DH11

OVERLAY FOR ITEP
CZDHLC0

AH-8489C-MC

FICHE 1 OF 1

JUN 1978
digital
MADE IN USA

IDENTIFICATION

PRODUCT CODE:	AC-8488C-MC
PRODUCT NAME:	CZDHLCO DH11 OVRLY FOR ITEP
PROGRAM DATE:	MARCH 1978
MAINTAINER:	DIAGNOSTICS
AUTHORS:	R A JONES JOHN EGOLF FAY BASHAW

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 CZDHLCO DH11 COMMUNICATION INTERFACE.

2.2 STORAGE.

4K OF CORE

3.0 LOADING PROCEDURE
THIS PROGRAM IS IN ABSOLUTE FORMAT.
THE ABS LOADER MUST BE USED TO LOAD THE PROGRAM.

4.0 OPERATING PROCEDURES.

A. TWO METHODS OF ENTERING PARAMETERS ARE PROVIDED
1. LOAD ADDRESS 200 AND START TO ENTER PARAMS FROM CONSOLE TTY, PROCEED TO SECTION B.
2. LOAD ADDRESS 200 AND SET SWITCH REGISTER BIT 15 BEFORE
STARTING TO ENTER PARAMS FROM CONSOLE SWITCHES, PROCEED TO SECTION C.
*THE PROGRAM MAY BE RESTARTED AT LOC 204 (ONCE PARAMETERS HAVE ALREADY BEEN SELECTED)

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

1. THE PROGRAM WILL TYPEOUT THE NAME OF THE VARIABLE OVERLAY.
A. IF YOU WISH TO SETUP JUST THE INDICATED OVERLAY, TYPE A CARAGE RETURN
B. IF YOU WISH TO SETUP A DN11, TYPE IN DN.
C. IF YOU WISH TO SETUP A DM11BB, TYPE IN DMB.

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

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

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

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

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

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

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

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

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

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

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

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

2. THE FOLLOWING HALTS ARE REPEATED FOR EACH ISR SPECIFIED.
SETUP SEQUENCE IS: DN11,DM11-BB THEN VARIABLE OVERLAY. (EACH ENTRY SET SWICHES THEN HIT CONTINUE.)
A. HALT FOR BUS ADDRESS OF INTERFACE
B. HALT FOR VECTOR ADDRESS OF INTERFACE
C. HALT FOR PRIORITY OF INTERFACE
D. HALT FOR INTERFACE PARAM #1 (SEE SECT. 10.0 IN OVERLAY LISTING FOR PARAMETER DESCRIPTION)
E. HALT FOR INTERFACE PARAM #2 (DN11 AND DMB8 PARAMETERS ARE DISCUSSED IN SECT. 10.0 OF THE MONITOR.
F. GO BACK TO STEP A IF THIS SETUP WAS FOR DN OR DMB.

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

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

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

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

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

D. OPERATIONAL SWITCH SETTINGS.

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

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

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

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

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

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

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

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

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

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

TEST MODES

INTERNAL LOOPBACK MODE

1. THE OVERLAY WAITS TO RECEIVE A MESSAGE (TERMINATED BY <001>)
2. VERFIES THE DATA AGAINST THE DATA SELECTED BY SW09 AND SW10(SW7=0)
3. TRANSMIT THE DATA SELECTED BY SW09 AND SW10 (SW8=0) OR
   TRANSMIT THE RECEIVED DATA (SW8=1)
4. RETURNS TO MONITOR FOR "END PASS" (SW4=1) OR
   GO TO STEP 1. (SW4=0)

EXTERNAL LOOPBACK MODE

1. THE OVERLAY SETS REQUEST TO SEND
2. WAIT FOR CLEAR TO SEND
3. TRANSMITS THE SELECTED DATA
4. RESETS REQUEST TO SEND
5. WAIT FOR MESSAGE TO BE RECEIVED
6. VERIFIES THE DATA (SW07=0)
7. RETURNS TO MONITOR FOR "END PASS". (SW04=1) OR
   GO TO STEP 1(SW04=0)

ONE-WAY-IN MODE

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

ONE-WAY-OUT MODE

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

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

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

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

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

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

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

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

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

*BXXXXXX TO PRINTOUT THE 16 BYTES
AFTER LOC XXXXXX.

*C TO CONTINUE

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

5.0 PROGRAM AND/OR OPERATOR ACTION

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

5.1 NORMAL HALTS

SEE SECTION 4.

6.0 ERRORS

6.1 ERROR REPORTING

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

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

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

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

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

THERE WAS A RECEIVER ERROR. RECEIVER DATA REGISTER =XXXXXX

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

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

7.0 RESTRICTIONS

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

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

WHEN THE COMMUNICATION LINK INVOLVES MODEMS THE FOLLOWING
RESTRICTION APPLY:

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

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

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

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

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

8.0 MISCELLANEOUS

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

9.0 PROGRAM DESCRIPTION

9.1 THE CZDHLCO DH11 INTERFACE SERVICE PARAMS ARE SETUP, AS SPECIFIED BY THE OPERATOR,
BY THE ITEP CONTROL PROGRAM.

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

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

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

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

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

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

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

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

9.4 THE RETURN TO MONITOR ROUTINE FOR END PASS AT EOP:
LOCKS OUT INTERUPTS AND SAVES THE TRANSMITTER INTERUPT ENABLE
BIT AND ALL GENERAL REGISTERS. IT THEN RETURNS TO THE MONITOR
TO TYPE "END PASS". THE MONITOR CHECKS SW14 IF UP IT RETURNS
TO ENTER:, OTHERWISE IT RESTARTS THE PROGRAM.

9.5 ENTER: IS ENTERED FROM THE MONITOR AFTER TYPEING "END PASS",
IT RESTORES THE GENERAL REGISTERS AND THE TRANSMITTER CSR
AS SAVED IN EOP. THE DELAY FLAG IS SET AND PROGRAM RETURNS TO
THE SCAN ROUTINE(OWO,OWI,ILB,XLB) WHERE IT CAME FROM.

