

DZ11

OVERLAY INTERPRO
CZDZBCO

AH-9031C-MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT © 76-82
MADE IN USA



DZ11 ITEP OVERLAY MACY11 30A(1052) 10-NOV-81 11:23 PAGE 2
CZDZBC.P11 10-NOV-81 11:13

IDENTIFICATION

PRODUCT CODE: AC-8785C-MC

PRODUCT NAME: CZDZBC0 DZ11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM

PROGRAM DATE: FEBRUARY 1982

MAINTAINER: DIAGNOSTICS

AUTHORS: R A JONES
JOHN EGOLF
JERRYL PAYNE

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

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

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

COPYRIGHT (C) 1976, 1982 BY DIGITAL EQUIPMENT CORPORATION

CZDZBC.P11

10-NOV-81 11:13

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 DZ11 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.

CZDZBC.P11

10-NOV-81 11:13

- 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.

CZDZBC.P11

10-NOV-81 11:13

- C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
1. THE PROGRAM HALTS FOR ISR (INTERFACE SERVICE ROUTINE) SPECIFICATION
SWR14=SETUP DM-11B ISR
SWR13=SETUP DN-11 ISR
SWR=000000=SETUP VARIABLE ISR
 2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
SETUP SEQUENCE IS: DN11, DM11-BB THEN VARIABLE OVERLAY. (EACH ENTRY SET SWITCHES THEN HIT CONTINUE.)
 - A. HALT FOR BUS ADDRESS OF INTERFACE
 - B. HALT FOR VECTOR ADDRESS OF INTERFACE
 - C. HALT FOR PRIORITY OF INTERFACE
 - D. HALT FOR INTERFACE PARAM #1 (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
 - E. HALT FOR INTERFACE PARAM #2 (DN11 AND 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).

CZDZBC.P11

10-NOV-81 11:13

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 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

CZDZBC.P11

10-NOV-81 11:13

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 (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.

CZDZBC.P11

10-NOV-81 11:13

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

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

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

TEST EXECUTION MAY BE INTERRUPTED BY TYPING THE FOLLOWING CHARACTERS ON THE CONSOLE TTY.

LINE FEED = RESTART PROGRAM AT LOCATION 200.

QUESTION MARK = PRINTOUT FIRST 8 WORDS OF INPUT BUFFER.(ASCII)

THEN TYPE EITHER:

*WXXXXXX

TO PRINTOUT THE 8 WORDS
AT LOC XXXXXX.

*BXXXXXX

TO PRINTOUT THE 16 BYTES
AFTER LOC XXXXXX.

*C

TO CONTINUE

PROGRAM MUST BE RESTARTED AT 200 AFTER PRINTING.

CARRIAGE RETURN = RESTART AT REQUEST FOR NEW OPERATIONAL SWITCHES.

5.0 PROGRAM AND/OR OPERATOR ACTION

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

5.1 NORMAL HALTS SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

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

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

RECEIVED DATA=RRRRRR
DATA SHOULD BE TTTTTT
DATA COMPARE ERROR; BAD DATA=BBB GOOD DATA=GGG

CZDZBC.P11

10-NOV-81 11:13

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 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-

CZDZBC.P11 10-NOV-81 11:13

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 DZ11 INTERFACE SERVICE PARAMS ARE SETUP, AS SPECIFIED BY THE OPERATOR, BY THE ITEP CONTROL PROGRAM.

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

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

9.3 THE OVERLAY THEN LOOPS IN ROUTINES: \$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

CZDZBC.P11

10-NOV-81 11:13

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

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

CZDZBC.P11

10-NOV-81 11:13

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.

10.0 PARAMETERS FOR THE DZ11

PARAM#1 IS LOADED INTO THE LINE PARAMETER REGISTER(DZLPR)
BITS 0-2 LINE NUMBER BEING USED, DEFAULT = LINE 0
BITS 3,4 CHARACTER LENGTH, DEFAULT = EIGHT BITS
BIT 5 STOP BIT COUNT, DEFAULT IS TWO STOP BITS
BITS 6,7 PARITY ENABLE AND SELECT, DEFAULT IS NO PARITY
BITS 8-11 BAUD RATE SELECT, DEFAULT IS 110 BAUD
BIT 12 RECEIVER ON (THIS SHOULD ALWAYS BE SET)

PARAM#2 IS NOT USED AT THIS POINT IN TIME

PARAM#3 IS NOT USED(177777).

DZ11 RESTRICTIONS

THE RTS MODEM SIGNAL ON THE DZ11 IS JUMPER SELECTABLE
AT THE TERMINATION PANEL. IT IS EITHER ALWAYS ASSERTED OR
ASSERTED WHEN DATA TERMINAL READY (DTR) IS SET.
CONSEQUENTLY, AT THIS POINT IN TIME, DZ11 ITEP CAN NOT BE USED
WITH SERIRES 200 AND OTHER HALF DUPLEX MODEMS. ALL ITEP
MODES ARE VALID WITH FULL DUPLEX MODEMS, AND ALL MODES
MAY BE USED TO TERMINALS (ONLY ONE WAY OUT AND IN ARE
RECOMMENDED HERE, HOWEVER).

