DQ11

OVERLAY FOR ITEP

MD-11-DZDQO-B

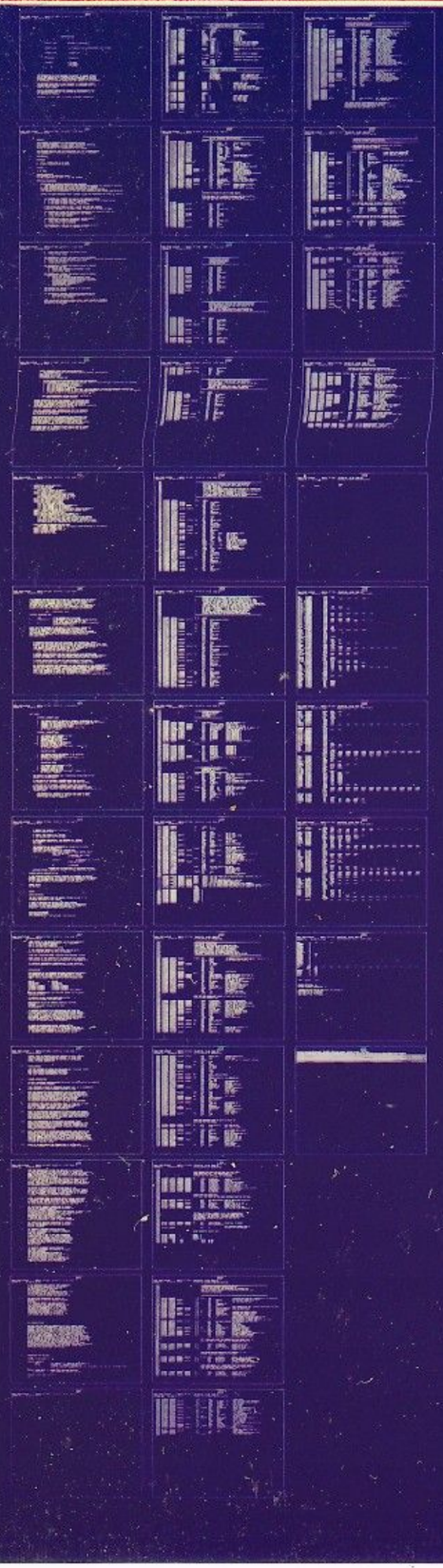
EP-DZDQO-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.



IDENTIFICATION

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

PRODUCT NAME: DQ11 OVERLAY FOR INTERPROCESSOR TEST PROGRAM

PROGRAM DATE: OCTOBER 1976

MAINTAINER: DIAGNOSTICS

AUTHORS: R A JONES
JOHN EGOLF
FAY BRSHAW

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 DM11BB, TYPE IN DMB.

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

2. THE PROGRAM WILL TYPE THE DEFAULT BUS ADDRESS OF THE INTERFACE UNDER TEST.
A. TYPE A CAR. RETURN TO USE DEFAULT BUS ADDRESS
B. TYPEIN ACTUAL BUS ADDRESS

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

4. THE PROGRAM WILL TYPE OUT THE DEFAULT INTERFACE PRIORITY
NOTE. 200=PRIO 4, 240=PRIO 5, 300=PRIO 6, ETC.

A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. TYPEIN ACTUAL VALUE

5. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#1
IF REQUIRED BY THE ISR.(SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. TYPEIN ACTUAL VALUE

6. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#2
IF REQUIRED BY THE ISR.
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
B. ENTER ACTUAL VALUE

7. THE PROGRAM WILL TYPEOUT THE DEFAULT VALUE OF PARAM#3
IF REQUIRED BY THE OVERLAY.
A. TYPE A CAR. RETURN TO USE DEFAULT VALUE
THE DN-11 WILL USE PARAM #3 AS THE # TO DIAL.
IF USING A MODEM WITHOUT AUTOMATIC HANDSHAKING,
THE NUMBER MUST TERMINATE WITH A
"END-OF-NUMBER" CHARACTER (:).
B. ENTER ACTUAL VALUE.

8. THE PROGRAM WILL RETURN TO STEP B1 IF THIS SETUP
WAS FOR DN11 OR DM11BB.

9. THE PROGRAM WILL REQUEST THAT SWITCH REGISTER BE SET.
A. SETUP SWITCH REGISTER AS SPECIFIED IN STEP D.
AND TYPE A CAR. RETURN.

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

C. MANUAL PARAMETER INPUT FROM SWITCH REGISTER
   1. THE PROGRAM HALTS FOR ISR(INTERFACE SERVICE ROUTINE) SPECIFICATION
   SWR14=SETUP DM-11B ISR
   SWR13=SETUP DM-11 ISR
   SWR=000000=SETUP VARIABLE ISR

   2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
      SETUP SEQUENCE IS: DN11,DM11-BB THEN VARIABLE OVERLAY. (EACH ENTRY SET SWICHES THEN HIT CONTINUE.)
      A. HALT FOR BUS ADDRESS OF INTERFACE
      B. HALT FOR VECTOR ADDRESS OF INTERFACE
      C. HALT FOR PRIORITY OF INTERFACE
      D. HALT FOR INTERFACE PARAM #1 (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
      E. HALT FOR INTERFACE PARAM #2 (DN11 AND DMBB PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.
      F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DMB.

   3. HALT FOR OPERATIONAL SWITCH SETTINGS. (SEE STEP D.)
      A. PRESS CONTINUE TO START TESTING

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

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

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

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

D. OPERATIONAL SWITCH SETTINGS.

SW15=1 HALT ON ERROR
SW14=1 SINGLE PASS
    SW14 HAS NO EFFECT IF SW04=0
SW13=1 INHIBIT ERROR TYPEOUTS
SW12=1 INHIBIT ALL TYPEOUTS EXCEPT ERRORS
    IF SW12-0 AND SW04=1 END PASS IS TYPED
    AND TRANSMITTED/RECEIVED DATA IS TYPED.
SW11=1 USE PREVIOUSLY SPECIFIED DATA
SW10=1 DATA SELECT (WITH SW09)
SW09=1 DATA SELECT (WITH SW10)
    00=1 GET DATA FROM OPERATOR
    01=1 TEST MESSAGE #1 (SA QUICK BROWN FOX)
    10=1 TEST MESSAGE #2 (SB NUMERICS)
    11=1 TEST MESSAGE #3 (SC COMTEST/QUICK BROWN FOX/NUMERICS)
SW08=1 TRANSMIT RECEIVED DATA (INTERNAL LOOPBACK MODE)
SW07=1 DO NOT TEST RECEIVED DATA
SW06=1 MONITOR TRANSMITTED DATA ON CONSOLE TTY.*
SW05=1 MONITOR RECEIVED DATA ON CONSOLE TTY.*
    * IN MANY CASES, NOT ALL DATA WILL APPEAR ON THE CONSOLE
    TTY. THIS IS ESPECIALLY TRUE WHEN THE COMM INTERFACE IS
    RUNNING AT A FASTER BAUD THAN THE CONSOLE, BUT EVEN AT EQUAL
    OR SLOWER BAUDS, ALL CHARACTERS MAY NOT APPEAR ON THE CONSOLE.

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

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

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

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

IF OPERATOR SPECIFIED DATA WAS INDICATED, THE PROGRAM WILL TYPE A
REQUEST FOR THE DATA. DATA MAY BE ENTERED AS ASCII CHARACTERS OR OCTAL CODE.
TYPE IN THE DATA TERMINATED WITH A CR. OCTAL CODE MAY BE ENTERED BY TYPING AN
↑(UP ARROW) FOLLOWED BY THE OCTAL CODE (IN THE RANGE 000 TO 377)
SEPERATED BY SPACES AND TERMINATED BY ↑(UP ARROW).
  I.E. ABCD↑ (000 123 377↑ EFG (CAR.RETURN)

A TYPICAL SWITCH SETTING FOR HALF-DUPLEX=003150 THIS SETTING USES
INTERNAL LOOPBACK MODE, LOOPS IN OVERLAY, MONITORS TRANSMITTED AND RECEIVED
DATA ON THE CONSOLE TTY, AND TESTS RECEIVED DATA USING TEST MESSAGE #3.

A TYPICAL SWITCH SETTING FOR FULL-DUPLEX=003144 THIS SETTING
IS THE SAME AS ABOVE EXCEPT IT USES THE EXTERNAL LOOPBACK MODE.

