IDENTIFICATION

PRODUCT CODE: AC-8532D-MC
PRODUCT NAME: CZDLODO DL11 OVRLY FOR ITEP
PROGRAM DATE: MARCH 1978
MAINTAINER: DIAGNOSTICS
AUTHORS: R. A. JONES, JOHN EGOLF
REVISED BY: FAY BASHAW 1/21/75

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.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1974, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:
DIGITAL PDP UNIBUS MASSBUSS
DEC DECUS DECTAPE
1.0 ABSTRACT.

THIS PROGRAM IS DESIGNED AS A MAINTENANCE AID FOR
FIELD SERVICE PERSONNEL. 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 CZDLO0O DL11 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 600 AND START TO ENTER PARAMS FROM CONSOLE TTY, PROCEED TO SECTION B.
   2. LOAD ADDRESS 600 AND SET SWITCH REGISTER BIT 1 IS BEFORE
      STARTING TO ENTER PARAMS FROM CONSOLE SWITCHES, PROCEED TO SECTION C.
      *THE PROGRAM MAY BE RESTARTED AT LOC 200 (ONCE PARAMETERS HAVE ALREADY BEEN SELECTED)

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

   1. THE PROGRAM WILL TYPE OUT THE NAME OF THE VARIABLE OVERLAY.
      A. IF YOU WISH TO SETUP JUST THE INDICATED OVERLAY, TYPE A RETURN
         B. IF YOU WISH TO SETUP A DNI1, TYPE IN DN
         C. IF YOU WISH TO SETUP A DM11, TYPE IN DM

      IF DN OR DM11 WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS
      VECTOR ETC. REFERED TO IN STEPS 2 THRU 7, PERTAIN TO THE DNI1 OR DM11.

   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. TYPE IN ACTUAL BUS ADDRESS

   3. THE PROGRAM WILL TYPE OUT THE DEFAULT VECTOR ADDRESS
      A. TYPE A CAR, RETURN TO USE DEFAULT ADDRESS
         B. TYPE IN ACTUAL VECTOR ADDRESS

   4. THE PROGRAM WILL TYPE OUT THE DEFAULT INTERFACE PRIORITY
      NOTE: 200=PRI0 4, 240=PRI0 5, 300=PRI0 6, ETC.
A. Type a car, return to use default value
B. Type in actual value

5. The program will typeout the default value of Param#1
   If required by the ISM (see sect. 10.0 in overlay listing for parameter description)
   A. Type a car, return to use default value
   B. Type in actual value

6. The program will typeout the default value of Param#2
   If required by the ISM
   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 DN11B.

9. The program will request that switch register be set.
   A. Set 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
   SMR1=SETUP DM-11B ISR
   SMR13=SETUP DM-11 ISR
   SMR=DDDDDD=SETUP VARIABLE ISR

2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED:
   SETUP SEQUENCE IS: DM11, 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 (DM11 AND DM1B PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.
   F. GO BACK TO STEP A IF THIS SETUP WAS FOR DM 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 PROCEDURES
TO BE USED, INCLUDING THE TYPE OF MODEMS, THE TYPE OF
INTERFACE BEING USED AT THE OTHER CPU AND THE MODES OF OPERATION,
DATA AND PARAMETERS TO BE USED AT EACH CPU.

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

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

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

SWI4 = 1 SINGLE PASS
SWI4 HAS NO EFFECT IF SWO4 = 0
SWI3 = 1 INHIBIT ERROR TYPEOUTS
SWI2 = 1 INHIBIT ALL TYPEOUTS EXCEPT ERRORS IF SWI2 = 0 AND SWI4 = 0 END PASS IS TYPED AND TRANSMITTED/RECEIVED DATA IS TYPED.
SWI1 = 1 USE PREVIOUSLY SPECIFIED DATA
SWI0 = 1 DATA SELECT (WITH SWO0)

SWO9 = 1 DATA SELECT (WITH SWO0)

DO = 1 GET DATA FROM OPERATOR

01 = TEST MESSAGE 01 (51 QUICK BROWN FOX)
10 = TEST MESSAGE 10 (51 NUMERICS)
11 = TEST MESSAGE 11 (51 COMTEST/QUICK BROWN FOX/NUMERICS)

SWO8 = 1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)

SWO7 = 1 DO NOT TEST RECEIVED DATA

SWO6 = 1 MONITOR TRANSMITTED 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.

SWO4 = 1 RETURN TO MONITOR FOR END PASS

WHEN SWO4 = 0 PROGRAM LOOPS IN THE OVERLAY NEVER RETURNING TO THE MONITOR.

