

DQ11

OVERLAY FOR ITEP
MD-11-DZDQO-B

EP-DZDQO-B-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN U.S.A.

This microfiche card contains a grid of frames on the left side, each containing technical data. The frames are arranged in approximately 15 rows and 3 columns. The data includes various tables, charts, and text blocks, likely representing flight performance or engine parameters for the MD-11 aircraft. The text is small and difficult to read due to the resolution of the scan.

11

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZD00-B-D

PRODUCT NAME: D011 OVERLAY FOR INTERPROCESSOR TEST PROGRAM

PROGRAM DATE: OCTOBER 1976

MAINTAINER: DIAGNOSTICS

AUTHORS: R A JONES
 JOHN EGOLF
 FAY BASHAW

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

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

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

COPYRIGHT (C) 1973, 1975, BY DIGITAL EQUIPMENT CORPORATION

1.0 ABSTRACT.

THIS PROGRAM IS DESIGNED AS A MAINTENANCE AID FOR FIELD SERVICE PERSONEL. IT WILL VERIFY THE PROPER OPERATION OF A COMPLETE COMMUNICATION LINK FROM ONE PDP-11 SYSTEM TO ANOTHER OR TO A COMMUNICATION TEST CENTER.

THIS PROGRAM MUST BE USED IN CONJUNCTION WITH THE INTERPROCESSOR TEST PROGRAM(DZITP) ON A PDP-11 SYSTEM WITH A DL-11 INTERFACE.

2.0 REQUIREMENTS.

2.1 EQUIPMENT

- A. PDP-11 SYSTEM WITH 4K OF CORE.
- B. A DQ11 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 DN11B, TYPE IN DMB.

IF DN OR DN3 WAS TYPED IN STEP 1 ABOVE THEN THE BUS ADDRESS, VECTOR ETC. REFERED TO IN STEPS 2 THRU 7, PERTAIN TO THE DN11 OR DMB.

- 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 ADDR 255
 - B. TYPE IN ACTUAL VECTOR ADDRESS
- 4. THE PROGRAM WILL TYPE OUT THE DEFAULT INTERFACE PRIORITY
NOTE. 200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.

- A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
 - B. TYPE IN ACTUAL VALUE
5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
IF REQUIRED BY THE ISR. (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
- A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
 - B. TYPE IN ACTUAL VALUE
6. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#2
IF REQUIRED BY THE ISR.
- A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
 - B. ENTER ACTUAL VALUE
7. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#3
IF REQUIRED BY THE OVERLAY.
- A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
THE DN-11 WILL USE PARAM #3 AS THE # TO DIAL.
IF USING A MODEM WITHOUT AUTOMATIC HANDSHAKING,
THE NUMBER MUST TERMINATE WITH A
"END-OF-NUMBER" CHARACTER (:).
 - B. ENTER ACTUAL VALUE.
8. THE PROGRAM WILL RETURN TO STEP B1 IF THIS SETUP
WAS FOR DN11 OR DN11B8.
9. THE PROGRAM WILL REQUEST THAT SWITCH REGISTER BE SET.
- A. SETUP SWITCH REGISTER AS SPECIFIED IN STEP D.
AND TYPE A CAR. RETURN.

NOTE: IF ANY OF THE ABOVE ITEMS 2 THRU 7 WERE CHANGED BY ENTERING
NEW VALUES, THE NEW VALUE BECOMES THE DEFAULT VALUE FOR SUBSEQUENT
RESTARTS OF THE PROGRAM.

- C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
1. THE PROGRAM HALTS FOR ISR (INTERFACE SERVICE ROUTINE) SPECIFICATION
SWR14=SETUP DN-11B ISR
SWR13=SETUP DN-11 ISR
SWR=000000=SETUP VARIABLE ISR
 2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
SETUP SEQUENCE IS: DN11, DN11-88 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 DN88 PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.)
 - F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DN8.
 3. HALT FOR OPERATIONAL SWITCH SETTINGS. (SEE STEP D.)
 - A. PRESS CONTINUE TO START TESTING

BEFORE ATTEMPTING TO RUN THIS PROGRAM, THE OPERATOR MUST ASCERTAIN THE COMPLETE COMMUNICATION LOOP AND PROCEDURES TO BE USED, INCLUDING THE TYPE OF MODEMS, THE TYPE OF INTERFACE BEING USED AT THE OTHER CPU AND THE MODES OF OPERATION, DATA AND PARAMETERS TO BE USED AT EACH CPU.

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

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

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

D. OPERATIONAL SWITCH SETTINGS.

SW15=1 HALT ON ERROR
 SW14=1 SINGLE PASS
 SW14 HAS NO EFFECT IF SW04=0
 SW13=1 INHIBIT ERROR TYPEOUTS
 SW12=1 INHIBIT ALL TYPEOUTS EXCEPT ERRORS
 IF SW12=0 AND SW04=1 END PASS IS TYPED
 AND TRANSMITTED/RECEIVED DATA IS TYPED.
 SW11=1 USE PREVIOUSLY SPECIFIED DATA
 SW10=1 DATA SELECT (WITH SW09)
 SW09=1 DATA SELECT (WITH SW10)
 SW08=1 GET DATA FROM OPERATOR
 SW07=1 TEST MESSAGE #1 (SA QUICK BROWN FOX)
 SW06=1 TEST MESSAGE #2 (SB NUMERICS)
 SW05=1 TEST MESSAGE #3 (SC CONTEST/QUICK BROWN FOX/NUMERICS)
 SW04=1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)
 SW03=1 DO NOT TEST RECEIVED DATA
 SW02=1 MONITOR TRANSMITTED DATA ON CONSOLE TTY.*
 SW01=1 MONITOR RECEIVED DATA ON CONSOLE TTY.*
 * IN MANY CASES, NOT ALL DATA WILL APPEAR ON THE CONSOLE
 TTY. THIS IS ESPECIALLY TRUE WHEN THE COMM INTERFACE IS
 RUNNING AT A FASTER BAUD THAN THE CONSOLE, BUT EVEN AT EQUAL
 OR SLOWER BAUDS, ALL CHARACTERS MAY NOT APPEAR ON THE CONSOLE.

 SW04=1 RETURN TO MONITOR FOR END PASS
 WHEN SW04=0 PROGRAM LOOPS IN THE OVERLAY NEVER RETURNING TO THE MONITOR.
 SW03=1 INTERNAL LOOPBACK MODE
 SW02=1 EXTERNAL LOOPBACK MODE
 SW01=1 ONE-WAY-IN MODE
 SW00=1 ONE-WAY-OUT MODE

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1. <CR> IF NO CHANGES ARE TO BE MADE
2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3. ↑U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

IF OPERATOR SPECIFIED DATA WAS INDICATED, THE PROGRAM WILL TYPE A REQUEST FOR THE DATA. DATA MAY BE ENTERED AS ASCII CHARACTERS OR OCTAL CODE. TYPE IN THE DATA TERMINATED WITH A CR. OCTAL CODE MAY BE ENTERED BY TYPING AN ↑(UP ARROW) FOLLOWED BY THE OCTAL CODE (IN THE RANGE 000 TO 377) SEPERATED BY SPACES AND TERMINATED BY ↑(UP ARROW).
I.E. ABCD↑ 00 123 377↑ 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 TRAILING TERMINATING CHARACTER(070). 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. TRANSMITS THE DATA SELECTED BY SW09 AND SW10 (SW8=0) OR
TRANSMITS 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. WAITS FOR CLEAR TO SEND
3. TRANSMITS THE SELECTED DATA
4. RESETS REQUEST TO SEND
5. WAITS FOR MESSAGE TO BE RECEIVED
6. VERIFIES THE DATA (SW7=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR
GO TO STEP 1 (SW04=0)

ONE-WAY-IN MODE

1. THE OVERLAY WAITS FOR MESSAGE TO BE RECEIVED.
2. VERIFIES THE DATA (SW07=0)
3. RETURNS TO MONITOR FOR "END PASS" (SW04=1) OR
GO TO STEP 1 (SW04=0)

ONE-WAY-OUT MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAITS FOR CLEAR TO SEND
3. TRANSMITS SELECTED DATA
4. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR
GO TO STEP 1 (SW04=0)

E. THE OVERLAY IS THEN ENTERED AND A CONNECTION ESTABLISHED EITHER MANUALLY OR AUTOMATICALLY.

IF ONE-WAY-IN OR INTERNAL LOOPBACK MODES ARE SELECTED,
THE OVERLAY WILL SET DATA TERMINAL READY AND WAIT FOR DATA.

IF ONE-WAY-OUT OR EXTERNAL LOOPBACK MODES WERE SELECTED,
THE OVERLAY WILL SET DATA TERMINAL READY AND REQUEST TO SEND.
THE OVERLAY WILL THEN WAIT FOR CLEAR TO SEND BEFORE ATTEMPTING TO
TRANSMIT DATA.