ALL STANDARD MESSAGES(TEST MESSAGES 1-3) ARE PRECEDED BY 2 FILL CHARACTERS(177),
AND ARE FOLLOWED BY A CR(015), LF(012), RECEIVE TERMINATING CHARACTER(001),
4 FILLS(177), AND A TRANSMIT TERMINATING CHARACTER(000). DURING TRANSMISSION,
WHEN A 000 CHARACTER IS SEEN THE TRANSMISSION IS STOPPED.DURING RECEPTION,
WHEN A 001 CHARACTER IS RECEIVED, THE RECEIVER IS SHUT OFF.
IF THE MESSAGE WAS INPUTED BY THE OPERATER, THE TERMINATING CHARACTERS ARE ADDED.

TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY <DO1>)
2. VERIFIES THE DATA AGAINST THE DATA SELECTED BY SW09 AND SW10(SW7=0)
3. TRANSMIT THE DATA SELECTED BY SW09 AND SW10 (SW8=0) OR
   TRANSMIT THE RECEIVED DATA (SW8=1)
4. RETURNS TO MONITOR FOR "END PASS" (SW4=1) OR
   GO TO STEP 1. (SW4=0)

EXTERNAL LOOPBACK MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAIT FOR CLEAR TO SEND
3. TRANSMITS THE SELECTED DATA
4. RESETS REQUEST TO SEND
5. WAIT FOR MESSAGE TO BE RECEIVED
6. VERIFIES THE DATA (SW7=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR
   GO TO STEP 1(SW04=0)

ONE-WAY-IN MODE

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

ONE-WAY-OUT MODE

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

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

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

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

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

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

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

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

TEST EXECUTION MAY BE INTERRUPTED BY TYPING THE FOLLOWING
CHARACTERS ON THE CONSOLE TTY.
LINE FEED = RESTART PROGRAM AT LOCATION 200.
QUESTION MARK = PRINTOUT FIRST 8 WORDS OF INPUT BUFFER.(ASCII)
THEN TYPE EITHER:

#Axxxxxx	TO PRINTOUT THE 8 WORDS AT LOC xxxxxx.
#Bxxxxxx	TO PRINTOUT THE 16 BYTES AFTER LOC xxxxxx.
#C	TO CONTINUE

PROGRAM MUST BE RESTARTED AT 200 AFTER PRINTING.
CARRIAGE RETURN = RESTART AT REQUEST FOR NEW OPERATIONAL SWITCHES.

5.0 PROGRAM AND/OR OPERATOR ACTION

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

5.1 NORMAL HALTS

SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

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

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

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

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

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

THERE WAS A RECEIVER ERROR. RECEIVER DATA REGISTER =XXXXXX

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

IF A RECEIVE TERMINATING CHARACTER(001) IS NOT DETECTED
WITHIN 512 CHARACTERS A "BUFFER FULL" PRINTOUT WILL OCCUR.

7.0 RESTRICTIONS

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

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

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTION APPLY:

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

BOTH SYSTEMS SHOULD BE RUNNING IDENTICAL ROUTINES.
EXAMPLE:
SWITCHES 14,13,7,4 SHOULD BE THE SAME
ON BOTH CPU'S

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

IF USING AN ASYNCRONOUS DEVICE, MODEMS AND THE
MAYNARD TEST STATION AND INITALIZE DOES NOT CLEAR THE
CONNECTION (EXAMPLE THE DJ11) IF THE PROGRAM IS RESTARTED
IN THE MIDDLE OF A MESSAGE AT LOC 204 OR BY HITTING CR
AN IMMEDEATE ERROR MESSAGE FROM MAYNARD WILL BE RE-

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

8.0 MISCELLANEOUS

ITEP HAS 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 DQ11 INTERFACE SERVICE PARAMS ARE SETUP, AS SPECIFIED BY THE OPERATOR,
BY THE ITEP CONTROL PROGRAM.

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

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

9.3 THE OVERLAY THEN LOOPS IN ROUTINES: $OWI ,IF "ONE WAY IN" MODE
WAS SELECTED. $OWO,IF "ONE WAY OUT" MODE WAS SELECTED.
$ILB, IF "INTERNAL LOOP BACK" MODE WAS SELECTED.
$XLB,IF "EXTERNAL LOOP BACK" WAS SELECTED.

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

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

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

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

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

9.4 THE RETURN TO MONITOR ROUTINE FOR END PASS AT EOP:
LOCKS OUT INTERUPTS AND SAVES THE TRANSMITTER 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 FULL DUPLEX (1), DEFAULT= HALF DUPLEX (0)
BIT 1 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)

```
                                ;*****************************************************************
                                ;       DQ11 INTERFACE SERVICE PARAMS
                                ;*****************************************************************
                 011000                 .=11000
011000  050504  000040         DQ11:   .ASCIZ  /DQ /           ;ISR NAME
011004  160010                 BA:     160010                  ;BUS ADDRESS
011006  000300                 RIV:    300                     ;VECTOR ADDRESS
011010  000240                 PRIOR:  240                     ;PRIORITY
011012  000000                 PARAM1: 0                       ;PARAM #1
011014  013000                 PARAM2: 013000                  ;PARAM #2
011016  177777                 PARAM3: 177777                  ;PARAM #3
011020  000000                 IRDA:   .WORD   0               ;INITIAL READ DATA ADDRESS
011022  000000                 IXDA:   .WORD   0               ;INITIAL XMIT DATA ADDRESS
011024  000000                 SETTLE: .WORD   0               ;LINE SETTLE DELAY FLAG
011026  000000                         .WORD   0               ;
011030  000000                 R2016:  .WORD   0               ;ADDR OF BIN TO OCT TYPE ROUTINE
011032  000000                 TIME:   .WORD   0               ;TIMER
011034  000000                         .WORD   0               ;
011036  011106                         .WORD   START           ;ADDR OF START OF PROGRAM
011040                         TX.TERM:
011040  000                            .BYTE   000             ;TRANSMITTER TERMINATING CHAR.
011041                         RX.TERM:
011041  001                            .BYTE   001             ;RECEIVER TERMINATING CHAR.
011042  000000                 FLAG:   .WORD   0
011044  177570                 SWR:    177570
011046  177570                 DISPLAY:177570

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

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

011062  000000         TXWC:0
011064  000000         RXWC:0
011066  000000         XCC:    0                       ;XMIT CHAR COUNT
011070  000000         RCC:    0                       ;RCV CHAR COUNT
011072  000000         RDA:    0                       ;RCV DATA ADDR.
011074  000000         XDA:    0                       ;XMIT DATA ADDR.

011076  177560         TKS:    177560
011100  177562         TKB:    177562
011102  177564         TPS:    177564
011104  177566         TPB:    177566

                 000001         FULL.DUPLEX=000001
```

```
                                        ;*****************************************************************
                                        ;       DQ11-X INTERFACE SERVICE ROUTINE
                                        ;*****************************************************************
011106  000240                  START:  NOP
011110  017700  177730                  MOV     @SWR,   R0      ;SETUP MODE IN R0
011114  042700  177400                  BIC     #177400,R0      ;STRIP JUNK
011120  013702  011006                  MOV     RIV,    R2      ;SETUP
011124  012722  015012                  MOV     #RISR,  (R2)+   ;INTERRUPT
011130  013722  011010                  MOV     PRIOR,  (R2)+   ;VECTORS
011134  012722  014170                  MOV     #XISR,  (R2)+   ;
011140  013722  011010                  MOV     PRIOR,  (R2)+   ;
011144  013704  011004                  MOV     BA,     R4      ;SETUP BUS ADDR INDEX
011150  013714  011012                  MOV     PARAM1, @RCSR   ;SETUP VARIABLES
011154  013702  011014                  MOV     PARAM2,R2       ;
011160  042702  000001                  BIC     #0001,R2        ;
011164  010264  000002                  MOV     R2,     XCSR(R4);IN CSR'S
011170  005014                          CLR     @RCSR           ;CLEAR THE RX CSR
011172  005064  000002                  CLR     XCSR(R4)        ;CLEAR TX CSR
011176  005064  000004                  CLR     ERR(R4)         ;CLEAR THE ERROR REGISTER
011202  012703  000020                  MOV     #16.,R3         ;SET COUNTER
011206  052764  010000  000004  1$:     BIS     #BIT12,ERR(R4)  ;SET WRITE ENABLE
011214  042764  060000  000004          BIC     #60000,ERR(R4)  ;CLEAR EXT MEM BITS
011222  005064  000006                  CLR     SEC(R4)         ;CLEAR THE SECONDARY REGISTER
011226  105264  000005                  INCB    REG(R4)         ;GET NEXT REGISTER
011232  005303                          DEC     R3              ;DONE YET??
011234  001364                          BNE     1$              ;KEEP CLEARING
011236  112764  000012  000005          MOVB    #12,REG(R4)     ;SELECT THE MISC REGISTER
011244  012764  000040  000006          MOV     #BIT5,SEC(R4)   ;ISSUE A MASTER CLEAR.
011252  005037  014160                  CLR     ERRORS
011256  052764  000010  000002          BIS     #BIT3,XCSR(R4)  ;ENABLE ERROR INTERUPTS
011264  005037  011032                  CLR     TIME            ;RESET TIMER
011270  005037  013246                  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.
                                        ;
011274                          X.X:
011274  013702  011022                          MOV     IXDA,R2
011300  005003                                  CLR     R3
011302  005203                  1$:             INC     R3
011304  123722  011040                          CMPB    TX.TERM,(R2)+
011310  001374                                  BNE     1$
011312  005403                                  NEG     R3
011314  010337  011062                          MOV     R3,TXWC
011320  013702  011022                          MOV     IXDA,R2
011324  005003                                  CLR     R3
011326  005203                  2$:             INC     R3
011330  123722  011041                          CMPB    RX.TERM,(R2)+
011334  001374                                  BNE     2$
011336  010302                                  MOV     R3,R2
011340  005403                                  NEG     R3
011342  010337  011064                          MOV     R3,RXWC
```