SWO3 = 1 INTERNAL LOOPBACK MODE

SWO2 = 1 EXTERNAL LOOPBACK MODE

SWO1 = 1 ONE-WAY-IN MODE

SWO0 = 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
   6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER
   VALUE, LAST DIGIT FOLLOWED BY <CR>
2. 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 1G
(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 ODD TO 377)
SEPERATED BY SPACES AND TERMINATED BY +UP ARROW).
I.E. ABCD ODD 123 377+ EFG (CR.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 ODD CHARACTER IS SEEN THE TRANSMISSION IS STOPPED. DURING RECEPTION,
WHEN A ODD CHARACTER IS RECEIVED THE RECEIVER IS SHUT OFF.
IF THE MESSAGE WAS INPUED BY THE OPERATOR, THE TERMINATING CHARACTERS ARE ADDED.
TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY \( \text{GO} \))
2. VERIFIES THE DATA AGAINST THE DATA SELECTED BY SW09 AND SW1D (SW7=0)
3. TRANSMITS THE DATA SELECTED BY SW09 AND SW1D (SWB=0) OR
   TRANSMITS THE RECEIVED DATA (SWB=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 (SWD7=0)
7. RETURNS TO MONITOR FOR "END PASS". (SWD4=1) OR
   GO TO STEP 1(SWD4=0)

ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA(SWD7=0)
3. RETURNS TO MONITOR FOR "END PASS"(SWD4=1) OR
   GO TO STEP 1(SWD4=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". (SWD4=1) OR
   GO TO STEP 1(SWD4=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 PRINT OUT 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 SW0=0 THE OVERLAY WILL CONTINUE TO
   TRANSMIT/RECEIVE DATA.

   IF SW0=1 THE OVERLAY WILL RETURN
   TO THE MONITOR AND TYPE "END PASS".

   IF BOTH SW0=1 AND SW1=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.
   THEN TYPE EITHER:
   #Wxxxxxxxx TO PRINTOUT THE 8 WORDS
   AT LOC xxxxxx.

   #Bxxxxxxxx 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 1000 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 R R R R R R IS THE RECEIVE BUFFER (UP TO 512 CHARACTERS)
T T T T T T IS THE TRANSMIT BUFFER (UP TO 512 CHARACTERS)
B B B IS THE BAD DATA CHARACTER
G G G 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 (0DH) IS NOT DETECTED
WITHIN 512 CHARACTERS A "BUFFER FULL" PRINTOUT WILL OCCUR.

7.0 RESTRICIONS

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

<table>
<thead>
<tr>
<th>CPU #1</th>
<th>CPU #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONE-WAY-OUT</td>
<td>ONE-WAY-IN</td>
</tr>
<tr>
<td>ONE-WAY-IN</td>
<td>ONE-WAY-OUT</td>
</tr>
<tr>
<td>EXTERNAL-LOOPBACK</td>
<td>INTERNAL-LOOPBACK</td>
</tr>
<tr>
<td>INTERNAL-LOOPBACK</td>
<td>EXTERNAL-LOOPBACK</td>
</tr>
<tr>
<td>EXTERNAL-LOOPBACK</td>
<td>EXTERNAL-LOOPBACK (FULL DUPLEX)</td>
</tr>
</tbody>
</table>

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 1, 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 OCCUR 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 0111) IF THE PROGRAM IS RESTARTED
IN THE MIDDLE OF A MESSAGE AT LOC 204 OR BY HITTING OR
AN IMMEDIATE ERROR MESSAGE FROM MAYNARD WILL BE RE-
8.0 MISCELLANEOUS
ITDP 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 CZILDOO DLI INTERFACE SERVICE PARMS ARE SETUP, AS SPECIFIED BY THE OPERATOR, BY THE ITDP CONTROL PROGRAM.
TIME: PROVIDES A MEANS OF MEASURING ELASPED TIME. IT IS INCREMENTED EVERY SECOND BY A CLOCK INTERRUPT ROUTINE IN ITDP.

9.2 WHEN THE OVERLAY IS FIRST ENTERED BY ITDP 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: $NOW, IF "ONE WAY IN" MODE WAS SELECTED. $NOW, IF "ONE WAY OUT" MODE WAS SELECTED. $ILB, IF "INTERNAL LOOP BACK" MODE WAS SELECTED. $XLB, IF "EXTERNAL LOOP BACK" WAS SELECTED.

9.31 $NOW: IN THIS ROUTINE THE RECEIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR THE RECEPTOR TO FINISH. IF NOTHING IS RECEIVED FOR 60 SECS A "WAITING" MESSAGE IS TYPED. WHEN THE TRANSACTION IS DONE, THE PROGRAM EITHER DATA IF SWITCHES PERMIT, AND TYPES END PASS DEPENDING ON A SETTING.