CZDZBC.P11

10-NOV-81 11:13

```

588
589
590
591
592      011000
593 011000 055104 000040
594 011004 160010
595 011006 000300
596 011010 000240
597 011012 011070
598 011014 177777
599 011016 177777
600 011020 000000
601 011022 000000
602 011024 000000
603 011026 000000
604 011030 000000
605 011032 000000
606 011034 000000
607 011036 011102
608 011040
609 011040 000
610 011041
611 011041 001
612 011042 000000
613 011044 177570
614 011046 177570
615
616
617
618
619      000000
620      100000
621      040000
622      020000
623      020000
624
625 011050 000000
626 011052 000000
627 011054 000000
628 011056 000000
629 011060 000000
630
631 011062 000000
632 011064 000000
633 011066 000000
634 011070 000000
635
636 011072 177560
637 011074 177562
638 011076 177564
639 011100 177566
640
641      000001

```

```

*****
:      DZ11 INTERFACE SERVICE PARAMS
*****

```

```

DZ11:  =11000
BA:    .ASCIZ  /DZ /
RIV:   300
PRIOR: 240
PARAM1: 11070
PARAM2: 177777
PARAM3: 177777
IRDA:  .WORD  0
IXDA:  .WORD  0
SETTLE: .WORD  0
B2016: .WORD  0
TIME:  .WORD  0
TX.TERM: .WORD  START
RX.TERM: .WORD  000
FLAG:   .WORD  001
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
RFLG=40000
DSFLG=20000
BIT13=20000
SXCSR: 0
SRCSR: 0
ERCSR: 0
ERDBR: 0
DSSTAT: 0
XCC: 0
RCC: 0
RDA: 0
XDA: 0
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
FULL.DUPLEX=000001

```

```

:XMIT COMPLETE FLAG
:RCV COMPLETE FLAG
:DATA SET STATUS CHANGE FLAG
:INHIBIT PRINTOUTS

```

```

:SAVED XMIT CSR
:SAVED RCV CSR
:RCV CSR SAVED ON ERROR
:RCV DATA REG SAVED ON ERROR
:RCV CSR SAVED ON DS CHANGE

```

```

:XMIT CHAR COUNT
:RCV CHAR COUNT
:RCV DATA ADDR.
:XMIT DATA ADDR.

```


CZDZBC.P11 10-NOV-81 11:13

```

642
643
644
645 011102 000240
646 011104 017700 177734
647 011110 042700 177400
648 011114 013702 011006
649 011120 012722 013666
650 011124 013722 011010
651 011130 012722 013354
652 011134 013722 011010
653 011140 013704 011004
654 011144 013714 011012
655 011150 013702 011014
656 011154 042702 000001
657 011160 010264 000000
658 011164 052714 000020
659 011170 032714 000020
660 011174 001375
661 011176 013737 011012 013512
662 011204 042737 010000 013512
663 011212 013764 013512 000002
664 011220 010046
665 011222 012700 000001
666 011226 013701 011012
667 011232 042701 177770
668 011236 001403
669 011240 006300
670 011242 005301
671 011244 000774
672 011246 010037 013516
673 011252 012600
674 011254 113764 013516 000005
675
676
677
678
679
680
681
682 011262 005037 011032
683 011266 005037 013152
684 011272 005037 013156
685 011276 032700 000001
686 011302 001402
687 011304 000137 011460
688 011310 032700 000002
689 011314 001402
690 011316 000137 011352
691 011322 032700 000010
692 011326 001402
693 011330 000137 011556
694 011334 032700 000004
695 011340 001402
696 011342 000137 012006
697 011346 000000

*****
: DZ11-X INTERFACE SERVICE ROUTINE
*****
START: NOP
MOV @SWR, R0 ;SETUP MODE IN R0
BIC #177400, R0 ;STRIP JUNK
MOV RIV, R2 ;SETUP
MOV #RISR, (R2)+ ;INTERRUPT
MOV PRIOR, (R2)+ ;VECTORS
MOV #XISR, (R2)+
MOV PRIOR, (R2)+
MOV BA, R4 ;SETUP BUS ADDR INDEX
MOV PARAM1, @RCR ;SETUP VARIABLES
MOV PARAM2, R2
BIC #0001, R2
MOV R2, XCSR(R4) ;IN CSR'S
BIS #DCLR, @RCR ;CLEAR SILO+UARTS
1$: BIT #DCLR, @RCR ;CLEAR PULSE DONE?
BNE 1$ ;BR IF NO
2$: MOV PARAM1, TEMP1
BIC #RCVON, TEMP1 ;DON'T TURN ON RECEIVER YET
MOV TEMP1, LPR(R4) ;LOAD LINE NUMBER AND PARAMETERS
3$: MOV R0, -(SP) ;SAVE R0
MOV #1, R0
MOV PARAM1, R1
BIC #^C<7>, R1 ;ISOLATE THE LINE NUMBER
4$: BEQ 5$ ;CALCULATE TCR BIT
ASL R0
DEC R1
BR 4$
5$: MOV R0, TCRTMP ;SAVE THE ACTIVE TCR BIT
MOV (SP)+, R0
MOVB TCRTMP, TCR+1(R4) ;SET DATA TERMINAL READY

*****
: ROUTINE USED TO GOTO
: SUBROUTINE DEPENDENT
: ON MODE SELECTED.
*****
GO: CLR TIME
CLR DELAY
CLR STOP
BIT #OWO, MODE
BEQ 1$
JMP $OWO
1$: BIT #OWI, MODE
BEQ 2$
JMP $OWI
2$: BIT #ILB, MODE
BEQ 3$
JMP $ILB
3$: BIT #XLB, MODE
BEQ 4$
JMP $XLB
4$: HALT