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

011346  005037  011032          GO:     CLR     TIME
011352  005037  013246                  CLR     DELAY
011356  005037  013252                  CLR     STOP
011362  032700  000001                  BIT     #OWO,MODE
011366  001402                          BEQ     1$
011370  000137  011544                  JMP     $OWO
011374  032700  000002          1$:     BIT     #WI,MODE
011400  001402                          BEQ     2$
011402  000137  011436                  JMP     $OWI
011406  032700  000010          2$:     BIT     #ILB,MODE
011412  001402                          BEQ     3$
011414  000137  011642                  JMP     $ILB
011420  032700  000004          3$:     BIT     #XLB,MODE
011424  001402                          BEQ     4$
011426  000137  012072                  JMP     $XLB
011432  000000          4$:     HALT
011434  000776                          BR      .-2


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


011436  104416          $OWI:   KBDIN
011440  004737  014516                  JSR     PC,STARTR
011444  032700  040000          1$:     BIT     #RFLG,STAT
011450  001013                          BNE     2$
011452  023727  011032  000100          CMP     TIME,#100
011460  103771                          BLO     1$
011462  011402                          MOV     @RCSR,R2
011464  016403  000002                  MOV     XCSR(R4),R3
011470  104001                          HLT     1
011472  005037  011032                  CLR     TIME
011476  000762                          BR      1$

011500  032777  000200  177336  2$:     BIT     #NODAT,@SWR
011506  001002                          BNE     3$
011510  004737  012462                  JSR     PC,TESTD
011514  042700  040000          3$:     BIC     #RFLG,STAT
```

```
   767  011520  032777  000020  177316          BIT     #LOOP,@SWR
   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
   772
   773
   774                                  ;*************************************************************************
   775                                  ;       ROUTINE USED IF "ONE WAY OUT" WAS SELECTED.
   776                                  ;       NOTE THAT WHEN IN THIS MODE HALF DUPLEX IS THE ONLY
   777                                  ;       MODE AVAILABLE.
   778                                  ;       "ONE WAY OUT" MEANS THAT ONLY THE TRANSMITTER IS
   779                                  ;       ENABLED. THE RECEIVER IS NEVER "TURNED ON.
   780                                  ;*************************************************************************
   781
   782  011544  104416                  $OWO:   KBDIN
   783  011546  004737  013254                  JSR     PC,STARTX
   784  011552  005037  011032                  CLR     TIME
   785  011556  032700  100000          1$:     BIT     #XFLG,STAT
   786  011562  001013                          BNE     2$
   787  011564  023727  011032  000100          CMP     TIME,#100
   788  011572  103771                          BLO     1$
   789  011574  011402                          MOV     @RCSR,R2
   790  011576  016403  000002                  MOV     XCSR(R4),R3
   791  011602  104001                          HLT     1
   792  011604  005037  011032                  CLR     TIME
   793  011610  000762                          BR      1$
   794  011612  042700  100000          2$:     BIC     #XFLG,STAT
   795  011616  032777  000020  177220          BIT     #LOOP,@SWR
   796  011624  001405                          BEQ     3$
   797  011626  012737  011640  013250          MOV     #3$,BACK
   798  011634  000137  012322                  JMP     EOP
   799  011640  000741                  3$:     BR      $OWO
   800
   801
   802
```

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

011642  104416                  $ILB:   KBDIN
011644  004737  014516                  JSR     PC,STARTR
011650  005037  011032                  CLR     TIME
011654  032700  040000          1$:     BIT     #RFLG,STAT
011660  001013                          BNE     2$
011662  023727  011032  000100          CMP     TIME,#100
011670  103771                          BLO     1$
011672  011402                          MOV     @RCSR,R2
011674  016403  000002                  MOV     XCSR(R4),R3
011700  104001                          HLT     1
011702  005037  011032                  CLR     TIME
011706  000742                          BR      1$
011710  032777  000200  177126  2$:     BIT     #NODAT,@SWR
011716  001002                          BNE     3$
011720  004737  012462                  JSR     PC,TESTD
011724  042700  040000          3$:     BIC     #RFLG,STAT
011730  032777  000020  177106          BIT     #LOOP,@SWR
011736  001405                          BEQ     4$
011740  012737  011752  013250          MOV     #4$,BACK
011746  000137  012322                  JMP     EOP
011752  032777  000400  177064  4$:     BIT     #400,   @SWR    ;USE EXTERNAL DATA?
011760  001416                          BEQ     7$              ;[R IF NO
011762  013702  011020                  MOV     IRDA,   R2      ;SET POINTER
011766  013703  011022                  MOV     IXDA,   R3      ;SET POINTER
011772  010337  011074                  MOV     R3,     XDA     ;SETUP XMIT DATA ADDR
011776  112223                          MOVB    (R2)+,  (R3)+   ;MOVE INPUT TO OUTPUT
012000  001376                          BNE     .-2             ;LOOP IF NOT ZERO CHAR
012002  112743  000177                  MOVB    #177,   -(R3)   ;INSERT A FILL CHAR
012006  005203                          INC     R3              ;BUMP ADDRESS
012010  112723  000177                  MOVB    #177,   (R3)+   ;INSERT ANOTHER FILL
012014  105023                          CLRB    (R3)+           ;INSERT ZERO CHAR
012016  005037  011032          7$:     CLR     TIME
012022  004737  013254                  JSR     PC,STARTX
012026  032700  100000          5$:     BIT     #XFLG,STAT
012032  001013                          BNE     6$
012034  023727  011032  000100          CMP     TIME,#100
012042  103771                          BLO     5$
012044  011402                          MOV     @RCSR,R2
012046  016403  000002                  MOV     XCSR(R4),R3
012052  104001                          HLT     1
012054  005037  011032                  CLR     TIME
012060  000762                          BR      5$
012062  042700  100000          6$:     BIC     #XFLG,STAT
012066  000137  011642                  JMP     $ILB
```

```
                                ;*****************************************************************
                                ;       ROUTINE USED IF "EXTERNAL LOOP BACK" WAS SELECTED.
                                ;       EITHER HALF OR FULL DUPLEX MAY BE SELECTED IN THIS MODE.
                                ;       "EXTERNAL LOOP BACK" MEANS THAT THE TRANSMITTER IS FIRST
                                ;       TURNED ON (IF HALF DUPLEX) AND THE WHOLE MESSAGE IS TRANSMITTED;
                                ;       THEN THE RECEIVER IS ENABLED. AFTER THE WHOLE MESSAGE IS RECEIVED
                                ;       DATA WILL THEN BE CHECKED IF DESIRED AND END PASS WILL
                                ;       BE GIVEN IF DESIRED.    THEN THE CYCLE IS REPEATED
                                ;       AS ABOVE.  IF RUNNING IN FULL DUPLEX THE PROGRAM
                                ;       WAITS FOR BOTH THE RECEIVER AND TRANSMITTER TO
                                ;       FINISH THEN RESTARTS THE RECEIVER AND TRANSMITTER.
                                ;*****************************************************************
012072  104416                  $XLB:   KBDIN
012074  032737  000001  011014          BIT     #FULL.DUPLEX,PARAM2
012102  001402                          BEQ     1$
012104  004737  014516                  JSR     PC,STARTR
012110  004737  013254          1$:     JSR     PC,STARTX
012114  005037  011032                  CLR     TIME
012120  032700  100000          2$:     BIT     #XFLG,STAT
012124  001016                          BNE     3$
012126  032700  040000          7$:     BIT     #RFLG,STAT
012132  001024                          BNE     4$
012134  023727  011032  000100          CMP     TIME,#100
012142  103766                          BLO     2$
012144  011402                          MOV     @RCSR,R2
012146  016403  000002                  MOV     XCSR(R4),R3
012152  104001                          HLT     1
012154  005037  011032                  CLR     TIME
012160  000757                          BR      2$
012162  032737  000001  011014  3$:     BIT     #FULL.DUPLEX,PARAM2
012170  001356                          BNE     7$
012172  042700  100000                  BIC     #XFLG,STAT
012176  004737  014516                  JSR     PC,STARTR
012202  000746                          BR      2$
012204  032737  000001  011014  4$:     BIT     #FULL.DUPLEX,PARAM2
012212  001420                          BEQ     8$
012214  032700  100000                  BIT     #XFLG,STAT
012220  001013                          BNE     6$
012222  023727  011032  000100          CMP     TIME,#100
012230  103765                          BLO     4$
012232  011402                          MOV     @RCSR,R2
012234  016403  000002                  MOV     XCSR(R4),R3
012240  104001                          HLT     1
012242  005037  011032                  CLR     TIME
012246  000756                          BR      4$
012250  042700  100020          6$:     BIC     #XFLG,STAT
012254  042700  040000          8$:     BIC     #RFLG,STAT
012260  005037  011032                  CLR     TIME
012264  032777  000200  176552          BIT     #MODAT,@SWR
012272  001002                          BNE     5$
012274  004737  012462                  JSR     PC,TESTD
012300  032777  000020  176536  5$:     BIT     #LOOP,@SWR
012306  001671                          BEQ     $XLB
012310  012737  012072  013250          MOV     #$XLB,BACK
012316  000137  012322                  JMP     EOP
```