9.32 $NOW: THE TRANSACTION IS INITIALIZED AND PROGRAM LOOPS WAITING FOR THE TRANSMITTER TO FINISH. A "WAITING" MESSAGE IS TYPED EVERY 30 SECS IF THERE IS NO ACTION. WHEN THE TRANSMITTER IS DONE, THE PROGRAM EITHER LOOPS BACK TO $NOW OR TYPES END PASS DEPENDING ON SWITCH SETTINGS.

9.33 $ILB: THE RECEIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR THE RECEPTOR TO FINISH. A "WAITING" MESSAGE IS TYPED EVERY 30 SECS IF THERE IS NO ACTION. WHEN THE RECEIVER IS DONE, PROGRAM CHECKS DATA IF SWITCH SETTINGS PERMIT, AND TYPES END PASS IF SWITCH SETTINGS PERMIT. THEN THE TRANSMITTER IS INITIALIZED, A "WAITING" MESSAGE IS TYPED EVERY 30 SECS IF NO ACTION. WHEN TRANSMITTER IS DONE, PROGRAM RETURNS TO START OF ROUTINE. ($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
MATCH 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 ($OM0, $OM0, $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 PARMS 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 STARTX:
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 XISR; 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 RSR1, THE
CSA AND OBA 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 DL11

PARAM#1 MUST BE ALL ZEROS.
PARAM#2 BIT 0 OF THIS PARAMETER IS CHECKED BY THE SOFTWARE TO RUN
EITHER FULL-DUPLEX OR HALF-DUPLEX. BIT0=1 SELECTS FULL-DUPLEX,
BIT0=0 SELECTS HALF-DUPEX. DEFAULT IS HALF-DUPLEX, ALL OTHER BITS MUST BE ZEROS.

PARAM#3 IS USED BY SOFTWARE TO DETERMINE IF TEST IS TO BE RUN A MODEM
OTHER THAN A STANDARD AMERICAN MODEM. FOR EXAMPLE, EUROPEAN MODEMS
EMPLOY A CABLE TO INHIBIT CARRIER DETECT (BIT2) WHEN REQUEST TO
SEND IS ASSERTED. IF THIS DIAGNOSTIC WILL BE RUN ON A EUROPEAN
MODEM, MANUALLY SET PARAM3 TO ZERO(0). OTHERWISE LEAVE VALUE AT
MINUS ONE(177777).
;*******************************************************************************
; DLL INTERFACE SERVICE PARAMS
;*******************************************************************************

; DLI: 11000 /DL /
; ISR NAME
; BA: 17610
; BUS ADDRESS
; RV: 300
; VECTOR ADDRESS
; PRIOR: 000
; PRIORITY
; PARAM1: 000000
; PARAM #1
; PARAM2: 000000
; PARAM #2
; PARAM3: 17777
; PARAM #3
; INRA: .WORD 0
; INITIAL READ DATA ADDRESS
; IDX: .WORD 0
; INITIAL XMIT DATA ADDRESS
; SETTLE: .WORD 0
; LINE SETTLE DELAY FLAG
; B2016: .WORD 0
; ADDR OF BIN TO OCT TYPE ROUTINE
; TIME: .WORD 0
; TIMER
; WSTART: .WORD 0
; ADDR OF START OF PROGRAM
; TXTERM: .BYTE 000
; TRANSMITTER TERMINATING CHAR.
; RXTERM: .BYTE 001
; RECEIVER TERMINATING CHAR.
; FLAG: .WORD 0
; DISPLAY: 177570

;*******************************************************************************
; CONSTANTS + WORKING STORAGE
;*******************************************************************************

;XRFLG=100000
;XMIT COMPLETE FLAG
;XRFLG=40000
;RCV COMPLETE FLAG
;DSFLO=20000
;DATA SET STATUS CHANGE FLAG
;BIT13=20000
;INHIBIT PRINTOUTS

;SXC: 0
;SXC: SAVED XMIT CSR
;SACR: 0
;SACR: SAVED RCV CSR
;ERCR: 0
;ERCR: RCV CSR SAVED ON ERROR
;RDCR: 0
;RDCR: RCV CSR SAVED ON DF CHANGE
;DSRSTAT: 0
;DSRSTAT: RCV DATA ADDR.

;XCC: 0
;XCC: XMIT CHAR COUNT
;RCC: 0
;RCC: RCV CHAR COUNT
;ROA: 0
;ROA: RCV DATA ADDR.
;XDA: 0
;XDA: XMIT DATA ADDR.