THE PROGRAM WILL PRINTOUT A "WAITING FOR CLEAR TO SEND"
MESSAGE AND THE CONTENTS OF THE XMIT CSR EVERY 60 SECS.
UNTIL CLEAR TO SEND IS ASSERTED.

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

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

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

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

LINE FEED = RESTART PROGRAM AT LOCATION 200.

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

THEN TYPE EITHER:

#XXXXXXX 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 ADDR 5 11020 AND EXAMINE. TO FIND THE STARTING ADDRESS OF THE TRANSMIT BUFFER, LOAD ADDRESS 11022 AND EXAMINE.

5.1 NORMAL HALTS SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

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

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

RECEIVED DATA=RRRRR
DATA SHOULD BE TTTTT
DATA COMPARE ERROR; BAD DATA=BBB GOOD DATA=GGG

WHERE RRRRRR IS THE RECEIVE BUFFER (UP TO 512 CHARACTERS)
TTTTT IS THE TRANSMIT BUFFER (UP TO 512 CHARACTERS)
BBB IS THE BAD DATA CHARACTER
GGG IS THE GOOD DATA CHARACTER

IF THE INTERFACE DETECTS A DATA ERROR, THE FOLLOWING
WILL BE PRINTED BEFORE THE DATA IS COMPARED:

THERE WAS A RECEIVER ERROR. RECEIVER DATA REGISTER =XXXXXX

WHERE XXXXXX IS THE CONTENTS OF THE RECEIVER DATA REGISTER
THE LOW BYTE IS THE DATA, AND THE HIGH BYTE IS THE ERROR BITS.

IF A RECEIVE TERMINATING CHARACTER (OOI) IS NOT DETECTED
WITHIN 512 CHARACTERS A "BUFFER FULL" PRINTOUT WILL OCCUR.

7.0 RESTRICTIONS

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

CPU #1	CPU #2
ONE-WAY-OUT	ONE-WAY-IN
ONE-WAY-IN	ONE-WAY-OUT
EXTERNAL-LOOPBACK	INTERNAL-LOOPBACK
INTERNAL-LOOPBACK	EXTERNAL-LOOPBACK
EXTERNAL-LOOPBACK	EXTERNAL-LOOPBACK (FULL DUPLEX)

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTION APPLY:

IF RUNNING IN FULL DUPLEX MODE BOTH SYSTEMS
MUST BE IN EXTERNAL LOOP BACK MODE.

BOTH SYSTEMS SHOULD BE RUNNING IDENTICAL ROUTINES.

EXAMPLE:

SWITCHES 14,13,7,4 SHOULD BE THE SAME
ON BOTH CPU'S

IF PROGRAM IS WAITING IN A SCAN ROUTINE AND TYPES OUT
A "WAITING MESSAGE", IF AN INCOMING MESSAGE STARTS DURING
THE TYPE OUT, IT WILL BE LOST BECAUSE THE TYPEOUT PRIORITY
IS AT LEVEL 7. THIS WILL RESULT IN OVERRUN OR SILO OVER-
RUN ERRORS, DEPENDING ON THE DEVICE. TO AVOID THIS SITUATION
RUN WITH SWITCH 13 UP. IF OVE RUN DOES OCCURE DURING A
TYPEOUT THE PROGRAM SHOULD BE RESTARTED.

IF USING AN ASYNCHRONOUS DEVICE, MODEMS AND THE
MAYNARD TEST STATION #10 INITIALIZE DOES NOT CLEAR THE
CONNECTION (EXAMPLE THE DQ11) IF THE PROGRAM IS RESTARTED
IN THE MIDDLE OF A MESSAGE AT LOC 204 OR BY HITTING CR
AN IMMEDIATE ERROR MESSAGE FROM MAYNARD WILL BE RE-

CEIVED. THIS IS BECAUSE THE TEST STATION IS STILL LOOKING FOR THE REST OF THE INTERRUPTED MESSAGE. TO AVOID THIS ERROR RESTART PROGRAM ONLY AT THE END OF THE MESSAGE CURRENTLY BEING TRANSMITTED.

8.0 MISCELLANEOUS

ITEP WAS CHECKED OUT USING THE FOLLOWING BELL TELEPHONE MODEMS.
201A (HALF-DUPLEX SYNCHRONOUS 2000 BAUD)
202C (HALF-DUPLEX ASYNCHRONOUS 1200 BAUD)
103A (FULL-DUPLEX ASYNCHRONOUS 110 BAUD)

9.0 PROGRAM DESCRIPTION

9.1 THE D011 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: SOWI, IF "ONE WAY IN" MODE WAS SELECTED. SOWO, IF "ONE WAY OUT" MODE WAS SELECTED. SILB, IF "INTF W/ LOOP BACK" MODE WAS SELECTED. SXLB, IF "EXTERNAL LOOP BACK" WAS SELECTED.

9.31 SOWI: IN THIS ROUTINE THE RECEIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR THE RECEIVER TO FINISH. IF NOTHING IS RECEIVED FOR 60 SECS A "WAITING" MESSAGE IS TYPED. WHEN THE RECEIVER IS DONE, THE PROGRAM CHECKS DATA IF SWITCHES PERMIT, AND TYPES END PASS DEPENDING ON SWITCH SETTINGS.

9.32 SOWO: THE TRANSMITTER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR THE TRANSMITTER TO FINISH. A "WAITING" MESSAGE IS TYPED EVERY 60 SECS IF THERE IS NO ACTION. WHEN THE TRANSMITTER IS DONE, THE PROGRAM EITHER LOOPS BACK TO SOWO OR TYPES END PASS DEPENDING ON SWITCH SETTINGS.

9.33 SILB: THE RECEIVER IS INITIALIZED AND PROGRAM LOOPS WAITING FOR RECEIVER TO FINISH. A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF NO ACTION. WHEN RECEIVER IS DONE PROGRAM CHECKS DATA IF SWITCH SETTINGS PERMIT, AND END PASS IS TYPED IF SWITCH SETTINGS PERMIT. THEN THE TRANSMITTER IS INITIALIZED, A "WAITING" MESSAGE IS TYPED EVERY 60 SEC IF NO ACTION. WHEN TRANSMITTER IS DONE PROGRAM RETURNS TO START OF ROUTINE. (SILB)

9.34 SXLB: IF IN HALF DUPLEX THE TRANSMITTER IS INITIALIZED, A "WAITING MESSAGE IS TYPED EVERY 60 SEC IF THERE IS NO ACTION

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

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

IN THE RECEIVED DATA (WITHIN 1000 CHARACTERS).
IF THE RECEIVED CHARACTER IS A LINE FEED,
THE RECEIVED LOGIC IS RESET AND THE
RECEIVE COMPLETE FLAG IS SET.
IF A 'RECEIVE ERROR' IS DETECTED AT RISR:, THE
CSR AND DBR WILL BE SAVED AND PRINTED OUT
AFTER THE COMPLETE MESSAGE HAS BEEN RECEIVED.

- 9.10 THE DATA TEST SUBROUTINE AT TESTD: IS
ENTERED AFTER A COMPLETE MESSAGE HAS BEEN
RECEIVED.
IF A 'RECEIVE ERROR' HAD BEEN DETECTED,
THE CONTENTS OF THE 'RECEIVE BUFFER' AT THE
TIME THE ERROR OCCURRED WILL BE PRINTED.
THE DATA IS COMPARED UNTIL A 'ALL ZEROS'
CHARACTER IS RECOGNIZED. 'FILL' (ALL ONES)
CHARACTERS ARE IGNORED. IF A MISMATCH
IS DETECTED, THE COMPLETE CONTENTS OF THE
INPUT BUFFER AND GOOD DATA IS PRINTED.

DQ11 RESTRICTIONS

THE DQ11 HAS TWO MODES OF OPERATION IN ITEP, NORMAL (WHICH
INTERUPTS EVERY TWO CHARACTERS), AND HI-BAUD (ONE INTERUPT PER
MESSAGE, THE WHOLE MESSAGE IS TRANSMITTED OR RECEIVED ON A
COMPLETE WORD COUNT.). WHEN IN HI-BAUD MODE, DATA CANNOT BE
MONITORED ON THE CONSOLE TTY. IF SW 5=1 (MONITOR RECEIVED DATA)
AN "R" WILL BE TYPED AFTER THE WHOLE MESSAGE IS RECEIVED.
IF SW 6=1 (MONITOR TRANSMIT DATA) A "T" WILL BE TYPED AFTER THE
WHOLE MESSAGE IS TRANSMITTED. IN NORMAL MODE, EVEN IF THE CONSOLE
TTY IS FASTER OR THE SAME BAUD AS THE DQ11, NOT EVERY CHARACTER
CAN BE TYPED BECAUSE OF THE TWO CHARACTER PER INTERUPT
OPERATION. IN THIS CASE EVERY OTHER CHARACTER WILL BE TYPED
IF DATA MONITORING IS SELECTED. HI-BAUD MODE (SELECTABLE IN PARAM#2)
SHOULD BE USED FOR BAUDS OF 40,000 OR HIGHER. NORMAL MODE IS
SUFFICIENT FOR BAUDS LOWER THAN 40,000 .