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

012322                                  EOP:
012322  104414  000340                          STPS,PRTY7              ;SET PS PRIORITY TO 7
012326  016437  000002  012460                  MOV     XCSR(R4),@TPIE  ;SAVE TX CSR
012334  042737  177737  012460                  BIC     #^C<TIE>,@TPIE  ;CLEAR ALL BUT TX IE,
012342  042764  000040  000002                  BIC     #TIE,XCSR(R4)   ;CLEAR TX IE (EVEN IF IT WASN'T SET)
012350  012766  012410  000002                  MOV     #ENTER,2(SP)    ;SET FOR RETURN IF SW 14=1
012356  010037  013232                          MOV     R0,SAVR0        ;SAVE REGISTER 0
012362  010137  013234                          MOV     R1,SAVR1        ;SAVE REGISTER 1
012366  010237  013236                          MOV     R2,SAVR2        ;SAVE REGISTER 2
012372  010337  013240                          MOV     R3,SAVR3        ;SAVE REGISTER 3
012376  010437  013242                          MOV     R4,SAVR4        ;SAVE REGISTER 4
012402  010537  013244                          MOV     R5,SAVR5        ;SAVE REGISTER 5
012406  000207                                  RTS     PC              ;RETURN TO CONTROL PROGRAM

012410                                  ENTER:
012410  013700  013232                          MOV     SAVR0,R0        ;RESTORE R0
012414  013701  013234                          MOV     SAVR1,R1        ;RESTORE R1
012420  013702  013236                          MOV     SAVR2,R2        ;RESTORE R2
012424  013703  013240                          MOV     SAVR3,R3        ;RESTORE R3
012430  013704  013242                          MOV     SAVR4,R4        ;RESTORE R4
012434  013705  013244                          MOV     SAVR5,R5        ;RESTORE R5
012440  012737  177777  013246                  MOV     #-1,DELAY
012446  053764  012460  000002                  BIS     @TPIE,XCSR(R4)  ;IF ORGINALLY SET; SET TX IE
012454  000177  000570                          JMP     @ADDR
012460  000000                          @TPIE:  000000


                                        ;*********************************
                                        ;       SUBROUTINE TO CHECK
                                        ;       RECEIVER DATA.
                                        ;*********************************
012462  013746  011056                  TESTD:  MOV     ERROR,  -(SP)   ;WAS THERE A RECEIVE ERROR?
012466  001413                                  BEQ     TSTDAT          ;BR IF NO
012470  032777  020000  176346                  BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS?
012476  001007                                  BNE     TSTDAT          ;BR IF YES
012500  104400  012672                          TYPE    ,MSG0           ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
012504  004077  176320                          JSR     R0,@#2016       ;PRINT CONTENTS OF RBUF
012510  005746                                  TST     -(SP)
012512  104400  012753                          TYPE    ,MSG1           ;<15><12>
012516  013701  011022                  TSTDAT: MOV     IXOA,   R1      ;SETUP XMIT DATA ADDR
012522  013702  011020                          MOV     IRDA,   R2      ;SETUP RCV DATA ADDR
012526  122122                          SCANH:  CMPB    (R1)+,  (R2)+   ;DATA OK ?
012530  001776                                  BEQ     SCANH           ;BR IF OK
012532  123741  011040                          CMPB    TX,TERM,-(R1)   ;IS IT END OF DATA
012536  001453                                  BEQ     TESTDX          ;BR IF YES
012540  122742  000002                          CMPB    #02,-(R2)
012544  001005                                  BNE     2$
012546  010237  012554                          MOV     R2,1$
012552  104400                                  TYPE
```

```
DQ11 ITEP OVERLAY       MACY11 27(1006)  29-OCT-76  14:38  PAGE 22
DZDQDB.P11    05-AUG-76 09:23

 970  012554  000000                    1$:     .WORD   0
 971  012556  000443                            BR      TESTDX
 972  012560                            2$:
 973  012560  105712                            TSTB    (R2)            ;
 974  012562  001441                            BEQ     TESTDX          ;BR IF YES
 975  012564  122721  000177                    CMPB    #177,   (R1)+   ;IS IT FILL CHAR?
 976  012570  001756                            BEQ     SCAN4           ;BR IF YES
 977  012572  005301                            DEC     R1              ;BACKUP
 978  012574  122722  000177                    CMPB    #177,   (R2)+   ;IS IT FILL?
 979  012600  001752                            BEQ     SCAN4           ;BR IF YES
 980  012602  105742                            TSTB    -(R2)           ;BACK UP POINTER
 981  012604  123722  011015                    CMPB    PARAM2+1,(R2)+
 982  012610  001746                            BEQ     SCAN4           ;BR IF CHAR WAS SYNC
 983  012612  000240                    SCAN5:  NOP                     ;DATA ERROR
 984  012614  032777  020000  176222            BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS
 985  012622  001016                            BNE     DERR            ;BR IF YES
 986  012624  104400  012756                    TYPE    ,MSG2           ;<15><12>RECEIVED DATA = <15><12>
 987  012630  013737  011020  012640            MOV     IRDA,   RDAX    ;SETUP DATA ADRESS
 988  012636  104400                            TYPE                    ;PRINT RECEIVED DATA
 989  012640  000000                    RDAX:   0                       ;RECEIVED DATA ADDR.
 990  012642  104400  013003                    TYPE    ,MSG3           ;<15><12>DATA SHOULD BE<15><12>
 991  012646  013737  011022  012656            MOV     IXDA,   .+10    ;SETUP ADDR.
 992  012654  104400                            TYPE                    ;PRINT GOOD DATA
 993  012656  011022                            IXDA
 994  012660  111103                    DERR:   MOVB    (R1),R3 ;SETUP XMIT DATA
 995  012662  114202                            MOVB    -(R2),R2        ;SETUP RCV DATA
 996  012664  104007                            HLT+7                   ;DATA ERROR HALT
 997  012666  005726                    TESTDX: TST     (SP)+           ;POP STACK
 998  012670  000207                            RTS     PC              ;RETURN FROM SUB/ROUT
 999
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)  013   3     015  042012  052101    MSG3:   .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
 (1)  013   6  005015  046120  040505    MSG4:   .ASCII <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
 (1)  013   5     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
 (1)  013232  000000                    SAVR0:  0
1001  013234  0     0                   SAVR1:  0
1002  013236  000000                    SAVR2:  0
1003  013240  000000                    SAVR3:  0
1004  013242  0     0                   SAVR4:  0
1005  013244  0     0                   SAVR5:  0
1006  013246  000000                    DELAY:  0
1007  013250  000000                    BACK:   0
1008  013252  000000                    STOP:   0
1009
```

