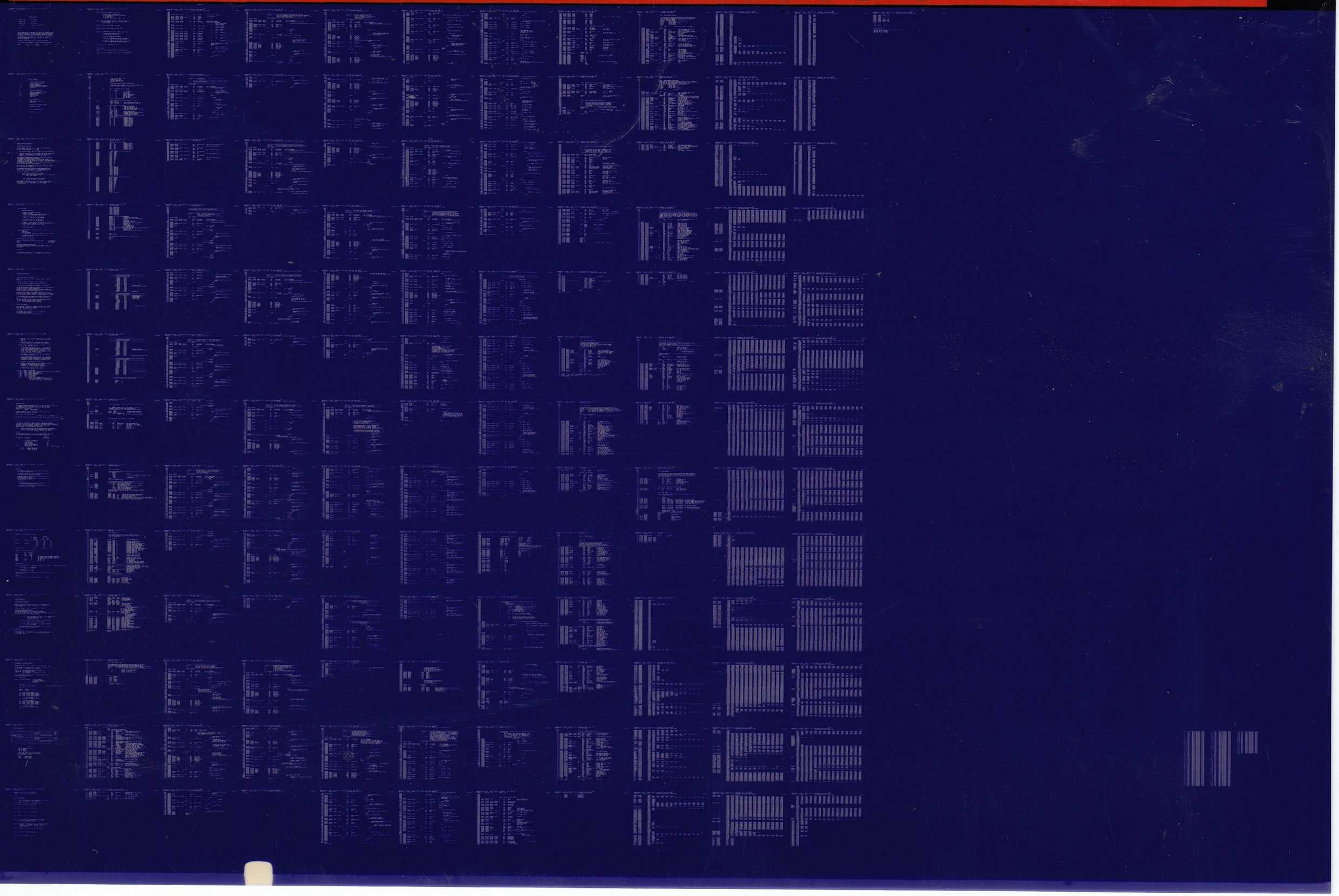
DLV11J

DLV11-J TEST
CNDLAA0

AH-T470A-MC
FICHE 1 OF 1



.REM %

IDENTIFICATION

PRODUCT CODE: AC-T469A-MC

PRODUCT NAME: CNDLAA0 DLV11-J TEST

PRODUCT DATE: DEC, 1982

MAINTAINER: DIAGNOSTIC SERVICES/ISS

AUTHOR: DIAG/ISS

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION.

1.1 PROGRAM PURPOSE (ABSTRACT).

THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE
DLV11-J SERIAL INTERFACE.
TESTING IS DONE IN TWO DISTINCT PHASES:

1. ALL SELECTED CHANNELS PER DLV11-J MODULE ARE TESTED INDIVIDUALLY

2. THE DLV11-J MODULE IS TESTED AS A WHOLE FOR CHANNEL INTERACTION
   PROBLEMS. THIS DIAGNOSTIC IS DESIGNED TO TEST AND DETECT
   ERRORS TO THE LOGIC LEVEL (NOT TO THE CHIP LEVEL).

THIS DIAGNOSTIC OPERATES UP TO 2 DLV11-J SERIAL LINE INTERFACES
CONFIGURED AT CONSECUTIVE BASE ADDRESSES.
THE PROGRAM WILL DO AUTO-SIZING IF THE DEVICE MAP '$DEVM' = 0.
ALTERNATELY, THE OPERATOR CAN SELECT OR DESELECT INDIVIDUAL MODULES
& CHANNELS BY SETTING THE PROPER BITS IN '$DEVM' (SEE PROGRAM OPTIONS SEC. 2.4)
BEFORE RUNNING THE PROGRAM.
BY THIS METHOD, THE PROGRAM WILL BYPASS AUTO-SIZING AND TEST ONLY
THOSE MODULES/CHANNELS IT FINDS IN $DEVM.

IN EITHER CASE, THE PROGRAM WILL PRINT OUT ALL MODULES & CHANNELS
TO BE TESTED BEFORE PROCEEDING.

THE OPERATOR MUST INSTALL DATA WRAP AROUND CONNECTORS TO DO
DATA TESTING. TO BYPASS DATA TESTS, THE OPERATOR MUST MODIFY
'$USWR' (USER SWITCH REGISTER) , SEE PROGRAM OPTIONS SEC. 2.4)

THE DEFAULT ADDRESSES & VECTORS ARE AS FOLLOWS:

```
        177560 -CONSOLE INTERFACE DEVICE ADDRESS
        176500 -FIRST SERIAL CHANNEL ADDRESS OF UP TO 8  CONSECUTIVE
                SERIAL LINE DEVICES.


           60 - VECTOR FOR CONSOLE DEVICE INTERFACE.
          300 - VECTOR FOR FIRST OF 16 DEVICES.
```

THIS PROGRAM IS DESIGNED TO RUN ON ANY Q-BUS PDP-11 WITH 4K OF
MEMORY AND A DLV11-J (Q-BUS) MODULE. IT CAN RUN UNDER XXDP &
APT MONITORS, AND ON PROCESSORS WITH NO HARDWARE
SWITCH REGISTER.

1.2 SYSTEM REQUIREMENTS.

HARDWARE REQUIREMENTS:

ANY Q-BUS PDP-11 FAMILY PROCESSOR
4K MEMORY - MINIMUM
A SPECIAL DATA WRAP AROUND CONNECTOR OR EQUIVALENT
(REQ'D IF DATA WRAP AROUND TESTS DESIRED)

IF CHANNEL 3 IS CONFIGURED AS THE CONSOLE:
TESTS 6-12, 14-17, 21, 22 ARE BYPASSED.

IF DATA WRAP AROUND TESTS ARE BYPASSED:
TESTS 7-12, 14-17, 21, 22 ARE BYPASSED.

IF CHANNEL 3 IS CONFIGURED AS THE CONSOLE & A VT-100 IS
THE CONSOLE DEVICE, SWITCH THE VT-100 TO "JUMP SCROLL" MODE
TO AVOID LOOSING CHARACTERS ON THE SCREEN.

SOFTWARE REQUIREMENTS:

THIS DIAGNOSTIC IS DESIGNED TO RUN IN ANY OF THE
FOLLOWING WAYS:
STAND ALONE
WITH APT MONITOR
WITH XXDP MONITOR (CHAINABLE IF RENAMED TO .BIC EXTENSION)

THIS DIAGNOSTIC IS NOT DESIGNED TO RUN WITH THE
DIAGNOSTIC SUPERVISOR.

1.3 RELATED DOCUMENTS AND STANDARDS.

DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS	175-003-009-02
APT	MD-11-DZZMA
SYSMAC	MD-11-DZQAC

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES.
NO SPECIAL DIAGNOSTICS ARE REQUIRED TO RUN BEFORE THIS, BUT
THE PROCESSOR, MEMORY, AND BUS ARE ASSUMED TO BE FULLY
OPERATIONAL.

1.5 ASSUMPTIONS.

THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR HAS INITIALIZED
LOCATION '$USWR' AND '$DEVM' TO THE PROPER VALUES. (SEE SEC. 2.4)

2.0 OPERATING INSTRUCTIONS.

2.1 LOADING AND STARTING PROCEDURES.

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED MEDIA.

THERE ARE 2 STARTING LOCATIONS FOR THIS DIAGNOSTIC:

1. LOCATION 200 FOR ALL NORMAL TESTING STARTS & RESTARTS.

2. LOCATION 400 FOR INTERRUPT VECTOR DEBUG... OFFLINE ONLY.

THE 400 START SHOULD BE USED ONLY TO DEBUG FAULTY VECTOR ADDRESSES PUT OUT BY THE DLV11-J DURING INTERRUPTS.
STARTING AT 400 WILL CAUSE THE ENTIRE VECTOR AREA, EXCEPT THAT NEEDED BY SYSMAC, TO BE OVERWRITTEN & TO POINT TO A COMMON SERVICE ROUTINE TO ENABLE LOOPING ON THE PROBLEM.

AS SOON AS TESTING STARTS, THE OPERATOR CAN CHANGE THE SWITCH REGISTER ONLY BY A 'BREAK' & MANUALLY LOADING LOCATION 176 (SWREG) WITH THE DESIRED CONTENTS (SEE SEC. 2.3) THEN DOING A 'P' TO PROCEED.

IT IS IMPORTANT THAT WHEN GOING BACK TO NORMAL TESTING A FRESH LOAD OF THE PROGRAM BE PERFORMED TO RESET THE VECTOR AREA.

THE USER CAN SELECT A SPECIFIC TEST TO BE EXECUTED BY SETTING BIT 8 IN SWREG AND THE TEST NUMBER (IN OCTAL) IN BITS <7:0>.
(NOTE: ALL TESTS PREVIOUS TO THE SELECTED ONE ARE EXECUTED WITHOUT ITERATIONS.)

2.2 SPECIAL ENVIRONMENTS.

THIS DIAGNOSTIC FOLLOWS THE STANDARD PROCEDURE FOR RUNNING UDER APT,XXDP MONITORS, AS DESCRIBED IN THEIR RESPECTIVE PROCEDURES MANUAL AND SYSMAC PACKAGE.

2.3 OPERATIONAL SWITCH SETTINGS

THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED FOR ALL OPERATIONAL SWITCH SETTINGS.
THIS CAN BE ACCOMPLISHED IN THE FOLLOWING WAY:

1) TYPE CONTROL G <^G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE:' SWR=XXXXXX NEW=' (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:

   A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED). LEADING ZEROS NEED NOT BE TYPED, AND IF MORE THAN 6 DIGITS ARE TYPED THE LAST 6 WILL BE USED. IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 3.

   C) IF THE INPUT CHARACTER IS NOT ONE OF THE CHARACTERS MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED FOLLOWED BY A CARRAGE RETURN AND A LINE FEED SEQUENCE THEN PROCEED FROM STEP 3 (ERASING ALL PREVIOUS INPUT).

4) THE DIAGNOSTIC WILL CONTINUE ON TYPING <CR>.

NOTE: BECAUSE OF FREQUENT BUS RESETS IN THE PROGRAM, IT MAY BE NECESSARY TO DO 'CONTROL-G' SEVERAL TIMES.
ALTERNATELY, A 'BREAK' & MANUALLY LOADING LOC. 176, FOLLOWED BY A 'P' TO PROCEED WILL ALSO WORK.

SOFTWARE SWITCH REGISTER OPTIONS (SWREG)

```
BIT 15 SET = 100000 = HALT ON ERROR
    14 SET =  40000 = LOOP ON TEST  (TO BE USED ONLY WHILE TESTING IN PROGRESS)
    13 SET =  20000 = INHIBIT ERROR TYPEOUTS
    12 SET =  10000 = ENABLE PERFORMANCE REPORTS
    11 SET =   4000 = INHIBIT ITERATIONS
    10 SET =   2000 = BELL ON ERROR
     9 SET =   1000 = LOOP ON ERROR
     8 SET =    400 = LOOP ON TEST IN SWR<7:0>
     7:0   =        NUMBER OF TEST TO LOOP ON (USED WITH BIT 8)
                      (ALL TESTS PREVIOIUS TO THE SELECTED TEST
                      ARE EXECUTED FIRST WITH 1 ITERATION ONLY)
```

2.4 PROGRAM OPTIONS.

THIS PROGRAM REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT '$BASE') AND ITS INTERRUPT VECTOR (STORED AT '$VECT1'). IT WILL BE ABLE TO ADDRESS ANY DLV11-J STARTING AT THE SPECIFIED BASE ADDRESS THRU 2 CONSECUTIVE MODULES.
EXAMPLES: $BASE: 176500 (DEFAULT)
$VECT1: 300 (DEFAULT)

LOCATION '$DEVM' IS USED AS A BIT MAP TO INDICATE WHICH UNIT NUMBERS ARE PRESENT AND WILL BE TESTED.

$DEVM (BITS 15 THRU 8 NOT USED)

								MODULE 2				MODULE 1				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	BIT
								3	2	1	0	3	2	1	0	CHANNEL

A BIT MAP CAN BE ENTERED AT '$DEVM' PRIOR TO STARTING THE PROGRAM. IF '$DEVM' IS LEFT AT ITS DEFAULT VALUE (0), THE PROGRAM WILL DO AUTO-SIZING BEGINNING AT THE DEVICE ADDRESS IN '$BASE' & WILL SETUP '$DEVM' ACCORDINGLY. IN EITHER CASE, THE PROGRAM WILL TYPE OUT A LIST OF ALL THE MODULES/CHANNELS TO BE TESTED BEFORE PROCEEDING.

NOTE: SIZING IS PERFORMED ONLY ONCE AT THE BEGINNING OF THE PROGRAM. IF $DEVM IS TO BE CHANGED, THE PROGRAM MUST BE RESTARTED AT 200 FOR IT TO BE EFFECTIVE. OTHERWISE, $DEVM WILL BE IGNORED.

OPTIONS

LOCATION $USWR CONTAINS ALL THE USER SELECTABLE OPTIONS. THE VALUES IN THIS WORD MUST CONFORM TO THE ACTUAL BOARD CONFIGURATION.

BIT POSITION	DEFINITION	$USWR DEFAULT VALUE	
0	# OF DATA BITS TRANSMITTED 0 = 7 BITS, 1 = 8 BITS	1	= 8 BITS
1	PARITY ENABLED	0	= NO
2	EVEN ODD PARITY	0	= ODD
3	BREAK DETECTION ENABLED	1	= YES
4	RUN DATA WRAP AROUND TESTS	1	= YES
<11:9>	CONSOLE DEVICE	1	= YES = CONSOLE ON MODULE 1, CHAN. 3

<11:9> = 0 = CONSOLE NOT ON DLV11-J
<11:9> = 1 = CONSOLE ON MODULE 1
<11:9> = 2 = CONSOLE ON MODULE 2

IMPORTANT:

1. FOR DIAGNOSTIC PURPOSES, ALL CHANNELS MUST BE CONFIGURED
   TO THE SAME BIT/WORD LENGTH.

2. THE 'SIZING' ROUTINE WILL SET THE APPROPRIATE CHAN 3 BIT
   IN $DEVM IF BITS <11:9> ARE SET IN $USWR.

3. IF RUNNING UNDER APT & THE CONSOLE IS ON THE DLV11-J, THE
   CONSOLE MUST NOT BE TESTED.
   THIS IS BECAUSE APT SENDS 'BREAKS' TO THE CONSOLE WHICH
   INTEREFERES WITH THE TESTS.

   FOR THIS CASE:

   A. ALLOW THE $DEVM TO STAY EITHER AT ITS DEFAULT OF 0
      OR CLEAR THE APPROPRIATE CHAN. 3 BIT.

   B. CLEAR BITS <11:9> IN $USWR SO THE CONSOLE BIT WILL NOT
      BE SET INTO $DEVM FOR TESTING.

SUMMARY OF DEVICE ADDRESSES & VECTORS.

			DEV ADDRESS (RCSR)	VECTORS REC	VECTORS XMIT
MODULE #1	CHANNEL	0	$BASE+00	$VECT1+00	$VECT1+4
		1	+10	+10	+14
		2	+20	+20	+24
		3	+30	+30	+34
MODULE #2	CHANNEL	0	+40	+40	+44
		1	+50	+50	+54
		2	+60	+60	+64
		3	+70	+70	+74
CONSOLE DEVICE		3	$TKS	TKVEC=60	TPVEC=64

SUMMARY OF USER LOCATIONS & DEFAULTS.

	LOC	DEFAULT	
$BASE	1250	176500	
$VECT1	1244	300	
$TKS	1144	177560	FOR CONSOLE DEVICE ON MODULE UNDER TEST
TKVEC		60	FOR CONSOLE DEVICE ON MODULE UNDER TEST
TPVEC		64	FOR CONSOLE DEVICE ON MODULE UNDER TEST
$USWR	1220	31	SEE SEC. 2.4
$DEVM	1252	17	SEE SEC. 2.4
SWREG	176	0	SEE SEC. 2.3

2.5 EXECUTION TIMES.

EXECUTION TIMES FOR AN LSI-11 PROCESSOR WITH ONE DLV11-J MODULE AT SHIPMENT CONFIGURATION:

CH. 0,1,2 AT 9600 BAUD.
CH. 3 (CONSOLE) AT 300 BAUD.

ARE: FIRST PASS- 30 SEC
ADDITIONAL PASSES 90 SEC

THE TEST TIME IS BAUD RATE DEPENDENT; HIGHER BAUD RATES RESULT IN SHORTER PASS TIMES.

2.6 POWER FAIL.

AUTO START FROM POWER FAIL IS NOT IMPLEMENTED IN THIS PROGRAM.

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROCEDURE.

SINCE THIS DIAGNOSTIC WAS DESIGNED TO FIT IN 4K OF MEMORY THE ERROR TYPEOUT IS VERY BRIEF. THE FORMAT OF THE ERROR TYPEOUT IS AS FOLLOWS:

TEST#_____,ERROR#_____,PC=_____,ADDRESS=______,VECTOR=_____

WHERE ALL VALUES TYPED ARE OCTAL.
THE ADDRESS AND VECTOR REFER TO THE FAILING CHANNEL.
FOR FURTHER INFORMATION THE LISTING MUST BE CONSULTED.
BITS 15,13,10 AND 9 OF THE SWITCH REGISTER (SWREG) CONTROL THE SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.

BIT 15 SET: CAUSES THE PROGRAM TO HALT IN THE ERROR ROUTINE. IF THE PROGRAM IS CONTINUED, IT WILL PROCEED FROM WHERE IT HALTED.

BIT 13 SET: DISABLES THE PRINTING OF THE ERROR MESSAGE.

BIT 10 SET: CAUSES THE BELL TO RING ON ERROR.

BIT 9 SET: CAUSES THE DIAGNOSTIC TO LOOP FROM BEGINNING OF TEST TO ERROR.

THE ERROR ROUTINE SUPPORTS THE CONTROL G <^G> FUNCTION.
REFER TO SECTION 2.3 FOR DETAILS.

3.2 ERROR HALTS.

THE ONLY HALT IN THIS DIAGNOSTIC IS IN THE ERROR ROUTINE, AND IS EXECUTED ONLY IF BIT 15 OF THE SWITCH REGISTER (SWREG) IS SET WHEN AN ERROR OCCURS.

