510  EM1:     .ASCIZ  <15><12>/CHAR PARITY ERROR/
1161  014334  020122  040520  044522
1162  014342  054524  042440  051122
1163  014350  051117     000
1164  014353     015  047412  042526  EM2:     .ASCIZ  <15><12>/OVERRUN ERROR/
1165  014360  051122  047125  042440
1166  014366  051122  051117     000
1167  014373     015  041412  040510  EM3:     .ASCIZ  <15><12>/CHAR PARITY ERROR + OVERRUN/
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  EM4:     .ASCIZ  <15><12>/RCV BYTE CNT=0/
1174  014436  041040  052131  020105
1175  014444  047103  036524  000060
1176  014452  005015  054116  020115  EM12:    .ASCIZ  <15><12>/NXM IN RCA/
1177  014460  047111  051040  040503
1178  014466     000
1179  014467     015  047012  046530  EM15:    .ASCIZ  <15><12>/NXM IN CONTROL BYTE/
1180  014474  044440  020116  047503
1181  014502  052116  047522  020114
1182  014510  054502  042524     000
1183  014515     015  046412  046505  EM16:    .ASCIZ  <15><12>/MEM PARITY ERROR/
1184  014522  050040  051101  052111
1185  014530  020131  051105  047522
1186  014536  000122
1187  014540  005015  040520  044522  EM17:    .ASCIZ  <15><12>/PARITY ERROR IN CONTROL BYTE/
1188  014546  054524  042440  051122
1189  014554  051117  044440  020116
```

```
 1190  014562  047503  052116  047522
 1191  014570  020114  054502  042524
 1192  014576     000
 1193  014577     015  052412  042116  UNDF:   .ASCIZ  <15><12>/UNDEFINED ERROR/
 1194  014604  043105  047111  042105
 1195  014612  042440  051122  051117
 1196  014620     000
 1197  014621     015  042412  051122  FATAL:  .ASCIZ  <15><12>/ERROR RIC 15:12 INDICATES/
 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  NOREC:  .ASCIZ  <15><12>/FATAL NON-RECOV-HALT!/
 1203  014662  046101  047040  047117
 1204  014670  051055  041505  053117
 1205  014676  044055  046101  020524
 1206  014704     000
 1207
 1208          014706                  .EVEN
 1209  014706  000000                  TEMP1:  0
 1210  014710  000000                  TEMP2:  0
 1211  014712  000000                  TEMP3:  0
 1212  014714  000000                  TEMP4:  0
 1213  014716  000000                  BCNT:   0
 1214  014720  000000                  SYNC:   0
 1215  014722  000000                          0
 1216  014724  000000                          0
 1217
```