```

CZDZBC.P11 10-NOV-81 11:13

698 011350 000776

BR .-2

699
700
701
702
703
704
705
706
707
708
709
710
711
712

```

*****
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'.
*****

```

713 011352 104416
 714 011354 004737 013524
 715 011360 032700 040000
 716 011364 001013
 717 011366 023727 011032 000100
 718 011374 103771
 719 011376 011402
 720 011400 016403 000000
 721 011404 104001
 722 011406 005037 011032
 723 011412 000762
 724
 725 011414 032777 000200 177422
 726 011422 001002
 727 011424 004737 012376
 728 011430 042700 040000
 729 011434 032777 000020 177402
 730 011442 001405
 731 011444 012737 011456 013154
 732 011452 000137 012236
 733 011456 000735

```

SOWI: KBDIN
      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 SOWI

```

734
735
736
737
738
739
740
741
742
743

```

*****
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'.
*****

```

744 011460 104416
 745 011462 004737 013160
 746 011466 005037 011032
 747 011472 032700 100000
 748 011476 001013
 749 011500 023727 011032 000100
 750 011506 103771
 751 011510 011402
 752 011512 016403 000000
 753 011516 104001

```

SOWO: KBDIN
      JSR PC,STARTX
      CLR TIME
1$:   BIT #XFLG,STAT
      BNE 2$
      CMP TIME,#100
      BLO 1$
      MOV @RCSR,R2
      MOV XCSR(R4),R3
      HLT 1

```


CZDZBC.P11 10-NOV-81 11:13

754	011520	005037	011032		CLR	TIME
755	011524	000762			BR	1\$
756	011526	042700	100000	2\$:	BIC	#XFLG,STAT
757	011532	032777	000020	177304	BIT	#LOOP,@SWR
758	011540	001405			BEQ	3\$
759	011542	012737	011554	013154	MOV	#3\$,BACK
760	011550	000137	012236		JMP	EOP
761	011554	000741		3\$:	BR	\$OWO
762						
763						
764						

CZDZBC.P11

10-NOV-81 11:13

```

765 *****
766 ROUTINE USED IF INTERNAL LOOP BACK" WAS SELECTED.
767 NOTE THAT WHEN IN THIS MODE; HALF DUPLEX IS THE
768 ONLY MODE AVAILABLE.
769 "INTERNAL LOOP BACK" MEANS THAT THE RECEIVER IS "TURNED ON"
770 AND A COMPLETE MESSAGE IS RECEIVED. IF DATA IS TO BE CHECKED
771 IT IS; IF "END PASS" IS DESIRED; IT IS GIVEN.
772 THEN THE TRANSMITTER IS ENABLED. AFTER THE WHOLE MESSAGE
773 IS TRANSMITTED; THE CYCLE IS REPETED AS ABOVE.
774 *****
775
776 011556 104416 $ILB: KBDIN
777 011560 004737 JSR PC,STARTR
778 011564 005037 CLR TIME
779 011570 032700 1$: BIT #RFLG,STAT
780 011574 001013 BNE 2$
781 011576 023727 011032 000100 CMP TIME,#100
782 011604 103771 BLO 1$
783 011606 011402 MOV @RCSR,R2
784 011610 016403 000000 MOV XCSR(R4),R3
785 011614 104001 HLT 1
786 011616 005037 011032 CLR TIME
787 011622 000762 BR 1$
788 011624 032777 000200 177212 2$: BIT #NODAT,@SWR
789 011632 001002 BNE 3$
790 011634 004737 JSR PC,TESTD
791 011640 042700 040000 3$: BIC #RFLG,STAT
792 011644 032777 000020 177172 BIT #LOOP,@SWR
793 011652 001405 BEQ 4$
794 011654 012737 011666 013154 MOV #4$,BACK
795 011662 000137 012236 JMP EOP
796 011666 032777 000400 177150 4$: BIT #400, @SWR ;USE EXTERNAL DATA?
797 011674 001416 BEQ 7$ ;BR IF NO
798 011676 013702 011020 MOV IRDA, R2 ;SET POINTER
799 011702 013703 011022 MOV IXDA, R3 ;SET POINTER
800 011706 010337 011070 MOV R3, XDA ;SETUP XMIT DATA ADDR
801 011712 112223 MOV (R2)+, (R3)+ ;MOVE INPUT TO OUTPUT
802 011714 001376 BNE -2 ;LOOP IF NOT ZERO CHAR
803 011716 112743 000177 MOV #177, -(R3) ;INSERT A FILL CHAR
804 011722 005203 INC R3 ;BUMP ADDRESS
805 011724 112723 000177 MOV #177, (R3)+ ;INSERT ANOTHER FILL
806 011730 105023 CLRB (R3)+ ;INSERT ZERO CHAR
807 011732 005037 011032 7$: CLR TIME
808 011736 004737 013160 JSR PC,STARTX
809 011742 032700 100000 5$: BIT #XFLG,STAT
810 011746 001013 BNE 6$
811 011750 023727 011032 000100 CMP TIME,#100
812 011756 103771 BLO 5$
813 011760 011402 MOV @RCSR,R2
814 011762 016403 000000 MOV XCSR(R4),R3
815 011766 104001 HLT 1
816 011770 005037 011032 CLR TIME
817 011774 000762 BR 5$
818 011776 042700 100000 6$: BIC #XFLG,STAT
819 012002 000137 011556 JMP $ILB