;TKS: 177560
;TKS: DATA ADDR.
;TKB: 177562
;TKB: DATA ADDR.
;TPB: 177564
;TPB: DATA ADDR.
;TBP: 177566
;TBP: DATA ADDR.

;FULL_DUPLEX=000001
DLII-X INTERFACE SERVICE ROUTINE

START:

NOC: MOV JSR, R0: SETUP MODE IN RD
BIC 8177400, R0: ;STRIP JUNK
MOV RTV, R2: SETUP
MOV U, (R2)+: INTERRUPT
MOV PRIOR, (R2)+: VECTORS
MOV #15, (R2)+: SETUP BUS ADDR INDEX
MOV PARAM1, #CSR: SETUP VARIABLES
MOV PARAM2, R2
BIC #0001, R2
MOV R2, #CSR(4): IN CSR'S

ROUTINE USED TO GOTO
SUBROUTINE, DEPENDENT
ON MODE SELECTED.

GO:

CLR TIME
CLR DELAY
CLR STOP
BIT #0,0,0, MODE
BEQ 1%
JMP $040
1%: BIT #0,0,1, MODE
BEQ 2%
JMP $014
2%: BIT #1L0, MODE
BEQ 3%
JMP $10B
3%: BIT #1L1, MODE
BEQ 4%
JMP $104
4%: HALT
BR -.2

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

SOWI: KBDIN
JSR PC, STARTR
C02

**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."**

**%W0: KBDM PC STARTX**

**JSR TIME**

**: BIT $XFLG,STAT**

**BNE 2%**

**CMP TIME, #100**

**BLO 3%**

**MOV $ACSR, R2**

**MOV $XCSR(A4), R3**

**HLT TIME**

**CLR TIME**

**BR 1%**

**%W: MOV $35, BACK**

**JMP EOP**

**3%: BR %W0**
ROUTINE USED IF INTERNAL LOOP BACK" HAS BEEN SELECTED.

WHEN IN THIS MODE; HALF DUPLEX IS THE
ONLY MODE AVAILABLE.

INTERNAL LOOP BACK MEANS THAT THE RECEIVER IS "TURNED ON"
AND A COMPLETE MESSAGE IS RECEIVED. IF DATA IS TO BE CHECKED
IT IS IF "END PASS" IS DESIRED; IT IS GIVEN.

THEN THE TRANSMITTER IS ENABLED, AFTER THE WHOLE MESSAGE
TRANSMITTED. THE CYCLE IS REPEATED AS ABOVE.

$ILB: KBIDN
JSR PC STARTX

1$: BIT @FLG, STAT

BNE 2$

JSR PC TESTD

3$: BIT @FLG, STAT

BIT @LOOP, ASWR

B EQ 4$

JSR @MPS, BACK

MOP @EQD

1$: BIT @MDO, ASWR: USE EXTERNAL DATA?

B EQ 7$

BR IF O

JSR @IQA, @R2

SET POINTER

MOV @IQA, @R3

MOV @XQA, XQA

MOV @R3, @R3

MCX @R2, @R3

JCX @R3

BLX @R3

INR @R3

INC @R3

JMP @MPS, BACK

7$: CLR @TIME

JSR PC STARTX

5$: BIT @FLG, STAT

BNE 8$

JSR @MPS, BACK

MOP @EQD

1$: BIT @MDO, ASWR

B NE 9$

BIT @XQA, @R2

MCX @R3, @R3

JCX @R3

JMP $ILB

8$: CLR @TIME

JSR PC STARTX

6$: BIT @FLG, STAT

BNE 11$

BIT @XQA, @R2

MCX @R3, @R3

JCX @R3

JMP $ILB
ROUTINE USED IF "EXTERNAL LOOP BACK" WAS SELECTED.

"EXTERNAL LOOP BACK" MEANS THAT THE TRANSMITTER IS FIRST TURNED ON IF FULL DUPLEX; AND THE WHOLE MESSAGE IS TRANSMITTED; THEN THE RECIIVER 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.