```
1218                                    ;*****************************************************************
1219                                    ;       DV11 CONTROL BYTE CORE TABLE
1220                                    ;*****************************************************************
1221
1222  014726    000                     CORTAB: .BYTE   0
1223  014727    001                             .BYTE   1
1224  014730    000                             .BYTE   0
1225  014731    000                             .BYTE   0
1226  014732    000                             .BYTE   0
1227  014733    000                             .BYTE   0
1228  014734    000                             .BYTE   0
1229  014735    000                             .BYTE   0
1230  014736    000                             .BYTE   0
1231  014737    000                             .BYTE   0
1232  014740    000                             .BYTE   0
1233  014741    000                             .BYTE   0
1234  014742    000                             .BYTE   0
1235  014743    000                             .BYTE   0
1236  014744    000                             .BYTE   0
1237  014745    000                             .BYTE   0
1238  014746    000                             .BYTE   0
1239  014747    000                             .BYTE   0
1240  014750    000                             .BYTE   0
1241  014751    000                             .BYTE   0
1242  014752    000                             .BYTE   0
1243  014753    000                             .BYTE   0
1244  014754    000                             .BYTE   0
1245  014755    000                             .BYTE   0
1246  014756    000                             .BYTE   0
1247  014757    000                             .BYTE   0
1248  014760    000                             .BYTE   0
1249  014761    000                             .BYTE   0
1250  014762    000                             .BYTE   0
1251  014763    000                             .BYTE   0
1252  014764    000                             .BYTE   0
1253  014765    000                             .BYTE   0
1254  014766    000                             .BYTE   0
1255  014767    000                             .BYTE   0
1256  014770    000                             .BYTE   0
1257  014771    000                             .BYTE   0
1258  014772    000                             .BYTE   0
1259  014773    000                             .BYTE   0
1260  014774    000                             .BYTE   0
1261  014775    000                             .BYTE   0
1262  014776    000                             .BYTE   0
1263  014777    000                             .BYTE   0
1264  015000    000                             .BYTE   0
1265  015001    000                             .BYTE   0
1266  015002    000                             .BYTE   0
1267  015003    000                             .BYTE   0
1268  015004    000                             .BYTE   0
1269  015005    000                             .BYTE   0
1270  015006    000                             .BYTE   0
1271  015007    000                             .BYTE   0
1272  015010    000                             .BYTE   0
1273  015011    000                             .BYTE   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
1480
1481                        ;*****************************************************************
1482                        ;       DV11 RAM CLEAR ROUTINE
1483                        ;*****************************************************************
1484
1485  015330  012714  004000  RAMCLR: MOV     #4000,(R4)      ;CLEAR PRIMARY REGS
1486  015334  010246                 MOV     R2,-(SP)        ;SAVE R2
1487  015336  010346                 MOV     R3,-(SP)        ;SAVE R3
1488  015340  012703  000017         MOV     #17,R3          ;SET UP LINE # COUNT IN R3
1489  015344  012702  000017  1$:    MOV     #17,R2          ;SET UP REGISTER # COUNT IN R2
1490  015350  110264  000007  2$:    MOVB    R2,7(R4)
1491  015354  110364  000006         MOVB    R3,6(R4)        ;SET UP SRS REGISTER
1492  015360  005064  000010         CLR     10(R4)          ;CLEAR IT
1493  015364  005302                 DEC     R2              ;FIRST CLEAR ALL REGS. FOR LN #
1494  015366  100370                 BPL     2$
1495  015370  005303                 DEC     R3              ;NOW CLEAR GO TO NEXT LN #
1496  015372  100364                 BPL     1$
1497  015374  012603                 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     R0,-(SP)        ;SAVE R0
1503  015406  013700  011022                MOV     IYDA,R0
1504  015412  005001                        CLR     R1
1505  015414  123720  011040      3$:       CMPB    TX.TERM,(R0)+   ;FIGURE OUT BYTE COUNT-
1506  015420  001402                        BEQ     4$              ;OF MESSAGE TO BE-
1507  015422  005201                        INC     R1              ;TRANSMITTED
1508  015424  000773                        BR      3$
1509  015426  010137  014716      4$:       MOV     R1,BCNT
1510  015432  005437  014716                NEG     BCNT
1511  015436  012700  014720                MOV     #SYNC,R0        ;SET UP CORE LABEL OF
1512  015442  012701  000006                MOV     #6,R1           ;SYNC CHARACTERS FOUND
1513  015446  113720  011015      5$:       MOVB    PARAM2+1,(R0)+  ;IN HIGH-BYTE OF PARAM2
1514  015452  005301                        DEC     R1
1515  015454  001374                        BNE     5$
1516  015456  012600                        MOV     (SP)+,R0        ;RESTORE R0
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#
LP0    = 000012            590#    982    1068
LS     = 000013            590#   1013    1048    1110
MSG0     012502            903     947#
MSG1     012563            906     947#
```