```
1010                                    ;*****************************************************************
1011                                    ;INITIALIZE TRANSMITTER.
1012                                    ;PURPOSE OF THIS ROUTINE IS TO SEND OUT FIVE
1013                                    ;SYNC CHARS. THE SYNC CHAR TO BE SENT IS FOUND
1014                                    ;IN LOCATION PARAM2+1 (HIGH BYTE OF PARAM2).
1015                                    ;WHEN THE SYNC CHARS. HAVE BEEN TRANSMITTED
1016                                    ;THE TRANSMITTER WILL BE ENABLED FOR REAL DATA
1017                                    ;TRANSFER. CHARACTERS WILL BE TRANSMITTED ONE AT A TIME.
1018                                    ;*****************************************************************
1019
1020  013254  005737  013246        STARTX: TST     DELAY           ;IF SW04=1 & SW14=0 WAIT BEFORE TURNING TX ON
1021  013260  001416                        BEQ     NDLY            ;NO GO AHEAD AND TURN ON TX
1022  013262  005037  014154                CLR     TEMP1
1023  013266  012737  000007  014156        MOV     #7,TEMP2
1024  013274  062737  000001  014154        ADD     #1,TEMP1
1025  013302  001374                        BNE     .-6
1026  013304  005337  014156                DEC     TEMP2
1027  013310  001371                        BNE     .-14
1028  013312  005037  013246                CLR     DELAY           ;ZERO DELAY
1029  013316  005037  011032        NDLY:   CLR     TIME
1030  013322  042764  000300  000002        BIC     #300,XCSR(R4)   ;CLEAR BOTH DONE BITS FROM TX
1031  013330  052764  000010  000002        BIS     #BIT3,XCSR(R4)  ;ENABLE ERRORS
1032  013336  032764  000004  000002        BIT     #BIT2,XCSR(R4)  ;WHERE IS THE POINTER POINTING
1033  013344  001403                        BEQ     1$              ;BR IF PRI IS NEXT.
1034  013346  004737  015366                JSR     PC,GETPRI       ;GO TOGGLE TO GET PRIMARY.
1035  013352  000740                        BR      STARTX          ;GOTO BEGGINING
1036  013354  012701  000005        1$:     MOV     #5,R1           ;SET FOR FIVE SYNC CHARS.
1037  013360  012705  014162                MOV     #SYNC,R5        ;SET POINTER LOCATION
1038  013364  113725  011015                MOVB    PARAM2+1,(R5)+  ;LOAD IN SYNC CHARS.
1039  013370  005301                        DEC     R1              ;ALL DONE??
1040  013372  001374                        BNE     .-6             ;BRANCH IF NOT DONE
1041  013374  105015                        CLRB    (R5)            ;SET LAST BYTE TO ZERO
1042
1043                                    ;NOW THE SYNC CHARS ARE LOADED IN CORE FOR THE DQ11 TO
1044                                    ;PICK UP AND TRANSMIT OUT.
1045
1046  013376  112764  000012  000005        MOVB    #12,REG(R4)     ;SELECT MISC REGISTER
1047  013404  012764  004000  000006        MOV     #4000,SEC(R4)   ;SET FOR EIGHT BITS PER CHAR.
1048  013412  112764  000002  000005        MOVB    #2,REG(R4)      ;SELECT THE TX BA PRI.
1049  013420  012764  014162  000006        MOV     #SYNC,SEC(R4)   ;LOAD TX BA PRI. WITH SYNC ADDR.
1050  013426  105364  000005                INCB    REG(R4)         ;SELECT THE TX WC PRI.
1051  013432  012764  177773  000006        MOV     #-5,SEC (R4)    ;SET FOR FIVE CHARS.
1052  013440  042700  100000                BIC     #XFLG,STAT      ;RESET TX COMPLETE FLAG
1053  013444  052764  001000  000002        BIS     #DTR,XCSR(R4)   ;SET DATA TERMINAL READY
1054  013452  005737  013252                TST     STOP            ;FIRST TIME HERE?
1055  013456  001004                        BNE     16$             ;BR IF NO
1056  013460  104400  013160                TYPE    ,MSG5           ;MAKE CONNECTION
1057  013464  005137  011032                COM     STOP
1058  013470  032764  002000  000002  16$:  BIT     #MRDY,XCSR(R4)
1059  013476  001037                        BNE     2$
1060  013500  032764  002000  000002  8$:   BIT     #MRDY,XCSR(R4)
1061  013506  001017                        BNE     3$
1062  013510  023727  011032  000036        CMP     TIME,#36        ;HAVE 30 SEC ELAPSED YET
1063  013516  103770                        BLO     8$              ;NO NOT YET
1064  013520  011402                        MOV     @RCSR,R2        ;LOAD FOR TYPEOUT
1065  013522  016403  000002                MOV     XCSR(R4),R3     ;LOAD FOR TYPEOUT
```

```
1066  013526  032777  010000  175310          BIT     @SW12,@SWR      ;INHIBIT PRINTOUTS?
1067  013534  001001                          BNE     12$             ;BR IF YES
1068  013536  104002                          HLT+2                   ;TYPE "WAITING TO TRANSMIT" MESSAGE.
1069  013540  005037  011032          12$:    CLR     TIME
1070  013544  000755                          BR      8$
1071  013546  005037  014154          3$:     CLR     TEMP1
1072  013552  012737  000005  014156          MOV     #5,TEMP2
1073  013560  062737  000001  014154          ADD     #1,TEMP1
1074  013566  001374                          BNE     .-6
1075  013570  005337  014156                  DEC     TEMP2
1076  013574  001371                          BNE     .-14
1077  013576  032737  000001  011014  2$:     BIT     #FULL.DUPLEX,PARAM2
1078  013604  001023                          BNE     9$
1079  013606  032764  010000  000002  10$:    BIT     #10000,XCSR(R4) ;IS CARRIER UP?
1080  013614  001417                          BEQ     9$
1081  013616  023727  011032  000036          CMP     TIME,#36        ;30 SECONDS UP ?
1082  013624  103770                          BLO     10$             ;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,@SWR      ;INHIBIT PRINTOUTS?
1086  013642  001001                          BNE     13$             ;BR IF YES
1087  013644  104001                          HLT     1               ;TYPE "WAITING "
1088  013646  005037  011032          13$:    CLR     TIME            ;ZERO TIMER
1089  013652  000755                          BR      10$
1090  013654  005037  011032          9$:     CLR     TIME            ;SET TIME=0
1091  013660  052764  000400  000002          BIS     #RQTS,XCSR(R4)
1092  013666  032764  020000  000002  11$:    BIT     #CTS,XCSR(R4)
1093  013674  001017                          BNE     6$
1094  013676  023727  011032  000036          CMP     TIME,#36        ;30 SECONDS UP ??
1095  013704  103770                          BLO     11$             ;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,@SWR      ;INHIBIT PRINTOUTS?
1099  013722  001001                          BNE     14$             ;BR IF YES
1100  013724  104002                          HLT     2               ;TYPE "WAITING "
1101  013726  005037  011032          14$:    CLR     TIME            ;ZERO TIMER
1102  013732  000755                          BR      11$
1103
1104
1105                                  ;HERE NOW BECAUSE CLEAR TO SEND HAS BEEN SET
1106                                  ;NOW SEND THE SYNC CHARS.
1107
1108  013734  052764  000001  000002  6$:     BIS     #.GO,XCSR(R4)   ;SET THE GO BIT
1109  013742  005037  011032                  CLR     TIME            ;SET TIME TO ZERO
1110
1111  013746  105764  000002          4$:     TSTB    XCSR(R4)        ;IS TX DONE WITH SYNCS
1112  013752  100417                          BMI     5$              ;BR IF YES
1113  013754  023727  011032  000036          CMP     TIME,#36        ;HAVE 30 SECOND GONE BY??
1114  013762  103771                          BLO     4$              ;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,@SWR      ;INHIBIT PRINTOUTS?
1118  014000  001001                          BNE     15$             ;BR IF YES
1119  014002  104002                          HLT     2               ;TYPE "AITING...
1120  014004  005037  011032          15$:    CLR     TIME            ;ZERO  IMER
1121  014010  000756                          BR      4$              ;GO W IT FOR DONE
```

```
                                        ;NOW TRANSMITTER DONE HAS SET AND THE SYNC CHARS
                                        ;HAVE BEEN SENT OUT. THE TRANSMITTER IS TO BE
                                        ;ENABLED AND THE DATA WILL BE TRANSFERED.
014012  112764  000002  000005  5$:     MOVB    #2,REG(R4)      ;SELECT TX BA PRI
014020  013764  011022  000006          MOV     IXDA,SEC(R4)    ;LOAD BA
014026  062764  000002  000006          ADD     #2,SEC(R4)      ;POINT BA TWO HIGHER
014034  112764  000003  000005          MOVB    #3,REG(R4)      ;SELECT TX WC PRI
014042  012764  177776  000006          MOV     #-2,SEC(R4)     ;SET TWO CHARS AT A TIME
014050  112764  000006  000005          MOVB    #6,REG(R4)      ;SELECT TX BA SEC
014056  013764  011022  000006          MOV     IXDA,SEC(R4)    ;LOAD TX BA SEC WITH DATA POINTER
014064  112764  000007  000005          MOVB    #7,REG(R4)      ;SELECT TX WC SEC
014072  012764  177776  000006          MOV     #-2,SEC(R4)     ;SET WC WITH -2

                                        ;CHECK FOR HIGH BAUD RATE.
                                        ;NOTE WELL THAT WHEN IN HIGH BAUD MESSAGES MUST BE THE SAME.