9.6 THE INITIALIZE TRANSMIT SUBROUTINE AT STARTX:
SETS UP THE INTERFACE AND POINTERS NECESSARY TO
INITIATE A TRANSMIT OPERATION.
AFTER SETTING "DATA TERMINAL READY" AND "REQUEST TO SEND" A CHECK
IS MADE ON PARAM2 TO DETERMINE IF HALF DUPLEX OPERATION
WAS SELECTED BY THE OPERATOR. IF IT WAS, THE
SUBROUTINE WAITS FOR CLEAR TO SEND.
A 'WAITING FOR CLEAR TO SEND' PRINTOUT OCCURS
EVERY 30 SECONDS UNTIL CLEAR TO SEND IS ASSERTED.

9.7 THE INITIALIZE RECEIVED SUBROUTINE AT STARTR:
SETS UP THE INTERFACE AND POINTERS NECESSARY TO
RECEIVE A MESSAGE.

9.8 THE TRANSMIT INTERRUPT SERVICE ROUTINE,
AT XISR:, IS ENTERED VIA TRANSMIT INTERRUPTS
FROM THE INTERFACE.
A TEST IS MADE TO SEE IF THE LAST CHARACTER
TRANSMITTED WAS A NULL (ALL ZEROS) CHARACTER.
IF IT WAS; THE TRANSMIT LOGIC IN THE INTERFACE
IS RESET AND THE TRANSMIT COMPLETE FLAG IS SET.
AT XISR1: THE NEXT CHARACTER IS TRANSMITTED
AND PRINTED ON THE TTY IF THE MONITOR TRANSMIT
SWITCH IS SET.

9.9 THE RECEIVE INTERRUPT SERVICE ROUTINE
AT RISR:, IS ENTERED VIA RECEIVER INTERRUPTS
FROM THE INTERFACE.
THE RECEIVED CHARACTER IS STORED IN
THE INPUT BUFFER AND PRINTED ON THE TTY IF
THE MONITOR RECEIVER SWITCH IS SET.
IF THE INPUT BUFFER IS FULL, A 'BUFFER FULL'
PRINTOUT WILL OCCUR. THIS INDICATES THAT A
LINE FEED CHARACTER WAS NOT RECOGNIZED

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

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

DH11 RESTRICTIONS

IF A DM11BB EXISTS IN THE SYSTEM WITH THE DH11 BEING
TESTED, BUT MODEM CONTROL IS NOT DESIRED AND THE DM11BB
WAS NOT INITALIZED BY ITEP THE PROGRAM WILL HANG IN THE
DH11 TRANSMITTER INITALIZATION ROUTINE. TO CORRECT THIS
LOAD LOCATION "DMBB" WITH AN ADDRESS THAT WILL TIME OUT(NO
SLAVE SYNC RESPONSE). THE ADDRESS OF DMBB CAN BE FOUND
IN THE CROSS REFERENCE TABLE IN THE BACK OF THIS LISTING.

576

10.0 PARAMETERS FOR THE DH11

PARAM#1 IS LOADED INTO THE SYSTEM CONTROL REGISTER.(SCR)
BITS 0-3 LINE SELECTION, DEFAULT= LINE 0 (0000)

PARAM#2 IS LOADED INTO THE LINE PARAMETER REGISTER.(LPR)
BITS 0,1 CHARACTER LENGTH, DEFAULT= 8 BITS (11)
BIT 2 STOP BITS, DEFAULT= 2 STOP BITS (1)
BIT 4 PARITY ENABLED (1), DEFAULT= (0)
BIT 5 ODD PARITY (1), DEFAULT= (0)
BITS 6-9 RECEIVER SPEED, DEFAULT= 110 BAUD (0011)
BITS 10-13 TRANSMIT SPEED, DEFAULT= 110 BAUD (0011)
BIT 14 HALF DUPLEX (1), DEFAULT= FULL DUPLEX (0)