4.0 PERFORMANCE AND PROGRESS REPORTS.

4.1 PERFORMANCE REPORTS. (BIT 12 SET IN THE SWITCH REGISTER 'SWREG')

AS EACH CHANNEL (4 CHANNELS/DLV11-J) COMPLETES ONE PASS OF THE DIAGNOSTIC THE FOLLOWING ITEMS ARE TYPED:

'CSR:_____' : THE BASE ADDRESS OF THE LINE UNDER TEST
'VECTOR:__' : THE ASSOCIATED VECTOR
'ERRORS:__' : THE TOTAL NUMBER OF ERRORS ON THIS DEVICE ON THIS PASS.

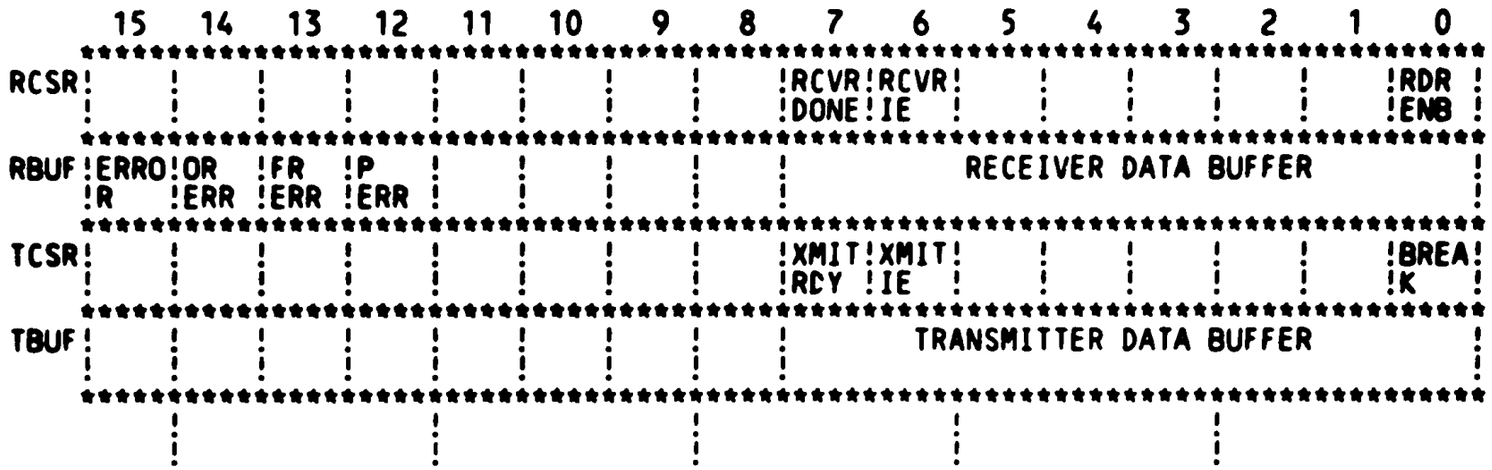
AFTER ALL MODULES & CHANNELS TO BE TESTED HAVE BEEN EXERCISED, AN END PASS STATEMENT IS TYPED:

'END PASS#_____.''

EXAMPLE OF PRINTOUT ASSUMING: 2 DLV11-J MODULES
ALL CHANNELS ENABLED
SEPERATE CONSOLE DEVICE
NO ERRORS

```
CNDLAA0 DLV11-J TEST

SWR= 000000     NEW= 10000<CR>                  (ENABLE PERFORMANCE REPORTS)

WILL TEST:

MODULE 1  CHANNEL 0  1  2  3
MODULE 2  CHANNEL 0  1  2  3

        ** PHASE 1 SUMMARY **
CSR: 176500, VECTOR: 000300, ERRORS: 0
CSR: 176510, VECTOR: 000310, ERRORS: 0
CSR: 176520, VECTOR: 000320, ERRORS: 0
CSR: 176530, VECTOR: 000330, ERRORS: 0

        ** PHASE 2 SUMMARY **
CSR: 176500, VECTOR: 000300, ERRORS: 0

        ** PHASE 1 SUMMARY **
CSR: 176540, VECTOR: 000340, ERRORS: 0
CSR: 176550, VECTOR: 000350, ERRORS: 0
CSR: 176560, VECTOR: 000360, ERRORS: 0
CSR: 176570, VECTOR: 000370, ERRORS: 0

        ** PHASE 2 SUMMARY **
CSR: 176540, VECTOR: 000340, ERRORS: 0

END PASS #      1
```

5.0 DEVICE REGISTERS.



NOTES:

1. RCSR AT BASE ADDRESS ($BASE)
   RBUF AT $BASE+2
   TCSR AT $BASE+4
   TBUF AT $BASE+6

2. BLANK BOXES INDICATE UNUSED AND RESERVED BIT POSITIONS. SEE THE LISTING FOR AN EXPLANATION OF THE BITS.

3. ORERR = OVERRUN ERROR
   FRERR = FRAMING ERROR
   PERR = PARITY ERROR

6.0 SUMMARY OF TESTS AND SPECIAL SUBROUTINES.

PHASE 1 TESTS

TEST 1 ADDRESSABILITY

THIS TEST VERIFIES THAT ALL 4 REGISTERS OF THE CHANNEL UNDER TEST RESPOND TO THEIR ADDRESSES.

THE FOLLOWING 3 TESTS TEST ALL 'READ WRITE' BITS

TEST 2 BREAK - TCSR 0 SET, CLEAR, RESET

TEST 3 XMITIE - TCSR 6 SET, CLEAR, RESET

TEST 4 RCVRIE - RCSR 6 SET, CLEAR, RESET

TEST 5 XMITRDY - TCSR 7 - IS SET BY INIT

TEST 6 XMIT RDY - TCSR 7 - CLEARS WHEN TBUF IS LOADED WITH A CHARACTER AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.

TEST 7 OUTPUTTING A CHAR FROM TBUF (WITH WRAP AROUND CONNECTED) RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME AND THAT RESET CLEARS THE BIT.

TEST 10 RCVRDONE IS CLEARED BY SETTING READER ENABLE

TEST 11 RCVRDONE IS CLEARED BY READING RBUF

TEST 12 OVERRUN & ERROR BIT - RBUF 14

TEST 13 TRANSMITTER INTERRUPT LOGIC TEST

LOGICALLLY THIS IS 4 SEPARATE TESTS
A) DOES TRANSMITTER INTERRUPT LOGIC WORK
B) AT PRIORITY OF 0
C) AND ONLY ONCE
D) BUT NOT WITH INTERRUPT ENABLE CLEAR

TEST 14 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL OF THE RECEIVER SIDE OF THE INTERRUPT LOGIC IN CHARACTER MODE.

TEST 15 TEST DATA WRAP AROUND: FLAG MODE.

TEST 16 TEST DATA WRAP AROUND: INTERRUPT MODE.

TEST 17 TEST BREAK DETECTION LOGIC TRANSMIT KNOWN CHAR

A) TRANSMIT KNOWN CHAR WITH BREAK SET AND COMPARE RECEIVED WITH 0

B) TEST FOR FRAMING ERROR ON BREAK

C) IF PARITY IS ENABLED AND ODD PARITY IS SELECTED, CHECK TO BE SURE PARITY ERROR WAS GENERATED

D) IF PARITY IS ENABLED AND EVEN PARITY IS SELECTED, CHECK TO BE SURE NO PARITY ERROR OCCURRED

TEST 20 NOT A TEST - SEND BACK TO LOOP

PHASE 2 TESTS

TEST 21 TEST THAT CHANNELS INTERRUPT AT ASSIGNED PRIORITY

TEST 22 TEST DATA TRANSFERS WITH ALL ACTIVE LINES INTERRUPTING.