WHEN THE PROGRAM IS RUNNING IN FULL DUMPLEX THE PROGRAM WAITS FOR BOTH THE RECIIVER AND TRANSMITTER TO FINISH THEN RESTARTS THE RECIIVER AND TRANSMITTER.
STARTX:
TST    DELAY   ; IF SWX$=1 & SWY$=0 WAIT BEFORE TURNING ON TX
        BEQ   18   ; NO GO AHEAD AND TURN ON TX
        CLR   TEMPL   ; PREPARE FOR DELAY
        MOV   #7,TEMP2
        ADD   #11,TEMP1   ; INCREMENT DELAY.
        DEC   TEMP2
        DEC   BNE   -14   ; ZERO POINTER.
        CLR   DELAY   ; SET UP XMT DATA ADDR.
        MOV   IXDA, XDA   ; SET REQUEST TO SEND
        BIS   #DTR, XRCSR
        STOP
        TST   BNE   "MSG4"
        TYPE   MSG4
        HALT
        STOP
        MOV   #02714, 000001
        MOV   #011014, 256
        BIT   #FULL Duplex, param2
        FULL Duplex?
        BNE   #10000, XRCSR   ; CARRIER UP?
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
        BR IF YES
        BNE   #10000, XRCSR   ; CARRIER UP?
TRANSMIT INTERRUPT SERVICE ROUTINE

ISR: NOP

TRUE XMITTING?
BNE XISAR \\
BIC RXFLG, STAT \\
SET XMIT COMPLETE FLAG \\
BIC RXS, XCSR \\
RESET REQUEST TO SEND \\
BIC RXS \\
XMIT INTERRUPT ENABLE \\
BNE XISAR

XISAR1: MOVX XDA, XBUF(R4) \\
XMIT NEXT CHAR. \\
BIT #100, JSP \\
MONITOR OUTPUT? \\
BEQ NOXMON \\
BR IF NO \\
TST RP \\
IS IT AVAILABLE? \\
BPL NOXMON \\
BR IF NO

NOXMON: INC XDA \\
INCREMENT ADDRESS \\
TST TXFLG \\
FIRST CHAR? \\
BEQ IS \\
BR IF NO \\
CLP TIME \\
CLEAR INTERRUPT ENABLE \\
CLR TXFLG \\
CLEAR FLAG \\
RTI \\
RETURN FROM INTERRUPT

RECEIVE INTERRUPT SERVICE ROUTINE

ISR: NOP

IS RECEIVER DONE BIT SET \\
BNE RISAR \\
STORE CHAR \\
MOVX RBUF(R4), R1 \\
STORE CHAR \\
BICB RDO, R1 \\
STRIP A BIT \\
MOVX R1, RXDA \\
MOVE CHAR TO INBUF \\
BIT 64H, JSP \\
MONITOR INPUT? \\
BEQ RXNOM \\
BR IF NO \\
TST RP \\
IS IT AVAILABLE? \\
BPL RXNOM \\
BR IF NO

RXNOM: INC RDA \\
BUMP POINTER \\
CLRB RDCA \\
CLEAR NEXT CHAR POSITION \\
DEC RCX \\
DECREMENT CHAR COUNTER \\
BNE IS \\
CHAR NOT FULL \\
BIC RXIE, XCSR \\
RESET INTERRUPT ENABLE \\
MOV RXIE(R4), R2 \\
SETUP RXIE CSRS \\
MOV RXS, XCSR(R4), R3 \\
SET RXTX CSRS \\
HLT+6 \\
WAIT FOR RECEIVE \\
JSR PC, START \\
initialize receive