10.0 PARAMETERS FOR THE DQ11

PARAM#1 IS NOT USED (0)

PARAM#2 (LOW BYTE)

BIT 0
BIT 1

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

HI-BAUD (1), DEFAULT= NORMAL (0)

(BITS 0,1 ARE NOT LOADED INTO ANY DQ11 REGISTERS, THEY ARE ONLY SOFTWARE FLAGS)

PARAM#2 (HIGH BYTE) IS LOADED INTO THE SYNC REGISTER.

BITS 8-15 SYNC CHARACTER, DEFAULT= 26 (26)

PARAM#3 IS NOT USED (177777)

NO1

DQ11 ITEP OVERLAY MACY11 27(1006) 29-OCT-76 14:38 PAGE 14
DZDQ08.P11 05-AUG-76 09:23

```

599
600
601
602
603 011000 011000
604 011000 050504 000040
605 011004 160010
606 011006 000300
607 011010 000740
608 011012 000300
609 011014 013300
610 011016 177570
611 011018 177570
612 011020 000000
613 011024 000000
614 011026 000300
615 011030 000300
616 011032 000300
617 011034 000300
618 011036 011106
619 011040
620 011040 000
621 011041
622 011041 001
623 011042 000000
624 011044 177570
625 011046 177570
626
627
628
629
630 000000
631 100000
632 040000
633 020000
634 020000
635
636 011050 000000
637 011052 000000
638 011054 000000
639 011056 000000
640 011060 000000
641
642 011062 000000
643 011064 000000
644 011066 000000
645 011070 000000
646 011072 000000
647 011074 000000
648
649 011076 177560
650 011100 177562
651 011102 177564
652 011104 177566
653
654 000001

```

```

*****
D011 INTERFACE SERVICE PARAMS
*****

```

```

D011:  =11000
      .ASCIZ  /D9 /
BA:    160010
RIV:   300
PRIOR: 240
PARAM1: 0
PARAM2: 013000
PARAM3: 17777
IPDA:  .WORD  0
IXDP:  .WORD  0
SETILE: .WORD  0
R2016: .WORD  0
TIME:  .WORD  0
TX.TERM: .WORD  START
      .BYTE  000
RX.TERM: .BYTE  001
FLAG:   .WORD  0
SWR:    177570
DISPLAY:177570

```

```

:ISR NAME
:BUS ADDRESS
:VECTOR ADDRESS
:PRIORITY
:PARAM #1
:PARAM #2
:PARAM #3
:INITIAL READ DATA ADDRESS
:INITIAL XMIT DATA ADDRESS
:LINE SETTLE DELAY FLAG
:ADDR OF BIN TO OCT TYPE ROUTINE
:TIMER
:ADDR OF START OF PROGRAM
:TRANSMITTER TERMINATING CHAR.
:RECEIVER TERMINATING CHAR.

```

```

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

```

```

STAT=RD
XFLG=100000
RFLG=40000
DSFLG=20000
BIT13=20000

```

```

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

```

```

SXCSR: 0
SRCR: 0
ERCER: 0
ERCR: 0
DSSTAT: 0

```

```

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

```

```

TXMC:0
RXMC:0
XCC: 0
RCC: 0
RDA: 0
XDA: 0

```

```

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

```

```

TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
FULL.DUPLEX=000001

```

```

655
656
657
659 011106 000240
659 011110 017700 177730
660 011114 042700 177400
661 011120 013702 011006
662 011124 012722 015012
663 011130 013722 011010
664 011134 012722 014170
665 011140 013722 011010
666 011144 013704 011004
667 011150 013714 011012
668 011154 013702 011014
669 011160 042702 000001
670 011164 010264 000002
671 011170 005014
672 011172 005064 000002
673 011176 005064 000004
674 011202 012703 000020
675 011206 052764 010000 000004 15:
676 011214 042764 050000 000004
677 011222 005064 000006
678 011226 105264 000005
679 011232 005303
680 011234 001364
681 011236 112764 000012 000005
682 011244 012764 000040 000006
683 011252 005037 014160
684 011256 052764 000010 000002
685 011264 005037 011032
686 011270 005037 013246
687
688
689
690
691
692
693
694
695 C11274
696 011274 013702 011022
697 011300 005003
698 011302 005203
699 011304 123722 011040
700 011310 001374
701 011312 005403
702 011314 010337 011062
703 011320 013702 011022
704 011324 005003
705 011326 005203
706 011330 123722 011041
707 011334 001374
708 011336 010302
709 011340 005403
710 011342 010337 011064

```

```

*****
D011-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, @RCSR ; SETUP VARIABLES
        MOV    @PARAM2, R2
        BIC    #0001, R2
        MOV    R2, @XCSR(R4) ; IN CSR'S
        CLR    @RCSR        ; CLEAR THE RX CSR
        CLR    @XCSR(R4)    ; CLEAR TX CSR
        CLR    @ERR(R4)     ; CLEAR THE ERROR REGISTER
        MOV    #16, R3      ; SET COUNTER
        BIS    @BIT12, @ERR(R4) ; SET WRITE ENABLE
        BIC    @#0000, @ERR(R4) ; CLEAR EXT MEM BITS
        CLR    @SEC(R4)     ; CLEAR THE SECONDARY REGISTER
        INCB   @REG(R4)     ; GET NEXT REGISTER
        DEC    R3           ; DONE YET?
        BNE   IS           ; KEEP CLEARING
        MOVB  #12, @REG(R4) ; SELECT THE MISC REGISTER
        MOV   @BIT5, @SEC(R4) ; ISSUE A MASTER CLEAR.
        CLR   @ERRORS
        BIS   @BIT3, @XCSR(R4) ; ENABLE ERROR INTERRUPTS
        CLR   @TIME        ; RESET TIMER
        CLR   @DELAY       ; RESET DELAY INDICATOR.

```

```

; ROUTINE TO FIGURE RX AND TX WORD COUNTS.
; FOR HIGH BAUD THE TX AND RX MESSAGES MUST BE IDENTICAL
; THE DATA WILL BE TRANSFERED AT A BURST MODE.
; THE TXWC AND RXWC SET FOR HOW MANY CHARS TO DEAL WITH.

```

```

X.X:
      MOV    IXDA, R2
      CLR    R3
      INC   R3
      IS:   INC   R3
           CMPB  TX. TERM, (R2)+
           BNE   IS
           NEG   R3
           MOV   R3, TXWC
           MOV   IXDA, R2
           CLR   R3
           IS:   INC   R3
           CMPB  RX. TERM, (R2)+
           BNE   IS
           MOV   R3, R2
           NEG   R3
           MOV   R3, RXWC

```

```

711
712
713
714
715
716
717
718
719
720 011346 005037 011032
721 011352 005037 013246
722 011356 005037 013252
723 011362 032700 000001
724 011366 001402
725 011370 000137 011544
726 011374 032700 000002
727 011400 001402
728 011402 000137 011436
729 011406 032700 000010
730 011412 001402
731 011414 000137 011642
732 011420 032700 000004
733 011424 001402
734 011426 000137 012072
735 011432 000000
736 011434 000776

```

```

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

```

```

GO: CLR TIME
CLR DELAY
CLR STOP
BIT @OWO,MODE
BEQ 1$
JMP SOWO
1$: BIT @WI,MODE
BEQ 2$
JMP SOWI
2$: BIT @ILB,MODE
BEQ 3$
JMP $ILB
3$: BIT @XLB,MODE
BEQ 4$
JMP $XLB
4$: HALT
BR .-2

```

```

737
738
739
740
741
742
743
744
745
746
747
748
749
750
751 011436 104416
752 011440 004737 014516
753 011444 032700 040000
754 011450 001013
755 011452 023727 011032 000100
756 011460 103771
757 011462 011402
758 011464 016403 000002
759 011470 104001
760 011472 005037 011032
761 011476 000762
762
763 011500 032777 000200 177336
764 011506 001002
765 011510 004737 012462
766 011514 042700 040000

```

```

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

```

```

SOWI: KBDIN
1$: JSR PC,STARTR
BIT @RFLG,STAT
BNE 2$
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