PARAM#3 IS NOT USED (177777)

```
                                ;*****************************************************************
                                ;       DH11 INTERFACE SERVICE PARAMS
                                ;*****************************************************************
                        011000          .=11000
011000  044104  000040        DH11:   .ASCIZ  /DH /                   ;ISR NAME
011004  160020                BA:     160020                          ;BUS ADDRESS
011006  000300                RIV:    300                             ;VECTOR ADDRESS
011010  000240                PRIOR:  240                             ;PRIORITY
011012  000000                PARAM1: 0                               ;PARAM #1
011014  006307                PARAM2: 006307                          ;PARAM #2
011016  177777                PARAM3: 177777                          ;PARAM #3
011020  000000                IRDA:   .WORD   0                       ;INITIAL READ DATA ADDRESS
011022  000000                IXDA:   .WORD   0                       ;INITIAL XMIT DATA ADDRESS
011024  000000                SETTLE: .WORD   0                       ;LINE SETTLE DELAY FLAG
011026  000000                        .WORD   0                       ;
011030  000000                B2016:  .WORD   0                       ;ADDR OF BIN TO OCT TYPE ROUTINE
011032  000000                TIME:   .WORD   0                       ;TIMER
011034  000000                        .WORD   0                       ;
011036  011102                        .WORD   START                   ;ADDR OF START OF PROGRAM
011040                        TX.TERM:
011040     000                        .BYTE   000                     ;TRANSMITTER TERMINATING CHAR.
011041                        RX.TERM:
011041     001                        .BYTE   001                     ;RECEIVER TERMINATING CHAR.
011042  000000                FLAG:   .WORD   0
011044  177570                SWR:    177570
011046  177570                DISPLAY:177570

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

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

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

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

                        000001          FULL.DUPLEX=000001
```

```
                                ;*****************************************************************
                                ;       DH11-X INTERFACE SERVICE ROUTINE
                                ;*****************************************************************
011102  000240                  START:  NOP
011104  017700  177734                  MOV     @SWR,   R0              ;SETUP MODE IN R0
011110  042700  177400                  BIC     #177400,R0              ;STRIP JUNK
011114  013702  011006                  MOV     RIV,    R2              ;SETUP
011120  012722  014040                  MOV     #RISR,  (R2)+           ;INTERRUPT
011124  013722  011010                  MOV     PRIOR,  (R2)+           ;VECTORS
011130  012722  013470                  MOV     #XISR,  (R2)+           ;
011134  013722  011010                  MOV     PRIOR,  (R2)+           ;
011140  013704  011004                  MOV     BA,     R4              ;SETUP BUS ADDR INDEX
011144  012714  004000                  MOV     #MC,@RCSR
011150  053714  011012                  BIS     PARAM1,@RCSR
011154  053764  011014  000004          BIS     PARAM2,LPR(R4)
011162  123727  011012  000017          CMPB    PARAM1,#17
011170  101402                          BLOS    1$
011172  000000                          HALT                            ;NUMBER IN PARAM1 IS TOO LARGE!
011174  000776                          BR      .-2                     ;MUST BE 17 OR LESS (LINE# IN OCTAL)
011176  010046                  1$:     MOV     R0,-(SP)                ;SAVE R0
011200  012700  000001                  MOV     #1,R0
011204  013701  011012                  MOV     PARAM1,R1
011210  005701                  2$:     TST     R1
011212  001403                          BEQ     3$                      ;CALCULATE BAR BIT
011214  006300                          ASL     R0
011216  005301                          DEC     R1
011220  000773                          BR      2$
011222  010037  013714          3$:     MOV     R0,BARTMP
011226  012600                          MOV     (SP)+,R0

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

011230  005037  011032          GO:     CLR     TIME
011234  005037  013120                  CLR     DELAY
011240  005037  013124                  CLR     STOP
011244  032700  000001                  BIT     #OWO,MODE
011250  001402                          BEQ     1$
011252  000137  011426                  JMP     $OWO
011256  032700  000002          1$:     BIT     #OWI,MODE
011262  001402                          BEQ     2$
011264  000137  011320                  JMP     $OWI
011270  032700  000010          2$:     BIT     #ILB,MODE
011274  001402                          BEQ     3$
011276  000137  011524                  JMP     $ILB
011302  032700  000004          3$:     BIT     #XLB,MODE
011306  001402                          BEQ     4$
011310  000137  011754                  JMP     $XLB
011314  000000                  4$:     HALT
011316  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".
                        ;*****************************************************************

011320  104416          $OWI:   KBDIN
011322  004737  013726          JSR     PC,STARTR
011326  032700  040000  1$:     BIT     #RFLG,STAT
011332  001013                  BNE     2$
011334  023727  011032  000100  CMP     TIME,#100
011342  103771                  BLO     1$
011344  011402                  MOV     @RCSR,R2
011346  016403  000000          MOV     XCSR(R4),R3
011352  104001                  HLT     1
011354  005037  011032          CLR     TIME
011360  000762                  BR      1$
011362  032777  000200  177454  2$:     BIT     #NODAT,@SWR
011370  001002                  BNE     3$
011372  004737  012344          JSR     PC,TESTD
011376  042700  040000  3$:     BIC     #RFLG,STAT
011402  032777  000020  177434          BIT     #LOOP,@SWR
011410  001405                  BEQ     4$
011412  012737  011424  013122          MOV     #4$,BACK
011420  000137  012204                  JMP     EOP
011424  000735          4$:     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.
                        ;*****************************************************************

011426  104416          $OWO:   KBDIN
011430  004737  013126          JSR     PC,STARTX
011434  005037  011032          CLR     TIME
011440  032700  100000  1$:     BIT     #XFLG,STAT
011444  001013                  BNE     2$
011446  023727  011032  000100  CMP     TIME,#100
011454  103771                  BLO     1$
011456  011402                  MOV     @RCSR,R2
011460  016403  000000          MOV     XCSR(R4),R3
011464  104001                  HLT     1
011466  005037  011032          CLR     TIME
011472  000762                  BR      1$
011474  042700  100000  2$:     BIC     #XFLG,STAT
011500  032777  000020  177336          BIT     #LOOP,@SWR
```

```
  762  011506  001405                        BEQ     3$
  763  011510  012737  011522  013122        MOV     #3$,BACK
  764  011516  000137  012204                JMP     EOP
  765  011522  000741              3$:       BR      $OWO
  766
  767
  768
```

```
                        ;*****************************************************************
                        ;       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 REPEATED AS ABOVE.
                        ;*****************************************************************

011524  104416          $ILB:   KBDIN
011526  004737  013726          JSR     PC,STARTR
011532  005037  011032          CLR     TIME
011536  032700  040000  1$:     BIT     #RFLG,STAT
011542  001013                  BNE     2$
011544  023727  011032  000100  CMP     TIME,#100
011552  103771                  BLO     1$
011554  011402                  MOV     @RCSR,R2
011556  016403  000000          MOV     XCSR(R4),R3
011562  104001                  HLT     1
011564  005037  011032          CLR     TIME
011570  000762                  BR      1$
011572  032777  000200  177244  2$:     BIT     #NODAT,@SWR
011600  001002                  BNE     3$
011602  004737  012344          JSR     PC,TESTD
011606  042700  040000  3$:     BIC     #RFLG,STAT
011612  032777  000020  177224  BIT     #LOOP,@SWR
011620  001405                  BEQ     4$
011622  012737  011634  013122  MOV     #4$,BACK
011630  000137  012204          JMP     EOP
011634  032777  000400  177202  4$:     BIT     #400,   @SWR    ;USE EXTERNAL DATA?
011642  001416                  BEQ     7$              ;BR IF NO
011644  013702  011020          MOV     IRDA,   R2      ;SET POINTER
011650  013703  011022          MOV     IXDA,   R3      ;SET POINTER
011654  010337  011070          MOV     R3,     XDA     ;SETUP XMIT DATA ADDR
011660  112223                  MOVB    (R2)+,  (R3)+   ;MOVE INPUT TO OUTPUT
011662  001376                  BNE     .-2             ;LOOP IF NOT ZERO CHAR
011664  112743  000177          MOVB    #177,   -(R3)   ;INSERT A FILL CHAR
011670  005203                  INC     R3              ;BUMP ADDRESS
011672  112723  000177          MOVB    #177,   (R3)+   ;INSERT ANOTHER FILL
011676  105023                  CLRB    (R3)+           ;INSERT ZERO CHAR
011700  005037  011032  7$:     CLR     TIME
011704  004737  013126          JSR     PC,STARTX
011710  032700  100000  5$:     BIT     #XFLG,STAT
011714  001013                  BNE     6$
011716  023727  011032  000100  CMP     TIME,#100
011724  103771                  BLO     5$
011726  011402                  MOV     @RCSR,R2
011730  016403  000000          MOV     XCSR(R4),R3
011734  104001                  HLT     1
011736  005037  011032          C_R     TIME
011742  000762                  BR      5$
011744  042700  100000  6$:     BIC     #XFLG,STAT
011750  000137  011524          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.
                        ;*************************************************************************

011754  104416          $XLB:   KBDIN
011756  032737  040000  011014          BIT     #HALF.DUPLEX,PARAM2
011764  001002                        BNE     1$
011766  004737  013726              JSR     PC,STARTR
011772  004737  013126      1$:     JSR     PC,STARTX
011776  005037  011032              CLR     TIME
012002  032700  100000      2$:     BIT     #XFLG,STAT
012006  001016                        BNE     3$
012010  032700  040000      7$:     BIT     #RFLG,STAT
012014  001024                        BNE     4$
012016  023727  011032  000100          CMP     TIME,#100
012024  103766                        BLO     2$
012026  011402                        MOV     @RCSR,R2
012030  016403  000000                MOV     XCSR(R4),R3
012034  104001                        HLT     1
012036  005037  011032                CLR     TIME
012042  000757                        BR      2$
012044  032737  040000  011014  3$:     BIT     #HALF.DUPLEX,PARAM2
012052  001756                        BEQ     7$
012054  042700  100000                BIC     #XFLG,STAT
012060  004737  013726              JSR     PC,STARTR
012064  000746                        BR      2$
012066  032737  040000  011014  4$:     BIT     #HALF.DUPLEX,PARAM2
012074  001020                        BNE     8$
012076  032700  100000                BIT     #XFLG,STAT
012102  001013                        BNE     6$
012104  023727  011032  000100          CMP     TIME,#100
012112  103765                        BLO     4$
012114  011402                        MOV     @RCSR,R2
012116  016403  000000                MOV     XCSR(R4),R3
012122  104001                        HLT     1
012124  005037  011032                CLR     TIME
012130  000756                        BR      4$
012132  042700  100000          6$:     BIC     #XFLG,STAT
012136  042700  040000          8$:     BIC     #RFLG,STAT
012142  005037  011032              CLR     TIME
012146  032777  000200  176670          BIT     #NODAT,@SWR
012154  001002                        BNE     5$
012156  004737  012344              JSR     PC,TESTD
012162  032777  000020  176654  5$:     BIT     #LOOP,@SWR
012170  001671                        BEQ     $XLB
012172  012737  011754  013122          MOV     #$XLB,BACK
012200  000137  012204                JMP     EOP
```