%

```
                                .TITLE  CNDLAA0 DLV11-J TEST
                                ;*COPYRIGHT (C) 1982
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY GREGORY GLEZMAN & GARY PAPAZIAN
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*

                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
                                ;*****  THE FOLLOWING ODT START ADDRESS FOR SBC 11/21 IS ADDED
170000                          ODTST=  170000
                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
000006                          SP=     %6              ;;STACK POINTER
000007                          PC=     %7              ;;PROGRAM COUNTER
```

```
(1)                             ;*PRIORITY LEVEL DEFINITIONS
(1)             000000          PR0=    0               ;;PRIORITY LEVEL 0
(1)             000040          PR1=    40              ;;PRIORITY LEVEL 1
(1)             000100          PR2=    100             ;;PRIORITY LEVEL 2
(1)             000140          PR3=    140             ;;PRIORITY LEVEL 3
(1)             000200          PR4=    200             ;;PRIORITY LEVEL 4
(1)             000240          PR5=    240             ;;PRIORITY LEVEL 5
(1)             000300          PR6=    300             ;;PRIORITY LEVEL 6
(1)             000340          PR7=    340             ;;PRIORITY LEVEL 7
(1)
(1)                             ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
(1)             100000          SW15=   100000
(1)             040000          SW14=   40000
(1)             020000          SW13=   20000
(1)             010000          SW12=   10000
(1)             004000          SW11=   4000
(1)             002000          SW10=   2000
(1)             001000          SW09=   1000
(1)             000400          SW08=   400
(1)             000200          SW07=   200
(1)             000100          SW06=   100
(1)             000040          SW05=   40
(1)             000020          SW04=   20
(1)             000010          SW03=   10
(1)             000004          SW02=   4
(1)             000002          SW01=   2
(1)             000001          SW00=   1
(1)                             .EQUIV  SW09,SW9
(1)                             .EQUIV  SW08,SW8
(1)                             .EQUIV  SW07,SW7
(1)                             .EQUIV  SW06,SW6
(1)                             .EQUIV  SW05,SW5
(1)                             .EQUIV  SW04,SW4
(1)                             .EQUIV  SW03,SW3
(1)                             .EQUIV  SW02,SW2
(1)                             .EQUIV  SW01,SW1
(1)                             .EQUIV  SW00,SW0
(1)
(1)                             ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)             100000          BIT15=  100000
(1)             040000          BIT14=  40000
(1)             020000          BIT13=  20000
(1)             010000          BIT12=  10000
(1)             004000          BIT11=  4000
(1)             002000          BIT10=  2000
(1)             001000          BIT09=  1000
(1)             000400          BIT08=  400
(1)             000200          BIT07=  200
(1)             000100          BIT06=  100
(1)             000040          BIT05=  40
(1)             000020          BIT04=  20
(1)             000010          BIT03=  10
(1)             000004          BIT02=  4
(1)             000002          BIT01=  2
(1)             000001          BIT00=  1
(1)                             .EQUIV  BIT09,BIT9
```

```
(1)                             .EQUIV  BIT08,BIT8
(1)                             .EQUIV  BIT07,BIT7
(1)                             .EQUIV  BIT06,BIT6
(1)                             .EQUIV  BIT05,BIT5
(1)                             .EQUIV  BIT04,BIT4
(1)                             .EQUIV  BIT03,BIT3
(1)                             .EQUIV  BIT02,BIT2
(1)                             .EQUIV  BIT01,BIT1
(1)                             .EQUIV  BIT00,BIT0
(1)
(1)                             ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)             000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
(1)             000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)             000014          TBITVEC=14              ;;"T" BIT
(1)             000014          TRTVEC= 14              ;;TRACE TRAP
(1)             000014          BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
(1)             000020          IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)             000024          PWRVEC= 24              ;;POWER FAIL
(1)             000030          EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
(1)             000034          TRAPVEC=34              ;;"TRAP" TRAP
(1)             000060          TKVEC=  60              ;;TTY KEYBOARD VECTOR
(1)             000064          TPVEC=  64              ;;TTY PRINTER VECTOR
(1)                             ;*****  THE FOLLOWING BREAK VECTOR AND LINE CLOCK VECTOR ARE INCLUDED
(1)             000100          LKVEC=  100             ;;LINE CLOCK VECTOR
(1)             000140          BRKVEC= 140             ;;BREAK VECTOR
(1)             000240          PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
 8725
 8726
 8727
 8728           000001          ADRS=   R1
 8729           176500          DLADDR= 176500
 8730
 8731                           ; THE FOLLOWING DEFINITIONS APPLY TO THE GLOBAL SUBS
 8732           177777          SET= -1
 8733           000000          CLR=  0
 8734           000001          TRUE= 1
```

```
                        ;;*****************************************************************
                        ;               RCSR REGISTER BIT NAMES
                        ;;*****************************************************************
                        ;               UNUSED          BIT15
                        ;               UNUSED          BIT14
                        ;               UNUSED          BIT13
                        ;               UNUSED          BIT12
                        ;               UNUSED          BIT11
                        ;               UNUSED          BIT10
                        ;               UNUSED          BIT09
                        ;               UNUSED          BIT08
000200                  RCVRDONE=       BIT07           ; RECEIVER DONE
000100                  RCVRIE=         BIT06           ; RECEIVER INTERRUPT ENABLE
                        ;               UNUSED          BIT05
                        ;               UNUSED          BIT04
                        ;               UNUSED          BIT03
                        ;               UNUSED          BIT02
                        ;               UNUSED          BIT01
000001                  RDRRUN=         BIT00           ; READER RUN

                        ;;*****************************************************************
                        ;               RBUF REGISTER BIT NAMES
                        ;;*****************************************************************
100000                  ERROR=          BIT15           ; ERROR INDICATOR
040000                  ORERR=          BIT14           ; OVERRUN ERROR
020000                  FRERR=          BIT13           ; FRAMING ERROR
010000                  PERR=           BIT12           ; PARITY ERROR
                        ;               UNUSED          BIT11
                        ;               UNUSED          BIT10
                        ;               UNUSED          BIT09
                        ;               UNUSED          BIT08
000200                  RDATA7=         BIT07           ; \
000100                  RDATA6=         BIT06           ;  !
000040                  RDATA5=         BIT05           ;  !
000020                  RDATA4=         BIT04           ;   \   RECEIVED DATA BITS
000010                  RDATA3=         BIT03           ;   /
000004                  RDATA2=         BIT02           ;  !
000002                  RDATA1=         BIT01           ;  !
000001                  RDATA0=         BIT00           ; /
```

```
                        ;;*****************************************************************
                        ; TCSR REGISTER BIT NAMES
                        ;;*****************************************************************
                        ; UNUSED       BIT15
                        ; UNUSED       BIT14
                        ; UNUSED       BIT13
                        ; UNUSED       BIT12
                        ; UNUSED       BIT11
                        ; UNUSED       BIT10
                        ; UNUSED       BIT09
                        ; UNUSED       BIT08
000200                  XMITRDY=       BIT07        ; TRANSMITTER READY
000100                  XMITIE=        BIT06        ; TRANSMITTER INTERRUPT ENABLE
                        ; UNUSED       BIT05
                        ; UNUSED       BIT04
                        ; UNUSED       BIT03
                        ; UNUSED       BIT02
                        ; UNUSED       BIT01
000001                  BREAK=         BIT00        ; SEND BREAK (CONTINUOUS SPACE)

                        ;;*****************************************************************
                        ; TBUF REGISTER BIT NAMES
                        ;;*****************************************************************
                        ; UNUSED       BIT15
                        ; UNUSED       BIT14
                        ; UNUSED       BIT13
                        ; UNUSED       BIT12
                        ; UNUSED       BIT11
                        ; UNUSED       BIT10
                        ; UNUSED       BIT09
                        ; UNUSED       BIT08
000200                  TDATA7=        BIT07        ; \
000100                  TDATA6=        BIT06        ; !
000040                  TDATA5=        BIT05        ; !
000020                  TDATA4=        BIT04        ; \ TRANSMITTER DATA BUFFER
000010                  TDATA3=        BIT03        ; /
000004                  TDATA2=        BIT02        ; !
000002                  TDATA1=        BIT01        ; !
000001                  TDATA0=        BIT00        ; /


                    ;;*****************************************************************
                    ; FLAG BITS TO BE USE OR CLEARED IN $USWR.

000002                  PARITY  =  2
000004                  EVENODD =  4
000010                  BRK     =  10
000020                  WRAP    =  20
```

```
 8831
 8832
 8833                                   ;;*****************************************************************
 8834                                   .SBTTL  TRAP CATCHER
  (1)
  (1)            000000                         .=0
  (1)                                   ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
  (1)                                   ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
  (1)                                   ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
  (1)            000174                         .=174
  (1)  000174  000000                   DISPREG: .WORD  0                ;;SOFTWARE DISPLAY REGISTER
  (1)  000176  000000                   SWREG:   .WORD  0                ;;SOFTWARE SWITCH REGISTER
  (1)                                   .SBTTL  STARTING ADDRESS(ES)
  (1)  000200  000137  001334                   JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
 8835
 8836
 8837
 8838                                   ;
 8839                                   ; OFFLINE TESTING FOR INTERRUPT VECTOR PROBLEMS
 8840                                   ;
 8841            000400                         .=400
 8842
 8843  000400  005000                         CLR     R0
 8844  000402  012720  013610         1$:     MOV     #INTSRV,(R0)+   ;SET INTR HANDLER PTR
 8845  000406  012720  00030(                 MOV     #PR6,(R0)+      ;SET PRIORITY TO 6 TO BE
 8846                                   ;                               ;SPECIFIC TO 11/21 PROCESSOR
 8847  000412  020027  000400                 CMP     R0,#400         ;ALL DONE?
 8848  000416  001371                         BNE     1$              ;BR IF NO
 8849  000420  005037  000176                 CLR     176             ;CLEAN UP SWREG
 8850  000424  000137  001334                 JMP     START           ;GO DO IT
```

```
                                        .SBTTL  ACT11 HOOKS

                                        ;;******************************************************************
                                        ;HOOKS REQUIRED BY ACT11
                 000430                         $SVPC=.                 ;SAVE PC
                 000046                         .=46
000046  014450                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                 000052                         .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
                 000430                         .=$SVPC                 ;; RESTORE PC
                 001000                 .=1000
                                        .SBTTL  APT PARAMETER BLOCK

                                        ;;******************************************************************
                                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                        ;;******************************************************************
                 001000                         .$X=.   ;;SAVE CURRENT LOCATION
                 000024                         .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                          200     ;;FOR APT START UP
                 000044                         .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                          $APTHDR ;;POINT TO APT HEADER BLOCK
                 001000                         .=.$X   ;;RESET LOCATION COUNTER
                                        ;;******************************************************************
                                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                        ;INTERFACE SPEC.

001000                                  $APTHD:
001000  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001174                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000031                  $TSTM:  .WORD   25.     ;;RUN TIM OF LONGEST TEST
001006  000144                  $PASTM: .WORD   100.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000144                  $UNITM: .WORD   100.    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000030                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
 8857                                   .SBTTL  COMMON TAGS
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
  (1)                                   ;*USED IN THE PROGRAM.
  (1)
  (1)                   001100                  .=1100
  (1)  001100                           $CMTAG:                         ;;START OF COMMON TAGS
  (1)  001100  000000                           .WORD   0
  (1)  001102     000   000377          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
  (1)  001103     000                   $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
  (1)  001104  000000                   $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
  (1)  001106  000000                   $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
  (1)  001110  000000                   $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
  (1)  001112  000000                   $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
  (1)  001114     000                   $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
  (1)  001115     001                   $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
  (1)  001116  000000                   $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
  (1)  001120  000000                   $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
  (1)  001122  000000                   $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
  (1)  001124  000000                   $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
  (1)  001126  000000                   $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
  (1)  001130  000000                           .WORD   0               ;;RESERVED--NOT TO BE USED
  (1)  001132  000000                           .WORD   0
  (1)  001134     000                   $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
  (1)  001135     000                   $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
  (1)  001136  000000                           .WORD   0
  (1)  001140  177570                   SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
  (1)  001142  177570                   DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
  (1)  001144  177560                   $TKS:   177560                  ;;TTY KBD STATUS
  (1)  001146  177562                   $TKB:   177562                  ;;TTY KBD BUFFER
  (1)  001150  177564                   $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
  (1)  001152  177566                   $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
  (1)  001154     000                   $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
  (1)  001155     002                   $FILLS: .BYTE   2
  (1)                                                   ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
  (1)  001156     012                   $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
  (1)  001157     000                   $TPFLG: .BYTE   0               ;;''TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
  (1)  001160  000000                   $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
  (1)  001162  000000                   $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
  (1)  001164  177607  000377           $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
  (1)  001170     077                   $QUES:  .ASCII  /?/             ;;QUESTION MARK
  (1)  001171     015                   $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
  (1)  001172  000012                   $LF:    .ASCIZ  <12>            ;;LINE FEED
  (2)                                   ;;*****************************************************************
  (2)                                   .SBTTL  APT MAILBOX-ETABLE
  (2)
  (3)                                   ;;*****************************************************************
  (2)                                   .EVEN
  (2)  001174                           $MAIL:                          ;;APT MAILBOX
  (2)  001174  000000                   $MSGTY: .WORD   AMSGTY           ;;MESSAGE TYPE CODE
  (2)  001176  000000                   $FATAL: .WORD   AFATAL           ;;FATAL ERROR NUMBER
  (2)  001200  000000                   $TESTN: .WORD   ATESTN           ;;TEST NUMBER
  (2)  001202  000000                   $PASS:  .WORD   APASS            ;;PASS COUNT
  (2)  001204  000000                   $DEVCT: .WORD   ADEVCT           ;;DEVICE COUNT
  (2)  001206  000000                   $UNIT:  .WORD   AUNIT            ;;I/O UNIT NUMBER
```

```
(2)  001210  000000                $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
(2)  001212  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
(2)  001214                        $ETABLE:                ;;APT ENVIRONMENT TABLE
(2)  001214     000                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
(2)  001215     000                $ENVM:  .BYTE   AENVM
(2)                                ;;ENVIRONMENT MODE BITS
(2)  001216  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
(2)  001220  000031                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
(2)  001222  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
(2)                                ;*                      BITS 15-11=CPU TYPE
(2)                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(2)                                ;*                              11/70=06,PDQ=07,Q=10
(2)                                ;*                      BIT 10=REAL TIME CLOCK
(2)                                ;*                      BIT  9=FLOATING POINT PROCESSOR
(2)                                ;*                      BIT  8=MEMORY MANAGEMENT
(2)  001224     000                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
(2)  001225     000                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
(2)                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
(2)                                ;*                              900 NSEC CORE=001
(2)                                ;*                              300 NSEC BIPOLAR=002
(2)                                ;*                              500 NSEC MOS=003
(2)  001226  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
(2)                                ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
(2)  001230     000                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
(2)  001231     000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
(2)  001232  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
(2)  001234     000                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
(2)  001235     000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
(2)  001236  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
(2)  001240     000                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
(2)  001241     000                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
(2)  001242  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
(2)  001244  000300                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
(2)  001246  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
(2)  001250  176500                $BASE:  .WORD   ABASE
(2)                                        ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(2)  001252  000017                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
(2)  001254                        $ETEND:
(2)                                .MEXIT
```

```
   (1)                                  .SBTTL  ERROR POINTER TABLE
   (1)
   (1)                                  ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
   (1)                                  ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
   (1)                                  ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
   (1)                                  ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
   (1)                                  ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
   (1)
   (1)                                  ;*      EM              ;;POINTS TO THE ERROR MESSAGE
   (1)                                  ;*      DH              ;;POINTS TO THE DATA HEADER
   (1)                                  ;*      DT              ;;POINTS TO THE DATA
   (1)                                  ;*      DF              ;;POINTS TO THE DATA FORMAT
   (1)
   (1)
   (1)  001254                          $ERRTB:
 8858                                   ;;              GLOBAL DATA
 8859  001254  176500                      DLADD:  DLADDR
 8860  001256  000300                      DLVEC:  300
 8861  001260  176500                      RCSR:   DLADDR  +  0
 8862  001262  176502                      RBUF:   DLADDR  +  2
 8863  001264  176504                      TCSR:   DLADDR  +  4
 8864  001266  176506                      TBUF:   DLADDR  +  6
 8865  001270  000000                      I:      0
 8866  001272  000020                      .BLKW 20    ;FOR R5 STACK
 8867  001332  000000                      R5STACK:    .WORD 0
```

```
 8870
 8871  001334                                 START:
  (1)                                         .SBTTL  INITIALIZE THE COMMON TAGS
  (1)                                         ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
  (1)  001334  012706  001100                         MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
  (1)  001340  005026                                 CLR     (R6)+           ;;CLEAR MEMORY LOCATION
  (1)  001342  022706  001140                         CMP     #SWR,R6 ;;DONE?
  (1)  001346  001374                                 BNE     .-6             ;;LOOP BACK IF NO
  (1)  001350  012706  001100                         MOV     #STACK,SP       ;;SETUP THE STACK POINTER
  (1)                                         ;;INITIALIZE A FEW VECTORS
  (1)  001354  012737  016210  000020                 MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
  (1)  001362  012737  000300  000022                 MOV     #PR6,@#IOTVEC+2 ;;LEVEL 6
  (1)  001370  012737  016010  000030                 MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
  (1)  001376  012737  000300  000032                 MOV     #PR6,@#EMTVEC+2 ;;LEVEL 6
  (1)                                         ;;BIT02
  (1)  001404  012737  017142  000034                 MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
  (1)  001412  012737  000300  000036                 MOV     #PR6,@#TRAPVEC+2;LEVEL 6
  (1)  001420  013737  014416  014410                 MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
  (1)  001426  005037  001160                         CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
  (1)  001432  005037  001162                         CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
  (1)  001436  112737  000001  001115                 MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
  (1)  001444  012737  001444  001106                 MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
  (1)  001452  012737  001452  001110                 MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
  (2)                                         ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
  (2)                                         ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
  (2)  001460  013746  000004                         MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
  (2)  001464  012737  001520  000004                 MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
  (2)  001472  012737  177570  001140                 MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
  (2)  001500  012737  177570  001142                 MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
  (2)  001506  022777  177777  177424                 CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
  (2)  001514  001012                                 BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
  (2)                                                                         ;;AND  THE HARDWARE SWR IS NOT = -1
  (2)  001516  000403                                 BR      65$             ;;BRANCH IF NO TIMEOUT
  (2)  001520  012716  001526          64$:           MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
  (2)  001524  000002                                 RTI
  (2)  001526  012737  000176  001140  65$:           MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
  (2)  001534  012737  000174  001142                 MOV     #DISPREG,DISPLAY
  (2)  001542  012637  000004          66$:           MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
  (1)
  (2)  001546  005037  001202                         CLR     $PASS           ;;CLEAR PASS COUNT
  (2)  001552  132737  000200  001215                 BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
  (2)  001560  001403                                 BEQ     67$             ;;YES,USE NON-APT SWITCH
  (2)  001562  012737  001216  001140                 MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
  (2)  001570                          67$:
 8872                                         .SBTTL  TYPE PROGRAM NAME
  (1)                                         ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
  (1)  001570  005227  177777                         INC     #-1             ;;FIRST TIME?
  (1)  001574  001043                                 BNE     68$             ;;BRANCH IF NO
  (1)  001576  022737  014450  000042                 CMP     #$ENDAD,@#42    ;;ACT-11?