```


CZDZBC.P11 10-NOV-81 11:13

```

820 *****
821 ROUTINE USED IF 'EXTERNAL LOOP BACK' WAS SELECTED.
822 EITHER HALF OR FULL DUPLEX MAY BE SELECTED IN THIS MODE.
823 'EXTERNAL LOOP BACK' MEANS THAT THE TRANSMITTER IS FIRST
824 TURNED ON (IF HALF DUPLEX) AND THE WHOLE MESSAGE IS TRANSMITTED;
825 THEN THE RECEIVER IS ENABLED. AFTER THE WHOLE MESSAGE IS RECEIVED
826 DATA WILL THEN BE CHECKED IF DESIRED AND END PASS WILL
827 BE GIVEN IF DESIRED. THEN THE CYCLE IS REPEATED
828 AS ABOVE. IF RUNNING IN FULL DUPLEX THE PROGRAM
829 WAITS FOR BOTH THE RECEIVER AND TRANSMITTER TO
830 FINISH THEN RESTARTS THE RECEIVER AND TRANSMITTER.
831 *****
832
833 012006 104416 $XLB: KBDIN
834 012010 032737 000001 011014 BIT #FULL.DUPLEX,PARAM2
835 012016 001402 BEQ 1$
836 012020 004737 013524 JSR PC,STARTR
837 012024 004737 013160 1$: JSR PC,STARTX
838 012030 005037 011032 CLR TIME
839 012034 032700 100000 2$: BIT #XFLG,STAT
840 012040 001016 BNE 3$
841 012042 032700 040000 7$: BIT #RFLG,STAT
842 012046 001024 BNE 4$
843 012050 023727 011032 000100 CMP TIME,#100
844 012056 103766 BLO 2$
845 012060 011402 MOV @RCSR,R2
846 012062 016403 000000 MOV XCSR(R4),R3
847 012066 104001 HLT 1
848 012070 005037 011032 CLR TIME
849 012074 000757 BR 2$
850 012076 032737 000001 011014 3$: BIT #FULL.DUPLEX,PARAM2
851 012104 001356 BNE 7$
852 012106 042700 100000 BIC #XFLG,STAT
853 012112 004737 013524 JSR PC,STARTR
854 012116 000746 BR 2$
855 012120 032737 000001 011014 4$: BIT #FULL.DUPLEX,PARAM2
856 012126 001420 BEQ 8$
857 012130 032700 100000 BIT #XFLG,STAT
858 012134 001013 BNE 6$
859 012136 023727 011032 000100 CMP TIME,#100
860 012144 103765 BLO 4$
861 012146 011402 MOV @RCSR,R2
862 012150 016403 000000 MOV XCSR(R4),R3
863 012154 104001 HLT 1
864 012156 005037 011032 CLR TIME
865 012162 000756 BR 4$
866 012164 042700 100000 6$: BIC #XFLG,STAT
867 012170 042700 040000 8$: BIC #RFLG,STAT
868 012174 005037 011032 CLR TIME
869 012200 032777 000200 176636 BIT #NODAT,@SWR
870 012206 001002 BNE 5$
871 012210 004737 012376 JSR PC,TESTD
872 012214 032777 000020 176622 5$: BIT #LOOP,@SWR
873 012222 001671 BEQ $XLB
874 012224 012737 012006 013154 MOV #$XLB,BACK
875 012232 000137 012236 JMP EOP