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

012204                                  EOP:
012204  104414  000340                          STPS,PRTY7              ;SET PS PRIORITY TO 7
012210  016437  000000  012342                  MOV     XCSR(R4),QTPIE  ;SAVE TX CSR
012216  042737  157777  012342                  BIC     #^C<TIE>,QTPIE  ;CLEAR ALL BUT TX IE.
012224  042764  020000  000000                  BIC     #TIE,XCSR(R4)   ;CLEAR TX IE (EVEN IF IT WASN'T SET)
012232  012766  012272  000002                  MOV     #ENTER,2(SP)    ;SET FOR RETURN IF SW 14=1
012240  010037  013104                          MOV     R0,SAVR0        ;SAVE REGISTER 0
012244  010137  013106                          MOV     R1,SAVR1        ;SAVE REGISTER 1
012250  010237  013110                          MOV     R2,SAVR2        ;SAVE REGISTER 2
012254  010337  013112                          MOV     R3,SAVR3        ;SAVE REGISTER 3
012260  010437  013114                          MOV     R4,SAVR4        ;SAVE REGISTER 4
012264  010537  013116                          MOV     R5,SAVR5        ;SAVE REGISTER 5
012270  000207                                  RTS     PC              ;RETURN TO CONTROL PROGRAM

012272                                  ENTER:
012272  013700  013104                          MOV     SAVR0,R0        ;RESTORE R0
012276  013701  013106                          MOV     SAVR1,R1        ;RESTORE R1
012302  013702  013110                          MOV     SAVR2,R2        ;RESTORE R2
012306  013703  013112                          MOV     SAVR3,R3        ;RESTORE R3
012312  013704  013114                          MOV     SAVR4,R4        ;RESTORE R4
012316  013705  013116                          MOV     SAVR5,R5        ;RESTORE R5
012322  012737  177777  013120                  MOV     #-1,DELAY
012330  053764  012342  000000                  BIS     QTPIE,XCSR(R4)  ;IF ORGINALLY SET; SET TX IE
012336  000177  000560                          JMP     @BACK
012342  000000                          QTPIE:  000000