```

767	011529	032777	000020	177316		BIT	#LOOP,2SWR
768	011526	001405				BEQ	4\$
769	011530	012737	011542	013250		MOV	#4\$,BACK
770	011536	000137	012322			JMP	EOP
771	011542	000735			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."
*****

```

782	011544	104416			\$OWO:	KBIN	
783	011546	004737	013254			JSR	PC,STARTX
784	011552	005037	011032			CLR	TIME
785	011556	032700	100000		1\$:	BIT	#XFLG,STAT
786	011552	001013				BNE	2\$
787	011554	023727	011032	000100		CMP	TIME,#100
788	011572	103771				BLO	1\$
789	011574	011402				MOV	2RCSR,R2
790	011576	016403	000002			MOV	XCSR(R4),R3
791	011572	104001				HLT	1
792	011604	001037	011032			CLR	TIME
793	011610	000762				BR	1\$
794	011612	042700	100000		2\$:	BIC	#XFLG,STAT
795	011616	032777	000020	177220		BIT	#LOOP,2SWR
796	011624	001405				BEQ	3\$
797	011626	012737	011640	013250		MOV	#3\$,BACK
798	011634	000137	012322			JMP	EOP
799	011640	000741			3\$:	BR	\$OWO
800							
801							
802							

```

803
804
805
806
807
808
809
810
811
812
813
814 011642 104416
815 011644 004737 014516
816 011650 005037 011032
817 011654 032700 040000
818 011660 001013
819 011662 023727 011032 000100
820 011670 103771
821 011672 011402
822 011674 016403 000002
823 011700 104301
824 011702 005037 011032
825 011706 000762
826 011710 032777 000200 177126
827 011716 001002
828 011720 004737 012462
829 011724 042700 040000
830 011730 032777 000020 177106
831 011736 001405
832 011740 012737 011752 013250
833 011746 000137 012322
834 011752 032777 000400 177064
835 011760 001416
836 011762 013702 011020
837 011766 013703 011022
838 011772 010337 011074
839 011776 112223
840 011780 001376
841 011782 112743 000177
842 011786 005203
843 011790 112723 000177
844 011794 104223
845 011796 005037 011032
846 011802 004737 013254
847 011806 032700 100000
848 011810 001013
849 011814 023727 011032 000100
850 011818 103771
851 011822 011402
852 011826 016403 000002
853 011830 104301
854 011834 004737 011032
855 012060 000762
856 012062 042700 100000
857 012066 000137 011642

```

```

*****
ROUTINE USED IF INTERNAL LOOP BACK" WAS SELECTED.
NOTE THAT WHEN IN THIS MODE; HALF DUPLEX IS THE
ONLY MODE AVAILABLE.
"INTERNAL LOOP BACK" MEANS THAT THE RECEIVER IS "TURNED ON"
AND A COMPLETE MESSAGE IS RECEIVED. IF DATA IS TO BE CHECKED
IT IS: IF "END PASS" IS DESIRED; IT IS GIVEN.
THEN THE TRANSMITTER IS ENABLED. AFTER THE WHOLE MESSAGE
IS TRANSMITTED; THE CYCLE IS REPETED AS ABOVE.
*****

```

```

SILB: KBDIN
      JSR PC,STARTR
      CLR TIME
15:   BIT #AFLG,STAT
      BVE 25
      CMP TIME,#100
      BLO 15
      MOV @RCSR,R2
      MOV XCSR(A4),R3
      HLT 1
      CLR TIME
      BR 15
25:   BIT #MODAT,@SWR
      BNE 35
      JSR PC,TESTD
35:   BIC #AFLG,STAT
      BIT #LOOP,@SWR
      BEQ 45
      MOV #45,BACK
      JMP EOP
45:   BIT #400,@SWR
      BEQ 75
      MOV IRDA,R2
      MOV IXDA,R3
      MOV R3,XDA
      MOVB (R2)+,(R3)+
      BNE -2
      MOVB #177,-(R3)
      INC R3
      MOVB #177,(R3)+
      CLRB (R3)+
75:   CLR TIME
      JSR PC,STARTX
55:   BIT #XFLG,STAT
      BNE 65
      CMP TIME,#100
      BLO 55
      MOV @RCSR,R2
      MOV XCSR(A4),R3
      HLT 1
      CLR TIME
      BR 55
65:   BIC #XFLG,STAT
      JMP SILB

```

```

;USE EXTERNAL DATA?
;IF NO
;SET POINTER
;SET POINTER
;SETUP XMIT DATA ADDR
;MOVE INPUT TO OUTPUT
;LOOP IF NOT ZERO CHAR
;INSERT A FILL CHAR
;BUMP ADDRESS
;INSERT ANOTHER FILL
;INSERT ZERO CHAR

```

858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913

012072 104416
012074 032737 000001 011014
012102 001402
012104 004737 014516
012110 004737 013254
012114 005037 011032
012120 032700 100000
012124 001016
012126 032700 040000
012132 001024
012134 023727 011032 000100
012142 103766
012144 011402
012146 016403 000002
012156 104001
012154 005037 011032
012160 000757
012162 032737 000001 011014
012170 001356
012172 042700 100000
012176 004737 014516
012202 000746
012204 032737 000001 011014
012212 001420
012214 032700 100000
012220 001013
012222 023727 011032 000100
012230 103766
012232 011402
012234 016403 000002
012236 104001
012238 000757 011032
012240 001356
012242 042700 100000
012244 042700 040000
012246 005037 011032
012248 000777 000200 176552
012272 001002
012274 004737 012462
012300 032777 000200 176536
012306 001671
012310 012737 012072 013250
012316 000137 012322

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 PA 3 WILL
BE GIVEN IF DESIRED. THEN THE CYCLE IS REPEATED
AS ABOVE. IF RUNNING IN FULL DUPLEX THE PRO 41
WAITS FOR BOTH THE RECEIVER AND TRANSMITTER TO
FINISH THEN RESTARTS THE RECEIVER AND TRANSMITTER.

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

914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

012322 104414 000340
012323 016437 000002 012460
012324 042737 177737 012460
012325 042764 000040 000032
012326 012766 012410 000002
012327 010037 013232
012328 010137 013234
012329 010237 013236
012330 010337 013240
012331 010437 013242
012332 010537 013244
012333 000207

013700 013700 013232
013701 013701 013234
013702 013702 013236
013703 013703 013240
013704 013704 013242
013705 013705 013244
012737 177777 013246
053764 012460 000002
000177 000570
000000

013746 011056
001413
032777 020000 176346
001007
104400 012672
004077 176320
005746
104400 012753
013701 011022
013702 011020
122122
001777
123741 011040
001453
122746 000002
001005
010237 012554
104400

ROUTINE TO RETURN
TO MONITOR FOR
END PASS.

EOP: STPS,PRTY7 ;SET PS PRIORITY TO 7
MOV XCSR(R4),@TPIE ;SAVE TX CSA
BIC #1C<TIE>,@TPIE ;CLEAR ALL BUT TX IE,
BIC @TIE,XCSR(R4) ;CLEAR TX IE (EVEN IF IT WASN'T SET)
MOV #ENTER,2(SP) ;SET FOR RETURN IF SW 14=1
MOV R0,SAVR0 ;SAVE REGISTER 0
MOV R1,SAVR1 ;SAVE REGISTER 1
MOV R2,SAVR2 ;SAVE REGISTER 2
MOV R3,SAVR3 ;SAVE REGISTER 3
MOV R4,SAVR4 ;SAVE REGISTER 4
MOV R5,SAVR5 ;SAVE REGISTER 5
RTS PC ;RETURN TO CONTROL PROGRAM

ENTER: MOV SAVR0,R0 ;RESTORE R0
MOV SAVR1,R1 ;RESTORE R1
MOV SAVR2,R2 ;RESTORE R2
MOV SAVR3,R3 ;RESTORE R3
MOV SAVR4,R4 ;RESTORE R4
MOV SAVR5,R5 ;RESTORE R5
MOV #-1,DELAY ;IF ORIGINALLY SET; SET TX IE
BIS @TPIE,XCSR(R4)
JMP @BACK
@TPIE: 000000

SUBROUTINE TO CHECK
RECEIVER DATA.

TESTD: MOV ERORR,-(SP) ;WAS THERE A RECEIVE ERROR?
BEQ TSTDAT ;BR IF NO
BIT #BIT13,@SWR ;INHIBIT PRINTOUTS?
BNE TSTDAT ;BR IF YES
TYPE MSG0 ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
JSR RD,202016 ;PRINT CONTENTS OF RBUF
TST -(SP)
TYPE MSG1 ;<15><12>
TSTDAT: MOV TXDA,R1 ;SETUP XMIT DATA ADDR
MOV RCDA,R2 ;SETUP RCV DATA ADDR
SCANH: CHPB (R1), (R2)+ ;DATA OK?
BEQ SCANH ;BR IF OK
CHPB TX,TERM,-(R1) ;IS IT END OF DATA
BEQ TESTOX ;BR IF YES
CHPB @002,-(R2)
BNE ZS
MOV R2,18
TYPE