IS: CMPB RXTERM, R1 \\
IS IT DTE FEED? \\
BNE RISAR \\
BR IF NO \\
BIC RXIE, RXCSR \\
DISABLE INTERRUPTS \\
BIC RXF, STAT \\
SET RXF COMPLETE FLAG \\
TST RXBUF(R4) \\
IS THERE A DATA ERROR? \\
BPL RISAR \\
BR IF NO
1040: MOV ACSR, ERCSR ; SAVE ACSR
1041: MOV ABUF(R4), ERDBR ; SAVE ADDR
1042: TST ACSR ; IS THERE A DATA SET STATUS CHANGE
1043: BPL RISR3 ; BR IF NO
1044: MOV ACSR, DSSTAT ; SAVE STATUS
1045: BIS DSFLG, STAT ; SET FLAG
1046: CLR SCNFLAG ; CLEAR FLAG
1047: CLR TIME ; RESET TIMER
1048: RTI ; RETURN FROM INTERRUPT
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OHI</td>
<td>00000002</td>
</tr>
<tr>
<td>OGO</td>
<td>00000001</td>
</tr>
<tr>
<td>PARAM1</td>
<td>011011</td>
</tr>
<tr>
<td>PARAM2</td>
<td>011014</td>
</tr>
<tr>
<td>PARAM3</td>
<td>011016</td>
</tr>
<tr>
<td>PE</td>
<td>011003</td>
</tr>
<tr>
<td>PRIOR</td>
<td>00000000</td>
</tr>
<tr>
<td>PRTY0</td>
<td>00000000</td>
</tr>
<tr>
<td>PRTY2</td>
<td>00000000</td>
</tr>
<tr>
<td>PRTY3</td>
<td>00000000</td>
</tr>
<tr>
<td>PRTY4</td>
<td>00000000</td>
</tr>
<tr>
<td>PRTY5</td>
<td>00000000</td>
</tr>
<tr>
<td>PRTY6</td>
<td>00000000</td>
</tr>
<tr>
<td>PRTY7</td>
<td>00000000</td>
</tr>
<tr>
<td>R / T</td>
<td>00000000</td>
</tr>
<tr>
<td>RBUF</td>
<td>00000000</td>
</tr>
<tr>
<td>R / C</td>
<td>011064</td>
</tr>
<tr>
<td>RD</td>
<td>00000000</td>
</tr>
<tr>
<td>RDA</td>
<td>011066</td>
</tr>
<tr>
<td>RDAX</td>
<td>012946</td>
</tr>
<tr>
<td>R / N</td>
<td>00000000</td>
</tr>
<tr>
<td>RFLG</td>
<td>042200</td>
</tr>
<tr>
<td>R / T</td>
<td>040000</td>
</tr>
<tr>
<td>RIE</td>
<td>00000000</td>
</tr>
<tr>
<td>R / S</td>
<td>013662</td>
</tr>
<tr>
<td>R / S1</td>
<td>014002</td>
</tr>
<tr>
<td>R / S2</td>
<td>014022</td>
</tr>
<tr>
<td>R / S3</td>
<td>014036</td>
</tr>
<tr>
<td>RIV</td>
<td>011036</td>
</tr>
<tr>
<td>R / T</td>
<td>00000000</td>
</tr>
<tr>
<td>R / T</td>
<td>100410</td>
</tr>
<tr>
<td>R / T</td>
<td>011041</td>
</tr>
<tr>
<td>SAVR1</td>
<td>013040</td>
</tr>
<tr>
<td>SAVR2</td>
<td>013040</td>
</tr>
<tr>
<td>SAVR3</td>
<td>013040</td>
</tr>
<tr>
<td>SAVR4</td>
<td>013040</td>
</tr>
<tr>
<td>SAVR5</td>
<td>013040</td>
</tr>
<tr>
<td>SAVR6</td>
<td>013040</td>
</tr>
<tr>
<td>SCAM1</td>
<td>012344</td>
</tr>
<tr>
<td>SCAM2</td>
<td>012420</td>
</tr>
<tr>
<td>SETTLE</td>
<td>011056</td>
</tr>
<tr>
<td>SDCFG</td>
<td>012540</td>
</tr>
<tr>
<td>SCRER</td>
<td>011053</td>
</tr>
<tr>
<td>SRO</td>