```

CZDZBC.P11 10-NOV-81 11:13

```

876
877
878
879
880
881
882 012236
883 012236 104414 000340
884 012242 016437 000000 012374
885 012250 042737 137777 012374
886 012256 042764 040000 000000
887 012264 012766 012324 000002
888 012272 010037 013136
889 012276 010137 013140
890 012302 010237 013142
891 012306 010337 013144
892 012312 010437 013146
893 012316 010537 013150
894 012322 000207
895
896 012324
897 012324 013700 013136
898 012330 013701 013140
899 012334 013702 013142
900 012340 013703 013144
901 012344 013704 013146
902 012350 013705 013150
903 012354 012737 177777 013152
904 012362 053764 012374 000000
905 012370 000177 000560
906 012374 000000
907
908
909
910
911
912
913
914 012376 013746 011056
915 012402 001413
916 012404 032777 020000 176432
917 012412 001007
918 012414 104400 012576
919 012420 004077 176404
920 012424 005746
921 012426 104400 012657
922 012432 013701 011022
923 012436 013702 011020
924 012442 122122
925 012444 001776
926 012446 123741 011040
927 012452 001447
928 012454 122742 000002
929 012460 001005
930 012462 010237 012470
931 012466 104400

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

EOP:
      STPS,PRTY7
      MOV     XCSR(R4),QTPIE
      BIC     #^C<TIE>,QTPIE
      BIC     #TIE,XCSR(R4)
      MOV     #ENTER,2(SP)
      MOV     R0,SAVR0
      MOV     R1,SAVR1
      MOV     R2,SAVR2
      MOV     R3,SAVR3
      MOV     R4,SAVR4
      MOV     R5,SAVR5
      RTS     PC

      ;SET PS PRIORITY TO 7
      ;SAVE TX CSR
      ;CLEAR ALL BUT TX IE.
      ;CLEAR TX IE (EVEN IF IT WASN'T SET)
      ;SET FOR RETURN IF SW 14=1
      ;SAVE REGISTER 0
      ;SAVE REGISTER 1
      ;SAVE REGISTER 2
      ;SAVE REGISTER 3
      ;SAVE REGISTER 4
      ;SAVE REGISTER 5
      ;RETURN TO CONTROL PROGRAM

ENTER:
      MOV     SAVR0,R0
      MOV     SAVR1,R1
      MOV     SAVR2,R2
      MOV     SAVR3,R3
      MOV     SAVR4,R4
      MOV     SAVR5,R5
      MOV     #-1,DELAY
      BIS     QTPIE,XCSR(R4)
      JMP     @BACK

      ;RESTORE R0
      ;RESTORE R1
      ;RESTORE R2
      ;RESTORE R3
      ;RESTORE R4
      ;RESTORE R5
      ;IF ORIGINALLY SET; SET TX IE

QTPIE: 000000

*****
SUBROUTINE TO CHECK
RECEIVER DATA.
*****
TESTD: MOV     ERDBR, -(SP)
      BEQ     TSTDAT
      BIT     #BIT13,@SWR
      BNE     TSTDAT
      TYPE    ,MSG0
      JSR     R0,@B2016
      TST     -(SP)
      TYPE    ,MSG1
      TSTDAT: MOV     IXDA, R1
      SCAN4: MOV     IRDA, R2
      CMPB    (R1)+, (R2)+
      BEQ     SCAN4
      CMPB    TX.TERM,-(R1)
      BEQ     TESTDX
      CMPB    #002,-(R2)
      BNE     2$
      MOV     R2,1$
      TYPE

      ;WAS THERE A RECEIVE ERROR?
      ;BR IF NO
      ;INHIBIT PRINTOUTS?
      ;BR IF YES
      ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
      ;PRINT CONTENTS OF RBUF
      ;<15><12>
      ;SETUP XMIT DATA ADDR
      ;SETUP RCV DATA ADDR
      ;DATA OK ?
      ;BR IF OK
      ;IS IT END OF DATA
      ;BR IF YES

```


CZDZBC.P11 10-NOV-81 11:13

```

932 012470 000000      1$: .WORD 0
933 012472 000437      BR TESTDX
934 012474          2$:
935 012474 105712      TSTB (R2)
936 012476 001435      BEQ TESTDX ;BR IF YES
937 012500 122721 000177 CMPB #177, (R1)+ ;IS IT FILL CHAR?
938 012504 001756      BEQ SCAN4 ;BR IF YES
939 012506 005301      DEC R1 ;BACKUP
940 012510 122722 000177 CMPB #177, (R2)+ ;IS IT FILL?
941 012514 001752      BEQ SCAN4 ;BR IF YES
942 012516 000240      NOP ;DATA ERROR
943 012520 032777 020000 176316 SCAN5: BIT #BIT13, @SWR ;INHIBIT PRINTOUTS
944 012526 001016      BNE DERR ;BR IF YES
945 012530 104400 012662 TYPE MSG2 ;<15><12>RECEIVED DATA = <15><12>
946 012534 013737 011020 012544 MOV IRDA, RDAX ;SETUP DATA ADDRESS
947 012542 104400      TYPE ;PRINT RECEIVED DATA
948 012544 000000      RDAX: 0 ;RECEIVED DATA ADDR.
949 012546 104400 012707 TYPE MSG3 ;<15><12>DATA SHOULD BE<15><12>
950 012552 013737 011022 012562 MOV IXDA, .+10 ;SETUP ADDR.
951 012560 104400      TYPE ;PRINT GOOD DATA
952 012562 011022      IXDA
953 012564 111103      DERR: MOVB (R1), R3 ;SETUP XMIT DATA
954 012566 114202      MOVB -(R2), R2 ;SETUP RCV DATA
955 012570 104007      HLT+7 ;DATA ERROR HALT
956 012572 005726      TESTDX: TST (SP)+ ;POP STACK
957 012574 000207      RTS PC ;RETURN FROM SUB/ROUT
958
959 012576 005015 044124 051105 MSG0: .ASCII <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
(1) 012657 015 000012 MSG1: .ASCII <15><12>
(1) 012662 005015 042522 042503 MSG2: .ASCII <15><12>/RECEIVED DATA = /<15><12>
(1) 012707 015 042012 052101 MSG3: .ASCII <15><12>/DATA SHOULD BE/<15><12>
(1) 012732 005015 046120 040505 MSG4: .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
(1) 013001 015 053412 042510 .ASCII <15><12>/WHEN CONNECTION COMPLETE; HIT CONTINUE SWITCH./<15><12>
(1) 013064 005015 046120 040505 MSG5: .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
(1)
(1) 013136 000000 .EVEN
960 013140 000000 SAVR0: 0
961 013142 000000 SAVR1: 0
962 013144 000000 SAVR2: 0
963 013146 000000 SAVR3: 0
964 013150 000000 SAVR4: 0
965 013152 000000 SAVR5: 0
966 013154 000000 DELAY: 0
967 013156 000000 BACK: 0
968 STOP: 0