                                        ;*************************************
                                        ;       SUBROUTINE TO CHECK
                                        ;       RECEIVER DATA.
                                        ;*************************************
012344  013746  011056                  TESTD:  MOV     ERDBR,  -(SP)   ;WAS THERE A RECEIVE ERROR?
012350  001413                                  BEQ     TSTDAT          ;BR IF NO
012352  032777  020000  176464                  BIT     #BIT13,@SWR     ;INHIBIT PRINTOUTS?
012360  001007                                  BNE     TSTDAT          ;BR IF YES
012362  104400  012544                          TYPE    ,MSGO           ;<15><12>THERE WAS A RECEIVE ERROR. RBUF=
012366  004077  176436                          JSR     R0,@B2016       ;PRINT CONTENTS OF RBUF
012372  005746                                  TST     -(SP)
012374  104400  012625                          TYPE    ,MSG1           ;<15><12>
012400  013701  011022                  TSTDAT: MOV     @XDA,   R1      ;SETUP XMIT DATA ADDR
012404  013702  011020                          MOV     @RDA,   R2      ;SETUP RCV DATA ADDR
012410  122122                          SCAN4:  CMPB    (R1)+,  (R2)+   ;DATA OK ?
012412  001776                                  BEQ     SCAN4           ;BR IF OK
012414  123741  011040                          CMPB    TX.TERM,-(R1)   ;IS IT END OF DATA
012420  001447                                  BEQ     TESTDX          ;BR IF YES
012422  122742  000002                          CMPB    #002,-(R2)
012426  001005                                  BNE     2$
012430  010237  012436                          MOV     R2,1$
012434  104400                                  TYPE
```

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

                                        MSG0:   .ASCIZ  <15><12>/THERE WAS A RECEIVER ERROR. REGISTER (SEL 2) =/
(1)                                     MSG1:   .ASCIZ  <15><12>
(1)                                     MSG2:   .ASCIZ  <15><12>/RECEIVED DATA = /<15><12>
(1)                                     MSG3:   .ASCIZ  <15><12>/DATA SHOULD BE/<15><12>
(1)                                     MSG4:   .ASCII  <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./
(1)                                             .ASCIZ  <15><12>/WHEN CONNECTION COMPLETE, HIT CONTINUE SWITCH./<15><12>
(1)                                     MSG5:   .ASCIZ  <15><12>/PLEASE MAKE CONNECTION (DIAL NUMBER)./<15><12>
(1)                                             .EVEN
(1)                                     $AVR0:  0
                                        $AVR1:  0
                                        $AVR2:  0
                                        $AVR3:  0
                                        $AVR4:  0
                                        $AVR5:  0
                                        DELAY:  0
                                        BACK:   0
                                        STOP:   0
```

```
                                ;*****************************************************************
                                ;       TRANSMITTER INITIALIZATION SUBROUTINE
                                ;*****************************************************************

013126  005737  011024        STARTX: TST     SETTLE
013132  001004                        BNE     6$
013134  005737  013120                TST     DELAY
013140  001015                        BNE     5$
013142  000434                        BR      1$
013144  005037  013710        6$:     CLR     TEMP1           ;PREPARE FOR DELAY
013150  012737  000007  013712        MOV     #7,TEMP2
013156  062737  000001  013710        ADD     #1,TEMP1        ;INC DELAY
013164  001374                        BNE     .-6
013166  005337  013712                DEC     TEMP2
013172  001371                        BNE     .-14
013174  005037  013710        5$:     CLR     TEMP1           ;PREPARE FOR DELAY
013200  012737  000007  013712        MOV     #7,TEMP2
013206  062737  000001  013710        ADD     #1,TEMP1        ;INC DELAY
013214  001374                        BNE     .-6
013216  005337  013712                DEC     TEMP2
013222  001371                        BNE     .-14
013224  005037  013120                CLR     DELAY
013230  005037  011024                CLR     SETTLE
013234  032737  040000  011014  1$:   BIT     #HALF.DUPLEX,PARAM2     ;HALF DUPLEX?
013242  001440                        BEQ     4$              ;BR IF NO
013244  013746  000004                MOV     @#4,-(SP)       ;SAVE LOC 4
013250  013746  000006                MOV     @#6,-(SP)       ;SAVE LOC 6
013254  012737  013332  000004        MOV     #3$,@#4         ;SET UP TRAP CATCHER
013262  005037  000006                CLR     @#6             ;CLEAR VECT+2
013266  005737  013124                TST     STOP            ;FIRST TIME HERE?
013272  001407                        BEQ     8$              ;BR IF YES
013274  012737  177777  013124        MOV     #-1,STOP
013302  032777  000100  000406        BIT     #100,@DMBB      ;CARRIER UP?
013310  001374                        BNE     .-6             ;BR IF YES
013312  052777  000004  000376  8$:   BIS     #BIT2,@DMBB     ;SET RQTS IN DMBB
013320  032777  000040  000370  2$:   BIT     #BIT5,@DMBB     ;SPIN ON CTS
013326  001774                        BEQ     2$
013330  024646                        CMP     -(SP),-(SP)     ;ADJUST STACK
013332  022626                3$:     CMP     (SP)+,(SP)+     ;POP STACK