```

000000 0127 000000 15: WORD 0
000443 0128 000443 25: BR TESTDX
105712 0129 105712 TSTB (R2)
001441 0130 001441 BEQ TESTDX ;BR IF YES
000177 0131 000177 CMPB #177 (R1)+ ;IS IT FILL CHAR?
001756 0132 001756 BEQ SCAN4 ;BR IF YES
005301 0133 005301 DEC R1 ;BACKUP
122722 0134 122722 CMPB #177 (R2)+ ;IS IT FILL?
001756 0135 001756 BEQ SCAN4 ;BR IF YES
105743 0136 105743 TSTB -(R2) ;BACK UP POINTER
123722 0137 123722 CMPB PARAM2+1,(R2)+ ;BR IF CHAR WAS SYNC
000240 0138 000240 BEQ SCAN4 ;DATA ERROR
020000 0139 020000 176222 SCANS: NOP ;INHIBIT PRINTOUTS
032777 0140 032777 BIT #BIT13,SWR ;INHIBIT PRINTOUTS
001016 0141 001016 BNE DEPR ;BR IF YES
104400 0142 104400 TYPE MSG2 ;<15><12>RECEIVED DATA = <15><12>
011020 0143 011020 012640 MOV IRDA, R0AX ;SETUP DATA ADDRESS
104400 0144 104400 TYPE ;PRINT RECEIVED DATA
000000 0145 000000 R0AX: 0 ;RECEIVED DATA ADDR.
013003 0146 013003 012640 TYPE MSG3 ;<15><12>DATA SHOULD BE<15><12>
011022 0147 011022 012656 MOV IXDA, .+10 ;SETUP ADDR.
104400 0148 104400 TYPE ;PRINT GOOD DATA
011022 0149 011022 IXDA
111103 0150 111103 DEPR: MOVB (R1),R3 ;SETUP XMIT DATA
114232 0151 114232 MOVB -(R2),R2 ;SETUP RCV DATA
104007 0152 104007 HLT+7 ;DATA ERROR HALT
005726 0153 005726 TESTDX: TST (SP)+ ;POP STACK
000207 0154 000207 RTS PC ;RETURN FROM SUB/ROUT

1000 012672 005015 044124 051105 MSG0: .ASCIZ <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
(1) 012753 015 000012 MSG1: .ASCIZ <15><12>
(1) 012756 005015 042522 042503 MSG2: .ASCIZ <15><12>/RECEIVED DATA = /<15><12>
(1) 013013 015 042012 052101 MSG3: .ASCIZ <15><12>/DATA SHOULD BE/<15><12>
(1) 013016 005015 046120 040505 MSG4: .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
(1) 013015 015 053412 042510 .ASCIZ <15><12>/WHEN CONNECTION COMPLETE: HIT CONTINUE SWITCH./<15><12>
(1) 013160 005015 046120 040505 MSG5: .ASCIZ <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
(1) .EVEN
10001 013232 000000 SAVR0: 0
10002 013234 000000 SAVR1: 0
10003 013236 000000 SAVR2: 0
10004 013238 000000 SAVR3: 0
10005 013240 000000 SAVR4: 0
10006 013242 000000 SAVR5: 0
10007 013244 000000 DELAY: 0
10008 013246 000000 BACK: 0
10009 013248 000000 STOP: 0

```

```

1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065

```

```

:*****
:INITIALIZE TRANSMITTER.
:PURPOSE OF THIS ROUTINE IS TO SEND OUT FIVE
:SYNC CHARS. THE SYNC CHAR TO BE SENT IS FOUND
:IN LOCATION PARAM2+1 (HIGH BYTE OF PARAM2).
:WHEN THE SYNC CHARS. HAVE BEEN TRANSMITTED
:THE TRANSMITTER WILL BE ENABLED FOR REAL DATA
:TRANSFER. CHARACTERS WILL BE TRANSMITTED ONE AT A TIME.
:*****
STARTX: TST      DELAY      ;IF SW04=1 & SW14=0 WAIT BEFORE TURNING TX ON
        BEQ      NOLY      ;NO GO AHEAD AND TURN ON TX
        CLR      TEMP1
        MOV      #7,TEMP2
        ADD      #1,TEMP1
        BNE      .-6
        DEC      TEMP2
        BNE      .-14
        CLR      DELAY      ;ZERO DELAY
NOLY:   CLR      TIME
        BIC      #300,XCSR(R4) ;CLEAR BOTH DONE BITS FROM TX
        BIS      #BIT3,XCSR(R4) ;ENABLE ERRORS
        BIT      #BIT2,XCSR(R4) ;WHERE IS THE POINTER POINTING
        BEQ      IS        ;BR IF PRI IS NEXT.
        JSR      PC,GETPRI  ;GO TOGGLE TO GET PRIMARY.
        BR      STARTX     ;GOTO EGGING
        MOV      #5,R1      ;SET FOR FIVE SYNC CHARS.
        MOV      #SYNC,R5  ;SET POINTER LOCATION
        MOVB    PARAM2+1,(R5) ;LOAD IN SYNC CHARS.
        DEC     R1          ;ALL DONE??
        BNE     .-6        ;BRANCH IF NOT DONE
        CLRB    (R5)       ;SET LAST BYTE TO ZERO

;NOW THE SYNC CHARS ARE LOADED IN CORE FOR THE D011 TO
;PICK UP AND TRANSMIT OUT.

        MOVB    #12,REG(R4) ;SELECT MISC REGISTER
        MOV     #4000,SEC(R4) ;SET FOR EIGHT BITS PER CHAR.
        MOVB   #2,REG(R4)   ;SELECT THE TX BA PRI.
        MOV     #SYNC,SEC(R4) ;LOAD TX BA PRI. WITH SYNC ADDR.
        INCB   REG(R4)     ;SELECT THE TX MC PRI.
        MOV     #5,SEC(R4)  ;SET FOR FIVE CHARS.
        BIC    #XFLG,STAT   ;RESET TX COMPLETE FLAG
        BIS    #DTR,XCSR(R4) ;SET DATA TERMINAL READY
        TST    STOP        ;FIRST TIME HERE?
        EIE    16$        ;BR IF NO
        TYPE   #1,65       ;MAKE CONNECTION
        COM    STOP
        BIT    #MRDY,XCSR(R4) ;
        BNE    2$
        BIT    #MRDY,XCSR(R4) ;
        EIE    3$
        CMP    TIME,#36    ;HAVE 30 SEC ELAPSED YET
        BLO   8$          ;NO NOT YET
        MOV    #DTR,R2    ;LOAD FOR TIMEOUT
        MOV    XCSR(R4),R3 ;LOAD FOR TIMEOUT

```

```

1066 013526 032777 010000 175310 BIT #SW12,2SWR ;INHIBIT PRINTOUTS?
1067 013534 001001 BNE 12S ;BR IF YES
1068 013536 104002 HLT+2 ;TYPE "WAITING TO TRANSMIT" MESSAGE.
1069 013540 005037 011032 12S: CLR TIME
1070 013544 000755 BR 8S
1071 013546 005037 014154 3S: CLR TEMP1
1072 013552 012737 000005 014156 MOV #5,TEMP2
1073 013560 062737 000001 014154 ADD #1,TEMP1
1074 013566 001374 BNE -6
1075 013570 005337 014156 DEC TEMP2
1076 013574 001371 BNE .-14
1077 013576 032737 000001 011014 2S: BIT #FULL.DUPLEX,PARAM2
1078 013604 001023 BNE 9S
1079 013606 032764 010000 000002 10S: BIT #10000,XCSR(R4) ;IS CARRIER UP?
1080 013614 001417 BEQ 9S
1081 013616 023727 011032 000036 CMP TIME,#36 ;30 SECONDS UP ?
1082 013624 103770 BLO 10S ;NOT YET
1083 013626 011402 MOV @RCSR,R2 ;PREPARE TYPE OUT
1084 013630 016403 000002 MOV XCSR(R4),R3 ;AS ABOVE
1085 013634 032777 010000 175202 BIT #SW12,2SWR ;INHIBIT PRINTOUTS?
1086 013642 001001 BNE 13S ;BR IF YES
1087 013644 104001 HLT ;TYPE "WAITING "
1088 013646 005037 011032 13S: CLR TIME ;ZERO TIMER
1089 013652 000755 BR 10S
1090 013654 005037 011032 9S: CLR TIME ;SET TIME=0
1091 013660 052764 000400 000002 BIS #RQTS,XCSR(R4)
1092 013666 032764 020000 000002 11S: BIT #CTS,XCSR(R4)
1093 013674 001017 BNE 6S
1094 013676 023727 011032 000036 CMP TIME,#36 ;30 SECONDS UP ??
1095 013704 103770 BLO 11S ;NOT YET
1096 013706 011402 MOV @RCSR,R2 ;PREPARE TYPE OUT
1097 013710 016403 000002 MOV XCSR(R4),R3 ;AS ABOVE
1098 013714 032777 010000 175122 BIT #SW12,2SWR ;INHIBIT PRINTOUTS?
1099 013722 001001 BNE 14S ;BR IF YES
1100 013724 104002 HLT ;TYPE "WAITING "
1101 013726 005037 011032 14S: CLR TIME ;ZERO TIMER
1102 013732 000755 BR 11S
1103
1104
1105 ;HERE NOW BECAUSE CLEAR TO SEND HAS BEEN SET
1106 ;NOW SEND THE SYNC CHARS.
1107
1108 013734 052764 000001 000002 6S: BIS #GO,XCSR(R4) ;SET THE GO BIT
1109 013742 005037 011032 CLR TIME ;SET TIME TO ZERO
1110
1111 013746 105764 000002 4S: TSTB XCSR(R4) ;IS TX DONE WITH SYNCs
1112 013752 100417 BMI 5S ;BR IF YES
1113 013754 023727 011032 000036 CMP TIME,#36 ;HAVE 30 SECOND GONE BY??
1114 013762 103771 BLO 4S ;BR IF NOT YET
1115 013764 011402 MOV @RCSR,R2 ;LOAD FOR TYPE OUT
1116 013766 016403 000002 MOV XCSR(R4),R3 ;LOAD FOR TYPE OUT
1117 013772 032777 010000 175044 BIT #SW12,2SWR ;INHIBIT PRINTOUTS?
1118 014000 001001 BNE 15S ;BR IF YES
1119 014002 104002 HLT ;TYPE "WAITING..."
1120 014004 005037 011032 15S: CLR TIME ;ZERO TIMER
1121 014010 000755 BR 4S ;GO V IT FOR DONE

```