<td>002000</td>
</tr>
<tr>
<td>START</td>
<td>011022</td>
</tr>
<tr>
<td>STARTX</td>
<td>013022</td>
</tr>
<tr>
<td>STOP</td>
<td>010501</td>
</tr>
<tr>
<td>STPS</td>
<td>100410</td>
</tr>
<tr>
<td>SWH</td>
<td>011044</td>
</tr>
<tr>
<td>WXCSR</td>
<td>011050</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMI</td>
<td>664</td>
</tr>
<tr>
<td>OD</td>
<td>661</td>
</tr>
<tr>
<td>P</td>
<td>647</td>
</tr>
<tr>
<td>Q</td>
<td>648</td>
</tr>
<tr>
<td>R</td>
<td>985</td>
</tr>
<tr>
<td>S</td>
<td>643</td>
</tr>
<tr>
<td>T</td>
<td>645</td>
</tr>
<tr>
<td>U</td>
<td>880</td>
</tr>
<tr>
<td>V</td>
<td>882</td>
</tr>
<tr>
<td>W</td>
<td>1033</td>
</tr>
<tr>
<td>X</td>
<td>1072</td>
</tr>
<tr>
<td>Y</td>
<td>1095</td>
</tr>
<tr>
<td>Z</td>
<td>1098</td>
</tr>
<tr>
<td>A</td>
<td>1026</td>
</tr>
<tr>
<td>B</td>
<td>1083</td>
</tr>
<tr>
<td>C</td>
<td>1029</td>
</tr>
<tr>
<td>D</td>
<td>1074</td>
</tr>
<tr>
<td>E</td>
<td>1081</td>
</tr>
<tr>
<td>F</td>
<td>1082</td>
</tr>
<tr>
<td>G</td>
<td>1034</td>
</tr>
<tr>
<td>H</td>
<td>1085</td>
</tr>
<tr>
<td>I</td>
<td>1093</td>
</tr>
<tr>
<td>J</td>
<td>1092</td>
</tr>
<tr>
<td>K</td>
<td>1095</td>
</tr>
<tr>
<td>L</td>
<td>1071</td>
</tr>
<tr>
<td>M</td>
<td>1096</td>
</tr>
<tr>
<td>N</td>
<td>1099</td>
</tr>
<tr>
<td>O</td>
<td>1100</td>
</tr>
<tr>
<td>P</td>
<td>1103</td>
</tr>
<tr>
<td>Q</td>
<td>641</td>
</tr>
<tr>
<td>R</td>
<td>1048</td>
</tr>
<tr>
<td>S</td>
<td>970</td>
</tr>
<tr>
<td>T</td>
<td>1091</td>
</tr>
<tr>
<td>U</td>
<td>935</td>
</tr>
<tr>
<td>V</td>
<td>873</td>
</tr>
<tr>
<td>W</td>
<td>874</td>
</tr>
<tr>
<td>X</td>
<td>875</td>
</tr>
<tr>
<td>Y</td>
<td>876</td>
</tr>
<tr>
<td>Z</td>
<td>877</td>
</tr>
<tr>
<td>A</td>
<td>878</td>
</tr>
<tr>
<td>B</td>
<td>940</td>
</tr>
<tr>
<td>C</td>
<td>914</td>
</tr>
<tr>
<td>D</td>
<td>917</td>
</tr>
<tr>
<td>E</td>
<td>938</td>
</tr>
<tr>
<td>F</td>
<td>990</td>
</tr>
<tr>
<td>G</td>
<td>914</td>
</tr>
<tr>
<td>H</td>
<td>917</td>
</tr>
<tr>
<td>I</td>
<td>938</td>
</tr>
<tr>
<td>J</td>
<td>1016</td>
</tr>
<tr>
<td>K</td>
<td>1027</td>
</tr>
<tr>
<td>L</td>
<td>1039</td>
</tr>
<tr>
<td>M</td>
<td>1103</td>
</tr>
<tr>
<td>N</td>
<td>1016</td>
</tr>
<tr>
<td>O</td>
<td>1027</td>
</tr>
<tr>
<td>P</td>
<td>1039</td>
</tr>
<tr>
<td>Q</td>
<td>1103</td>
</tr>
<tr>
<td>R</td>
<td>980</td>
</tr>
<tr>
<td>S</td>
<td>980</td>
</tr>
</tbody>
</table>
# CROSS REFERENCE TABLE -- USER SYMBOLS