013334  012637  000006                MOV     (SP)+,@#6       ;RESTORE LOC 6
013340  012637  000004                MOV     (SP)+,@#4       ;RESTORE LOC 4
013344  013737  011022  011070  4$:   MOV     IXDA,XDA        ;SET UP XMIT DATA ADD
013352  042700  100000                BIC     #XFLG,STAT      ;CLEAR XFLG
013356  013764  011070  000006        MOV     XDA,CAR(R4)     ;LOAD CURRENT ADDRESS REG
013364  032737  040000  011014        BIT     #HALF.DUPLEX,PARAM2     ;HALF DUPLEX?
013372  001022                        BNE     7$              ;BR IF YES
013374  032700  000004                BIT     #XLB,MODE       ;XLB MODE?
013400  001417                        BEQ     7$              ;BR IF NO
013402  012737  177777  013722        MOV     #-1,TRNFLG      ;SET SOFTWARE FLAG
013410  012764  177777  000010        MOV     #-1,BCR(R4)
013416  052714  020000                BIS     #TIE,@RCSR
013422  013764  013714  000012        MOV     BARTMP,BAR(R4)
013430  000001                        WAIT
013432  005737  013720                TST     SNCFLG          ;HAS RECEIVER GOT FIRST CHAR?
013436  001375                        BNE     .-4             ;NO WAIT FOR IT
013440  013764  011070  000006  7$:   MOV     XDA,CAR(R4)     ;LOAD CURRENT ADDRESS REG
```

```
1029  013446  012764  177777  000010          MOV     #-1,BCR(R4)      ;LOAD BYTE COUNT REG
1030  013454  052714  020000                  BIS     #TIE,@RCSR       ;SET INTERUPT ENABLE
1031  013460  013764  013714  000012          MOV     BARTMP,BAR(R4)   ;LOAD BAR REG
1032  013466  000207                          RTS     PC
1033
1034  013470  042714  100000          XISR:   BIC     #TI,@RCSR        ;CLEAR XMIT DONE
1035  013474  032714  002000                  BIT     #NEM,@RCSR       ;NON-EXISTENT MEM ERROR?
1036  013500  001407                          BEQ     1$               ;BR IF NO
1037  013502  011402                          MOV     @RCSR,R2         ;SAVE CSR FOR TYPE OUT
1038  013504  005003                          CLR     R3
1039  013506  104010                          HLT     10               ;ERROR HLT
1040  013510  104400  014317                  TYPE    ,NONEX           ;TYPE ERROR MESS
1041  013514  000000                          HALT
1042  013516  000776                          BR      .-2              ;BR HALT
1043  013520  127737  175344  011040  1$:     CMPB    @XDA,TX.TERM     ;IS CHAR TERMINATION CHAR?
1044  013526  001033                          BNE     XISR1            ;BR IF NO
1045  013530  052700  100000                  BIS     #XFLG,STAT       ;SET XMIT DONE FLAG
1046  013534  042714  020000                  BIC     #TIE,@RCSR       ;CLEAR INTERUPT ENABLE
1047  013540  032737  040000  011014          BIT     #HALF.DUPLEX,PARAM2      ;HALF DUPLEX?
1048  013546  001422                          BEQ     3$               ;BR IF NO
1049  013550  013746  000004                  MOV     @#4,-(SP)        ;SAVE LOC 4
1050  013554  013746  000006                  MOV     @#6,-(SP)        ;SAVE LOC 6
1051  013560  012737  013602  000004          MOV     #2$,@#4          ;SET UP TRAP CATCHER
1052  013566  005037  000006                  CLR     @#6              ;CLEAR VECT+2
1053  013572  042777  000004  000116          BIC     #BIT2,@DMBB      ;CLEAR RQTS
1054  013600  024646                          CMP     -(SP),-(SP)      ;ADJUST STACK
1055  013602  022626                  2$:     CMP     (SP)+,(SP)+      ;POP STACK
1056  013604  012637  000006                  MOV     (SP)+,@#6        ;RESTORE LOC 6
1057  013610  012637  000004                  MOV     (SP)+,@#4        ;RESTORE LOC 4
1058  013614  000430                  3$:     BR      XISR2
1059  013616  032777  000100  175220  XISR1:  BIT     #100,@SWR        ;MONITOR XMIT DATA?
1060  013624  001406                          BEQ     NOXMON           ;BR IF NO
1061  013626  105777  175244                  TSTB    @TPS             ;TTY READY?
1062  013632  100003                          BPL     NOXMON           ;BR IF NO
1063  013634  117777  175230  175236          MOVB    @XDA,@TPB        ;TYPE CHAR
1064  013642  005237  011070          NOXMON: INC     XDA              ;INC TXBUF POINTER
1065  013646  013764  011070  000006          MOV     XDA,CAR(R4)      ;LOAD CURRENT ADDRESS REG
1066  013654  005737  013722                  TST     TRNFLG           ;IS THIS FIRST TIME?
1067  013660  001006                          BNE     XISR2            ;BR IF YES
1068  013662  012764  177777  000010          MOV     #-1,BCR(R4)      ;LOAD BYTE COUNT REG
1069  013670  013764  013714  000012          MOV     BARTMP,BAR(R4)   ;SET BAR BIT
1070  013676  005037  011032          XISR2:  CLR     TIME
1071  013702  005037  013722                  CLR     TRNFLG
1072  013706  000002                          RTI
1073  013710  000000                  TEMP1:  0
1074  013712  000000                  TEMP2:  0
1075  013714  000000                  BARTMP: 0
1076  013716  170502                  DMBB:   170502                   ;LINE STATUS REG IN DMBB
1077  013720  000000                  SNCFLG: 0
1078  013722  000000                  TRNFLG: 0
1079  013724     177                  FILL:   .BYTE   177
1080        013726                    .EVEN
```

```
1081                                    ;*****************************************************************
1082                                    ;       RECEIVER INITIALIZATION SUBROUTINE
1083                                    ;*****************************************************************
1084
1085  013726  032737  040000  011014  STARTR: BIT     #HALF.DUPLEX,PARAM2     ;HALF DUPLEX?