```

CZDZBC.P11 10-NOV-81 11:13

```

969
970
971
972
973 013160 005737 013152
974 013164 001415
975 013166 005037 013512
976 013172 012737 000007 013514
977 013200 005237 013512
978 013204 001375
979 013206 005337 013514
980 013212 001372
981 013214 005037 013152
982 013220 042700 100000
983 013224 013737 011022 011070
984 013232 005737 013156
985 013236 001020
986 013240 104400 012732
987 013244 000000
988 013246 005137 013156
989 013252 005037 013512
990 013256 012737 000030 013514
991 013264 005237 013512
992 013270 001375
993 013272 005337 013514
994 013276 001372
995 013300 113764 013516 000004 2$:
996
997
998 013306 113764 013516 000005
999 013314 032700 000004
1000 013320 001412
1001 013322 012737 177777 013522
1002 013330 052714 040040
1003 013334 000001
1004 013336 000240
1005 013340 005737 013520
1006 013344 001375
1007 013346 052714 040040 3$:
1008 013352 000207
1009
1010 013354 127737 175510 011040 XISR:
1011 013362 001005
1012 013364 052700 100000
1013 013370 042714 040000
1014 013374 000440
1015 013376 116405 000001 XISR1:
1016 013402 042705 177770
1017 013406 013701 011012
1018 013412 042701 177770
1019 013416 120501
1020 013420 001407
1021 013422 011402
1022 013424 005003
1023 013426 104010
1024 013430 104400 014107

*****
TRANSMITTER INITIALIZATION SUBROUTINE
*****

STARTX: TST DELAY ;IF SW04=1 & SW14=0 DELAY
        BEQ 1$ ;NO DELAY START TRANSMITTER
        CLR TEMP1 ;PREPARE FOR DELAY
        MOV #7,TEMP2
        INC TEMP1 ;INCREMENT DELAY
        BNE -4
        DEC TEMP2
        BNE -12
        CLR DELAY
        BIC #XFLG,STAT
        MOV IXDA,XDA ;SET UP XMIT DATA ADD
        TST STOP ;FIRST TIME HERE?
        BNE 2$ ;NO
        TYPE ,MSG4 ;MAKE CONNECTION
        COM STOP ;COMPLEMENT STOP
        CLR TEMP1 ;YES PREPARE FOR DELAY
        MOV #14*2,TEMP2
        INC TEMP1 ;INCREMENT DELAY
        BNE -4
        DEC TEMP2
        BNE -12
        MOVB TCRTMP,TCR(R4) ;SET LINE # IN TCR - CHANGE IN REV C TO A 'MOVB' TO
        ;CORRECT AN AIDS REPORT (HH0001070) - PROBLEM OCCURED
        ;WHEN DTR WAS DROPPED ON A DF03.
        MOVB TCRTMP,TCR+1(R4) ;SET DATA TERMINAL READY
        BIT #XLB,MODE ;XLB MODE?
        BEQ 3$ ;BR IF NO
        MOV #-1,TRNFLG ;SET FLAG
        BIS #TIE+MSENAB,@RCR ;SET INTERRUPT ENABLE
        WAIT
        NOP
        TST SNCFLG ;FIRST CHAR RECEIVED YET?
        BNE -4 ;BR IF NO
        BIS #TIE+MSENAB,@RCR ;SET INTERRUPT ENABLE,SCAN ENABLE
        RTS PC

XISR: CMPB @XDA,TX.TERM ;IS CHAR TRANSMITTER TERMINATION CHAR
      BNE XISR1 ;BR IF NO
      BIS #XFLG,STAT ;SET XMIT DONE FLAG
      BIC #TIE,@RCR ;CLEAR ENABLES
      BR XISR3
XISR1: MOVB 1(R4),R5 ;GET LINE NUMBER OF READY LINE
      BIC #^C<7>,R5 ;ISOLATE THE LINE NUMBER
      MOV PARAM1,R1 ;GET THE EXPECTED LINE NUMBER
      BIC #^C<7>,R1 ;ISOLATE IT
      CMPB R5,R1 ;ARE THEY EQUAL?
      BEQ XISR2 ;IF SO, GO TRANSMIT A CHARACTER
      MOV @RCR,R2 ;SET UP R2 WITH CSR CONTENTS
      CLR R3
      HLT 10 ;ERROR WRONG LINE
      TYPE ,SCANE ;TYPE ERROR MESSAGE