  (1)  001604  001437                                 BEQ     68$             ;;BRANCH IF YES
  (1)  001606  104401  001654                         TYPE    ,69$            ;;TYPE ASCIZ STRING
  (2)                                         .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
  (2)  001612  005737  000042                         TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
  (2)  001616  001012                                 BNE     70$             ;;BRANCH IF YES
  (2)  001620  123727  001214  000001                 CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
  (2)  001626  001406                                 BEQ     70$             ;;BRANCH IF YES
```

```
(2)  001630  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
(2)  001636  001005                          BNE     71$             ;;BRANCH IF NO
(2)  001640  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
(2)  001642  000403                          BR      71$
(2)  001644  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
(2)  001652                          71$:
(1)  001652  000414                          BR      68$             ;;GET OVER THE ASCIZ
(1)                                  ;;69$:  .ASCIZ  <CRLF>*CNDLAA0 DLV11-J TEST*<CRLF>
(1)  001704                          68$:
```

```
 8874
 8875
 8876
 8877  001704  004737  012700                  JSR     PC,SIZE         ;SIZE THE BUS IF $DEVM = 0
 8878
 8879  001710  005037  014166                  CLR     PHASE2          ;INIT FOR TESTS
 8880  001714  012737  000001  001206          MOV     #1,$UNIT
 8881  001722  005037  014164                  CLR     CHAN
 8882  001726  005037  014170                  CLR     P1CNT
 8883
 8884  001732  012706  001100        LOOP:     MOV     #STACK,SP       ;RESET STACK POINTER
 8885  001736  004737  013620                  JSR     PC,CYCLE        ;SETUP MODULE AND CHANNEL
 8890                                                                  ;                ADRS+2 -> NEXT VECTOR
 8891                                                                                   ;GET UNIT ADDRESS
 8892  001742                                                                  LET     DLADD := (ADRS)+
   (4) 001742  012137  001254                  MOV     (ADRS)+,DLADD
 8893                                                                                   ;GET UNIT VECTOR
 8894  001746                                                                  LET     DLVEC := (ADRS)
   (4) 001746  011137  001256                  MOV     (ADRS),DLVEC
 8895                                                                                   ;RCSR = DLADD  +  0
 8896  001752                                                                  LET     RCSR  := DLADD
   (4) 001752  013737  001254  001260          MOV     DLADD,RCSR
 8897  001760                                                                  LET     RBUF  := DLADD + #2
   (4) 001760  013737  001254  001262          MOV     DLADD,RBUF
   (7) 001766  062737  000002  001262          ADD     #2,RBUF
 8898  001774                                                                  LET     TCSR  := DLADD + #4
   (4) 001774  013737  001254  001264          MOV     DLADD,TCSR
   (7) 002002  062737  000004  001264          ADD     #4,TCSR
 8899  002010                                                                  LET     TBUF  := DLADD + #6
   (4) 002010  013737  001254  001266          MOV     DLADD,TBUF
   (7) 002016  062737  000006  001266          ADD     #6,TBUF
 8900  002024                                                                  LET     R5    := #R5STACK
   (4) 002024  012705  001332                  MOV     #R5STACK,R5
 8901
 8902  002030  012700  001000                  MOV     #1000,R0        ;DELAY TO ALLOW PRINT TO FINISH
 8903  002034  077001                          SOB     R0,.            ;BEFORE RESET
 8904  002036                                                          BRESET
   (1) 002036  000005                          RESET
 8905  002040                                                                  LET     $DEVCT := $DEVCT + #1
   (7) 002040  005237  001204                  INC     $DEVCT
 8906  002044                                                                  IFB     PHASE2  EQ  #TRUE  THEN
   (6) 002044  123727  014166  000001          CMPB    PHASE2,#TRUE
   (9) 002052  001003                          BNE     $1
 8907  002054  000137  007000                  JMP     MODTST              ;GO TO PHASE 2 TESTING
 8908  002060                                                          ELSE
   (4) 002060  000402                          BR      $2
   (3) 002062                          $1:
 8909  002062                                                              LET $TSTNM := #0
   (4) 002062  005037  001102                  CLR     $TSTNM
 8910  002066                                                          ENDIF
   (4) 002066                          $2:
```

```
                                ;;*****************************************************************
                                ;*TEST 1        ADDRESSABILITY
                                ;*
                                ;*      THIS TEST VERIFIES THAT ALL 4 REGISTERS OF THE CHANNEL UNDER TEST
                                ;*      RESPOND TO THEIR ADDRESSES.
                                ;*
                                ;;*****************************************************************
002066  000004                  TST1:   SCOPE

002070  012737  000002  001160          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
002076  012737  000001  001200          MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        LET     ADRS := DLADD
002104  013701  001254                  MOV     DLADD,ADRS
                                                                        ; SET UP INTERRUPT
002110  012737  013610  000004          MOV     #INTSRV,ERRVEC
002116  012737  000300  000006          MOV     #PR6,ERRVEC+2           ;PRIORITY LOWERED TO 6
                                ;                                       ;TO BE SPECIFIC TO PDP 11/21
002124                                                                  LET     I := #0
002124  005037  001270                  CLR     I
002130                                                                  REPEAT
002130                          $3:
002130                                                                      BGNSUB
002130  012737  002136  001110          MOV     #64$,$LPERR
                                                                            ;CLEAR FLAG
                                                                            LET INTFLAG := #0
002136  005037  013616                  CLR     INTFLAG

                                                                            ;READ FLAG
002142  005711                  TST     @ADRS
002144                                                                      IF INTFLAG NE #0 THEN
002144  005737  013616          TST     INTFLAG
002150  001401                  BEQ     $4

002152                                                                      ; FATAL ERROR
                                                                              ERRDF 1
002152  104001                  ERROR   1
002154                                                                      ENDIF
002154                          $4:
002154                                                                  ENDSUB
002154                                                                  LET     I := I + #2
002154  062737  000002  001270          ADD     #2,I
002162                                                                  LET     ADRS := DLADD + I
002162  013701  001254                  MOV     DLADD,ADRS
002166  063701  001270                  ADD     I,ADRS

002172                                                          UNTIL I EQ #8.
002172  023727  001270  000010          CMP     I,#8.
002200  001353                  BNE     $3
```

```
8950
8951
8952  002202  032777  177476  177050          BIT     #177476,@RCSR   ;CHECK THAT ALL UNUSED BITS ARE 0
8953  002210  001401                          BEQ     1$
8954  002212  104027                          ERROR   27              ;RCSR HAS UNUSED BITS SET
8955
8956  002214  032777  007400  177040  1$:     BIT     #7400,@RBUF
8957  002222  001401                          BEQ     2$
8958  002224  104030                          ERROR   30              ;RBUF HAS UNUSED BITS SET
8959
8960  002226  032777  177476  177030  2$:     BIT     #177476,@TCSR
8961  002234  001401                          BEQ     3$
8962  002236  104031                          ERROR   31              ;TCSR HAS UNUSED BITS SET
8963
8964  002240  032777  177400  177020  3$:     BIT     #177400,@TBUF
8965  002246  001401                          BEQ     4$
8966  002250  104032                          ERROR   32              ;TBUF HAS ;UNUSED BITS SET
8967
8968  002252  012737  000006  000004  4$:     MOV     #6,ERRVEC       ;RESTORE
8969  002260  005037  000006                  CLR     ERRVEC+2
8970
```

```
 8972
 8977                                   ;;*********************************************************************
 8978                                   ;* THE FOLLOWING 3 TESTS  TEST ALL 'READ WRITE' BITS
 8979                                   ;;*********************************************************************
 8980
 8981
 8986                                   ;;*********************************************************************
  (3)                                   ;*TEST 2        BREAK - TCSR 0 SET, CLEAR, RESET
  (4)                                   ;*              THIS BIT IS THE ONLY ONE IN THIS POSITION
  (4)                                   ;*              THAT IS READ AND WRITE.
  (3)                                   ;;*********************************************************************
  (2) 002264  000004                    TST2:   SCOPE
  (2)
  (1) 002266  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 002274  012737  000002  001200            MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 8987
 8992 002302                                                    IF #BRK NOTSETIN $USWR THEN
  (6) 002302  032737  000010  001220            BIT     #BRK,$USWR
  (9) 002310  001001                            BNE     $5
 8993 002312                                                        EXIT                    ; BREAK NOT INSTALLED
  (3) 002312  000452                            BR      TST3            ;;EXIT THIS TEST
 8994 002314                                                    ENDIF
  (4) 002314                            $5:
 8995
 8996                                                   ; SEE IF IT IS CLEAR
 8997 002314                                                    BGNSUB
  (5) 002314  012737  002322  001110            MOV     #64$,$LPERR
 8998
 8999 002322                                                        IF      #BREAK SETIN @TCSR THEN
  (6) 002322  032777  000001  176734            BIT     #BREAK,@TCSR
  (9) 002330  001401                            BEQ     $6
 9000                                                               ; BREAK DID NOT RESET IN TCSR
 9001 002332                                                        ERRHRD 2
  (1) 002332  104002                            ERROR   2
 9002 002334                                                    ENDIF
  (4) 002334                            $6:
 9003 002334                                                ENDSUB
 9004
 9005                                               ; TRY TO SET BREAK BIT
 9006 002334                                                BGNSUB
  (5) 002334  012737  002342  001110            MOV     #64$,$LPERR
 9007 002342                                                        LET     @TCSR := @TCSR SET.BY #BREAK
  (7) 002342  052777  000001  176714            BIS     #BREAK,@TCSR
 9008                                                               ; STUCK TO 0
 9009 002350                                                        IF      #BREAK NOTSETIN @TCSR THEN
  (6) 002350  032777  000001  176706            BIT     #BREAK,@TCSR
  (9) 002356  001001                            BNE     $7
 9010                                                               ; BREAK DID NOT SET IN TCSR
 9011 002360                                                        ERRHRD 3
  (1) 002360  104003                            ERROR   3
 9012 002362                                                    ENDIF
  (4) 002362                            $7:
 9013 002362                                                ENDSUB
 9014
 9015                                               ; TRY TO CLEAR A SET BIT
 9016 002362                                                BGNSUB
```

```
  (5)  002362  012737  002370  001110          MOV     #64$,$LPERR
9017
9018  002370                                                        LET    @TCSR := @TCSR CLR.BY #BREAK
 (7)  002370  042777  000001  176666          BIC     #BREAK,@TCSR
9019                                                                ; SHOULD HAVE CLEARED
9020  002376                                                        IF     #BREAK SETIN @TCSR THEN
 (6)  002376  032777  000001  176660          BIT     #BREAK,@TCSR
 (9)  002404  001401                          BEQ     $10
9021                                                                ; BREAK  DID NOT CLEAR IN TCSR
9022  002406                                                        ERRHRD 4
 (1)  002406  104004                          ERROR   4
9023  002410                                                        ENDIF
 (4)  002410                          $10:
9024  002410                                                      ENDSUB
9025
9026                                                        ; NOW SEE IF RESET CLEARS IT
9027  002410                                                      BGNSUB
 (7)  002410  012737  002416  001110          MOV     #64$,$LPERR
9028
9029  002416                                                        LET    @TCSR := @TCSR SET.BY #BREAK
 (7)  002416  052777  000001  176640          BIS     #BREAK,@TCSR
9030                                                                ; ISSUE BUS RESET
9031  002424                                                        BRESET
 (1)  002424  000005                          RESET
9032  002426                                                        IF     #BREAK SETIN @TCSR THEN
 (6)  002426  032777  000001  176630          BIT     #BREAK,@TCSR
 (9)  002434  001401                          BEQ     $11
9033                                                                ; BREAK DID NOT RESET IN TCSR
9034  002436                                                        ERRHRD 5
 (1)  002436  104005                          ERROR   5
9035  002440                                                        ENDIF
 (4)  002440                          $11:
9036  002440                                                      ENDSUB
```

```
 9038
 9039
 9044                                ;;*****************************************************************
  (3)                                ;*TEST 3         XMITIE - TCSR 6        SET, CLEAR, RESET
  (3)                                ;;*****************************************************************
  (2) 002440  000004                 TST3:   SCOPE
  (2)
  (1) 002442  012737  000010  001160         MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 002450  012737  000003  001200         MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 9045                                                                        ; USE PRIORITY OF 6
 9050 002456                                 SETPRI #PR6
 9051
 9052                                                ; SEE IF IT IS CLEAR
 9053 002470                                                             BGNSUB
  (5) 002470  012737  002476  001110         MOV     #65$,$LPERR
 9054
 9055 002476                                                                 IF      #XMITIE SETIN @TCSR THEN
  (6) 002476  032777  000100  176560         BIT     #XMITIE,@TCSR
  (9) 002504  001401                         BEQ     $12
 9056                                                                        ; XMITIE DID NOT RESET IN TCSR
 9057 002506                                                                ERRHRD 12
  (1) 002506  104012                         ERROR   12
 9058 002510                                                             ENDIF
  (4) 002510                         $12:
 9059 002510                                                         ENDSUB
 9060
 9061                                                ; TRY TO SET XMITIE BIT
 9062 002510                                                         BGNSUB
  (5) 002510  012737  002516  001110         MOV     #64$,$LPERR
 9063 002516                                                             LET     @TCSR := @TCSR SET.BY #XMITIE
  (7) 002516  052777  000100  176540         BIS     #XMITIE,@TCSR
 9064                                                                    ; STUCK TO 0
 9065 002524                                                             IF      #XMITIE NOTSETIN @TCSR THEN
  (6) 002524  032777  000100  176532         BIT     #XMITIE,@TCSR
  (9) 002532  001001                         BNE     $13
 9066                                                                        ; XMIT DID NOT SET IN TCSR
 9067 002534                                                                ERRHRD 13
  (1) 002534  104013                         ERROR   13
 9068 002536                                                             ENDIF
  (4) 002536                         $13:
 9069 002536                                                         ENDSUB
 9070
 9071                                                ; TRY TO CLEAR A SET BIT
 9072 002536                                                         BGNSUB
  (5) 002536  012737  002544  001110         MOV     #64$,$LPERR
 9073
 9074 002544                                                             LET     @TCSR := @TCSR CLR.BY #XMITIE
  (7) 002544  042777  000100  176512         BIC     #XMITIE,@TCSR
 9075                                                                    ; SHOULD HAVE CLEARED
 9076 002552                                                             IF      #XMITIE SETIN @TCSR THEN
  (6) 002552  032777  000100  176504         BIT     #XMITIE,@TCSR
  (9) 002560  001401                         BEQ     $14
 9077                                                                        ;  XMIT DID NOT CLEAR IN TCSR
 9078 002562                                                                ERRHRD 14
  (1) 002562  104014                         ERROR   14
 9079 002564                                                             ENDIF
```

```
  (4)  002564                          $14:
 9080  002564                                                          ENDSUB
 9081
 9082                                                          ; NOW SEE IF RESET CLEARS IT
 9083  002564                                                          BGNSUB
  (5)  002564  012737  002572  001110          MOV     #64$,$LPERR
 9084
 9085  002572                                                              LET     @TCSR := @TCSR SET.BY #XMITIE
  (7)  002572  052777  000100  176464          BIS     #XMITIE,@TCSR
 9086                                                                  ; ISSUE BUS RESET
 9087  002600                                                          BRESET
  (1)  002600  000005                          RESET
 9088  002602                                                          IF      #XMITIE SETIN @TCSR THEN
  (6)  002602  032777  000100  176454          BIT     #XMITIE,@TCSR
  (9)  002610  001401                          BEQ     $15
 9089                                                                     ; XMIT DID NOT RESET IN TCSR
 9090  002612                                                             ERRHRD 15
  (1)  002612  104015                          ERROR   15
 9091  002614                                                          ENDIF
  (4)  002614                          $15:
 9092  002614                                                          ENDSUB
```

```
 9094
 9095
 9104                                   ;;*****************************************************************
  (3)                                   ;*TEST 4          RCVRIE - RCSR 6        SET, CLEAR, RESET
  (4)                                   ;*               THIS BIT IS THE ONLY ONE IN THIS POSITION
  (4)                                   ;*               THAT IS READ AND WRITE.
  (3)                                   ;;*************************************** *********************
  (2) 002614  000004                    TST4:   SCOPE
  (2)
  (1) 002616  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 002624  012737  000004  001200            MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 9109                                                   ; SEE IF IT IS CLEAR
 9110 002632                                                            BGNSUB
  (5) 002632  012737  002640  001110            MOV     #64$,$LPERR
 9111
 9112 002640                                                                IF      #RCVRIE SETIN @RCSR THEN
  (6) 002640  032777  000100  176412            BIT     #RCVRIE,@RCSR
  (9) 002646  001401                            BEQ     $16
 9113                                                                           ; RCVRIE DID NOT RESET IN RCSR
 9114 002650                                                                    ERRHRD 35
  (1) 002650  104035                            ERROR   35
 9115 002652                                                                ENDIF
  (4) 002652                            $16:
 9116 002652                                                            ENDSUB
 9117
 9118                                                   ; TRY TO SET RCVRIE BIT
 9119 002652                                                            BGNSUB
  (5) 002652  012737  002660  001110            MOV     #64$,$LPERR
 9120 002660                                                                LET     @RCSR := @RCSR SET.BY #RCVRIE
  (7) 002660  052777  000100  176372            BIS     #RCVRIE,@RCSR
 9121                                                                       ; STUCK TO 0
 9122 002666                                                                IF      #RCVRIE NOTSETIN @RCSR THEN
  (6) 002666  032777  000100  176364            BIT     #RCVRIE,@RCSR
  (9) 002674  001001                            BNE     $17
 9123                                                                           ; RCVRIE DID NOT SET IN RCSR
 9124 002676                                                                    ERRHRD 36
  (1) 002676  104036                            ERROR   36
 9125 002700                                                                ENDIF
  (4) 002700                            $17:
 9126 002700                                                            ENDSUB
 9127
 9128                                                   ; TRY TO CLEAR A SET BIT
 9129 002700                                                            BGNSUB
  (5) 002700  012737  002706  001110            MOV     #64$,$LPERR
 9130
 9131 002706                                                                LET     @RCSR := @RCSR CLR.BY #RCVRIE
  (7) 002706  042777  000100  176344            BIC     #RCVRIE,@RCSR
 9132                                                                       ; SHOULD HAVE CLEARED
 9133 002714                                                                IF      #RCVRIE SETIN @RCSR THEN
  (6) 002714  032777  000100  176336            BIT     #RCVRIE,@RCSR
  (9) 002722  001401                            BEQ     $20
 9134                                                                           ; RCVRIE DID NOT CLEAR IN RCSR
 9135 002724                                                                    ERRHRD 37
  (1) 002724  104037                            ERROR   37
 9136 002726                                                                ENDIF
  (4) 002726                            $20:
```

```
 9137  002726                                                          ENDSUB
 9138
 9139                                                  ; NOW SEE IF RESET CLEARS IT
 9140  002726                                                          BGNSUB
  (5)  002726  012737  002734  001110          MOV     #64$,$LPERR
 9141
 9142  002734                                                              LET     @RCSR := @RCSR SET.BY #RCVRIE
  (7)  002734  052777  000100  176316          BIS     #RCVRIE,@RCSR
 9143                                                                      ; ISSUE BUS RESET
 9144  002742                                                              BRESET
  (1)  002742  000005                          RESET
 9145  002744                                                              IF      #RCVRIE SETIN @RCSR THEN
  (6)  002744  032777  000100  176306          BIT     #RCVRIE,@RCSR
  (9)  002752  001401                          BEQ     $21
 9146                                                                          ; RCVRIE DID NOT RESET IN RCSR
 9147  002754                                                                  ERRHRD 40
  (1)  002754  104040                          ERROR   40
 9148  002756                                                              ENDIF
  (4)  002756                          $21:
 9149  002756                                                          ENDSUB
```

```
 9151
 9152
 9157                                   ;;*****************************************************************
  (3)                                   ;*TEST 5          TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
  (3)                                   ;;*****************************************************************
  (2)  002756  000004                   TST5:   SCOPE
  (2)
  (1)  002760  012737  000010  001160           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  002766  012737  000005  001200           MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 9158
 9163  002774                                                                  BGNSUB
  (5)  002774  012737  003002  001110           MOV     #64$,$LPERR
 9164
 9165  003002                                                                      IF      #XMITRDY NOTSETIN @TCSR THEN
  (6)  003002  032777  000200  176254           BIT     #XMITRDY,@TCSR
  (9)  003010  001002                           BNE     $22
 9166
 9167                                                                                ;RESET SHOULD HAVE SET BIT.
 9168                                                                                ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
 9169  003012                                                                       ERRHRD 42
  (1)  003012  104042                           ERROR   42
 9170                                                                                ;ISSUE ANOTHER RESET
 9171  003014                                                                       BRESET
  (1)  003014  000005                           RESET
 9172  003016                                                                   ENDIF
  (4)  003016                           $22:
 9173                                                                                ;ALLOW LOOPING ON ERROR
 9174  003016                                                                  ENDSUB
```

```
 9176
 9177
 9186                                   ;;*********************************************************************
  (3)                                   ;*TEST 6         TEST THAT XMIT RDY - TCSR 7 - CLEARS
  (4)                                   ;*              WHEN TBUF IS LOADED WITH A CHARACTER
  (4)                                   ;*              AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
  (3)                                   ;;*********************************************************************
  (2)  003016  000004                   TST6:   SCOPE