1086  013734  001010                         BNE     2$                      ;BR IF YES
1087  013736  032700  000004                 BIT     #XLB,MODE               ;XLB MODE?
1088  013742  001405                         BEQ     2$                      ;BR IF NO
1089  013744  005037  013710                 CLR     TEMP1                   ;START DELAY
1090  013750  005237  013710          1$:    INC     TEMP1
1091  013754  001375                         BNE     1$
1092  013756  042700  040000          2$:    BIC     #RFLG,STAT              ;CLEAR RFLG
1093  013762  013737  011020  011066         MOV     IRDA,RDA                ;SET UP RECEIVER DATA ADD
1094  013770  012737  001000  011064         MOV     #1000,RCC               ;SET UP BUFFER LIMIT
1095  013776  012737  177777  013720         MOV     #-1,SNCFLG              ;SET SOFTWARE FLAG
1096  014004  005037  011054                 CLR     ERCSR                   ;CLEAR ERROR RECORDS
1097  014010  005037  011056                 CLR     ERDBR
1098  014014  052714  004000                 BIS     #BIT11,@RCSR            ;MASTER CLEAR
1099  014020  053714  011012                 BIS     PARAM1,@RCSR            ;SET LINE NUMBER
1100  014024  053764  011014  000004         BIS     PARAM2,LPR(R4)          ;LINE PARAMETERS
1101  014032  052714  010100                 BIS     #RIE+SIE,@RCSR          ;SET INTERUPT ENABLES
1102  014036  000207                         RTS     PC
1103
1104  014040  032714  040000          RISR:  BIT     #SI,@RCSR               ;SILO OVERFLOW?
1105  014044  001407                         BEQ     1$                      ;BR IF NO
1106  014046  011402                         MOV     @RCSR,R2                ;SAVE CSR FOR TYPEOUT
1107  014050  005003                         CLR     R3
1108  014052  104010                         HLT     10                      ;ERROR HLT
1109  014054  104400  014270                 TYPE    ,SILO                   ;TYPE ERROR MESS
1110  014060  000000                         HALT
1111  014062  000776                         BR      .-2                     ;BR HALT
1112  014064  016401  000002          1$:    MOV     NRCR(R4),R1             ;PUT CHAR IN R1
1113  014070  042701  000200                 BIC     #200,R1                 ;STRIP A BIT
1114  014074  005701                         TST     R1                      ;VALID DATA?
1115  014076  100403                         BMI     4$                      ;BR IF YES
1116  014100  011402                         MOV     @RCSR,R2                ;SAVE CSR FOR TYPEOUT
1117  014102  005003                         CLR     R3
1118  014104  104010                         HLT     10                      ;ERROR HLT
1119  014106  032701  070000          4$:    BIT     #DO+FE+PE,R1            ;OVERRUN,FRAMING OR PARITY ERROR?
1120  014112  001404                         BEQ     3$                      ;BR IF NO
1121  014114  011437  011054                 MOV     @RCSR,ERCSR             ;SAVE CSR
1122  014120  010137  011056                 MOV     R1,ERDBR                ;SAVE CHAR
1123  014124  110177  174736          3$:    MOVB    R1,@RDA                 ;STORE CHAR IN BUFFER
1124  014130  032777  000040  174706         BIT     #BIT5,@SWR              ;MONITOR RECEIVE DATA?
1125  014136  001405                         BEQ     NORMON                  ;BR IF NO
1126  014140  105777  174732                 TSTB    @TPS                    ;TTY READY?
1127  014144  100002                         BPL     NORMON                  ;BR IF NO
1128  014146  110177  174726                 MOVB    R1,@TPB                 ;TYPE CHAR
1129  014152  005237  011066          NORMON: INC     RDA                     ;INC RECEIVER BUFFER POINTER
1130  014156  105077  174704                 CLRB    @RDA                    ;CLEAR NEXT LOCATION
1131  014162  005337  011064                 DEC     RCC                     ;DEC CHAR COUNT
1132  014166  001005                         BNE     1$                      ;BR IF BUFFER NOT FULL
1133  014170  000005                         RESET
1134  014172  104000                         HLT     0                       ;STOP THE SHOW,BUFFER OVERFLOWED!
1135  014174  104006                         HLT+6                           ;RECEIVER BUFFER FULL
1136  014176  000000                         HALT
```

```
1137  014200  000776                          BR      .-2                 ;BR HALT
1138  014202  123701  011041          1$:     CMPB    RX.TERM,R1          ;IS CHAR RCV TERMINATION CHAR
1139  014206  001004                          BNE     RISR1               ;BR IF NOT
1140  014210  042714  010100                  BIC     #RIE+SIE,@RCSR      ;CLEAR INTERUPT ENABLES
1141  014214  052700  040000                  BIS     #RFLG,STAT          ;SET RCV DONE FLAG
1142  014220  005037  011032          RISR1:  CLR     TIME
1143  014224  005037  013720                  CLR     SNDFLG              ;CLEAR FLAG
1144  014230  000002                          RTI
1145  014232  005015  051105  047522  MFULL:  .ASCIZ<15><12>/ERROR! RECEIVER BUFFER FULL/
      014270  005015  051105  047522  SILO:   .ASCIZ<15><12>/ERROR! SILO OVERFLOW/
      014317     015  047012  047117  NONEX:  .ASCIZ<15><12>/NON EXISTENT MEMORY ERROR/
      014353     015  050012  042514  LINES:  .ASCIZ<15><12>/PLEASE SELECT ONLY ONE LINE AT A TIME(PARAM3)/
                014434                  .EVEN
                000001                  .END
```

```
BA       011004  602#  661
BACK     013122  735*  763*  798*  878*  909  970#