```

1122
1123
1124
1125
1126
1127 014012 112764 000002 000005 55:   MOVB   #2,REG(R4)   ;SELECT TX BA PRI
1128 014020 013764 011022 000006   MOV    IXDA,SEC(R4) ;LOAD BA
1129 014026 062764 000002 000006   ADD    #2,SEC(R4)   ;POINT BA TWO HIGHER
1130 014034 112764 000003 000005   MOVB   #3,REG(R4)   ;SELECT TX WC PRI
1131 014042 012764 177776 000006   MOV    #-2,SEC(R4)  ;SET TWO CHARS AT A TIME
1132 014050 112764 000006 000005   MOVB   #6,REG(R4)   ;SELECT TX BA SEC
1133 014056 013764 011022 000006   MOV    IXDA,SEC(R4) ;LOAD TX BA SEC WITH DATA POINTER
1134 014064 112764 000007 000005   MOVB   #7,REG(R4)   ;SELECT TX WC SEC
1135 014072 012764 177776 000006   MOV    #-2,SEC(R4)  ;SET WC WITH -2
1136
1137
1138
1139
1140 014100 032737 000002 011014   BIT    #HI.BAUD,PARAM2 ;HAS HI BAUD BEEN SELECTED?
1141 014106 001413   BEQ    7$           ;BR IF NO
1142 014110 112764 000003 000005   MOVB   #3,REG(R4)   ;SELECT THE TX WC PRI
1143 014116 005064 000006   CLR    SEC(R4)      ;ZERO THE PRI WC
1144 014122 112764 000007 000005   MOVB   #7,REG(R4)   ;SELECT THE TX WC SEC
1145 014130 013764 011062 000006   MOV    TXWC,SEC(R4) ;SET FOR MAX CHAR TRANSFER.
1146
1147
1148
1149
1150
1151
1152
1153
1154 014136 042764 000300 000002 7$:   BIC    #300,XCSR(R4) ;CLEAR ALL TX DONES
1155 014144 052764 000041 000002   BIS    #IE+.GO,XCSR(R4)
1156
1157 014152 000207   RTS    PC           ;SET INTERRUPT ENABLE AND GO
1158
1159
1160
1161
1162
1163
1164
1165
1166
; SYNC CHARACTER BUFFER AREA.
TEMP1: 0
TEMP2: 0
ERRORS: 0
SYNC:  .BYTE 26,26
      .BYTE 26,26
      .BYTE 26,26

```

M02