  (2)
  (1)  003020  012737  000010  001160           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  003026  012737  000006  001200           MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 9191
 9192  003034                                                           IF CONSOL EQ #TRUE THEN
  (6)  003034  023727  013604  000001           CMP     CONSOL,#TRUE
  (9)  003042  001001                           BNE     $23
 9193  003044                                                              EXIT
  (3)  003044  000521                           BR      TST7            ;;EXIT THIS TEST
 9194  003046                                                           ENDIF
  (4)  003046                           $23:
 9195
 9196  003046                                                           LET PASS := #0  ;INIT COUNT OF TIMES THRU
  (4)  003046  005037  003302                   CLR     PASS
 9197  003052                                                           LOOP            ; START OF LOOP
  (3)  003052                           $24:
 9198                                                                                   ;  MAX OF 2 TIMES THRU
 9199  003052                                                              LET ERRORFLAG := #0
  (4)  003052  005037  003304                   CLR     ERRORFLAG
 9200  003056                                                               LET EXITFLAG := #0
  (4)  003056  005037  003306                   CLR     EXITFLAG
 9201                                                       ; LOAD TBUF WITH ONE CHARACTER
 9202                                                       ; WAIT FOR READY TO SET
 9203                                                       ; (SHOULD BE VERY SHORT WAIT
 9204                                                       ; SINCE UART DOUBLE BUFFERS ITS INPUT)
 9205
 9206                                                                   ;SEND A CHARACTER
 9207  003062                                                        LET @TBUF :B= #0
  (4)  003062  105077  176200                   CLRB    @TBUF
 9208                                                                   ;WAIT A MAXIMUM
 9209                                                                   ;OF 50 MSEC FOR
 9210                                                                   ;XMIT RDY TO SET IN TCSR
 9211  003066                                                        CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
  (3)  003066  010546                           MOV     R5,-(SP)
  (7)  003070  012745  177777                   MOV     #SET,-(R5)
  (6)  003074  013745  001264                   MOV     TCSR,-(R5)
  (5)  003100  012745  000200                   MOV     #XMITRDY,-(R5)
  (4)  003104  012745  000500                   MOV     #500,-(R5)
  (3)  003110  004737  012446                   JSR     PC,TIMER
  (3)  003114  012605                           MOV     (SP)+,R5
 9212                                                                   ;TIMER RETURNS AN ERROR IF BIT DID
 9213                                                                   ;NOT MEET CONDITION WITHIN TIME LIMIT
 9214  003116                                                        IF.ERROR THEN
  (6)  003116  103003                           BCC     $26
 9215                                                                   ;XMIT RDY DID NOT SET IN TCSR
 9216  003120                                                           ERRHRD 66
  (1)  003120  104066                           ERROR   66
 9217  003122  000137  003310                   JMP     TST7
```

```
                                                                                    ENDIF
 9218  003126
  (4)  003126                                $26:
 9219  003126  105077  176134                      CLRB    @TBUF   ;SHIP 1'ST CHAR
 9220  003132  105777  176126                1$:   TSTB    @TCSR   ;WAIT FOR RDY
 9221  003136  100375                            BPL     1$
 9222                                                   ; LOAD TBUF WITH A SECOND CHARACTER (TO DOUBLE BUFFER)
 9223                                                   ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
 9224                                                   ; AND THEN WAIT FOR IT TO SET
 9225
 9226                                                                        ;SEND SECOND CHARACTER
 9227  003140                                                              LET @TBUF :B= #0
  (4)  003140  105077  176122                      CLRB    @TBUF
 9228  003144  000240                                      NOP             ; GIVE IT TIME TO CLEAR
 9229                                                                        ; XMITRDY SHOULD HAVE CLEARED UPON
 9230                                                                        ; RECEIPT OF A CHARACTER
 9231  003146                                                              IF #XMITRDY SETIN @TCSR THEN
  (6)  003146  032777  000200  176110              BIT     #XMITRDY,@TCSR
  (9)  003154  001404                              BEQ     $27
 9232
 9233                                                                        ; XMITRDY DID NOT CLEAR IN TCSR
 9234  003156                                                                ; WILL RESULT IN ERR 67 IF FAILS 2X
  (4)  003156  012737  177777  003304              MOV     #SET,ERRORFLAG    LET ERRORFLAG := #SET
 9235
 9236                                                                                ; DEFER ERROR TYPEOUT
 9237  003164                                                              ELSE
  (4)  003164  000416                              BR      $30
  (3)  003166                                $27:
 9238
 9239                                                                        :WAIT A MAXIMUM
 9240                                                                        ;OF 100 MSEC FOR
 9241  003166                                                              ;XMIT RDY TO SET IN TCSR
  (3)  003166  010546                              MOV     R5,-(SP)        CALL TIMER IN <#1000,#XMITRDY,TCSR,#SET>
  (7)  003170  012745  177777                      MOV     #SET,-(R5)
  (6)  003174  013745  001264                      MOV     TCSR,-(R5)
  (5)  003200  012745  000200                      MOV     #XMITRDY,-(R5)
  (4)  003204  012745  001000                      MOV     #1000,-(R5)
  (3)  003210  004737  012446                      JSR     PC,TIMER
  (3)  003214  012605                              MOV     (SP)+,R5
 9242  003216                                                              IF.ERROR THEN
  (6)  003216  103001                              BCC     $31
 9243                                                                        ;XMIT RDY DID NOT SET IN TCSR
 9244  003220                                                              ERRHRD 70
  (1)  003220  104070                              ERROR   70
 9245  003222                                                              ENDIF
  (4)  003222                                $31:
 9246  003222                                                             ENDIF ; OF DEFERED ERROR CALL
  (4)  003222                                $30:
 9247  003222                                                            IF ERRORFLAG EQ #SET THEN
  (6)  003222  023727  003304  177777              CMP     ERRORFLAG,#SET
  (9)  003230  001013                              BNE     $32
 9248  003232                                                              LET PASS := PASS + #1
  (7)  003232  005237  003302                      INC     PASS
 9249  003236                                                               IF PASS GT #1 THEN
  (6)  003236  023727  003302  000001              CMP     PASS,#1
  (9)  003244  003404                              BLE     $33
 9250                                                                           ; CALL ERROR IF 2ND TRY
```

```
 9251
 9252  003246
  (1)  003246  104067                    ERROR   67                          ; ON XMIT RDY NOT CLEARING
                                                                             ERRHRD 67
 9253  003250
  (4)  003250  012737  177777  003306     MOV     #SET,EXITFLAG                   LET EXITFLAG := #SET
 9254  003256                                                                  ENDIF
  (4)  003256                         $33:
 9255  003256                                                              ELSE            ; NO ERROR
  (4)  003256  000403                    BR      $34
  (3)  003260                         $32:
 9256  003260                                                                  LET EXITFLAG := #SET
  (4)  003260  012737  177777  003306     MOV     #SET,EXITFLAG
 9257  003266                                                              ENDIF
  (4)  003266                         $34:
 9258  003266                                                          EXIF        EXITFLAG EQ #SET
  (3)  003266  023727  003306  177777     CMP     EXITFLAG,#SET
  (5)  003274  001401                     BEQ     $25
 9259  003276                                                          ENDLOOP
  (4)  003276  000665                     BR      $24
  (3)  003300                         $25:
 9260  003300                                                              EXIT ; SKIP AROUND FLAG WORDS
  (3)  003300  000403                     BR      TST7            ;;EXIT THIS TEST
 9261  003302  000000                PASS:       0
 9262  003304  000000                ERRORFLAG:  0
 9263  003306  000000                EXITFLAG:   0
```

```
 9265
 9266
 9275                                   ;;*****************************************************************
  (3)                                   ;*TEST 7        TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH WRAP CONNECTED)
  (4)                                   ;*             RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
  (4)                                   ;*             AND THAT RESET CLEARS THE BIT.
  (3)                                   ;;*****************************************************************
  (2) 003310  000004                    TST7:   SCOPE
  (2)
  (1) 003312  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 003320  012737  000007  001200            MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 9276
 9281 003326                                                            IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
  (6) 003326  032737  000020  001220            BIT     #WRAP,$USWR
  (8) 003334  001404                            BEQ     $35
  (6) 003336  023727  013604  000001            CMP     CONSOL,#TRUE
  (9) 003344  001001                            BNE     $36
  (6) 003346                            $35:
 9282 003346                                                                    EXIT
  (3) 003346  000434                            BR      TST10           ;;EXIT THIS TEST
 9283 003350                                                            ENDIF
  (4) 003350                            $36:
 9284
 9285 003350                                                            BGNSUB
  (5) 003350  012737  003356  001110            MOV     #64$,$LPERR
 9286                                                       ; SEND A CHARACTER AND LET IT WRAP AROUND
 9287
 9288 003356                                                                LET @TBUF :B= #0
  (4) 003356  105077  175704                    CLRB    @TBUF
 9289
 9290                                                                       ; WAIT A MAXIMUM OF 50 MSEC
 9291                                                                       ; FOR RCVR DONE TO SET IN
 9292                                                                       ; RCSR
 9293 003362                                                                CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3) 003362  010546                            MOV     R5,-(SP)
  (7) 003364  012745  177777                    MOV     #SET,-(R5)
  (6) 003370  013745  001260                    MOV     RCSR,-(R5)
  (5) 003374  012745  000200                    MOV     #RCVRDONE,-(R5)
  (4) 003400  012745  000500                    MOV     #500,-(R5)
  (3) 003404  004737  012446                    JSR     PC,TIMER
  (3) 003410  012605                            MOV     (SP)+,R5
 9294
 9295                                                                       ;DIDN'T SET IN TIME
 9296 003412                                                            IF.ERROR THEN
  (6) 003412  103001                            BCC     $37
 9297                                                                       ; RCVRDONE DID NOT SET IN RCSR
 9298 003414                                                                ERRHRD 71
  (1) 003414  104071                            ERROR   71
 9299 003416                                                            ENDIF
  (4) 003416                            $37:
 9300
 9301 003416                                                    ENDSUB
 9302
 9303 003416                                                    BGNSUB
  (5) 003416  012737  003424  001110            MOV     #64$,$LPERR
 9304                                                       ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
```

```
 9305  003424                                                          BRESET
  (1)  003424  000005                    RESET
 9306
 9307  003426                                                          IF #RCVRDONE SETIN @RCSR THEN
  (6)  003426  032777  000200  175624    BIT     #RCVRDONE,@RCSR
  (9)  003434  001401                    BEQ     $40
 9308                                                                      ; RCVRDONE DID NOT RESET IN RCSR.
 9309  003436                                                              ERRHRD 72
  (1)  003436  104072                    ERROR   72
 9310  003440                                                          ENDIF
  (4)  003440                        $40:
 9311  003440                                                      ENDSUB
```

```
 9313
 9314
 9319                                   ;;*********************************************************************
  (3)                                   ;*TEST 10        TEST THAT RCVRDONE IS CLEARED BY SETTING READER ENABLE
  (3)                                   ;;*********************************************************************
  (2)  003440  000004                   TST10:  SCOPE
  (2)
  (1)  003442  012737  000010  001160           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  003450  012737  000010  001200           MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9320
 9325  003456                                                           IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
  (6)  003456  032737  000020  001220           BIT     #WRAP,$USWR
  (8)  003464  001404                           BEQ     $41
  (6)  003466  023727  013604  000001           CMP     CONSOL,#TRUE
  (9)  003474  001001                           BNE     $42
  (6)  003476                           $41:
 9326  003476                                                                   EXIT
  (3)  003476  000433                           BR      TST11           ;;EXIT THIS TEST
 9327  003500                                                                   ENDIF
  (4)  003500                           $42:
 9328
 9329  003500                                                           BGNSUB
  (5)  003500  012737  003506  001110           MOV     #64$,$LPERR
 9330                                               ; OUTPUT A CHARACTER AND WAIT FOR XMITRDY TO SET.
 9331
 9332                                                                       ; OUTPUT A CHARACTER
 9333  003506                                                           LET @TBUF :B= #0
  (4)  003506  105077  175554                   CLRB    @TBUF
 9334                                                                       ; WAIT MAXIMUM OF 500 MSEC
 9335                                                                       ; FOR RCVRDONE TO SET IN
 9336                                                                       ; RCSR
 9337  003512                                                           CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3)  003512  010546                           MOV     R5,-(SP)
  (7)  003514  012745  177777                   MOV     #SET,-(R5)
  (6)  003520  013745  001260                   MOV     RCSR,-(R5)
  (5)  003524  012745  000200                   MOV     #RCVRDONE,-(R5)
  (4)  003530  012745  000500                   MOV     #500,-(R5)
  (3)  003534  004737  012446                   JSR     PC,TIMER
  (3)  003540  012605                           MOV     (SP)+,R5
 9338                                                                       ; DID IT BECOME READY?
 9339  003542                                                           IF.ERROR  THEN
  (6)  003542  103001                           BCC     $43
 9340                                                                       ;RCVRDONE DID NOT SET IN RCSR
 9341  003544                                                               ERRHRD 25
  (1)  003544  104025                           ERROR   25
 9342                                                                       ; SET IT BACK TO CONTINUE
 9343  003546                                                           ENDIF
  (4)  003546                           $43:
 9344  003546                                                      ENDSUB
 9345
 9346                                       ;NOW THAT IT IS SET LETS SEE IF SETTING
 9347                                       ;READER ENABLE CLEARS RCVRDONE
 9348
 9349  003546                                                      LET @RCSR := @RCSR SET.BY #RDRRUN
  (7)  003546  052777  000001  175504           BIS     #RDRRUN,@RCSR
 9350  003554                                                           IF #RCVRDONE SETIN @RCSR THEN
```

```
   (6)  003554  032777  000200  175476          BIT     #RCVRDONE,@RCSR
   (9)  003562  001401                          BEQ     $44
 9351                                                        ;RCVRDONE DID NOT CLEAR IN RCSR
 9352  003564                                                ERRHRD 26
   (1)  003564  104026                          ERROR   26
 9353                                                        ; SET IT BACK TO CONTINUE
 9354  003566                                              ENDIF
   (4)  003566                        $44:
```

```
 9356
 9357
 9362                                ;;*****************************************************************
  (3)                                ;*TEST 11        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
  (3)                                ;;*****************************************************************
  (2)  003566  000004                TST11:  SCOPE
  (2)
  (1)  003570  012737  000010  001160        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  003576  012737  000011  001200        MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9363
 9368  003604                                                              IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
  (6)  003604  032737  000020  001220        BIT     #WRAP,$USWR
  (8)  003612  001404                        BEQ     $45
  (6)  003614  023727  013604  000001        CMP     CONSOL,#TRUE
  (9)  003622  001001                        BNE     $46
  (6)  003624                        $45:
 9369  003624                                                                  EXIT
  (3)  003624  000432                        BR      TST12           ;;EXIT THIS TEST
 9370  003626                                                                  ENDIF
  (4)  003626                        $46:
 9371
 9372  003626                                                              BGNSUB
  (5)  003626  012737  003634  001110        MOV     #64$,$LPERR
 9373                                            ; OUTPUT A CHARACTER AND WAIT FOR RCVRDONE TO SET.
 9374
 9375                                                                        ; OUTPUT A CHARACTER
 9376  003634                                                              LET @TBUF :B= #0
  (4)  003634  105077  175426                CLRB    @TBUF
 9377                                                                       ; WAIT MAXIMUM OF 500 MSEC
 9378                                                                       ; FOR RCVRDONE TO SET IN
 9379                                                                       ; RCSR
 9380  003640                                                              CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3)  003640  010546                        MOV     R5,-(SP)
  (7)  003642  012745  177777                MOV     #SET,-(R5)
  (6)  003646  013745  001260                MOV     RCSR,-(R5)
  (5)  003652  012745  000200                MOV     #RCVRDONE,-(R5)
  (4)  003656  012745  000500                MOV     #500,-(R5)
  (3)  003662  004737  012446                JSR     PC,TIMER
  (3)  003666  012605                        MOV     (SP)+,R5
 9381                                                                       ; DID IT BECOME READY?
 9382  003670                                                              IF.ERROR  THEN
  (6)  003670  103001                        BCC     $47
 9383                                                                       ;RCVRDONE DID NOT SET IN RCSR
 9384  003672                                                              ERRHRD 73
  (1)  003672  104073                        ERROR   73
 9385                                                                       ; SET IT BACK TO CONTINUE
 9386  003674                                                              ENDIF
  (4)  003674                        $47:
 9387  003674                                                          ENDSUB
 9388
 9389                                            ; NOW THAT IT IS SET LETS SEE IF READING THE
 9390                                            ; BUFFER CLEARS RCVRDONE.
 9391
 9392                                                                       ;READ BUFFER
 9393  003674                                                              LET R0 :B= @RBUF
  (4)  003674  117700  175362                MOVB    @RBUF,R0
```

```
 9394
 9395  003700                                                    IF #RCVRDONE SETIN @RCSR THEN
  (6)  003700  032777  000200  175352          BIT     #RCVRDONE,@RCSR
  (9)  003706  001401                          BEQ     $50
 9396
 9397  003710                                                        ;RCVRDONE DID NOT CLEAR IN RCSR
  (1)  003710  104074                          ERROR   74           ERRHRD 74
 9398                                                               ; SET IT BACK TO CONTINUE
 9399  003712                                                    ENDIF
  (4)  003712                          $50:
```

```
 9401
 9402
 9407                                ;;*********************************************************************
  (3)                                ;*TEST 12       TEST THE OVERRUN & ERROR BITS - RBUF 14
  (3)                                ;;*********************************************************************
  (2) 003712  000004                 TST12:  SCOPE
  (2)
  (1) 003714  012737  000010  001160         MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 003722  012737  000012  001200         MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9408
 9413 003730                                                                IF CONSOL EQ #0 THEN
  (6) 003730  005737  013604                 TST     CONSOL
  (9) 003734  001007                         BNE     $51
 9414 003736                                                                    IF #WRAP SETIN $USWR THEN
  (6) 003736  032737  000020  001220         BIT     #WRAP,$USWR
  (9) 003744  001401                         BEQ     $52
 9415                                                                           ;; NULL ---EXECUTE TEST
 9416 003746                                                                ELSE
  (4) 003746  000401                         BR      $53
  (3) 003750                         $52:
 9417 003750                                                                    EXIT
  (3) 003750  000524                         BR      TST13           ;;EXIT THIS TEST
 9418 003752                                                                ENDIF
  (4) 003752                         $53:
 9419 003752                                                            ELSE
  (4) 003752  000401                         BR      $54
  (3) 003754                         $51:
 9420 003754                                                                EXIT
  (3) 003754  000522                         BR      TST13           ;;EXIT THIS TEST
 9421 003756                                                            ENDIF
  (4) 00375<                         $54:
 9422
 9423
 9424
 9425 003756                                                        BGNSUB
  (5) 003756  012737  003764  001110         MOV     #64$,$LPERR
 9426                                            ;OUTPUT 2 CHARACTERS
 9427                                            ;THIS SHOULD AN CAUSE OVERRUN ERROR.
 9428
 9429                                                                   ;OUTPUT 1 CHARACTER
 9430 003764                                                         LET @TBUF :B= #0
  (4) 003764  105077  175276                 CLRB    @TBUF
 9431
 9432 003770                                                        CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3) 003770  010546                         MOV     R5,-(SP)
  (7) 003772  012745  177777                 MOV     #SET,-(R5)
  (6) 003776  013745  001260                 MOV     RCSR,-(R5)
  (5) 004002  012745  000200                 MOV     #RCVRDONE,-(R5)
  (4) 004006  012745  000500                 MOV     #500,-(R5)
  (3) 004012  004737  012446                 JSR     PC,TIMER
  (3) 004016  012605                         MOV     (SP)+,R5
 9433
 9434                                                                   ;DID IT SET IN TIME?
 9435 004020                                                        IF.ERROR THEN
  (6) 004020  103001                         BCC     $55
 9436                                                                   ;RCVRDONE DID NOT SET IN RCSR
```

```
                                                                                    ERRHRD 16
 9437  004022
  (1)  004022  104016                    ERROR   16
                                                                                ENDIF
 9438  004024
  (4)  004024                    $55:
 9439
 9440                                                                               ;OUTPUT 2ND CHARACTER
 9441  004024                                                                    LET @TBUF :B= #0
  (4)  004024  105077  175236            CLRB    @TBUF
 9442                                                                               ;LET OVERRUN HAPPEN
 9443  004030                                                                      WAITMS 300.
  (4)  004030  010546                    MOV     R5,-(SP)
  (5)  004032  012745  000454            MOV     #300.,-(R5)
  (4)  004036  004737  012620            JSR     PC,WAIT
  (4)  004042  012605                    MOV     (SP)+,R5
 9444
 9445                                                                               ;READ BUFFER AND ERROR BITS
 9446  004044                                                                    LET R4 := @RBUF
  (4)  004044  017704  175212            MOV     @RBUF,R4
 9447
 9448                                                                               ;IT DIDN'T SET
 9449  004050                                                                    IF #ORERR NOTSETIN R4 THEN
  (6)  004050  032704  040000            BIT     #ORERR,R4
  (9)  004054  001002                    BNE     $56