BAR    = 000012  596#  1024*  1031*  1069*
BARTMP   013714  677*  1024  1031  1069  1075#
BCR    = 000010  596#  1022*  1029*  1068*
BIT0   = 000001  596#
BIT1   = 000002  596#
BIT10  = 002000  596#
BIT11  = 004000  596#  1098
BIT12  = 010000  596#
BIT13  = 020000  596#  631#  920  947
BIT14  = 040000  596#
BIT15  = 100000  596#
BIT2   = 000004  596#  1007  1053
BIT3   = 000010  596#
BIT4   = 000020  596#
BIT5   = 000040  596#  1008  1124
BIT6   = 000100  596#
BIT7   = 000200  596#
BIT8   = 000400  596#
BIT9   = 001000  596#
B2016    011030  612#  923
CAR    = 000006  596#  1016*  1028*  1065*
DELAY    013120  687*  907*  969#  979  994*
DERR     012532  948  957#
DH11     011000  601#
DISPLA   011046  622#
DMBB     013716  1005  1007*  1008  1053*  1076#
DO     = 040000  596#  1119
DSFLG  = 020000  596#  630#
DSSTAT   011060  637#
ENTER    012272  891  900#
EOP      012204  736  764  799  879  886#
ERCSR    011054  635#  1096*  1121*
ERDBR    011056  636#  918  1097*  1122*
FE     = 020000  596#  1119
FILL     013724  1079#
FLAG     011042  620#
FULL.D=  000001  649#
GO       011230  686#
HALF.D=  040000  596#  838  854  859  996  1017  1047  1085
ILB    = 000010  596#  695
IRDA     011020  608#  802  927  950  1093
IXDA     011022  609#  803  926  954  956  1014
KBDIN  = 104416  596#  717  748  780  837
LINES    014353  1145#
LOOP   = 000020  596#  733  761  796  876
LPR    = 000004  596#  664*  1100*
MC     = 004000  596#  662
MFULL    014232  1145#
MSG0     012544  922  963#
MSG1     012625  925  963#
MSG2     012630  949  963#
MSG3     012655  953  963#
MSG4     012700  963#
MSG5     013032  963#
```

```
NEM    = 002000   596#  1035
NODAT  = 000200   596#   729    792    873
NONEX    014317  1040   1145#
NORMON   014152  1125   1127   1129#
NOXMON   013642  1060   1062   1064#
NRCR   = 000002   596#  1112
OWI    = 000002   596#   692
OWO    = 000001   596#   689
PARAM1   011012   605#   663    665    671   1099
PARAM2   011014   606#   664    838    854    859    996   1017   1047   1085   1100
PARAM3   011016   607#
PE     = 010000   596#  1119
PRIOR    011010   604#   658    660
PRTY0  = 000000   596#
PRTY1  = 000040   596#
PRTY2  = 000100   596#
PRTY3  = 000140   596#
PRTY4  = 000200   596#
PRTY5  = 000240   596#
PRTY6  = 000300   596#
PRTY7  = 000340   596#   887
QTPIE    012342   888*   889*   908    910#
RCC      011064   640#  1094*  1131*
RDA      011066   641#  1093*  1123*  1129*  1130*
RDAX     012512   950*   952#
RFLG   = 040000   629#   719    732    783    795    845    871   1092   1141
RIE    = 000100   596#  1101   1140
RISR     014040   657   1104#
RISR1    014220  1139   1142#
RIV      011006   603#   656
RWAIT  = 104410   596#
RX.TER   011041   618#  1138
SAVR0    013104   892*   901    963#
SAVR1    013106   893*   902    964#
SAVR2    013110   894*   903    965#
SAVR3    013112   895*   904    966#
SAVR4    013114   896*   905    967#
SAVR5    013116   897*   906    968#
SCAN4    012410   928#   929    942    945
SCAN5    012464   946#
SETTLE   011024   610#   977    995*
SI     = 040000   596#  1104
SIE    = 010000   596#  1101   1140
SILO     014270  1109   1145#
SNCFLG   013720  1026   1077#  1095*  1143*
SRCSR    011052   634#
SSR    = 000016   596#
START    011102   615    653#
STARTR   013726   718    781    840    857   1085#
STARTX   013126   749    812    841    977#
STOP     013124   688*   971#  1002   1004*
STPS   = 104414   596#   887
SWR      011044   621#   654    729    733    761    792    796    800    873    876    920    947   1059
                 1124
SW12   = 010000   596#
SXCSR    011050   633#
```

```
TEMP1   013710          982*   984*   988*   990*  1073*  1089*  1090*
TEMP2   013712          983*   986*   989*   992*  1074*
TESTD   012344           731    794    875    918*
TESTDX  012540           931    937    940    960*
TI    = 100000           596*  1034
TIE   = 020000           596*   889    890   1023   1030   1046
TIME    011032           613*   686*   721    726*   750*   753    758*   782*   785    790*   811*   815    820*
                         842*   847    852*   863    868*   872*  1070*  1142*
TKB     011074           645*
TKS     011072           644*
TPB     011100           647*  1063*  1128*
TPS     011076           646*  1061   1126
TRNFLG  013722          1021*  1066   1071*  1078*
TSTDAT  012400           919    921    926*
TX.TER  011040           616*   930   1043
TYPE  = 104400           596*   922    925    935    949    951    953    955   1040   1109
XCC     011062           639*
XCSR  = 000000           596*   724    756    788    818    850    866    888    890*   908*
XDA     011070           642*   804*  1014*  1016   1028   1043   1063   1064*  1065
XFLG  = 100000           628*   751    760    813    822    843    856    861    870   1015   1045
XISR    013470           659   1034*
XISR1   013616          1044   1059*
XISR2   013676          1058   1067   1070*
XL8   = 000004           596*   698   1019   1087
XWAIT = 104412           596*
$ILB    011524           697    780*   823
$OWI    011320           694    717*   737
$OWO    011426           691    748*   765
$XLB    011754           700    837*   877    878
.     = 014434           600*   668    702    806    954*   985    987    991    993   1006   1027   1042   1080*
                        1111   1137   1145*

. ABS.   014434     000

 ERRORS DETECTED:  0

 DSKZ:CZDHLC,DSKZ:CZDHLC.SEQ=DSKZ:ITEP1.MAC,DSKZ:CZDHLC.P11
 RUN-TIME: 3 4 .2 SECONDS
 RUN-TIME RATIO: 15/8=1.7
 CORE USED:  16K  (31 PAGES)

 DOCUMENT PAGES:  27
```

C03