```
MSG2    012566            933     947*
MSG3    012613            937     947*
MSG4    012636            947*    992
MSG5    012770            947*   1076
NODAT = 000200            590*    710     773     854
NOREC   014655           1122    1202*
OWI   = 000002            590*    673
OWO   = 000001            590*    670
PARAM1  011012            599*    972     973    1059    1060
PARAM2  011014            600*    819     835     840     928     984     995    1030    1078    1513
PARAM3  011016            601*
PRIOR   011010            598*    652     654
PRTY0 = 000000            590*
PRTY1 = 000040            590*
PRTY2 = 000100            590*
PRTY3 = 000140            590*
PRTY4 = 000200            590*
PRTY5 = 000240            590*
PRTY6 = 000300            590*
PRTY7 = 000340            590*    868
QTPIE   012270            869*    870*    889     891*
RABCC = 000007            590*
RAMCLR  015330            656    1485*
RBC   = 000005            590*   1064
RCA   = 000004            590*   1062
RCBH  = 000017            590*
RCC     011064            634*
RCTBA = 000011            590*   1066
RDA     011066            635*
RDAX    012450            934*    936*
RFLG  = 040000            623*    700     713     764     776     826     852    1061    1102
RISR    014016            651    1091*
RIV     011006            597*    650
RMB   = 000015            590*   1070
RWAIT = 104410            590*
RX.TER  011041            612*   1095
SAVR0   013042            873*    882     947*
SAVR1   013044            874*    883     948*
SAVR2   013046            875*    884     949*
SAVR3   013050            876*    885     950*
SAVR4   013052            877*    886     951*
SAVR5   013054            878*    887     952*
SCAN4   012336            909*    910     923     926     929
SCAN5   012422            930*
SETTLE  011024            604*
SETUP   015402            657    1501*
SRCSR   011052            628*
START   011102            609     647*
STARTR  013620            699     762     821     838    1059*
STARTX  013064            730     793     822     961*   1050
STOP    013062            669*    955*    990     994*   1074    1077*
STPS  = 104414            590*    868
SWR     011044            615*    648     710     714     742     773     777     781     854     857     901     931    1007
                         1033    1097
SWI2  = 010000            590*   1007
SXCSR   011050            627*
```

```
SYNC    014720            975    1214#   1511
TABC  = 000003            590#    980
TABCC = 000006            590#
TACA  = 000002            590#    978
TCTBA = 000010            590#
TEMP1   014706            964#    966#   1022#   1023    1025    1027    1040    1092#   1093    1095    1114    1120    1129#
                         1209#
TEMP2   014710            965#    968#   1210#
TEMP3   014712           1211#
TEMP4   014714           1212#
TESTD   012272            712     775     856     899#
TESTDX  012476            912     918     921     944#
TIE   = 020000            590#    870     871
TIME    011032            607#    667#    702     707#    731#    734     739#    763#    766     771#    792#    796     801#
                          823#    828     833#    844     849#    853#    971#    999#   1003    1010#   1051#
TKB     011074            639#
TKS     011072            638#
TMB   = 000014            590#
TPB     011100            641#   1037#   1101#
TPBC  = 000001            590#    976
TPCA  = 000000            590#    974
TPS     011076            640#   1035    1099
TSTDAT  012326            900     902     907#
TX.TER  011040            610#    911    1505
TYPE  = 104400            590#    903     906     916     933     935     937     939     992    1076    1116    1117    1122
UNDF    014577           1144    1145    1146    1148    1150    1151   1193#
XCC     011062            633#
XCSR  = 000000            590#    705     737     769     799     831     847     869    871#    889#
XDA     011070            636#    785#
XFLG  = 100000            622#    732     741     794     803     824     837     842     851     961    1029
XISR    013436            653    1021#
XLB   = 000004            590#    679
XWAIT = 104412            590#
$ILB    011452            678     761#    804
$OWI    011246            675     698#    718
$OWO    011354            672     729#    746
$XLB    011702            681     818#    858     859
.     = 015464            594#    683     787     938#   1208#
```

CZDVOB.P11 29-DEC-77 12:35 CROSS REFERENCE TABLE -- MACRO NAMES

```
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
$EQUAT       1#    590
$INTF        1#    590
$ITEP        1#    660
$SERV        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)
```