```


CZDZBC.P11 10-NOV-81 11:13

1025	013434	000000				HALT			
1026	013436	000776				BR			;HALT
1027	013440	117764	175424	00C006	XISR2:	MOVB	;-2		
1028	013446	032777	000100	175370		BIT	@XDA,TDR(R4)		;TRANSMIT DATA
1029	013454	001406				BEQ	#100,@SWR		;MONITOR TX DATA?
1030	013456	105777	175414			TSTB	NOXMON		;BR IF NO
1031	013462	100003				BPL	@TPS		;TTY READY?
1032	013464	117777	175400	175406		MOVB	NOXMON		;BR IF NO
1033	013472	005237	011070		NOXMON:	INC	@XDA,@TPB		;TYPE CHAR
1034	013476	005037	011032		XISR3:	CLR	XDA		;INC TTDR POINTER
1035	013502	005037	013522			CLR	TIME		
1036	013506	000002				RTI	TRNFLG		
1037	013510	000000			ERROR1:	0			
1038	013512	000000			TEMP1:	0			
1039	013514	000000			TEMP2:	0			
1040	013516	000000			TCRTMP:	0			
1041	013520	000000			SNCFLG:	0			
1042	013522	000000			TRNFLG:	0			

CZDZBC.P11 10-NOV-81 11:13

```

1043
1044
1045
1046
1047 013524 005737 013156
1048 013530 001005
1049 013532 104400 012732
1050 013536 005137 013156
1051 013542 000000
1052 013544 032700 000004
1053 013550 001405
1054 013552 005037 013512
1055 013556 005237 013512
1056 013562 001375
1057 013564 042700 040000
1058 013570 013737 011020 011066
1059 013576 012737 001000 011064
1060 013604 012737 177777 013520
1061 013612 005037 011054
1062 013616 005037 011056
1063 013622 005764 000002
1064 013626 100775
1065 013630 013737 011012 013512
1066 013636 052737 010000 013512
1067 013644 013764 013512 000002
1068 013652 113764 013516 000005
1069 013660 052714 000140
1070 013664 000207
1071
1072 013666 105714
1073 013670 100403
1074 013672 011402
1075 013674 005003
1076 013676 104010
1077 013700 016401 000002
1078 013704 100403
1079 013706 011402
1080 013710 005003
1081 013712 104010
1082 013714 042701 000200
1083 013720 032701 070000
1084 013724 001404
1085 013726 011437 011054
1086 013732 010137 011056
1087 013736 110177 175124
1088 013742 032777 000040 175074
1089 013750 001405
1090 013752 105777 175120
1091 013756 100002
1092 013760 110177 175114
1093 013764 005237 011066
1094 013770 105077 175072
1095 013774 005337 011064
1096 014000 001007
1097 014002 000005
1098 014004 005002

*****
: RECEIVER INITIALIZATION SUBROUTINE
*****
STARTR: TST STOP :FIRST TIME HERE?
        BNE 1$ :BR IF NO
        TYPE ,MSG4 :TYPE 'MAKE CONNECTION'
        COM STOP :COMPLEMENT STOP
        HALT
1$: BIT #XLB,MODE :XLB MODE?
    BEQ 2$ :BR IF NO
    CLR TEMP1 :START DELAY
    INC TEMP1
    BNE .-4
2$: BIC #RFLG,STAT
    MOV IRDA,RDA :SET UP RECEIVER DATA ADD
    MOV #1000,RCC :SET UP BUFFER LIMIT
    MOV #-1,SNCFLG
    CLR ERCSR :CLEAR ERROR RECORDS
    CLR ERDBR
3$: TST RBUF(R4) :CLEAR SILO
    BMI 3$ :KEEP CLEARING UNTIL BIT 15 CLEAR
    MOV PARAM1,TEMP1 :GET READY TO LOAD PARAMETERS
    BIS #RCVON,TEMP1 :BE SURE TO TURN RECEIVER ON
    MOV TEMP1,LPR(R4) :LOAD PARAMETERS, ENABLE RECEIVER
    MOVB TCRTMP,TCR+1(R4) :SET DATA TERMINAL READY
    BIS #RIE!MSENAB,ERCSR :SET INTERRUPT ENABLE,RECEIVER ENABLE
    RTS PC

RISR: TSTB @RCSR :DID RECEIVER DONE SET?
    BMI 1$ :BR IF YES
    MOV @RCSR,R2 :SAVE CSR
    CLR R3
    HLT 10 :ERROR RECEIVER INTERRUPTED BUT DONE NOT SET
1$: MOV RBUF(R4),R1 :GET CHAR
    BMI 2$ :BR IF YES
    MOV @RCSR,R2 :SAVE CSR
    CLR R3
    HLT 10 :ERROR CHAR PRESENT NOT SET
2$: BIC #200,R1 :STRIP A BIT
    BIT #ORUN+FRME+PARE,R1 :CHECK FOR RECEIVER ERRORS
    BEQ 3$ :BR IF NO ERRORS
    MOV @RCSR,ERCSR :SAVE CSR
    MOV R1,ERDBR :SAVE RBUF
    MOVB R1,@RDA :STORE CHAR
    BIT #BITS,@SWR :MONITOR RXDATA?
    BEQ NORMON :BR IF NO
    TSTB @TPS :IS TTY READY?
    BPL NORMON :BR IF NO
    MOVB R1,@TPB :TYPE CHAR
    INC RBUF POINTER
    CLRB @RDA :CLEAR NEXT POSITION
    DEC RCC :DEC CHAR COUNT
    BNE 1$ :BUFFER FULL YET?
    RESET :STOP THE SHOW,BUFFER OVERFLOWED
    CLR R2
NORMON: INC RDA
        CLRB @RDA
        DEC RCC
        BNE 1$
        RESET
        CLR R2