 9450                                                                               ;ORERR DID NOT SET IN RBUF
 9451  004056                                                                      ERRHRD 101
  (1)  004056  104101                    ERROR   101
 9452
 9453                                                                               ;NO USE COMPOUNDING ERRORS
 9454  004060                                                                      EXIT
  (3)  004060  000460                    BR      TST13           ;;EXIT THIS TEST
 9455  004062                                                                    ENDIF
  (4)  004062                    $56:
 9456  004062                                                                 ENDSUB
 9457
 9458                                    ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR:
 9459  004062                                                                 BGNSUB
  (5)  004062  012737  004070  001110            MOV     #64$,$LPERR
 9460  004070                                                                     IF #ERROR NOTSETIN R4 THEN
  (6)  004070  032704  100000            BIT     #ERROR,R4
  (9)  004074  001002                    BNE     $57
 9461
 9462                                                                                ;ERROR DID NOT SET IN RBUF
 9463  004076                                                                       ERRHRD 102
  (1)  004076  104102                    ERROR   102
 9464
 9465                                                                                ;-WHEN ORERR SET.
 9466                                                                                ;GET OUT NOW.
 9467  004100                                                                       EXIT
  (3)  004100  000450                    BR      TST13           ;;EXIT THIS TEST
 9468  004102                                                                    ENDIF
  (4)  004102                    $57:
 9469  004102                                                                 ENDSUB
 9470
 9471  004102                                                                 BGNSUB
  (5)  004102  012737  004110  001110            MOV     #64$,$LPERR
 9472                                        ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.
```

```
 9473
 9474  004110                                                          IF #ORERR NOTSETIN @RBUF THEN
  (6)  004110  032777  040000  175144          BIT     #ORERR,@RBUF
  (9)  004116  001002                          BNE     $60
 9475
 9476                                                                      ;READING RBUF CLEARED ORERR.
 9477  004120                                                             ERRHRD 103
  (1)  004120  104103                          ERROR   103
 9478                                                                      ;SKIP REST OF TEST
 9479  004122                                                             EXIT
  (3)  004122  000437                          BR      TST13          ;;EXIT THIS TEST
 9480  004124                                                          ENDIF
  (4)  004124                          $60:
 9481  004124                                                      ENDSUB
 9482
 9483  004124                                                      BGNSUB
  (5)  004124  012737  004132  001110          MOV     #64$,$LPERR
 9484                                              ;READING RBUF ABOVE SHOULD ENABLE ERROR TO BE CLEARED
 9485                                              ;BY NEXT TRANSFER.
 9486                                              ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED
 9487
 9488                                                                      ;SEND A CHARACTER AROUND.
 9489  004132                                                          LET @TBUF :B= #0
  (4)  004132  105077  175130                  CLRB    @TBUF
 9490  004136                                                      CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3)  004136  010546                          MOV     R5,-(SP)
  (7)  004140  012745  177777                  MOV     #SET,-(R5)
  (6)  004144  013745  001260                  MOV     RCSR,-(R5)
  (5)  004150  012745  000200                  MOV     #RCVRDONE,-(R5)
  (4)  004154  012745  000500                  MOV     #500,-(R5)
  (3)  004160  004737  012446                  JSR     PC,TIMER
  (3)  004164  012605                          MOV     (SP)+,R5
 9491
 9492                                                                      ;DID IT SET IN TIME?
 9493  004166                                                      IF.ERROR THEN
  (6)  004166  103001                          BCC     $61
 9494                                                                      ;RCVRDONE DID NOT SET IN RCSR
 9495  004170                                                             ERRHRD 20
  (1)  004170  104020                          ERROR   20
 9496  004172                                                      ENDIF
  (4)  004172                          $61:
 9497
 9498  004172                                                          IF #ORERR SETIN @RBUF THEN
  (6)  004172  032777  040000  175062          BIT     #ORERR,@RBUF
  (9)  004200  001402                          BEQ     $62
 9499                                                                      ;ORERR DID NOT CLEAR IN RBUF
 9500  004202                                                             ERRHRD 104
  (1)  004202  104104                          ERROR   104
 9501
 9502                                                                      ;-AFTER RECEIVING ANOTHER CHAR
 9503                                                                      ;SKIP AROUND REST
 9504  004204                                                             EXIT
  (3)  004204  000406                          BR      TST13          ;;EXIT THIS TEST
 9505  004206                                                          ENDIF
  (4)  004206                          $62:
 9506
```

```
 9507  004206                                                    IF #ERROR SETIN @RBUF THEN
  (6)  004206  032777  100000  175046        BIT     #ERROR,@RBUF
  (9)  004214  001401                        BEQ     $63
 9508                                                          ;ERROR DID NOT CLEAR IN RBUF
 9509  004216                                                  ERRHRD 105
  (1)  004216  104105                        ERROR   105
 9510
 9511  004220                                                  ENDIF
  (4)  004220                          $63:
 9512  004220                                                ENDSUB
 9513  004220                                                EXIT
  (3)  004220  000400                        BR      TST13          ;;EXIT THIS TEST
```

```
 9515
 9516
 9528                                   ;;*****************************************************************
  (3)                                   ;*TEST 13       TRANSMITTER INTERRUPT LOGIC TEST
  (4)                                   ;*             LOGICALLY THIS IS 4 SEPARATE TESTS
  (4)                                   ;*             A) DOES TRANSMITTER INTERRUPT LOGIC WORK
  (4)                                   ;*             B) AT PRIORITY OF 0
  (4)                                   ;*             C) AND ONLY ONCE
  (4)                                   ;*             D) BUT NOT WITH INTERRUPT ENABLE CLEAR
  (3)                                   ;*****************************************************************
  (2)  004222  000004                   TST13:  SCOPE
  (2)
  (1)  004224  012737  000010  001160           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2)  004232  012737  000013  001200           MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9533                                                                           ;CLEAR 'INTERRUPT OCCURED' FLAG
 9534  004240                                                           LET INTFLAG := #0
  (4)  004240  005037  013616                   CLR     INTFLAG
 9535
 9536                                                                           ;GET VECTOR ADDRESS
 9537  004244                                                           LET R3 := DLVEC
  (4)  004244  013703  001256                   MOV     DLVEC,R3
 9538                                                                           ;FOR THE TRANSMITTER
 9539  004250                                                           LET R3 := R3 + #4
  (7)  004250  062703  000004                   ADD     #4,R3
 9540                                                                           ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
 9541  004254  012723  013610                   MOV     #INTSRV,(R3)+
 9542  004260  012713  000300                   MOV     #PR6,(R3)
 9543  004264                                                           BGNSUB
  (5)  004264  012737  004272  001110           MOV     #64$,$LPERR
 9544                                                                   ;; MAKE SURE THAT TRANSMITTER READY IS SET
 9545  004272                                                           CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
  (3)  004272  010546                           MOV     R5,-(SP)
  (7)  004274  012745  177777                   MOV     #SET,-(R5)
  (6)  004300  013745  001264                   MOV     TCSR,-(R5)
  (5)  004304  012745  000200                   MOV     #XMITRDY,-(R5)
  (4)  004310  012745  000500                   MOV     #500,-(R5)
  (3)  004314  004737  012446                   JSR     PC,TIMER
  (3)  004320  012605                           MOV     (SP)+,R5
 9546                                                                                    ;
 9547                                                                           ;CLEAR INTERRUPT ENABLE
 9548  004322                                                           LET @TCSR := @TCSR CLR.BY #XMITIE
  (7)  004322  042777  000100  174734           BIC     #XMITIE,@TCSR
 9549
 9550                                                                           ;SET IT TO 0
 9551  004330                                  SETPRI #PR0
 9552
 9553                                                                           ;NOW SET I.E. BIT
 9554  004342                                                           LET @TCSR := @TCSR SET.BY #XMITIE
  (7)  004342  052777  000100  174714           BIS     #XMITIE,@TCSR
 9555                                                                                    !
 9556  004350                                                           CALL TIMER IN <#500,#1,#INTFLAG,#SET>
  (3)  004350  010546                           MOV     R5,-(SP)
  (7)  004352  012745  177777                   MOV     #SET,-(R5)
  (6)  004356  012745  013616                   MOV     #INTFLAG,-(R5)
  (5)  004362  012745  000001                   MOV     #1,-(R5)
  (4)  004366  012745  000500                   MOV     #500,-(R5)
```

```
  (3)  004372  004737  012446              JSR     PC,TIMER
  (3)  004376  012605                      MOV     (SP)+,R5
9557
9558                                                        ;DID IT SET IN TIME?
9559  004400                                              IF.ERROR THEN
  (6)  004400  103001                      BCC     $64
9560                                                        ;INTERRUPT DID NOT OCCUR
9561  004402                                                ERRHRD 106
  (1)  004402  104106                      ERROR   106
9562  004404                                              ENDIF
  (4)  004404                      $64:
9563
9564                                                        ;LET POSSIBLE 2'ND INTERR OCCUR
9565  004404                                              WAITMS 500.
  (4)  004404  010546                      MOV     R5,-(SP)
  (5)  004406  012745  000764              MOV     #500.,-(R5)
  (4)  004412  004737  012620              JSR     PC,WAIT
  (4)  004416  012605                      MOV     (SP)+,R5
9566
9567                                                        ;DID EXACTLY 1 INTERRUPT OCCUR
9568  004420                                              IF INTFLAG GT #1 THEN
  (6)  004420  023727  013616  000001      CMP     INTFLAG,#1
  (9)  004426  003401                      BLE     $65
9569                                                        ;TRANSMITTER INTERRUPTED TWICE
9570  004430                                                  ERRHRD 107
  (1)  004430  104107                      ERROR   107
9571  004432                                              ENDIF
  (4)  004432                      $65:
9572  004432                                            ENDSUB
9573                                          ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
9574  004432                                            BGNSUB
  (5)  004432  012737  004440  001110      MOV     #64$,$LPERR
9575                                                        ;CLEAR INTERRUPT ENABLE
9576  004440                                              LET @TCSR := @TCSR CLR.BY #XMITIE
  (7)  004440  042777  000100  174616      BIC     #XMITIE,@TCSR
9577                                                        ;CLEAR 'INTERRUPT OCCURED' FLAG
9578  004446                                              LET INTFLAG := #0
  (4)  004446  005037  013616              CLR     INTFLAG
9579                                        ;NO INTERRUPTS SHOULD OCCUR, PSW STILL AT 0.
9580                                        ;DARE IT TO HAPPEN
9581  004452                                                  LET @TBUF := #0
  (4)  004452  005077  174610              CLR     @TBUF
9582                                                      ;SEE IF INT FLAG EVER SETS
9583  004456                                              CALL TIMER IN <#1000,#1,#INTFLAG,#SET>
  (3)  004456  010546                      MOV     R5,-(SP)
  (7)  004460  012745  177777              MOV     #SET,-(R5)
  (6)  004464  012745  013616              MOV     #INTFLAG,-(R5)
  (5)  004470  012745  000001              MOV     #1,-(R5)
  (4)  004474  012745  001000              MOV     #1000,-(R5)
  (3)  004500  004737  012446              JSR     PC,TIMER
  (3)  004504  012605                      MOV     (SP)+,R5
9584                                                      ;DID IT SET DURING TIMER?
9585  004506  103401                      BCS     1$        ;BR IF NO
9586                                                        ;INTERR STILL OCCURED WITH IE DISABLED
9587  004510                                                  ERRHRD 110
  (1)  004510  104110                      ERROR   110
```

```
 9588  004512                          1$:
```

```
 9590
 9591
 9601                                   ;;*****************************************************************
  (3)                                   ;*TEST 14       RECEIVER INTERRUPT LOGIC TEST
  (4)                                   ;*             THIS TEST COVERS ALL OF THE RECEIVER
  (4)                                   ;*             SIDE OF THE INTERRUPT LOGIC IN
  (4)                                   ;*             CHARACTER MODE.
  (3)                                   ;;*****************************************************************
  (2) 004512  000004                    TST14:  SCOPE
  (2)
  (1) 004514  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 004522  012737  000014  001200            MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9606 004530                                                            IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
  (6) 004530  032737  000020  001220            BIT     #WRAP,$USWR
  (8) 004536  001404                            BEQ     $66
  (6) 004540  023727  013604  000001            CMP     CONSOL,#TRUE
  (9) 004546  001002                            BNE     $67
  (6) 004550                            $66:
 9607 004550  000137  005046                    JMP     TST15                   ;       EXIT TEST
 9608 004554                                                                    ENDIF
  (4) 004554                            $67:
 9609
 9610                                                                           ;CLEAR INTERRUPT OCCURED FLAG
 9611                                                                           ;SET UP RECEIVER INTER.VECTOR
 9612 004554                                                                    SETVEC DLVEC,#INTSRV,#PR6
  (5) 004554  010146                            MOV     R1,-(SP)
  (5) 004556  013701  001256                    MOV     DLVEC,R1
  (5) 004562  012721  013610                    MOV     #INTSRV,(R1)+
  (5) 004566  012711  000300                    MOV     #PR6,(R1)
  (5) 004572  012601                            MOV     (SP)+,R1
 9613                                           ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
 9614 004574                                                                    BGNSUB
  (5) 004574  012737  004602  001110            MOV     #64$,$LPERR
 9615 004602
  (4) 004602  005037  013616                    CLR     INTFLAG
 9616
 9617 004606
  (7) 004606  042777  000100  174444            BIC     #RCVRIE,@RCSR
 9618
 9619
 9620 004614                            SETPRI #PR0
 9621
 9622
 9623 004626
  (4) 004626  105077  174434                    CLRB    @TBUF
 9624
 9625
 9626
 9627 004632
  (3) 004632  010546                            MOV     R5,-(SP)
  (7) 004634  012745  177777                    MOV     #SET,-(R5)
  (6) 004640  013745  001264                    MOV     TCSR,-(R5)
  (5) 004644  012745  000200                    MOV     #RCVRDONE,-(R5)
  (4) 004650  012745  000500                    MOV     #500,-(R5)
  (3) 004654  004737  012446                    JSR     PC,TIMER
  (3) 004660  012605                            MOV     (SP)+,R5
```

```
                                        BGNSUB

                                        LET INTFLAG := #0

                                            ;CLEAR INTERRUPTS
                                        LET @RCSR := @RCSR CLR.BY #RCVRIE

                                            ;CHANGE PRIORITY
                                            ;..TO 0


                                            ;SEND A CHARACTER
                                        LET @TBUF :B= #0

                                            ;WAIT A MAXIMUM
                                        ;OF 50 MSEC FOR
                                        ;XMIT RDY TO SET IN TCSR
                                        CALL TIMER IN <#500,#RCVRDONE,TCSR,#SET>
```

```
                                                                        ;SET INTERRUPT ENABLE
                                                                    LET @RCSR := @RCSR SET.BY #RCVRIE
                                                                        ;LET IT COME IN.
                                                                    CALL TIMER IN <#500,#1,#INTFLAG,#SET>
9628
9629  004662
 (7)  004662  052777  000100  174370          BIS     #RCVRIE,@RCSR
9630
9631
9632  004670
 (3)  004670  010546                          MOV     R5,-(SP)
 (7)  004672  012745  177777                  MOV     #SET,-(R5)
 (6)  004676  012745  013616                  MOV     #INTFLAG,-(R5)
 (5)  004702  012745  000001                  MOV     #1,-(R5)
 (4)  004706  012745  000500                  MOV     #500,-(R5)
 (3)  004712  004737  012446                  JSR     PC,TIMER
 (3)  004716  012605                          MOV     (SP)+,R5
9633
9634
9635  004720                                                        ;DID IT SET IN TIME?
                                                                    IF.ERROR THEN
 (6)  004720  103001                          BCC     $70
9636                                                                    ;INTERRUPT DID NOT OCCUR
9637  004722                                                            ERRHRD 111
 (1)  004722  104111                          ERROR   111
9638  004724                                                        ENDIF
 (4)  004724                          $70:
9639                                                                ;LET POSSIBLE 2'ND INTERR OCCUR
9640  004724                                                        WAITMS 500
 (4)  004724  010546                          MOV     R5,-(SP)
 (5)  004726  012745  000500                  MOV     #500,-(R5)
 (4)  004732  004737  012620                  JSR     PC,WAIT
 (4)  004736  012605                          MOV     (SP)+,R5
9641
9642                                                                    ;EXACTLY 1 INTERRUPT?
9643  004740                                                        IF  INTFLAG GT #1 THEN
 (6)  004740  023727  013616  000001          CMP     INTFLAG,#1
 (9)  004746  003401                          BLE     $71
9644                                                                    ;RECEIVER INTERRUPTED TWICE
9645  004750                                                                ERRHRD 112
 (1)  004750  104112                          ERROR   112
9646  004752                                                        ENDIF
 (4)  004752                          $71:
9647  004752                                                    ENDSUB
9648
9649
9650
9651
9652
9653
9654                                    ;INTERRUPT WITHOUT IE SET.
9655  004752                                                    BGNSUB
 (5)  004752  012737  004760  001110          MOV     #64$,$LPERR
9656
9657                                                                    ;CLEAR INTERRUPT
9658  004760                                                        LET @RCSR := @RCSR CLR.BY #RCVRIE
 (7)  004760  042777  000100  174272          BIC     #RCVRIE,@RCSR
9659                                                                    ;CLEAR INTERRUPT FLAG
9660  004766                                                        LET INTFLAG := #0
 (4)  004766  005037  013616                  CLR     INTFLAG
9661                                                                    ; SEND A CHARACTER
```

```
 9662  004772
  (4)  004772  105077  174270          CLRB    @TBUF                          LET @TBUF :B= #0
 9663
 9664
 9665  004776                                                                 CALL TIMER IN <#500,#1,#INTFLAG,#CLR>
  (3)  004776  010546                  MOV     R5,-(SP)
  (7)  005000  012745  000000          MOV     #CLR,-(R5)
  (6)  005004  012745  013616          MOV     #INTFLAG,-(R5)
  (5)  005010  012745  000001          MOV     #1,-(R5)
  (4)  005014  012745  000500          MOV     #500,-(R5)
  (3)  005020  004737  012446          JSR     PC,TIMER
  (3)  005024  012605                  MOV     (SP)+,R5
 9666                                                                           ;DID IT CLEAR IN TIME?
 9667  005026                                                                 IF.ERROR THEN
  (6)  005026  103001                  BCC     $72
 9668                                                                           ;INTERR STILL OCCURED WITH IE DISABLED
 9669  005030                                                                   ERRHRD 113
  (1)  005030  104113                  ERROR   113
 9670  005032                                                                 ENDIF
  (4)  005032                  $72:    .
 9671  005032                                                                 ENDSUB
 9672  005032                          SETPRI  #PR6    ;RAISE CPU PRIORITY
 9673                                                                         ;CLEAR THE WORLD
 9674  005044                                                                   BRFSET
  (1)  005044  000005                  RESET
 9675
 9676
```

```
 9678
 9679
 9684                                   ;;*****************************************************************
  (3)                                   ;*TEST 15       TEST DATA  WRAP AROUND: FLAG MODE
  (3)                                   ;;*****************************************************************
  (2)  005046  000004                   TST15:  SCOPE
  (2)
  (1)  005050  012737  000001  001160           MOV     #1,$TIMES       ;;DO 1 ITERATION
  (2)  005056  012737  000015  001200           MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9689  005064                                                           IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
  (6)  005064  032737  000020  001220           BIT     #WRAP,$USWR
  (8)  005072  001404                           BEQ     $73
  (6)  005074  023727  013604  000001           CMP     CONSOL,#TRUE
  (9)  005102  001001                           BNE     $74
  (6)  005104                           $73:                                    ;
 9690                                                                                ;CAN'T TEST WITHOUT A WRAP
 9691  005104                                                                  EXIT
  (3)  005104  000506                           BR      TST16           ;;EXIT THIS TEST
 9692  005106                                                           ENDIF
  (4)  005106                           $74:
 9693  005106                                                           BRESET
  (1)  005106  000005                           RESET
 9694                                                                        ;BINARY COUNT PATTERN
 9695  005110                                                           INCR R2 FROM #0 TO #377 BY #1
  (4)  005110  005002                           CLR     R2
  (5)  005112  000401                           BR      $75
  (4)  005114                           $76:
  (8)  005114  005202                           INC     R2
  (5)  005116                           $75:
  (5)  005116  020227  000377                   CMP     R2,#377
  (7)  005122  003077                           BGT     $77
 9696
 9697
 9698
 9699                                                                   ; MAKE SURE IT'S READY
 9700  005124                                                          CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
  (3)  005124  010546                           MOV     R5,-(SP)
  (7)  005126  012745  177777                   MOV     #SET,-(R5)
  (6)  005132  013745  001264                   MOV     TCSR,-(R5)
  (5)  005136  012745  000200                   MOV     #XMITRDY,-(R5)
  (4)  005142  012745  000500                   MOV     #500,-(R5)
  (3)  005146  004737  012446                   JSR     PC,TIMER
  (3)  005152  012605                           MOV     (SP)+,R5
 9701  005154                                                           IF.ERROR THEN
  (6)  005154  103002                           BCC     $100
 9702                                                                       ; TRANSMITTER NEVER BECAME READY
 9703  005156                                                               ERRHRD 123
  (1)  005156  104123                           ERROR   123
 9704  005160                                                               EXIT
  (3)  005160  000460                           BR      TST16           ;;EXIT THIS TEST
 9705  005162                                                           ENDIF
  (4)  005162                           $100:
 9706
 9707                                                                       ;START IT ON ITS WAY
 9708  005162                                                           LET @TBUF :B= R2
  (4)  005162  110277  174100                   MOVB    R2,@TBUF
```

```
 9709                                                                   ; NOW WAIT FOR RECIEVER DONE
 9710  005166                                                           CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3)  005166  010546                    MOV     R5,-(SP)
  (7)  005170  012745  177777            MOV     #SET,-(R5)
  (6)  005174  013745  001260            MOV     RCSR,-(R5)
  (5)  005200  012745  000200            MOV     #RCVRDONE,-(R5)
  (4)  005204  012745  000500            MOV     #500,-(R5)
  (3)  005210  004737  012446            JSR     PC,TIMER