<table>
<thead>
<tr>
<th>TEMP1</th>
<th>013534</th>
<th>952#</th>
<th>954#</th>
<th>992#</th>
<th>994#</th>
<th>1024#</th>
<th>1025#</th>
<th>1037#</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEMP2</td>
<td>013536</td>
<td>954#</td>
<td>956#</td>
<td>993#</td>
<td>996#</td>
<td>1038#</td>
<td>1039#</td>
<td>1041#</td>
</tr>
<tr>
<td>TEST01</td>
<td>012300</td>
<td>703</td>
<td>756</td>
<td>857#</td>
<td>890#</td>
<td>932#</td>
<td>934#</td>
<td>967#</td>
</tr>
<tr>
<td>TEST02</td>
<td>124370</td>
<td>903</td>
<td>909</td>
<td>912#</td>
<td>917#</td>
<td>922#</td>
<td>925#</td>
<td>930#</td>
</tr>
<tr>
<td>TIE</td>
<td>000001</td>
<td>851</td>
<td>861</td>
<td>862</td>
<td>1004</td>
<td>1008</td>
<td>1049</td>
<td>1062</td>
</tr>
<tr>
<td>TIME</td>
<td>011032</td>
<td>854#</td>
<td>859#</td>
<td>863#</td>
<td>864#</td>
<td>884#</td>
<td>896#</td>
<td>983#</td>
</tr>
<tr>
<td>TB8</td>
<td>011074</td>
<td>919</td>
<td>919</td>
<td>834#</td>
<td>835</td>
<td>840#</td>
<td>844#</td>
<td>976#</td>
</tr>
<tr>
<td>TK8</td>
<td>011074</td>
<td>630#</td>
<td>630#</td>
<td>1057#</td>
<td>1057#</td>
<td>1062#</td>
<td>1063#</td>
<td>1104#</td>
</tr>
<tr>
<td>TPS</td>
<td>011100</td>
<td>632#</td>
<td>632#</td>
<td>1057#</td>
<td>1057#</td>
<td>1062#</td>
<td>1063#</td>
<td>1104#</td>
</tr>
<tr>
<td>TRNFLG</td>
<td>013532</td>
<td>1000#</td>
<td>1004#</td>
<td>1010#</td>
<td>1060#</td>
<td>1064#</td>
<td>1065#</td>
<td>1069#</td>
</tr>
<tr>
<td>TSTQRT</td>
<td>012334</td>
<td>891#</td>
<td>893#</td>
<td>894#</td>
<td>895#</td>
<td>907</td>
<td>921</td>
<td>923</td>
</tr>
<tr>
<td>TXhtar</td>
<td>011040</td>
<td>901#</td>
<td>902#</td>
<td>1065#</td>
<td>1074#</td>
<td>1076#</td>
<td>1077#</td>
<td>1078#</td>
</tr>
<tr>
<td>TYPE</td>
<td>010400</td>
<td>891#</td>
<td>894#</td>
<td>897#</td>
<td>907#</td>
<td>921#</td>
<td>923#</td>
<td>925#</td>
</tr>
<tr>
<td>XBUF</td>
<td>000006</td>
<td>891#</td>
<td>891#</td>
<td>1052#</td>
<td>1052#</td>
<td>1053#</td>
<td>1053#</td>
<td>1054#</td>
</tr>
<tr>
<td>XCC</td>
<td>000001</td>
<td>634#</td>
<td>634#</td>
<td>550#</td>
<td>550#</td>
<td>553#</td>
<td>553#</td>
<td>554#</td>
</tr>
<tr>
<td>XCSR</td>
<td>000004</td>
<td>1008#</td>
<td>1049#</td>
<td>1062#</td>
<td>1062#</td>
<td>1067#</td>
<td>1067#</td>
<td>1067#</td>
</tr>
<tr>
<td>XDA</td>
<td>011070</td>
<td>581#</td>
<td>581#</td>
<td>696#</td>
<td>728#</td>
<td>729#</td>
<td>790#</td>
<td>822#</td>
</tr>
<tr>
<td>XFLG</td>
<td>100000</td>
<td>613#</td>
<td>723#</td>
<td>732#</td>
<td>785#</td>
<td>794#</td>
<td>815#</td>
<td>826#</td>
</tr>
<tr>
<td>XISR</td>
<td>013534</td>
<td>644#</td>
<td>1044#</td>
<td>1054#</td>
<td>1054#</td>
<td>1055#</td>
<td>1055#</td>
<td>1056#</td>
</tr>
<tr>
<td>XISR</td>
<td>013537</td>
<td>1046#</td>
<td>1056#</td>
<td>1056#</td>
<td>1056#</td>
<td>1056#</td>
<td>1056#</td>
<td>1056#</td>
</tr>
<tr>
<td>XISR2</td>
<td>013534</td>
<td>1050#</td>
<td>1050#</td>
<td>1050#</td>
<td>1050#</td>
<td>1050#</td>
<td>1050#</td>
<td>1050#</td>
</tr>
<tr>
<td>XLB</td>
<td>000004</td>
<td>581#</td>
<td>670#</td>
<td>1001#</td>
<td>1022#</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XWAIT</td>
<td>014012</td>
<td>581#</td>
<td>581#</td>
<td>581#</td>
<td>581#</td>
<td>581#</td>
<td>581#</td>
<td>581#</td>
</tr>
<tr>
<td>ILB</td>
<td>011410</td>
<td>669#</td>
<td>752#</td>
<td>795#</td>
<td>795#</td>
<td>795#</td>
<td>795#</td>
<td>795#</td>
</tr>
<tr>
<td>ISW</td>
<td>011430</td>
<td>666#</td>
<td>689#</td>
<td>709#</td>
<td>709#</td>
<td>709#</td>
<td>709#</td>
<td>709#</td>
</tr>
<tr>
<td>ISN</td>
<td>12352</td>
<td>663#</td>
<td>720#</td>
<td>737#</td>
<td>737#</td>
<td>737#</td>
<td>737#</td>
<td>737#</td>
</tr>
<tr>
<td>JXLB</td>
<td>011710</td>
<td>62#</td>
<td>80#</td>
<td>84#</td>
<td>84#</td>
<td>84#</td>
<td>84#</td>
<td>84#</td>
</tr>
<tr>
<td>JZB</td>
<td>014050</td>
<td>86#</td>
<td>87#</td>
<td>78#</td>
<td>78#</td>
<td>78#</td>
<td>78#</td>
<td>78#</td>
</tr>
</tbody>
</table>

ABS. 014050 900

ERRORS DETECTED: 0

DSKZ: CZDLOD.DSK2: CZDLOD.SEQ=DSKZ: ITP1.MAC, DSKZ: CZDLOD.P11
RUN-TIME: 3.5 .3 SECONDS
RUN-TIME RATIO: 66/9=7.1
CORE USED: 16K (31 PAGES)

DOCUMENT PAGES: 26