```

1167
1168
1169
1170
1171
1172
1173
1174
1175 014170 000240          1.5  XISR:  NOP                ;LOCATION SET FOR HALT INSTRUCTION IN DEBUGGING.
1176 014172 005764 000004  4.4  TST      ERR(R4)          ;IS THE DQ11 ERROR FLAG SET??
1177 014176 100012          2.6  BPL      2$              ;NOT SET GOOD SO FAR.
1178 014200 016403 000002  5.0  MOV     XCSR(R4),R3
1179 014204 011402          3.8  MOV     @RCSR,R2
1180 014206 104000          9.3  HLT     0
1181 014210 016437 000004 014160  6.4  MOV     ERR(R4),ERRORS ;DQ11 ERROR FLAG SET. EXAMINE REGISTERS FOR ERRO
1182 014216 042764 000377 000004  7.0  BIC     #377,ERR(R4)    ;SAVE DA ERROR REG
1183 014224 011402          3.8  2$:    MOV     @RCSR,R2    ;PREPARE FOR ERROR TYPEOUT
1184 014226 016403 000002          5.0  MOV     XCSR(R4),R3    ;PREPARE FOR ERROR TYPEOUT
1185 014232 032764 000300 000002  6.5  BIT     #300,XCSR(R4)  ;IS EITHER TX DONE SET??
1186 014240 001001          2.6  BNE     .+4            ;GOOD EITHER PRI OR SEC DONE IS SET
1187 014242 104000          9.3  HLT     0              ;REPORT ERROR. INTERRUPT WAS TO
1188                                     ;BE CAUSED BY TX DONE; AND TX IS NOT DONE.
1189 014244 032737 000002 011014  5.3  BIT     #HI.BAUD,PARAM2 ;IS HIGH BAUD RATE SELECTED?
1190 014252 001402          2.6  BEQ     1$
1191 014254 005746          4.4  TST     -(SP)          ;FAKE STACK
1192 014256 000474          2.6  BR      HB,CHK        ;XFER ALL DONE
1193 014260 032764 000004 000002  6.5  1$:    BIT     #BIT2,XCSR(R4) ;WHERE IS POINTER??
1194 014266 001424          2.6  BEQ     TX,SEC        ;SECONDARY IS NEXT.
1195 014270 112764 000002 000005  6.4  TX.PRI: MOVB    #2,REG(R4)  ;SELECT TX BA PRIMARY
1196 014276 016401 000006          5.0  MOV     SEC(R4),R1    ;GET NEXT ADDRESS TO TRANSMIT FROM
1197 014302 062764 000002 000006  6.4  ADD     #2,SEC(R4)    ;UPDATE CURRENT ADDRESS.
1198 014310 004737 014410  5.8  JSR     PC,TX,CK     ;GO CHECK FOR END CHAR.
1199
1200                                     ;IF I COME BACK FROM THE SUBROUTINE THAT MEANS THAT THE
1201                                     ;END CHAR WAS NOT FOUND AND THE TRANSMISSON GOES ON.
1202
1203 014314 112764 000003 000005  6.4  MOVB    #3,REG(R4)    ;SELECT TX WC SEC
1204 014322 012764 177776 000006  6.4  MOV     #-2,SEC(R4)   ;LOAD WITH A -2
1205 014330 042764 000200 000002  7.0  BIC     #BIT7,XCSR(R4) ;CLEAR PRI DONE FROM TX
1206 014336 000002          4.8  RTI
1207
1208                                     ;THE ABOVE ROUTINE SERVICED THE INTERRUPT IF THE
1209                                     ;PRIMARY REGISTERS CAUSED THE INTERRUPT.
1210
1211 014340 112764 000006 000005  6.4  TX.SEC: MOVB    #6,REG(R4) ;SELECT THE TX BA SECONDARY.
1212 014346 016401 000006          5.0  MOV     SEC(R4),R1    ;GET ADDRESS POINTER.
1213 014352 062764 000002 000006  6.4  ADD     #2,SEC(R4)    ;UPDATE CURRENT ADDRESS.
1214 014360 004737 014410  5.8  JSR     PC,TX,CK     ;GO CHECK THE LAST CHARACTER.
1215
1216                                     ;JUST LIKE ABOVE; IF I COME BACK FROM THE ABOVE SUBROUTIN
1217                                     ;THAT MEANS THE THE LAST CHAR HASN'T BEEN TRANSMITTED
1218                                     ;AND THAT TRANSMISSON SHOULD CONTINUE.
1219
1220 014364 112764 000007 000005  6.4  MOVB    #7,REG(R4)    ;SELECT THE TX WC SEC
1221 014372 012764 177776 000006  6.4  MOV     #-2,SEC(R4)   ;LOAD WITH A -2
1222 014400 042764 000100 000002  7.0  BIC     #BIT6,XCSR(R4) ;CLEAR SEC DONE
  
```


1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299

014516 000240
014520 042714 000300
014524 032714 000004
014530 001403
014532 004737 015366
014536 000767
014540 113737 011015 014162
014546 113737 011015 014163
014554 112764 000011 000005
014562 013764 014162 000006
014570 112764 000000 000005
014576 013764 011020 000006
014604 105264 000005
014610 012764 177776 000006
014616 112764 000004 000005
014624 013764 011020 000006
014632 062764 000002 000006
014640 105264 000005
014644 012764 177776 000006
014652 112764 000012 000005
014660 012764 004000 000006
014666 012737 000750 011070
014674 042700 040000
014700 005037 011054
014704 005037 011056
014710 032737 000002 011014
014716 001413
014720 112764 000001 000005
014726 013764 011064 000006
014734 112764 000005 000005
014742 000064 000006
014746 000064 001000 000002
014754 005737 013252
014760 001004
014762 104400 013150
014766 005137 013252
014772
014772 000240
014774 000240
014776 000240
015000 000240
015002 000240
015004 052714 000041
015010 000207

: INITIALIZE RECEIVER ROUTINE

STARTR: NOP ; LEAVE OPEN FOR TEST PURPOSES.
BIC #300, @RCSR ; CLEAR ALL DONES
BIT #BIT2, @RCSR ; CHECK FOR POINTER TO SEC.
BEQ #1 ; BR IF POINTING TO PRI
JSR PC, GETPRI ; POINT REGISTERS TO PRI
BR STARTR ; BEGIN AGAIN
15: MOV #PARAM2+1, SYNC ; GET SYNC CHAR
MOV #PARAM3+1, SYNC+1 ; GET SECOND SYNC CHAR
MOV #11, REG(R4) ; SELECT SYNC REGISTER
MOV SYNC, SEC(R4) ; LOAD SYNC REGISTER
MOV #0, REG(R4) ; SELECT RX BA PRI.
MOV IRDA, SEC(R4) ; LOAD RX BA FOR DATA
INCB REG(R4) ; GET MC REGISTER
MOV #2, SEC(R4) ; SET FOR ONE CHAR.
MOV #4, REG(R4) ; SELECT RX BA SEC.
MOV IRDA, SEC(R4) ; LOAD RX BA SEC.
ADD #2, SEC(R4) ; UPDATE DATA POINTER BY ONE
INCB REG(R4) ; GET MC REGISTER
MOV #2, SEC(R4) ; SET FOR TWO CHAR.
MOV #12, REG(R4) ; SELECT MISC REGISTER
MOV #4000, SEC(R4) ; SELECT EIGHT BITS PER CHAR.
MOV #750, ACC ; SET FOR MAX 750 CHARS TO BE RXED
BIC #PFLG, STAT ; RESET RX COMPLETE FLAG
CLR ERCSR
CLR ERDRA
25: BIT #HI.BAUD, PARAM2 ; IS HIGH BAUD RATE SELECTED?
BEQ #25 ; BR IF NO
MOV #1, REG(R4) ; SELECT THE RX MC PRI
MOV RXMC, SEC(R4) ; RECEIVE CHARS MAX.
MOV #5, REG(R4) ; SELECT THE RX MC SEC.
CLR SEC(R4) ; ZERO THE RX MC SECONDARY.
35: BIS #DTR, XCSR(R4) ; SET DATA TERMINAL READY.
TST STOP ; FIRST TIME HERE?
BNE #35
TYPE MSG5
COM STOP
35: NOP
NOP
NOP
NOP
NOP
BIS #IE+.GO, @RCSR ; SET INTERRUPT ENABLE AND GO
RTS PC ; GO TO MAINLINE.....

: THE ABOVE SHOULD HAVE PREPARED THE RECEIVER TO
: RECEIVE ALL THE CHARS NEEDED FOR TESTING.
: THE RECEIVER WILL INTERRUPT AFTER EACH CHAR IS RECEIVED.
: CHECKING FOR THE END CHARACTER.

1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355

015012 000240
015014 005764 000004
015020 100012
015022 011402
015024 016403 000002
015030 104000
015032 016437 000004 014160
015040 042764 000377 000004
015046 005337 011070
015052 001004
015054 000005
015056 104006
015060 000000
015062 000776
015064 011402
015066 016403 000002
015072 032714 000300
015076 001001
015100 104000
015102 032737 000002 011014
015110 011404
015112 032714 000041
015116 005746
015120 000507
015122 032714 000004
015126 001423
015130 112764 000000 000005
015136 016401 000006
015142 062764 000002 000006
015150 004737 015244

;RECEIVER INTERRUPT SERVICE ROUTINE.

;THIS ROUTINE WILL SERVICE THE PRIMARY AND SECONDARY
;REGISTERS WHEN THEY INTERRUPT.
;CHECKING FOR THE END CHAR IS PERFORMED.

RISR: NOP ;LEAVE SPACE FOR BEBUGGING
TST ERR(R4) ;CHECK THE D011 ERROR FLAG
BPL 25 ;BR IF ERROR FLAG NOT SET
MOV 2RCSR,R2
MOV XCSR(R4),R3
HLT 0 ;CHECK ERROR REGISTER FOR ERROR.
MOV ERR(R4),ERRORS
BIC #377,ERR(R4)
25: DEC RCC ;CHECK THE BUFFER SIZE
BNE 15 ;BR IF OK
RESET ;STOP THE SHOW
HLT+6 ;RECEIVER BUFFER FULL
HALT ;STOP EVERY THING
BR -2 ;DISABLE CONT SWITCH
15: MOV 2RCSR,R2 ;PREPARE FOR ERROR TIMEOUT
MOV XCSR(R4),R3 ;PREPARE FOR ERROR TIMEOUT
BIT #300,2RCSR ;IS EITHER DONE SET??
BNE +4 ;BR IF AT LEAST ONE DONE IS SET
HLT 0 ;NOBODY IS DONE. BUT AN INTERRUPT OCCURED.
BIT #HI.BAUD,PARAM2 ;HIGH BAUD??
BEQ 35 ;BR IF NO
BIC #IE+.GO,2RCSR ;CLEAR GO AND INTERRUPT ENABLE
TST -(SP) ;FAKE STACK
BR HBRCHK ;XFER ALL DONE
35: BIT #BIT2,2RCSR ;WHERE IS THE POINTER.
BEQ RX.SEC ;POINTING TO SECONDARY.
RX.PRI: MOVB #0,REG(R4) ;SELECT THE RX BA PRIMARY
MOV SEC(R4),R1 ;GET THE ADDRESS OF DATA.
ADD #2,SEC(R4) ;UPDATE CURRENT ADDRESS
JSR PC,RX.CK ;GO CHECK THE DATA.

;IF ? RETURN BACK FROM THE ABOVE SUBROUTINE THE END CHAR
;WAS NOT FOUND AND RECEIVING WILL CONTINUE.

MOVB #1,REG(R4) ;SELECT THE RX WC PRI.
MOV #-2,SEC(R4) ;LOAD WITH -2
BIC #BIT7,2RCSR ;CLEAR RX DONE.
RTI ;LEAVE HERE NOW.

;THE ABOVE ROUTINE WAS FOR THE PRIMARY REGISTER INTERRUPT.
;THE BELOW ROUTINE IS FOR THE SECONDARY REGISTERS.

RX.SEC: MOVB #4,REG(R4) ;SELECT THE RX BA SEC.
MOV SEC(R4),R1 ;GET DATA ADDRESS
ADD #2,SEC(R4) ;UPDATE CURRENT ADDRESS
JSR PC,RX.CK ;GO CHECK THE DATA

```

1356                                     ;IF I COME BACK THAT MEANS THAT THE END CHARACTER WASN'T
1357                                     ;FOUND AND I SHOULD CONTINUE RECEIVING.
1358
1359 015222 112764 000005 000005 6.4      MOVB    #5,REG(R4)      ;SELECT THE RX I/O SEC
1360 015230 012764 177776 000006 6.4      MOV     #2,SEC(R4)     ;LOAD WITH A -2
1361 015236 042714 000100 5.8      BIC     #BIT6,IRCSR    ;CLEAR RX DONE
1362 015242 000002 4.8      RTI                                     ;GO BACK TO MAINLINE ....
1363
1364                                     ;THE BELOW ROUTINE CHECKS THE DATA FOR END CHAR.
1365
1366 015244 142741 000200 7.0      RX.CK: BICB    #BIT7,-(R1)  ;CLEAR PARITY BIT.
1367 015250 142741 000200 7.0      BICB    #BIT7,-(R1)  ;CLEAR PARITY BIT.
1368 015254 123721 011041 4.7      CMPB    RX.TERM,(R1)+ ;WAS THE CHAR =0x11
1369 015260 001403 2.6      BEQ     RXISRDN      ;BR IF 001
1370 015262 123721 011041 4.7      CMPB    RX.TERM,(R1)+ ;001?
1371 015266 001012 2.6      BNE     NO.12       ;NO 001
1372
1373 015270 042714 000041 5.8      RXISRDN: BIC     #IE+.GO,IRCSR ;CLEAR GO AND INTERRUPT ENABLE
1374 015274 042714 000300 5.8      BIC     #300,IRCSR   ;CLEAR ALL RX DONE'S
1375 015300 052700 040000 3.8      BIS     #RFLG,STAT  ;SET RX COMPLETE FLAG
1376 015304 005037 011032 3.7      CLR     TIME        ;ZERO TIMER
1377 015310 005726 3.2      TST     (SP)+       ;POP SUBROUTINE POINTER.
1378 015312 000002 4.8      RTI                                     ;ALL DONE GO HOME.....
1379 015314 032777 000040 173522 7.7      NO.12: BIT     #BITS,IRSWR ;MONITOR RX DATA??
1380 015322 001405 2.6      BEQ     IS          ;DON'T MONITOR
1381 015324 105777 173552 5.6      TSTB   @TPS        ;TTY READY??
1382 015330 100002 2.6      BPL     IS          ;NOT READY GO ON WITH TEST
1383 015332 114177 173546 8.8      MOVB   -(R1),@TPB  ;PRINT CHARACTER
1384 015336 000207 3.5      IS:     RTS        PC ;GO TO RX ISR
1385 015340 032777 000040 173476 7.7      HBRCHK: BIT     #BITS,IRSWR ;MONITOR RX DATA?
1386 015346 001750 2.6      BEQ     RXISRDN    ;BR IF NO
1387 015350 105777 173526 5.6      TSTB   @TPS        ;TTY READY?
1388 015354 100345 2.6      BPL     RXISRDN    ;NO GO ON WITH TEST
1389 015356 112777 000122 173520 7.6      MOVB   #'R,@TPB   ;TYPE "R"
1390 015364 000741 2.6      BR     RXISRDN

```

E03

D011 ITEP OVERLAY MACY11 27(1006) 29-OCT-76 14:38 PAGE 31
 DZD008.P11 05-AUG-76 09:23 RECEIVER INTERRUPT SERVICE ROUTINE.