  (3)  005214  012605                    MOV     (SP)+,R5
 9711  005216                                                               IF.ERROR THEN
  (6)  005216  103002                    BCC     $101
 9712  005220                                                                   ERRHRD 124
  (1)  005220  104124                    ERROR   124
 9713                                                                       ; RECIEVER NEVER BECAME READY
 9714  005222                                                               EXIT
  (3)  005222  000437                    BR      TST16           ;;EXIT THIS TEST
 9715  005224                                                           ENDIF
  (4)  005224                    $101:
 9716
 9717                                                                    ;RETRIEVE
 9718  005224                                                           LET R3 := @RBUF
  (4)  005224  017703  174032            MOV     @RBUF,R3
 9719                                                                    ;CHECK FOR ERROR DURING TRANSFER
 9720  005230                                                           IF #ERROR SETIN R3 THEN
  (6)  005230  032703  100000            BIT     #ERROR,R3
  (9)  005234  001401                    BEQ     $102
 9721                                                                     ;ERROR BIT SET IN HIGH BYTE OF RBUF
 9722  005236                                                           ERRHRD 200
  (1)  005236  104200                    ERROR   200
 9723  005240                                                            ENDIF
  (4)  005240                    $102:
 9724
 9725
 9726                                                                            ;COMPARE DATA
 9727  005240                                                                 IF R3 NE R2 THEN
  (6)  005240  020302                    CMP     R3,R2
  (9)  005242  001415                    BEQ     $103
 9728  005244                                                                    IF #BIT0 NOTSETIN $USWR THEN
  (6)  005244  032737  000001  001220    BIT     #BIT0,$USWR
  (9)  005252  001006                    BNE     $104
 9729  005254                                                                         IF R2 LT #200 THEN
  (6)  005254  020227  000200            CMP     R2,#200
  (9)  005260  002002                    BGE     $105
 9730                                                                                    ;DATA COMPARE ERR IN 7 BIT WORD
 9731  005262                                                                          ERRHRD 117
  (1)  005262  104117                    ERROR   117
 9732  005264                                                                          EXIT
  (3)  005264  000416                    BR      TST16           ;;EXIT THIS TEST
 9733  005266                                                                  ENDIF
  (4)  005266                    $105:
 9734  005266                                                                ELSE
  (4)  005266  000402                    BR      $106
  (3)  005270                    $104:
 9735                                                                            ;DATA COMP ERR IN 8 BIT WORD
 9736  005270                                                                   ERRHRD 17
  (1)  005270  104017                    ERROR   17
```

```
 9737  005272                                                                  EXIT
  (3)  005272  000413                    BR      TST16          ;;EXIT THIS TEST
 9738  005274                                                                    ENDIF
  (4)  005274                    $106:
 9739  005274                                                          ELSE IF #BIT0 NOTSETIN $USWR AND R2 GT #177 THEN
  (4)  005274  000411                    BR      $107
  (3)  005276                    $103:
  (7)  005276  032737  000001  001220     BIT     #BIT0,$USWR
  (10) 005304  001005                     BNE     $110
  (7)  005306  020227  000177             CMP     R2,#177
  (10) 005312  003402                     BLE     $110
 9740                                                                          ;GETTING 8 BITS ON 7 BIT XMIT
 9741                                                                          ;MAKE SURE $USWR SETUP CORRECTLY.
 9742  005314                                                                  ERRHRD 22
  (1)  005314  104022                    ERROR   22
 9743  005316                                                                  EXIT
  (3)  005316  000401                    BR      TST16          ;;EXIT THIS TEST
 9744  005320                                                                    ENDIF
  (4)  005320                    $110:
  (4)  005320                    $107:
 9745  005320                                                  ENDINC  ;R2
  (4)  005320  000675                    BR      $76
  (3)  005322                    $77:
```

```
 9747
 9748
 9753                                   ;;*****************************************************************
  (3)                                   ;*TEST 16       TEST DATA WRAP AROUND: INTERRUPT MODE
  (3)                                   ;;*****************************************************************
  (2)  005322  000004                   TST16:  SCOPE
  (2)
  (1)  005324  012737  000001  001160            MOV     #1,$TIMES       ;;DO 1 ITERATION
  (2)  005332  012737  000016  001200            MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9758  005340                                                            IF #WRAP NOTSETIN $USWR OR CONSOL EQ #TRUE THEN
  (6)  005340  032737  000020  001220            BIT     #WRAP,$USWR
  (8)  005346  001404                            BEQ     $111
  (6)  005350  023727  013604  000001            CMP     CONSOL,#TRUE
  (9)  005356  001002                            BNE     $112
  (6)  005360                           $111:
 9759  005360  000137  005772                    JMP     TST17                   ;EXIT TEST
 9760  005364                                                            ENDIF
  (4)  005364                           $112:
 9761
 9762                                           ; THIS TEST WILL RUN BOTH TRANSMITTER AND
 9763                                           ; RECIEVER AT FULL SPEED TESTING
 9764                                           ; THE ABILITY OF THE MODULE
 9765                                           ; TO HANDLE INTERRUPTS FROM BOTH SIDES AT ONCE.
 9766                                           ;
 9767                                           ;DOUBLE BUFFERING IS NOT FULLY TESTED BECAUSE OF
 9768                                           ;APT CONSIDERATIONS. I.E. 'BREAK' FROM APT
 9769                                           ;CAUSES OVERRUN ERRORS. THEREFORE TRANSMIT INTR IS
 9770                                           ;ENABLED ONLY AFTER THE RECVR HAS OBTAINED THE LAST WORD
 9771
 9772                                           ;  THIS TEST WILL TRANSFER A MAXIMUM OF 400(8)
 9773                                           ; CHARACTERS THROUGH THE MODULE, BUT IF AN ERROR
 9774                                           ; IS DETECTED BY THE TEST A PREMATURE SHUTDOWN OCCURS.
 9775
 9776
 9777                                                                    ;CHANGE PRIORITY
 9778  005364                           SETPRI #PR0                     ;..TO 0
 9779
 9780  005376                                                            ;GET VECTOR ADDRESS
  (4)  005376  013701  001256                    MOV     DLVEC,R1        LET R1 := DLVEC
 9781
 9782  005402                                                            ;RCVR VECTOR
  (4)  005402  012721  005664                    MOV     #REC,(R1)+      LET (R1)+ := #REC
 9783  005406
  (4)  005406  012721  000300                    MOV     #PR6,(R1)+      LET (R1)+ := #PR6
 9784
 9785                                                                    ;POINT TO TRANSMITTER VECTOR
 9786  005412                                                            ;AND SET IT UP ALSO
  (4)  005412  012721  005626                    MOV     #TRAN,(R1)+     LET (R1)+ := #TRAN
 9787  005416
  (4)  005416  012711  000300                    MOV     #PR6,(R1)       LET (R1)  := #PR6
 9788
 9789                                                                    ; CLEAR ERROR COUNTER
 9790  005422                                                            LET ERRCNT := #0
  (4)  005422  005037  005622                    CLR     ERRCNT
 9791  005426                                                            LET DATA := #0  ;XMIT DATA
  (4)  005426  005037  005662                    CLR     DATA
```

```
 9792  005432
  (6)  005432  032737  000001  001220        BIT     #BIT0,$USWR
  (9)  005440  001004                        BNE     $113
 9793  005442
  (4)  005442  012737  000200  005624        MOV     #200,NUMBER
 9794  005450
  (4)  005450  000403                        BR      $114
  (3)  005452                        $113:
 9795  005452
  (4)  005452  012737  000400  005624        MOV     #400,NUMBER
 9796  005460
  (4)  005460                        $114:
 9797
 9798  005460
  (1)  005460  000005                        RESET
 9799
 9800
 9801  005462
  (7)  005462  052777  000100  173574        BIS     #XMITIE,@TCSR
 9802
 9803  005470
  (7)  005470  052777  000100  173562        BIS     #RCVRIE,@RCSR
 9804
 9805
 9806
 9807  005476
  (3)  005476                        $115:
 9808  005476
  (4)  005476  023737  005662  005624        CMP     DATA,NUMBER
  (6)  005504  001403                        BEQ     $116
  (4)  005506  005737  005622                TST     ERRCNT
  (7)  005512  003771                        BLE     $115
  (4)  005514                        $116:
 9809
 9810
 9811
 9812  005514
  (7)  005514  042777  000100  173542        BIC     #XMITIE,@TCSR
 9813  005522
  (7)  005522  042777  000100  173530        BIC     #RCVRIE,@RCSR
 9814  005530
  (6)  005530  005737  005622                TST     ERRCNT
  (9)  005534  001431                        BEQ     $117
 9815  005536
  (6)  005536  032737  100000  005770        BIT     #ERROR,RHLD
  (9)  005544  001424                        BEQ     $120
 9816  005546
  (6)  005546  032737  040000  005770        BIT     #ORERR,RHLD
  (9)  005554  001402                        BEQ     $121
 9817
 9818  005556
  (1)  005556  104220                        ERROR   220
 9819  005560
  (4)  005560  000415                        BR      $122
  (3)  005562                        $121:
  (7)  005562  032737  020000  005770        BIT     #FRERR,RHLD

IF #BIT0 NOTSETIN $USWR THEN

   LET NUMBER := #200

ELSE

   LET NUMBER := #400

ENDIF

BRESET  ;SET UP ALL REGISTERS

        ;SET I.E. IN TRANSMITTER
LET @TCSR := @TCSR SET.BY #XMITIE

        ;AND RECEIVER
LET @RCSR := @RCSR SET.BY #RCVRIE


;NOW WE WAIT
REPEAT

UNTIL DATA EQ NUMBER OR ERRCNT GT #0


;NOW LETS CHECK.
;TURN OFF ALL INTR ENABLE
LET @TCSR := @TCSR CLR.BY #XMITIE

LET @RCSR := @RCSR CLR.BY #RCVRIE

IF ERRCNT NE #0 THEN

   IF #ERROR SETIN RHLD THEN

      IF #ORERR SETIN RHLD THEN

        ;OVERRUN ERROR
        ERRHRD 220

      ELSE IF #FRERR SETIN RHLD THEN
```

```
  (10) 005570  001402                    BEQ     $123
9820                                                          ;FRAMING ERROR
9821  005572                                                  ERRHRD 221
 (1)  005572  104221                    ERROR   221
9822  005574                                                ELSE IF #PERR SETIN RHLD THEN
 (4)  005574  000407                    BR      $124
 (3)  005576                    $123:
 (7)  005576  032737  010000  005770    BIT     #PERR,RHLD
 (10) 005604  001402                    BEQ     $125
9823                                                          ;PARITY ERROR
9824  005606                                                  ERRHRD 222
 (1)  005606  104222                    ERROR   222
9825  005610                                                ELSE
 (4)  005610  000401                    BR      $126
 (3)  005612                    $125:
9826                                                          ;UNKNOWN ERROR
9827  005612                                                  ERRHRD 24
 (1)  005612  104024                    ERROR   24
9828  005614                                                ENDIF
 (4)  005614                    $126:
 (4)  005614                    $124:
 (4)  005614                    $122:
9829  005614                                              ELSE
 (4)  005614  000401                    BR      $127
 (3)  005616                    $120:
9830                                                            ;DATA COMPARE ERROR
9831  005616                                                    ERRHRD 120
 (1)  005616  104120                    ERROR   120
9832  005620                                              ENDIF
 (4)  005620                    $127:
9833  005620                                                 ENDIF
 (4)  005620                    $117:
9834  005620                                                 EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
 (3)  005620  000464                    BR      TST17   ;;EXIT THIS TEST
9835
9836
9837  005622  000000            ERRCNT: 0
9838  005624  000000            NUMBER: 0
```

```
 9840
 9841
 9846                                   ;;*****************************************************************
 9847                                   ;                TRANSMIT INTERRUPT HANDLER
 9848                                   ;;*****************************************************************
 9853  005626                                                    BGNSRV  TRAN
  (1)  005626                           TRAN:
 9854  005626                                                       IF DATA NE NUMBER AND ERRCNT EQ #0 THEN
  (6)  005626  023737  005662  005624          CMP     DATA,NUMBER
  (9)  005634  001406                          BEQ     $130
  (6)  005636  005737  005622                  TST     ERRCNT
  (9)  005642  001003                          BNE     $130
 9855
 9856  005644                                                            ;SHIP OUT WORD
                                                                        LET @TBUF := DATA
  (4)  005644  013777  005662  173414          MOV     DATA,@TBUF
 9857  005652                                                       ENDIF
  (4)  005652                           $130:
 9858                                                               ;STOP INTERR, NOT EXER DOUBL BUFFER
 9859  005652                                                       LET @TCSR := @TCSR CLR.BY #XMITIE
  (7)  005652  042777  000100  173404          BIC     #XMITIE,@TCSR
 9860  005660                                                    ENDSRV
  (1)  005660  000002                          RTI
 9861
 9862  005662  000000                 DATA:   0
 9863
 9868                                   ;;*****************************************************************
 9869                                   ;                RECEIVER INTERRUPT HANDLER
 9870                                   ;;*****************************************************************
 9875  005664                                                    BGNSRV  REC
  (1)  005664                           REC:
 9876
 9877  005664                                                    ;GET CHAR
  (4)  005664  017737  173372  005770          MOV     @RBUF,RHLD      LET RHLD := @RBUF
 9878
 9879  005672                                                    ;CHECK ERROR
  (6)  005672  032737  100000  005770          BIT     #ERROR,RHLD     IF #ERROR SETIN RHLD OR RHLD NE  DATA THEN
  (8)  005700  001004                          BNE     $131
  (6)  005702  023737  005770  005662          CMP     RHLD,DATA
  (9)  005710  001411                          BEQ     $132
  (6)  005712                           $131:
 9880
 9881  005712                                                       ;STOP ALL INTERR PROC & GET OUT
  (4)  005712  013737  005624  005662          MOV     NUMBER,DATA     LET DATA := NUMBER
 9882  005720
  (7)  005720  042777  000100  173332          BIC     #RCVRIE,@RCSR   LET @RCSR := @RCSR CLR.BY #RCVRIE
 9883  005726
  (7)  005726  005237  005622                  INC     ERRCNT          LET ERRCNT := ERRCNT + #1
 9884  005732                                                    ELSE
  (4)  005732  000415                          BR      $133
  (3)  005734                           $132:
 9885  005734                                                       LET DATA := DATA + #1
  (7)  005734  005237  005662                  INC     DATA
 9886  005740                                                          IF DATA EQ NUMBER THEN
  (6)  005740  023737  005662  005624          CMP     DATA,NUMBER
  (9)  005746  001004                          BNE     $134
 9887  005750                                                             LET @RCSR := @RCSR CLR.BY #RCVRIE
```

```
  (7)  005750  042777  000100  173302          BIC     #RCVRIE,@RCSR
9888   005756                                                          ELSE
  (4)  005756  000403                          BR      $135
  (3)  005760                          $134:
9889
9890   005760                                                            ;ALLOW NEXT XMIT INTERR
                                                                         LET @TCSR := @TCSR SET.BY #XMITIE
  (7)  005760  052777  000100  173276          BIS     #XMITIE,@TCSR
9891   005766                                                          ENDIF
  (4)  005766                          $135:
9892   005766                                                      ENDIF
  (4)  005766                          $133:
9893   005766                                                  ENDSRV
  (1)  005766  000002                          RTI
9894
9895   005770  000000                  RHLD:   0
9896
```

```
 9898
 9899
 9910                                   ;;*****************************************************************
  (3)                                   ;*TEST 17       TEST BREAK DETECTION LOGIC
  (4)                                   ;*             TRANSMIT KNOWN CHAR WITH BREAK SET
  (4)                                   ;*             AND COMPARE RECEIVED WITH 0.
  (4)                                   ;*             FRAMING ERROR WILL ALSO BE CHECKED
  (4)                                   ;*             IF ERROR BITS ARE ENABLED.
  (3)                                   ;;*****************************************************************
  (2) 005772  000004                    TST17:  SCOPE
  (2)
  (1) 005774  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 006002  012737  000017  001200            MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 9915                                                                   ;DONT DO THIS TEST IF 'BREAK' GENERATION
 9916                                                                   ;ENABLED, ELSE WILL HALT TO CONSOLE ODT.
 9917                                                                   ;DO IF BREAK 'DETECTION' IS ENABLED.
 9918 006010                                                            IF #WRAP NOTSETIN $USWR OR #BRK NOTSETIN $USWR THEN
  (6) 006010  032737  000020  001220            BIT     #WRAP,$USWR
  (8) 006016  001404                            BEQ     $136
  (6) 006020  032737  000010  001220            BIT     #BRK,$USWR
  (9) 006026  001003                            BNE     $137
  (6) 006030                            $136:
 9919 006030  000137  006502                    JMP     TST20                           ;EXIT TEST
 9920 006034                                                            ELSE
  (4) 006034  000406                            BR      $140
  (3) 006036                            $137:
 9921 006036                                                                    IF CONSOL EQ #TRUE THEN
  (6) 006036  023727  013604  000001            CMP     CONSOL,#TRUE
  (9) 006044  001002                            BNE     $141
 9922                                                                               ;CAN'T TEST CONSOLE
 9923 006046  000137  006502                    JMP     TST20                      ;EXIT TEST
 9924 006052                                                                    ENDIF
  (4) 006052                            $141:
 9925 006052                                                            ENDIF
  (4) 006052                            $140:
 9926 006052                                                            BGNSUB
  (5) 006052  012737  006060  001110            MOV     #64$,$LPERR
 9927
 9928 006060                                                                    LET ERRCHK := #0        ; CLEAR ERROR WORD
  (4) 006060  005037  006500                    CLR     ERRCHK
 9929                                                                           ;SET BREAK BIT
 9930 006064                                                                     LET @TCSR := @TCSR SET.BY #BREAK
  (7) 006064  052777  000001  173172            BIS     #BREAK,@TCSR
 9931                                                                           ;NON-ZERO CHAR. '*'
 9932 006072                                                                    LET @TBUF := #125
  (4) 006072  012777  000125  173166            MOV     #125,@TBUF
 9933                                                                           ;WAIT FOR DONE
 9934 006100                                                                    CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
  (3) 006100  010546                            MOV     R5,-(SP)
  (7) 006102  012745  177777                    MOV     #SET,-(R5)
  (6) 006106  013745  001260                    MOV     RCSR,-(R5)
  (5) 006112  012745  000200                    MOV     #RCVRDONE,-(R5)
  (4) 006116  012745  000500                    MOV     #500,-(R5)
  (3) 006122  004737  012446                    JSR     PC,TIMER
  (3) 006126  012605                            MOV     (SP)+,R5
 9935 006130                                                                    IF.ERROR THEN
```

```
  (6)  006130  103001                    BCC     $142
9936
9937  006132
  (1)  006132  104115                    ERROR   115
9938  006134
  (4)  006134                    $142:
9939  006134
  (4)  006134  017700  173122            MOV     @RBUF,R0
9940  006140
  (6)  006140  105700                    TSTB    R0
  (9)  006142  001403                    BEQ     $143
9941
9942  006144
  (7)  006144  052737  000001  006500    BIS     #BIT0,ERRCHK
9943  006152
  (4)  006152                    $143:
9944  006152
  (6)  006152  032700  020000            BIT     #FRERR,R0
  (9)  006156  001003                    BNE     $144
9945  006160
  (7)  006160  052737  000002  006500    BIS     #BIT1,ERRCHK
9946  006166
  (4)  006166                    $144:
9947  006166
  (6)  006166  032737  000002  001220    BIT     #PARITY,$USWR
  (9)  006174  001421                    BEQ     $145
9948
9949  006176
  (6)  006176  032737  000004  001220    BIT     #EVENODD,$USWR
  (9)  006204  001007                    BNE     $146
9950
9951  006206
  (6)  006206  032700  010000            BIT     #PERR,R0
  (9)  006212  001003                    BNE     $147
9952
9953  006214
  (7)  006214  052737  000004  006500    BIS     #BIT2,ERRCHK
9954  006222
  (4)  006222                    $147:
9955  006222
  (4)  006222  000406                    BR      $150
  (3)  006224                    $146:
9956  006224
  (6)  006224  032700  010000            BIT     #PERR,R0
  (9)  006230  001403                    BEQ     $151
9957  006232
  (7)  006232  052737  000010  005622    BIS     #BIT3,ERRCNT
9958  006240
  (4)  006240                    $151:
9959  006240
  (4)  006240                    $150:
9960  006240
  (4)  006240                    $145:
9961
9962  006240
  (1)  006240  000005                    RESET
```

```
    ; RECIEVER DONE DID NOT SET
    ERRHRD 115
ENDIF

LET R0 := @RBUF

IFB R0 NE #0 THEN

        ; BREAK DID NOT EQUAL 0
    LET ERRCHK := ERRCHK SET.BY #BIT0

ENDIF

IF #FRERR NOTSETIN R0 THEN

        LET ERRCHK := ERRCHK SET.BY #BIT1

ENDIF

IF #PARITY SETIN $USWR THEN

            ;ODD PARITY ENABLED
    IF #EVENODD NOTSETIN $USWR THEN

                ;BREAK SHOULD GENERATE A PARITY ERRO
        IF #PERR NOTSETIN R0 THEN

                    ;NO PAR ERROR WHEN THERE SHOULD
            LET ERRCHK := ERRCHK SET.BY #BIT2

        ENDIF

    ELSE

        IF #PERR SETIN R0 THEN

            LET ERRCNT := ERRCNT SET.BY #BIT3

        ENDIF

    ENDIF

ENDIF

BRESET  ;CLEAN UP
```

```
 9963  006242  032777  170000  173012          BIT     #170000,@RBUF
 9964  006250  001401                          BEQ     1$
 9965  006252  104033                          ERROR   33              ;RESET DID NOT CLEAR ERROR,FR ERR,OR PERR IN RBUF
 9966  006254                          1$:
 9967
 9968  006254                                                  IF #BIT0 SETIN ERRCHK THEN
  (6)  006254  032737  000001  006500          BIT     #BIT0,ERRCHK
  (9)  006262  001401                          BEQ     $152
 9969  006264                                                      ERRHRD 121  ;BREAK ERROR
  (1)  006264  104121                          ERROR   121
 9970  006266                                                  ENDIF
  (4)  006266                          $152:
 9971  006266                                                  IF #BIT1 SETIN ERRCHK THEN
  (6)  006266  032737  000002  006500          BIT     #BIT1,ERRCHK
  (9)  006274  001401                          BEQ     $153
 9972  006276                                                      ERRHRD 122  ; FRAMING ERROR
  (1)  006276  104122                          ERROR   122
 9973  006300                                                  ENDIF
  (4)  006300                          $153:
 9974  006300                                                  IF #BIT2 SETIN ERRCHK THEN
  (6)  006300  032737  000004  006500          BIT     #BIT2,ERRCHK
  (9)  006306  001401                          BEQ     $154
 9975  006310                                                      ERRHRD  235
  (1)  006310  104235                          ERROR   235
 9976                                                                 ;NO PARITY ERROR WHEN
 9977                                                                 ;THERE SHOULD BE
 9978  006312                                                  ENDIF
  (4)  006312                          $154:
 9979  006312                                                  IF #BIT3 SETIN ERRCHK THEN
  (6)  006312  032737  000010  006500          BIT     #BIT3,ERRCHK
  (9)  006320  001401                          BEQ     $155
 9980  006322                                                      ERRHRD 236
  (1)  006322  104236                          ERROR   236
 9981                                                                 ;PARITY ERROR SHOULD NOT HAVE
 9982                                                                 ;OCCURED WITH EVEN PARITY
 9983                                                                 ;ENABLED AND BREAK SET
 9984  006324                                                  ENDIF
  (4)  006324                          $155:
 9985  006324                                              ENDSUB
 9986  006324                                              BGNSUB
  (5)  006324  012737  006332  001110          MOV     #64$,$LPERR
 9987                                                                 ;SET BREAK BIT
 9988  006332                                                  LET @TCSR := @TCSR SET.BY #BREAK
  (7)  006332  052777  000001  172724          BIS     #BREAK,@TCSR
 9989
 9990  006340                                                  CALL TIMER IN <#500,#BREAK,TCSR,#SET>
  (3)  006340  010546                          MOV     R5,-(SP)
  (7)  006342  012745  177777                  MOV     #SET,-(R5)
  (6)  006346  013745  001264                  MOV     TCSR,-(R5)
  (5)  006352  012745  000001                  MOV     #BREAK,-(R5)
  (4)  006356  012745  000500                  MOV     #500,-(R5)