```


CZDZBC.P11 10-NOV-81 11:13

1099	014006	005003			CLR	R3	
1100	014010	104000			HLT	0	
1101	014012	104006			HLT	6	;RECEIVER BUFFER FULL
1102	014014	000000			HALT		
1103	014016	000776			BR	.-2	
1104	014020	123701	011041	1\$:	CMPB	RX.TERM,R1	;IS CHAR 001?
1105	014024	001004			BNE	RISR1	;BR IF NO
1106	014026	042714	000100		BIC	#RIE,@RCSR	;CLEAR RECEIVER INTERRUPT ENABLE
1107	014032	052700	040000		BIS	#RFLG,STAT	;SET R DONE FLAG
1108	014036	005037	011032	RISR1:	CLR	TIME	
1109	014042	005037	013520		CLR	SNCFLG	
1110	014046	000002			RTI		;GO HOME
1111	014050	005015	042522	042503	MFULL:	.ASCIZ<15><12>/RECEIVER	BUFFER FULL ERROR!!/
	014107	015	042412	051122	SCANE:	.ASCIZ<15><12>/ERROR!	TRANSMITTER SCAN STOPPED ON WRONG LINE/
		014170			.EVEN		
		000001			.END		

CZDZBC.P11

10-NOV-81 11:13

CROSS REFERENCE TABLE -- USER SYMBOLS

BA	011004	594#	653					
BACK	013154	731*	759*	794*	874*	905	966#	
BIT0	= 000001	588#						
BIT1	= 000002	588#						
BIT10	= 002000	588#						
BIT11	= 004000	588#						
BIT12	= 010000	588#						
BIT13	= 020000	588#	623#	916	943			
BIT14	= 040000	588#						
BIT15	= 100000	588#						
BIT2	= 000004	588#						
BIT3	= 000010	588#						
BIT4	= 000020	588#						
BIT5	= 000040	588#	1088					
BIT6	= 000100	588#						
BIT7	= 000200	588#						
BIT8	= 000400	588#						
BIT9	= 001000	588#						
B2016	011030	604#	919					
DCLR	= 000020	588#	658	659				
DELAY	013152	683*	903*	965#	973	981*		
DERR	012564	944	953#					
DISPLA	011046	614#						
DSFLG	= 020000	588#	622#					
DSSTAT	011060	629#						
DZ11	011000	593#						
ENTER	012324	887	896#					
EOP	012236	732	760	795	875	882#		
ERCSR	011054	627#	1061*	1085*				
ERDBR	011056	628#	914	1062*	1086*			
ERROR1	013510	1037#						
FLAG	011042	612#						
FRME	= 020000	588#	1083					
FULL.D	= 000001	641#	834	850	855			
GO	011262	682#						
ILB	= 000010	588#	691					
IRDA	011020	600#	798	923	946	1058		
IXDA	011022	601#	799	922	950	952	983	
KBDIN	= 104416	588#	713	744	776	833		
LOOP	= 000020	588#	729	757	792	872		
LPR	= 000002	588#	663*	1067*				
MFULL	014050	1111#						
MSENAB	= 000040	588#	1002	1007	1069			
MSG0	012576	918	959#					
MSG1	012657	921	959#					
MSG2	012662	945	959#					
MSG3	012707	949	959#					
MSG4	012732	959#	986	1049				
MSG5	013064	959#						
NODAT	= 000200	588#	725	788	869			
NORMON	013764	1089	1091	1093#				
NOXMON	013472	1029	1031	1033#				
ORUN	= 040000	588#	1083					
OWI	= 000002	588#	688					
OWO	= 000001	588#	685					
PARAM1	011012	597#	654	661	666	1017	1065	

CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

CROSS REFERENCE TABLE -- USER SYMBOLS

[illegible]

CZDZBC.P11

10-NOV-81 11:13

CROSS REFERENCE TABLE -- MACRO NAMES

BOX	1#	589	616	642	969	1043								
DCPARM	1#													
DHDOC1	1#													
DHPARM	1#													
DJPARM	1#													
DLPARM	1#													
DPPARM	1#													
DQDOC1	1#													
DQPARM	1#													
DUPARM	1#													
DUPPAR	1#													
DVDOC1	1#													
DVPARM	1#													
DZPARM	1#	560												
HELLO	1#													
HLT	588#	721	753	785	815	847	863	955	1023	1076	1081	1100	1101	
SEQUAT	1#	588												
SINTF	1#	588												
SITEP	1#	675												
SSERV	1#	631												

. ABS. 014170 000

ERRORS DETECTED: 0

CZDZBC,CZDZBC/CRF/SOL/NL:TOC=ITEP1.MAC,CZDZBC.P11

RUN-TIME: 3 3 .3 SECONDS

RUN-TIME RATIO: 55/7=7.7

CORE USED: 16K (31 PAGES)