```

1391                                     ; ROUTINE TO GET TRANSMITTER OR RECEIVER ONTO
1392                                     ; THE PRIMARY REGISTERS IF THEY ARE ON THE SECONDARY.
1393
1394 015366 032764 000004 000002 GETPRI: BIT    #BIT2,XCSR(R4)  ;IS THE TX ON THE SEC??
1395 015374 001424                BEQ     15          ;BR IF NO
1396 015376 112764 000012 000005        MOVB   #12,REG(R4)  ;SELECT THE MISC REG
1397 015404 012764 004010 000006        MOV    #4010,SEC(R4) ;SET EIGHT BITS AND TEST LOOP FOR CLK
1398 015412 112764 000007 000005        MOVB   #7,REG(R4)   ;SELECT TX WC SEC
1399 015420 012764 177777 000006        MOV    #-1,SEC(R4)  ;TX ONE CHAR
1400 015426 052764 000001 000002        BIS    #.GO,XCSR(R4) ;SET GO
1401 015434 032764 000100 000002        BIT    #BIT6,XCSR(R4) ;HANG HERE FOR SEC DONE
1402 015442 001774                BEQ     -6          ;KEEP WAITING FOR TX DONE SEC.
1403 015444 000750                BR     GETPRI      ;GO RECHECK AND CKECK RX
1404
1405 015446 032714 000004                15:   BIT    #BIT2,@RCSR  ;IS RX ON SECONDARY REG
1406 015452 001430                BEQ     25          ;BR IF ON PRI.
1407 015454 112764 000012 000005        MOVB   #12,REG(R4)  ;SELECT MISC REGISTERS
1408 015462 012764 004010 000006        MOV    #4010,SEC(R4) ;EIGHT BITS AND TEST LOOP (CLK)
1409 015470 112764 000005 000005        MOVB   #5,REG(R4)   ;SELECT RX WC SEC
1410 015476 012764 177777 000006        MOV    #-1,SEC(R4)  ;RX ONE CHAR
1411 015504 112764 000004 000005        MOVB   #4,REG(R4)   ;SEL RX BA SEC.
1412 015512 012764 015564 000006        MOV    #NO.DAT,SEC(R4) ;LOAD RX BA SEC.
1413 015520 052714 010001                BIS    #BIT12+.GO,@RCSR ;SET GO!!!!!!+ACTIVE
1414 015524 032714 000100                BIT    #BIT6,@RCSR  ;HANG HERE FOR RX SEC DONE.
1415 015530 001775                BEQ     -4          ;KEEP WAITING
1416 015532 000715                BR     GETPRI      ;GO CHECK EVERY ONE
1417
1418 015534 112764 000012 000005        25:   MOVB   #12,REG(R4)  ;SELECT MISC REGISTER
1419 015542 042764 004010 000006        BIC    #4010,SEC(R4) ;CLEAR EIGHT BITS AND TEST LOOP
1420 015550 042714 000300                BIC    #300,@RCSR   ;CLEAR RX DONES
1421 015554 042764 000300 000002        BIC    #300,XCSR(R4) ;CLEAR TX DONES
1422 015562 000207                RTS     PC          ;GO HOME
1423 015564 000000                NO.DAT: 0
1424
1425 015566 005015 042522 042503 MFULL:  .ASCIZ <15><12>/RECEIVER BUFFER FILLED. ERROR!!/
1426 015630 005015 042522 042503 MRXDA:  .ASCIZ <15><12>/RECEIVED DATA = /<15><12>
1427 015655 015 051012 041505 MRXSB:  .ASCIZ <15><12>/RECEIVED DATA SHOULD BE /<15><12>

```

F03

DQ11 ITP OVERLAY MACY11 27(1006) 29-OCT-76 14:38 PAGE 32
DZD008.P11 05-AUG-76 09:23 RECEIVER INTERRUPT SERVICE ROUTINE.

1425

000001

.EVEN
.END

STPS = 104414	599#	921													
SWR 011044	624#	659	763	767	795	826	830	834	907	910	954	984	1066		
	1005	1038	1117	1228	1234	1379	1385								
SM12 = 010000	599#	1066	1085	1098	1117										
SXCSR 011050	636#														
SYNC 014162	1037	1049	1164#	1256*	1257*	1259									
TEMP1 014154	1022*	1024*	1071*	1073*	1161#										
TEMP2 014156	1023*	1026*	1072*	1075*	1162#										
TESTD 012452	765	828	909	952#											
TESTDX 012566	965	971	974	997#											
TIE = 000040	599#	923	924												
TIME 011032	616#	685*	720*	755	760*	784*	787	792*	816*	819	824*	845*	849		
	854*	876*	881	886*	897	902*	906*	1029*	1062	1069*	1081	1088*	1090*		
	1094	1101*	1109*	1113	1120*	1242*	1376*								
TKB 011100	650#														
TKS 011076	649#														
TFB 011104	652#	1232*	1238*	1383*	1389*										
TPS 011102	651#	1230	1236	1381	1387										
TSTDAT 012516	953	955	960#												
TXOUT 014514	1244#														
TXWC 011062	642#	702*	1145												
TX.CK 014410	1198	1214	1224#												
TX.PRI 014270	1195#														
TX.SFC 014340	1194	1211#													
TX.TER 011040	619#	699	964	1224	1226										
TYPE = 104400	599#	956	959	969	966	988	990	992	1056	1284					
XCC 011056	644#														
XCSR = 000002	599#	670*	672*	684*	758	790	822	852	884	900	922	924*	942*		
	1030*	1031*	1032	1053*	1058	1060	1065	1079	1034	1091*	1092	1097	1108*		
	1111	1116	1154*	1155*	1178	1184	1185	1193	1205*	1222*	1240*	1281*	1313		
	1324	1394	1400*	1401	1421*										
XDA 011074	647#	838*													
XFLG = 100000	631#	785	794	847	856	877	890	895	904	1052	1239				
XIFR 014170	664	1175#													
XIS.ON 014474	1225	1227	1235	1237	1239#										
XLB = 000004	599#	732													
XWAIT = 104412	599#														
X.X 011274	695#														
\$ILB 011642	731	814#	857												
\$JMI 011436	728	751#	771												
\$MO 011544	725	722#	799												
\$XLB 012072	734	871#	911	912											
. = 015712	603#	736	840	991*	1025	1027	1040	1074	1076	1186	1322	1326	1402		
	1415														
.GO = 000001	599#	1108	1155	1240	1292	1330	1373	1400	1413						

BOX	10	600	627	655															
D00P1	10																		
D00J1	10																		
D00M1	10																		
D00N1	10																		
D00OC1	10	560																	
D00PAR1	10	581																	
D00PAR2	10																		
D00PAR3	10																		
D00OC1	10																		
D00PAR1	10																		
D00PAR2	10																		
HELLO	10																		
HLT	599 1320	759 1327	791	823	853	885	901	996	1068	1087	1100	1119	1180	1187	1314				
SEQUAT	10	599																	
SINTF	10	599																	
SITEP	10	713																	
SSERV	10	644																	

. ABS. 015712 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZDQ08.SEQ/SOL/CRF/NL:TOC=ITEP1.MAC,DZDQ08.P11
RUN-TIME: 11 16 1 SECONDS
RUN-TIME RATIO: 65/29=2.2
CORE USED: 16K (31 PAGES)