014100  032737  000002  011014          BIT     #HI.BAUD,PARAM2 ;HAS HI BAUD BEEN SELECTED?
014106  001413                          BEQ     7$              ;BR IF NO
014110  112764  000003  000005          MOVB    #3,REG(R4)      ;SELECT THE TX WC PRI
014116  005064  000006                  CLR     SEC(R4)         ;ZERO THE PRI WC
014122  112764  000007  000005          MOVB    #7,REG(R4)      ;SELECT THE TX WC SEC
014130  013764  011062  000006          MOV     TXWC,SEC(R4)    ;SET FOR MAX CHAR TRANSFER.


                                        ;THE DQ11 IS TOTALLY ENABLE TO GO. THE TRANSMISSION
                                        ;WILL BEGIN ON THE SECONDAY (THE SYNC CHARS WENT ON
                                        ;THE PRIMARY). THE DQ11 WILL INTERUPT AFTER EACH CHAR
                                        ;AND CONTINUE TRANSMITTING WHILE THE END CHAR IS
                                        ;SEARCHED FOR.

014136  042764  000300  000002  7$:     BIC     #300,XCSR(R4)   ;CLEAR ALL TX DONES
014144  052764  000041  000002          BIS     #IE+.GO,XCSR(R4)
                                                                ;SET INTERUPT ENABLE AND GO
014152  000207                          RTS     PC              ;GO BACK TO MAIN PROGRAM

                                        ;SYNC CHARACTER BUFFER AREA.

014154  000000          TEMP1:  0
014156  000000          TEMP2:  0
014160  000000          ERRORS: 0
014162    026     026   SYNC:   .BYTE   26,26
014164    026     026           .BYTE   26,26
014166    026     026           .BYTE   26,26
```

```
1167                                    ;*************************************************************************
1168                                    ;TRANSMITTER INTERUPT SERVICE ROUTINE
1169                                    ;IN THIS ROUTINE THE END CHARACTER IS CHECKED FOR
1170                                    ;AND IF IT WASN'T FOUND THE TRANSMITTER CHARACTER COUNT
1171                                    ;REGISTER IS ADJUSTED.
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                  ;DQ11 ERROR FLAG SET. EXAMINE REGISTERS FOR ERRO
1181  014210  016437  000004  014160  6.4      MOV   ERR(R4),ERRORS     ;SAVE DQ ERROR REG
1182  014216  042764  000377  000004  7.0      BIC   #377,ERR(R4)
1183  014224  011402                3.8  2$:   MOV   @RCSR,R2          ;PREPARE FOR ERROR TYPEOUT
1184  014226  016403  000002        5.0        MOV   XCSR(R4),R3        ;PREPARE FOR ERROR TYPEOUT
1185  014232  032764  000300  000002  6.5      BIT   #300,XCSR(R4)      ;IS EITHER TX DONE SET??
1186  014240  001001                2.6        BNE   .+4                ;GOOD EITHER PRI OR SEC DONE IS SET
1187  014242  104000                9.3        HLT   0                  ;REPORT ERROR. INTERUPT WAS TO
1188                                          ;BE CAUSED BY TX DONE; AND TX IS NOT DONE.
1189  014244  032737  000002  011014  5.3      BIT   #HI.BAUD,PARAM2    ;IS HIGH BAUD RATE SELECTED?
1190  014252  001402                2.6        BEQ   1$                 ;BR IF NO
1191  014254  005746                4.4        TST   -(SP)              ;FAKE STACK
1192  014256  000474                2.6        BR    HBICHK             ;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                      ;LEAVE TX ISR
1207
1208                                    ;THE ABOVE ROUTINE SERVICED THE INTERUPT IF THE
1209                                    ;PRIMARY REGISTERS CAUSED THE INTERUPT.
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
```

TRANSMITTER INTERUPT SERVICE ROUTINE.

```
1223  014406  000002                    4.8          RTI                ;LEAVE HERE
1224  014410  123741  011040            5.9  TX.CK:  CMPB   TX.TERM,-(R1) ;WAS THAT THE LAST CHARACTER.
1225  014414  001427                    2.6          BEQ    XISRDN      ;ISR IS DONE IF BR IS MADE
1226  014416  123741  011040            5.9          CMPB   TX.TERM,-(R1) ;LAST CHAR?
1227  014422  001424                    2.6          BEQ    XISRDN      ;BR IF YES
1228  014424  032777  000100  174412    7.7  MONDAT: BIT    #BIT6,@SWR  ;CHECK FOR DATA MONITOR.
1229  014432  001405                    2.6          BEQ    1$          ;DON'T MONITOR DATA
1230  014434  105777  174442            5.6          TSTB   @TPS        ;TTY READY??
1231  014440  100002                    2.6          BPL    1$          ;TTY NOT READY GO ON WITH TEST
1232  014442  111177  174436            7.6          MOVB   (R1),@TPB   ;TYPE CHARACTER
1233  014446  000207                    3.5  1$:     RTS    PC          ;GO BACK TO ISR
1234  014450  032777  000100  174366    7.7  HBTCHK: BIT    #BIT6,@SWR  ;CHECK FOR DATA MONITOR
1235  014456  001406                    2.6          BEQ    XISRDN      ;DON'T MONITOR
1236  014460  105777  174416            5.6          TSTB   @TPS        ;IS TTY READY
1237  014464  100003                    2.6          BPL    XISRDN      ;TTY NOT READY GO ON WITH TEST
1238  014466  112777  000124  174410    7.6          MOVB   #'T,@TPB    ;TYPE "T"
1239  014474  052700  100000            3.8  XISRDN: BIS    #XFLG,STA1  ;SET TX COMPLETE FLAG
1240  014500  042764  000441  000002    7.0          BIC    #IE+RQTS+.GO,XCSR(R4)
1241                                                                     ;CLEAR ALL CONDITIONS FOR TX
1242  014506  005037  011032            3.7          CLR    TIME        ;ZERO TIMER
1243  014512  005726                    3.2          TST    (SP)+       ;POP SUBROUTINE PC FROM STACK
1244  014514  000002                    4.8  TXOUT.  RTI                ;LEAVE HERE FOR GOOD.
1245
```

```
1246                                    ;*****************************************************************
1247                                    ;INITIALIZE RECEIVER ROUTINE
1248                                    ;*****************************************************************
1249
1250  014516  000240                    STARTR: NOP                             ;LEAVE OPEN FOR TEST PURPOSES.
1251  014520  042714  000300                    BIC     #300,@RCSR              ;CLEAR ALL DONES
1252  014524  032714  000004                    BIT     #BIT2,@RCSR             ;CHECK FOR POINTER TO SEC.
1253  014530  001403                            BEQ     1$                      ;BR IF POINTING TO PRI
1254  014532  004737  015366                    JSR     PC,GETPRI               ;POINT REGISTERS TO PRI
1255  014536  000767                            BR      STARTR                  ;BEGIN AGAIN
1256  014540  113737  011015  014162  1$:       MOVB    PARAM2+1,SYNC           ;GET SYNC CHAR
1257  014546  113737  011015  014163            MOVB    PARAM2+1,SYNC+1         ;GET SECOND SYNC CHAR
1258  014554  112764  000011  000005            MOVB    #11,REG(R4)             ;SELECT SYNC REGISTER
1259  014562  013764  014162  000006            MOV     SYNC,SEC(R4)            ;LOAD SYNC REGISTER
1260  014570  112764  000000  000005            MOVB    #0,REG(R4)              ;SELECT RX BA PRI.
1261  014576  013764  011020  000006            MOV     IRDA,SEC(R4)            ;LOAD RX BA FOR DATA
1262  014604  105264  000005                    INCB    REG(R4)                 ;GET WC REGISTER
1263  014610  012764  177776  000006            MOV     #-2,SEC(R4)             ;SET FOR ONE CHAR.
1264  014616  112764  000004  000005            MOVB    #4,REG(R4)              ;SELECT RX BA SEC.
1265  014624  013764  011020  000006            MOV     IRDA,SEC(R4)            ;LOAD RX BA SEC.
1266  014632  062764  000002  000006            ADD     #2,SEC(R4)              ;UPDATE DATA POINTER BY ONE
1267  014640  105264  000005                    INCB    REG(R4)                 ;GET WC REGISTER
1268  014644  012764  177776  000006            MOV     #-2,SEC(R4)             ;SET FOR TWO CHAR.
1269  014652  112764  000012  000005            MOVB    #12,REG(R4)             ;SELECT MISC REGISTER
1270  014660  012764  004000  000006            MOV     #4000,SEC(R4)           ;SELECT EIGHT BITS PER CHAR.
1271  014666  012737  000750  011070            MOV     #750,RCC                ;SET FOR MAX 750 CHARS TO BE RXED
1272  014674  042700  040000                    BIC     #PFLG,STAT              ;RESET RX COMPLETE FLAG
1273  014700  005037  011054                    CLR     ERCSR
1274  014704  005037  011056                    CLR     ERDBR
1275  014710  032737  000002  011014            BIT     #HI.BAUD,PARAM2         ;IS HIGH BAUD RATE SELECTED?
1276  014716  001413                            BEQ     2$                      ;BR IF NO
1277  014720  112764  000001  000005            MOVB    #1,REG(R4)              ;SELECT THE RX WC PRI
1278  014726  013764  011064  000006            MOV     RXWC,SEC(R4)            ;RECEIVE CHARS MAX.
1279  014734  112764  000005  000005            MOVB    #5,REG(R4)              ;SELECT THE RX WC SEC.
1280  014742  005064  000006                    CLR     SEC(R4)                 ;ZERO THE RX WC SECONDARY.
1281  014746  052764  001000  000002  2$:       BIS     #DTR,XCSR(R4)           ;SET DATA TERMINAL READY.
1282  014754  005737  013252                    TST     STOP                    ;FIRST TIME HERE?
1283  014760  001004                            BNE     3$
1284  014762  104400  013150                    TYPE    ,MSG5
1285  014766  005137  013252                    COM     STOP
1286  014772                            3$:
1287  014772  000240                            NOP
1288  014774  000240                            NOP
1289  014776  000240                            NOP
1290  015000  000240                            NOP
1291  015002  000240                            NOP
1292  015004  052714  000041                    BIS     #IE+.GO,@RCSR           ;SET INTERUPT ENABLE AND GO
1293  015010  000207                            RTS     PC                      ;GO TO MAINLINE.....
1294
1295                                    ;THE ABOVE SHOULD HAVE PREPARED THE RECEIVER TO
1296                                    ;RECEIVE ALL THE CHARS NEEDED FOR TESTING.
1297                                    ;THE RECEIVER WILL INTERUPT AFTER EACH CHAR IS RECEIVED.
1298                                    ;CHECKING FOR THE END CHARACTER.
1299
```

```
1300
1301                                    ;*************************************************************************
1302                                    ;RECEIVER INTERUPT SERVICE ROUTINE.
1303                                    ;*************************************************************************
1304
1305                                    ;THIS ROUTINE WILL SERVICE THE PRIMARY AND SECONDARY
1306                                    ;REGISTERS WHEN THEY INTERUPT.
1307                                    ;CHECKING FOR THE END CHAR IS PERFORMED.
1308
1309  015012  000240                1.5  RISR:   NOP                     ;LEAVE SPACE FOR DEBUGGING
1310  015014  005764  000004        4.4         TST     ERR(R4)         ;CHECK THE DQ11 ERROR FLAG
1311  015020  100012                2.6         BPL     2$              ;BR IF ERROR FLAG NOT SET
1312  015022  011402                3.8         MOV     @RCSR,R2
1313  015024  016403  000002        5.0         MOV     XCSR(R4),R3
1314  015030  104000                9.3         HLT     0               ;CHECK ERROR REGISTER FOR ERROR.
1315  015032  016437  000004 014160 6.4         MOV     ERR(R4),ERRORS
1316  015040  042764  000377 000004 7.0         BIC     #377,ERR(R4)
1317  015046  005337  011070        3.7  2$:    DEC     RCC             ;CHECK THE BUFFER SIZE
1318  015052  001004                2.6         BNE     1$              ;BR IF OK
1319  015054  000005                1.5         RESET                   ;STOP THE SHOW
1320  015056  104006                9.3         HLT+6                   ;RECEIVER BUFFER FULL
1321  015060  000000                1.8         HALT                    ;STOP EVERY THING
1322  015062  000776                2.6         BR      .-2             ;DISABLE CONT SWITCH
1323  015064  011402                3.8  1$:    MOV     @RCSR,R2        ;PREPARE FOR ERROR TYPEOUT
1324  015066  016403  000002        5.0         MOV     XCSR(R4),R3     ;PREPARE FOR ERROR TYPEOUT
1325  015072  032714  000300        5.3         BIT     #300,@RCSR      ;IS EITHER DONE SET??
1326  015076  001001                2.6         BNE     .+4             ;BR IF AT LEAST ONE DONE IS SET
1327  015100  104000                9.3         HLT     0               ;NOBODY IS DONE. BUT AN INTERUPT OCCURED.
1328  015102  032737  000002 011014 5.3         BIT     #HI.BAUD,PARAM2 ;HIGH BAUD??
1329  015110  001404                2.6         BEQ     3$              ;BR IF NO
1330  015112  042714  000041        5.8         BIC     #IE+.GO,@RCSR   ;CLEAR GO AND INTERUPT ENABLE
1331  015116  005746                4.4         TST     -(SP)           ;FAKE STACK
1332  015120  000507                2.6         BR      HBRCHK          ;XFER ALL DONE
1333  015122  032714  000004        5.3  3$:    BIT     #BIT2,@RCSR     ;WHERE IS THE POINTER.
1334  015126  001423                2.6         BEQ     RX.SEC          ;POINTING TO SECONDARY.
1335  015130  112764  000000 000005 6.4  RX.PRI: MOVB    #0,REG(R4)      ;SELECT THE RX BA PRIMARY
1336  015136  016401  000006        5.0         MOV     SEC(R4),R1      ;GET THE ADDRESS OF DATA.
1337  015142  062764  000002 000006 6.4         ADD     #2,SEC(R4)      ;UPDATE CURRENT ADDRESS
1338  015150  004737  015244        5.8         JSR     PC,RX.CK        ;GO CHECK THE DATA.
1339
1340                                    ;IF I RETURN BACK FROM THE ABOVE SUBROUTINE THE END CHAR
1341                                    ;WAS NOT FOUND AND RECEIVING WILL CONTINUE.
1342
1343  015154  112764  000001 000005 6.4         MOVB    #1,REG(R4)      ;SELECT THE RX WC PRI.
1344  015162  012764  177776 000006 6.4         MOV     #-2,SEC(R4)     ;LOAD WITH -2
1345  015170  042714  000200        5.8         BIC     #BIT7,@RCSR     ;CLEAR RX DONE
1346  015174  000002                4.8         RTI                     ;LEAVE HERE NOW.
1347
1348                                    ;THE ABOVE ROUTINE WAS FOR THE PRIMARY REGISTER INTERUPT.
1349                                    ;THE BELOW ROUTINE IS FOR THE SECONDARY REGISTERS.
1350
1351  015176  112764  000004 000005 6.4  RX.SEC: MOVB    #4,REG(R4)      ;SELECT THE RX BA SEC.
1352  015204  016401  000006        5.0         MOV     SEC(R4),R1      ;GET DATA ADDRESS
1353  015210  062764  000002 000006 6.4         ADD     #2,SEC(R4)      ;UPDATE CURRENT ADDRESS
1354  015216  004737  015244        5.8         JSR     PC,RX.CK        ;GO CHECK THE DATA
1355
```

```
1356                                                    ;IF I COME BACK THAT MEANS THAT THE END CHARACTER WASN'T
1357                                                    ;FOUND AND I SHOULD CONTINUE RECEVING.
1358
1359  015222  112764  000005  000005    6.4           MOVB    #5,REG(R4)      ;SELECT THE RX MC SEC
1360  015230  012764  177776  000006    6.4           MOV     #-2,SEC(R4)     ;LOAD WITH A -2
1361  015236  042714  000100            5.8           BIC     #BIT6,@RCSR     ;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 =011
1369  015260  001403                    2.6           BEQ     RXISRDN         ;BR IF 011
1370  015262  123721  011041            4.7           CMPB    RX.TERM,(R1)+   ; 011?
1371  015266  001012                    2.6           BNE     NO.12           ; NO 011
1372  015270                            RXISRDN:
1373  015270  042714  000041            5.8           BIC     #IE+.GO,@RCSR   ;CLEAR GO AND INTERUPT ENABLE
1374  015274  042714  000300            5.8           BIC     #300,@RCSR      ;CLEAR ALL RX DONES
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     #BIT5,@SWR      ;MONITOR RX DATA??
1380  015322  001405                    2.6           BEQ     1$              ;DON'T MONITOR
1381  015324  105777  173552            5.6           TSTB    @TPS            ;TTY READY??
1382  015330  100002                    2.6           BPL     1$              ;NOT READY GO ON WITH TEST
1383  015332  114177  173546            8.8           MOVB    -(R1),@TPB      ;PRINT CHARACTER
1384  015336  000207                    3.5  1$:      RTS     PC              ;GO TO RX ISR
1385  015340  032777  000040  173476    7.7  HBRCHK:  BIT     #BIT5,@SWR      ;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
```

```
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     1$              ;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          1$:     BIT     #BIT2,@RCSR     ;IS RX ON SECONDARY REG
1406  015452  001430                          BEQ     2$              ;BR IF ON PRI.
1407  015454  112764  000C12  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  2$:     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
      015566  005015  042522  042503  MFULL:  .ASCIZ  <15><12>/RECEIVER BUFFER FILLED. ERROR!!/
      015630  005015  042522  042503  MRXDA:  .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
      015655     015  051012  041505  MRXSB:  .ASCIZ  <15><12>/RECEIVED DATA SHOULD BE /<15><12>
```

F03

```
DQ11 ITEP OVERLAY       MACY11 27(1006)  29-OCT-76  14:38  PAGE 32
DZD908.P11    05-AUG-76 09:23           RECEIVER INTERUPT SERVICE ROUTINE.


                                        .EVEN
  1425          000001                  .END
```

```
BA       011074            605#    666
BACK     013250            769*    797*    832*    912*    943    1007#
BIT0   = 000001            599#
BIT1   = 000002            599#
BIT10  = 002000            599#
BIT11  = 004000            599#
BIT12  = 010000            599#    675    1413
BIT13  = 020000            599#    634#    954     984
BIT14  = 040000            599#
BIT15  = 100000            599#
BIT2   = 000004            599#   1032    1193    1252    1333    1394    1405
BIT3   = 000010            599#    684    1031
BIT4   = 000020            599#
BIT5   = 000040            599#    682    1379    1385
BIT6   = 000100            599#   1222    1228    1234    1361    1401    1414
BIT7   = 000200            599#   1205    1345    1366    1367
BIT8   = 000400            599#
BIT9   = 001000            599#
B2016    011030            615#    957
CTS    = 020000            599#   1092
DELAY    013246            686*    721*    941*   1006#   1020    1028*
DERR     012660            985     994#
DISPLA   011046            625#
DQ11     011000            604#
DSFLG  = 020000            599#    633#
DSSTAT   011060            640#
DTR    = 001000            599#   1053    1281
ENTER    012410            925     934#
EOP      012322            770     798     833     913     920#
E[illegible]SR   011054            639#   1273*
E[illegible]SR   011056            639#    952    1274*
E[illegible]   = 000004            599#    673*    675*    676*   1176    1181    1182*   1310    1315    1316*
E[illegible]ORS  014160            683*   1163#   1181*   1315*
FLAG     011042            623#
FULL.D=  000001            654#    872     898     893    1077
GETPRI   015366           1034    1254    1394#   1403    1416
GO       011346            720#
H[illegible]CHK  015340           1332    1395#
H[illegible]CHK  014150           1192    1234#
HI.BAU=  000002            599#   1140    1189    1275    1328
IE     = 000040            599#   1155    1240    1292    1330    1373
ILB    = 000010            599#    729
I[illegible]A    011020            611#    836     961     997    1261    1265
I[illegible]A    011022            612#    696     703     837     960     991     993    1128    1133
KBDIN  = 104416            593#    751     782     814     871
LOOP   = 000020            599#    767     795     830     910
MFULL    01[illegible]6            1424#
MO[illegible]AT  014124           1273#
MRDY   = 002000            599#   1058    1060
MRXDA    015630           1424#
MR[illegible]SB  015655           1424#
M[illegible]D    012572            996    1000#
MSG1     012753            969    1000#
MSG2     012756            986    1000#
MSG3     013003            990    1000#
MSG4     013026           1000#
```

```
MSGS    013160          1000#   1056    1284
NOLY    013316          1021    1029#
NODAT = 000200           599#    763     826     907
NO.DAT  015264          1412    1423#
NO.12   015314          1371    1379#
OWI   = 000002           599#    726
OWO   = 000001           599#    723
PPRMI   011012           608#    667
PRMM2   011014           609#    668     872     888     893     981    1038    1077    1140    1189    1256    1257    1275
                        1328
PRMM3   011016           610#
PR10R   011010           607#    663     665
PRTY0 = 000030           599#
PRTY1 = 000040           599#
PRTY2 = 000100           599#
PRTY3 = 000140           599#
PRTY4 = 000200           599#
PRTY5 = 000240           599#
PRTY6 = 000300           599#
PRTY7 = 000340           599#    921
QTPIE   012460           922#    923#    942     944#
RCC     011070           645#   1271#   1317#
RDA     011072           646#
RDAX    012640           987#    989#
REG   = 000005           599#    678#    681#   1046#   1048#   1050#   1127#   1130#   1132#   1134#   1142#   1144#   1195#
                        1203#   1211#   1220#   1258#   1260#   1262#   1264#   1267#   1269#   1277#   1279#   1335#   1343#
                        1351#   1359#   1396#   1398#   1407#   1409#   1411#   1418#
RFLG  = 040000           632#    753     766     817     829     879     905    1272    1375
RISR    015012           662    1309#
RIV     011006           606#    661
RQTS  = 000400           599#   1091    1240
R[illegible]IT = 104410  593#
RXISRD  015270          1369    1372#   1396    1388    1390
RXMC    011064           643#    710#   1278
RX.CK   015244          1353    1354    1366#
RX.PRI  015130          13[illegible]#
RX.SEC  015176          1354    1351#
RX.TER  011041           621#    706    1368    1370
SAVRD   013232           926#    935    1000#
[illegible]             927#    936    1001#
[illegible]             928#    937    1002#
[illegible]             929#    938    1003#
[illegible]             930#    939    1004#
SAVRS   013244           931#    940    1005#
SCAN4   01[illegible]    962#    963     976     979     982
SCANS   01[illegible]    963#
SEC   = 000006           599#    677#    682#   1047#   1049#   1051#   1128#   1129#   1131#   1133#   1135#   1143#   1145#
                        1196    1197#   1204#   1212    1213#   1221#   1259#   1261#   1263#   1265#   1266#   1268#   1270#
                        1278#   1280#   1336    1337#   1344#   1352    1353#   1360#   1397#   1399#   1408#   1410#   1412#
                        1419#
SETTLE  011024           613#
SPCSR   011052           637#
START   011106           618     658#
STARTR  014516           752     815     874     891    1250#   1255
STARTX  013[illegible]   783     846     875    1020#   1035
STOP    013[illegible]   722#   1008#   1054    1057#   1282    1285#
```

```
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
SW12  = 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#
TPB     011104          652#  1232*  1238*  1383*  1389*
TPS     011102          651#  1230   1236   1381   1387
TSTDAT  012516          953   955   960#
TXOUT   014514         1244#
TXUC    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   986   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  1084  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
XISR    014170          664   1175#
XISCON  014474         1225   1227  1235  1237  1239#
XLB   = 000004          599#  732
XWAIT = 104412          599#
X.X     011274          695#
$ILB    011642          731   814#  857
$XWI    011436          728   751#  771
$XWO    011544          725   782#  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
```

```
DQ11 ITEP OVERLAY       MACY11 27(1006)  29-OCT-76  14:38  PAGE 38
DZDQ08.P11    05-AUG-76 09:23          CROSS REFERENCE TABLE -- MACRO NAMES

BOX          1#    600     627     655
DCP[illegible]  1#
D[illegible]DC1  1#
D[illegible]M  1#
DJP[illegible]  1#
DLP[illegible]  1#
D[illegible]M  1#
DQDOC1       1#    560
DQPARM       1#    581
DUPARM       1#
DUPPAR       1#
DVDOC1       1#
DVPARM       1#
DZPARM       1#
HELLO        1#
HLT        599#    759     791     823     853     885     901     996    1068    1087    1100    1119    1180    1187    1314
          1320    1327
SEQUAT       1#    599
SINTF        1#    599
SITEP        1#    713
SSERV        1#    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)
```

K03