  (3)  006362  004737  012446                  JSR     PC,TIMER
  (3)  006366  012605                          MOV     (SP)+,R5
 9991
 9992                                                                 ;DID IT SET?
 9993  006370                                                  IF.ERROR THEN
```

```
   (6)  006370  103001                    BCC     $156
 9994
 9995  006372
   (1)  006372  104021                    ERROR   21
 9996  006374                                                          ENDIF
   (4)  006374                          $156:
 9997
 9998                                                                  ;CLEAR  BREAK BIT
 9999  006374                                                          LET @TCSR := @TCSR CLR.BY #BREAK
   (7)  006374  042777  000001  172662    BIC     #BREAK,@TCSR
10000  006402                                                          WAITMS 100.
   (4)  006402  010546                    MOV     R5,-(SP)
   (5)  006404  012745  000144            MOV     #100.,-(R5)
   (4)  006410  004737  012620            JSR     PC,WAIT
   (4)  006414  012605                    MOV     (SP)+,R5
10001
10002
10003                                                                  ;READ RBUF TO CLEAR ERRORS & REC DONE
10004  006416                                                          LET R0 := @RBUF
   (4)  006416  017700  172640            MOV     @RBUF,R0
10005                                                                  ;SEND CHAR
10006  006422                                                          LET @TBUF := #125
   (4)  006422  012777  000125  172636    MOV     #125,@TBUF
10007                                                                  ;WAIT FOR DONE BIT
10008  006430                                                          CALL  TIMER IN <#500,#RCVRDONE,RCSR,#SET>
   (3)  006430  010546                    MOV     R5,-(SP)
   (7)  006432  012745  177777            MOV     #SET,-(R5)
   (6)  006436  013745  001260            MOV     RCSR,-(R5)
   (5)  006442  012745  000200            MOV     #RCVRDONE,-(R5)
   (4)  006446  012745  000500            MOV     #500,-(R5)
   (3)  006452  004737  012446            JSR     PC,TIMER
   (3)  006456  012605                    MOV     (SP)+,R5
10009  006460                                                          IF.ERROR THEN
   (6)  006460  103001                    BCC     $157
10010                                                                  ;RECEIVER NEVER CAME READY
10011  006462                                                          ERRHRD 230
   (1)  006462  104230                    ERROR   230
10012  006464                                                          ENDIF
   (4)  006464                          $157:
10013
10014                                                                  ;WAS CHAR AFTER BREAK RECEIVED
10015  006464                                                          IFB @RBUF NE #125 THEN
   (6)  006464  127727  172572  000125    CMPB    @RBUF,#125
   (9)  006472  001401                    BEQ     $160
10016                                                                  ;CHAR AFTER BREAK NOT RECEIVED CORRECTLY
10017  006474                                                          ERRHRD 231
   (1)  006474  104231                    ERROR   231
10018  006476                                                          ENDIF
   (4)  006476                          $160:
10019  006476                                                  ENDSUB
10020  006476                                                  EXIT
   (3)  006476  000401                    BR      TST20           ;;EXIT THIS TEST
10021  006500  000000                  ERRCHK: .WORD 0
```

```
10023
10024
10029                                   ;;*******************************************************************
  (3)                                   ;*TEST 20       NOT A TEST - SEND BACK TO LOOP
  (3)                                   ;;*******************************************************************
  (2)  006502  000004                   TST20:  SCOPE
  (2)
  (1)  006504  012737  000001  001160           MOV     #1,$TIMES       ;;DO 1 ITERATION
10034  006512                                                           IF #BIT12 SETIN @SWR THEN
  (6)  006512  032777  010000  172420           BIT     #BIT12,@SWR
  (9)  006520  001513                           BEQ     $161
10035  006522                                                               IF PHASE2 EQ #TRUE  THEN
  (6)  006522  023727  014166  000001           CMP     PHASE2,#TRUE
  (9)  006530  001021                           BNE     $162
10036  006532                                   TYPTXT  <<CRLF>/        ** PHASE 2 SUMMARY **/<CRLF>>
10037  006572                                                             ELSE
  (4)  006572  000425                           BR      $163
  (3)  006574                           $162:
10038  006574                                                               IF P1CNT EQ #0 THEN
  (6)  006574  005737  014170                   TST     P1CNT
  (9)  006600  001022                           BNE     $164
10039  006602                                   TYPTXT  <<CRLF>/        ** PHASE 1 SUMMARY **/<CRLF>>
10040  006642                                                                 LET P1CNT := P1CNT + #1
  (7)  006642  005237  014170                   INC     P1CNT
10041  006646                                                               ENDIF
  (4)  006646                           $164:
10042  006646                                                             ENDIF
  (4)  006646                           $163:
10043  006646                                   TYPTXT  <*CSR: *>
10044  006662                                   TYPOCT  DLADD
10045  006670                                   TYPTXT  <*,VECTOR: *>
10046  006710                                   TYPOCT  DLVEC
10047  006716                                   TYPTXT  <*,ERRORS: *>
10048  006736                                   TYPDEC  $ERTTL
10049  006744  104401  001171                   TYPE    ,$CRLF
10050  006750                                                           ENDIF
  (4)  006750                           $161:
10051  006750  005037  001112                   CLR     $ERTTL          ; RESET FOR NEXT DEVICE/PASS
10052  006754                                                           IF PHASE2 EQ #TRUE THEN
  (6)  006754  023727  014166  000001           CMP     PHASE2,#TRUE
  (9)  006762  001004                           BNE     $165
10053  006764                                                             LET PHASE2 := #0
  (4)  006764  005037  014166                   CLR     PHASE2
10054  006770                                                             LET $UNIT := $UNIT + #1
  (7)  006770  005237  001206                   INC     $UNIT
10055  006774                                                           ENDIF
  (4)  006774                           $165:
10056  006774  000137  001732                   JMP     LOOP            ; BACK UP TO THE BEGINNING
```

```
10058
10059
10060  007000                          MODTST:
10070                                  ;;*****************************************************************
  (3)                                  ;*TEST 21       TEST THAT CHANNELS INTR AT ASSIGNED PRIORITY
  (4)                                  ;*             INTERRUPTS WILL BE ENABLED ON ALL ACTIVE CHANNELS.
  (4)                                  ;*             RECEIVER AND TRANSMITTER. THEN WE'LL CHECK TO
  (4)                                  ;*             SEE IF THEY INTERRUPTED IN THE ASSIGNED SEQUENCE.
  (3)                                  ;;*****************************************************************
  (2)  007000  000004                  TST21:  SCOPE
  (2)
  (1)  007002  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
  (2)  007010  012737  000021  001200          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10071
10076
10077                                                                 ;CLEAR OUT INTERRUPT TABLE
10078  007016                                                         LET R0 := #INTRTABLE
  (4)  007016  012700  007744                MOV     #INTRTABLE,R0
10079  007022                                                         REPEAT
  (3)  007022                          $166:
10080  007022                                                             LET (R0)+ := #0
  (4)  007022  005020                        CLR     (R0)+
10081  007024                                                         UNTIL R0 EQ #TABEND
  (3)  007024  020027  007764                CMP     R0,#TABEND
  (6)  007030  001374                        BNE     $166
10082
10083                                                                 ;SET PRIORITY TO 6
10084  007032                              SETPRI #PR6
10085
10086                                                                 ;SET UP ALL INTERRUPT VECTORS
10087  007044                                                         LET R0 := DLVEC
  (4)  007044  013700  001256                MOV     DLVEC,R0
10088  007050                                                         LET R1 := #RCVR0SRV
  (4)  007050  012701  007424                MOV     #RCVR0SRV,R1
10089  007054                                                         REPEAT
  (3)  007054                          $167:
10090  007054                                                             LET (R0)+ := R1
  (4)  007054  010120                        MOV     R1,(R0)+
10091  007056                                                             LET (R0)+ := #PR6
  (4)  007056  012720  000300                MOV     #PR6,(R0)+
10092  007062                                                             LET R1 := R1 + #30
  (7)  007062  062701  000030                ADD     #30,R1
10093  007066                                                         UNTIL R1 EQ #SRVEND
  (3)  007066  020127  007724                CMP     R1,#SRVEND
  (6)  007072  001370                        BNE     $167
10094
10095                                                                 ;ENABLE INTERRUPTS ON ALL ACTIVE LINES,
10096                                                                 ;ALSO, XMIT CHAR'S TO PRIME RECEIVERS
10097
10098                                                                 ;COPY MASK
10099  007074                                                         LET CHMASK := MASK
  (4)  007074  013737  014162  007764        MOV     MASK,CHMASK
10100  007102                                                         LET CHCNT := #0
  (4)  007102  005037  007766                CLR     CHCNT
10101  007106                                                         LET R0 := DLADD
  (4)  007106  013700  001254                MOV     DLADD,R0
```

```
10102  007112                                                  REPEAT
  (3)  007112                          $170:
10103  007112                                                      IF #BIT0 SETIN CHMASK THEN
  (6)  007112  032737  000001  007764        BIT     #BIT0,CHMASK
  (9)  007120  001430                        BEQ     $171
10104
10105  007122                                                          ;SET RCSR IE
                                                                       LET (R0) := (R0) SET.BY #BIT6
  (7)  007122  052710  000100               BIS     #BIT6,(R0)
10106  007126                                                          LET R0 := R0 + #4
  (7)  007126  062700  000004               ADD     #4,R0
10107                                                                  ;SET XCSR IE
10108  007132                                                          LET (R0) := (R0) SET.BY #BIT6
  (7)  007132  052710  000100               BIS     #BIT6,(R0)
10109                                                                  ;LOAD XBUF
10110  007136                                                          LET 2(R0) := #252
  (4)  007136  012760  000252  000002       MOV     #252,2(R0)
10111                                                                  ;GO BACK TO RCSR
10112  007144                                                          LET R0 := R0 - #4
  (7)  007144  162700  000004               SUB     #4,R0
10113                                                                  ;LOOK FOR RCVR DONE
10114  007150                                                          CALL TIMER IN <#500,#RCVRDONE,R0,#SET>
  (3)  007150  010546                       MOV     R5,-(SP)
  (7)  007152  012745  177777               MOV     #SET,-(R5)
  (6)  007156  010045                       MOV     R0,-(R5)
  (5)  007160  012745  000200               MOV     #RCVRDONE,-(R5)
  (4)  007164  012745  000500               MOV     #500,-(R5)
  (3)  007170  004737  012446               JSR     PC,TIMER
  (3)  007174  012605                       MOV     (SP)+,R5
10115                                                                          ;DID IT SET IN TIME?
10116  007176                                                          IF.ERROR THEN
  (6)  007176  103001                       BCC     $172
10117                                                                          ;RCVRDONE DID NOT SET IN RCSR
10118  007200                                                                  ERRHRD 23
  (1)  007200  104023                       ERROR   23
10119  007202                                                              ENDIF
  (4)  007202                          $172:
10120  007202                                                      ENDIF
  (4)  007202                          $171:
10121
10122  007202                                                          LET R0 := R0 + #10
  (7)  007202  062700  000010               ADD     #10,R0
10123                                                              ;SETUP FOR NEXT CHANNEL
10124  007206                                                      LET CHMASK := CHMASK SHIFT -1
  (8)  007206  006237  007764               ASR     CHMASK
10125  007212                                                      LET CHCNT := CHCNT + #1
  (7)  007212  005237  007766               INC     CHCNT
10126  007216                                                      IF CHCNT EQ #3 THEN
  (6)  007216  023727  007766  000003       CMP     CHCNT,#3
  (9)  007224  001007                       BNE     $173
10127  007226                                                          CALL TSTCON
  (3)  007226  004737  013530               JSR     PC,TSTCON
10128  007232                                                          IF CONSOLE NE #0 THEN
  (6)  007232  005737  013604               TST     CONSOLE
  (9)  007236  001402                       BEQ     $174
10129  007240                                                                  LET CHCNT := CHCNT + #1
  (7)  007240  005237  007766               INC     CHCNT
```

```
10130  007244                                                          ENDIF
  (4)  007244                          $174:
10131  007244                                                        ENDIF
  (4)  007244                          $173:
10132  007244                                                      UNTIL CHCNT EQ #4
  (3)  007244  023727  007766  000004          CMP     CHCNT,#4
  (6)  007252  001317                         BNE     $170
10133
10134                                         ;ALL XMIT & REC INTERRUPTS SHOULD BE
10135                                         ;IN CONTENTION.
10136                                         ;RECEIVE INTERRUPTS HAVE PRIORITY
10137                                         ;OVER XMIT INTERRUPTS.
10138                                         ;LOW CHANNELS HAVE PRIORITY OVER HIGH CHANNELS.
10139                                         ;THEREFORE, ONCE CPU PRIORITY IS LOWERED,
10140                                         ;INTERRUPTS S/B IN THE ORDER SHOWN IN THE
10141                                         ;CHANNEL IDENTIFIER TABLE (RCH0:)
10142
10143                                                  ;SET UP POINTER FOR SERVICE ROUTINE
10144  007254                                        LET R4 := #INTRTABLE
  (4)  007254  012704  007744                MOV     #INTRTABLE,R4
10145
10146                                                  ;LET EM GO
10147  007260                                SETPRI #PR0
10148
10149                                                  ;DISABLE ALL INTERRUPTS
10150  007272                                        BRESET
  (1)  007272  000005                         RESET
10151
10152  007274                                        LET TCNT := #0
  (4)  007274  005037  007772                CLR     TCNT
10153                                                 ;CLEAR ERR FLAG
10154  007300                                        LET PRIERR := #0
  (4)  007300  005037  007770                CLR     PRIERR
10155  007304                                        LET R0 := #INTRTABLE
  (4)  007304  012700  007744                MOV     #INTRTABLE,R0
10156                                                 ;SETUP EXPECTED VALUE
10157  007310                                        LET R1 := #INTRTABLE+2
  (4)  007310  012701  007746                MOV     #INTRTABLE+2,R1
10158  007314  013737  014162  007764         MOV     MASK,CHMASK     ;GET # OF CHANNELS IN TCNT
10159
10160  007322  006237  007764          1$:   ASR     CHMASK
10161  007326  103002                       BCC     2$
10162  007330  005237  007772               INC     TCNT            ;GET TOTAL CHANNEL CT
10163  007334  005737  007764          2$:   TST     CHMASK
10164  007340  001370                       BNE     1$
10165
10166  007342  004737  013530               JSR     PC,TSTCON       ;CORRECT CHANNEL CT FOR CONSOLE
10167  007346  005737  013604               TST     CONSOLE
10168  007352  001402                       BEQ     3$
10169  007354  005337  007772               DEC     TCNT
10170  007360  006337  007772          3$:  ASL     TCNT            ;APPLY X2 MULT FACTOR
10171
10172  007364  005337  007772          4$:  DEC     TCNT
10173  007370  001410                       BEQ     5$
10174  007372  023727  007772  177777       CMP     TCNT,#-1        ;ALL 1'S ?
10175  007400  001404                       BEQ     5$              ;BR IF YES
```

```
10176  007402  022120                  CMP     (R1)+,(R0)+     ;R1-R0
10177  007404  003367                  BGT     4$              ;BR IF R1>R0
10178  007406  005237  007770          INC     PRIERR          ;ELSE ERROR
10179
10180  007412                  5$:
10181  007412                                                  IF PRIERR NE #0 THEN
   (6) 007412  005737  007770          TST     PRIERR
   (9) 007416  001401                  BEQ     $175
10182  007420                                                      ERRHRD 250
   (1) 007420  104250                  ERROR   250
10183                                                           ;CHANNELS DID NOT INTR ACCORDING TO
10184                                                           ;ASSIGNED PRIORITY. THE BYTE ENTRIES
10185                                                           ;IN INTRTABLE: SHOULD BE IN THE ORDER
10186                                                           ;THAT THEY APPEAR IN THE CHANNEL #
10187                                                           ;TABLE .(EXCLUDING CHANNELS NOT ACTIVE)
10188  007422                                                  ENDIF
   (4) 007422                  $175:
10189
10190  007422                                                  EXIT
   (3) 007422  000565                  BR      TST22           ;;EXIT THIS TEST
```

```
10192
10193
10198                                   ;;*****************************************************************
10199                                   ;*              START OF SERVICE ROUTINES
10200                                   ;;*****************************************************************
10205
10206  007424                                                       BGNSRV RCVROSRV
  (1)  007424                           RCVROSRV:
10207
10208  007424                                                               ;PUT INTR IDENTIFIER IN INTRTABLE
  (4)  007424  013724  007724                 MOV     RCH0,(R4)+     LET (R4)+ := RCH0
10209
10210  007430                                                               ;GENERATE CSR ADDRESS
  (4)  007430  013737  001260  007774         MOV     RCSR,TEMP      LET TEMP := RCSR
10211  007436
  (7)  007436  062737  000000  007774         ADD     #0,TEMP        LET TEMP := TEMP + #0
10212
10213  007444                                                               ;ONE INTR IS ALL WE WANT FROM HERE
  (7)  007444  042777  000100  000322         BIC     #BIT6,@TEMP    LET @TEMP := @TEMP CLR.BY #BIT6
10214  007452                                                       ENDSRV
  (1)  007452  000002                         RTI
10215
10216
10217  007454                                                       BGNSRV  XMITOSRV
  (1)  007454                           XMITOSRV:
10218
10219  007454                                                               ;PUT INTR IDENTIFIER IN INTRTABLE
  (4)  007454  013724  007734                 MOV     TCH0,(R4)+     LET (R4)+ := TCH0
10220
10221  007460                                                               ;GENERATE CSR ADDRESS
  (4)  007460  013737  001260  007774         MOV     RCSR,TEMP      LET TEMP := RCSR
10222  007466
  (7)  007466  062737  000004  007774         ADD     #4,TEMP        LET TEMP := TEMP + #4
10223
10224  007474                                                               ;ONE INTR IS ALL WE WANT FROM HERE
  (7)  007474  042777  000100  000272         BIC     #BIT6,@TEMP    LET @TEMP := @TEMP CLR.BY #BIT6
10225  007502                                                       ENDSRV
  (1)  007502  000002                         RTI
10226
10227  007504                                                       BGNSRV  RCVR1SRV
  (1)  007504                           RCVR1SRV:
10228
10229  007504                                                               ;PUT INTR IDENTIFIER IN INTRTABLE
  (4)  007504  013724  007726                 MOV     RCH1,(R4)+     LET (R4)+ := RCH1
10230
10231  007510                                                               ;GENERATE CSR ADDRESS
  (4)  007510  013737  001260  007774         MOV     RCSR,TEMP      LET TEMP := RCSR
10232  007516
  (7)  007516  062737  000010  007774         ADD     #10,TEMP       LET TEMP := TEMP + #10
10233
10234  007524                                                               ;ONE INTR IS ALL WE WANT FROM HERE
  (7)  007524  042777  000100  000242         BIC     #BIT6,@TEMP    LET @TEMP := @TEMP CLR.BY #BIT6
10235  007532                                                       ENDSRV
  (1)  007532  000002                         RTI
10236
10237  007534                                                       BGNSRV  XMIT1SRV
```

```
  (1)  007534                          XMIT1SRV:
10238
10239  007534
  (4)  007534  013724  007736                MOV     TCH1,(R4)+
10240
10241  007540
  (4)  007540  013737  001260  007774        MOV     RCSR,TEMP
10242  007546
  (7)  007546  062737  000014  007774        ADD     #14,TEMP
10243
10244  007554
  (7)  007554  042777  000100  000212        BIC     #BIT6,@TEMP
10245  007562
  (1)  007562  000002                        RTI
10246
10247  007564
  (1)  007564                          RCVR2SRV:
10248
10249  007564
  (4)  007564  013724  007730                MOV     RCH2,(R4)+
10250
10251  007570
  (4)  007570  013737  001260  007774        MOV     RCSR,TEMP
10252  007576
  (7)  007576  062737  000020  007774        ADD     #20,TEMP
10253
10254  007604
  (7)  007604  042777  000100  000162        BIC     #BIT6,@TEMP
10255  007612
  (1)  007612  000002                        RTI
10256
10257  007614
  (1)  007614                          XMIT2SRV:
10258
10259  007614
  (4)  007614  013724  007740                MOV     TCH2,(R4)+
10260
10261  007620
  (4)  007620  013737  001260  007774        MOV     RCSR,TEMP
10262  007626
  (7)  007626  062737  000024  007774        ADD     #24,TEMP
10263
10264  007634
  (7)  007634  042777  000100  000132        BIC     #BIT6,@TEMP
10265  007642
  (1)  007642  000002                        RTI
10266
10267  007644
  (1)  007644                          RCVR3SRV:
10268
10269  007644
  (4)  007644  013724  007732                MOV     RCH3,(R4)+
10270
10271  007650
  (4)  007650  013737  001260  007774        MOV     RCSR,TEMP
10272  007656
```

```
                    ;PUT INTR IDENTIFIER IN INTRTABLE
        LET (R4)+ := TCH1

                    ;GENERATE CSR ADDRESS
        LET TEMP := RCSR

        LET TEMP := TEMP + #14

                    ;ONE INTR IS ALL WE WANT FROM HERE
        LET @TEMP := @TEMP CLR.BY #BIT6
ENDSRV

BGNSRV  RCVR2SRV
                    ;PUT INTR IDENTIFIER IN INTRTABLE
        LET (R4)+ := RCH2

                    ;GENERATE CSR ADDRESS
        LET TEMP := RCSR

        LET TEMP := TEMP + #20

                    ;ONE INTR IS ALL WE WANT FROM HERE
        LET @TEMP := @TEMP CLR.BY #BIT6
ENDSRV

BGNSRV  XMIT2SRV
                    ;PUT INTR IDENTIFIER IN INTRTABLE
        LET (R4)+ := TCH2

                    ;GENERATE CSR ADDRESS
        LET TEMP := RCSR

        LET TEMP := TEMP + #24

                    ;ONE INTR IS ALL WE WANT FROM HERE
        LET @TEMP := @TEMP CLR.BY #BIT6
ENDSRV

BGNSRV RCVR3SRV
                    ;PUT INTR IDENTIFIER IN INTRTABLE
        LET (R4)+ := RCH3

                    ;GENERATE CSR ADDRESS
        LET TEMP := RCSR

        LET TEMP := TEMP + #30
```

```
  (7)  007656  062737  000030  007774          ADD     #30,TEMP
10273                                                        ;ONE INTR IS ALL WE WANT FROM HERE
10274  007664                                             LET @TEMP := @TEMP CLR.BY #BIT6
  (7)  007664  042777  000100  000102          BIC     #BIT6,@TEMP
10275  007672                                     ENDSRV
  (1)  007672  000002                RTI
10276
10277  007674                                     BGNSRV  XMIT3SRV
  (1)  007674                XMIT3SRV:
10278                                                        ;PUT INTR IDENTIFIER IN INTRTABLE
10279  007674                                             LET (R4)+ := TCH3
  (4)  007674  013724  007742          MOV     TCH3,(R4)+
10280                                                        ;GENERATE CSR ADDRESS
10281  007700                                             LET TEMP := RCSR
  (4)  007700  013737  001260  007774          MOV     RCSR,TEMP
10282  007706                                             LET TEMP := TEMP + #34
  (7)  007706  062737  000034  007774          ADD     #34,TEMP
10283                                                        ;ONE INTR IS ALL WE WANT FROM HERE
10284  007714                                             LET @TEMP := @TEMP CLR.BY #BIT6
  (7)  007714  042777  000100  000052          BIC     #BIT6,@TEMP
10285  007722                                     ENDSRV
  (1)  007722  000002                RTI
10286  007724                     SRVEND:
```

```
10288
10289
10290
10291
10292                                   ;**CHANNEL IDENTIFIER TABLE**
10293                                   ;THE ENTRIES FROM THIS
10294                                   ;TABLE ARE PLACED IN THE INTRTABLE
10295                                   ;IN THE ORDER THAT THE INTR'S
10296                                   ;OCCUR. (EXCLUDING NON-ACTIVE CH'S)
10297  007724  000000             RCH0:    .WORD 0
10298  007726  000001             RCH1:    .WORD 1
10299  007730  000002             RCH2:    .WORD 2
10300  007732  000003             RCH3:    .WORD 3
10301  007734  000010             TCH0:    .WORD 10
10302  007736  000011             TCH1:    .WORD 11
10303  007740  000012             TCH2:    .WORD 12
10304  007742  000013             TCH3:    .WORD 13
10305
10306                                   ;THIS TABLE WILL CONTAIN ENTRIES
10307                                   ;REPRESENTING EACH INTR IN THE ORDER
10308                                   ;THAT IT OCCURED
10309                                   ; S/B IN THE ABOVE ORDER
10310                                   ;
10311  007744  000010             INTRTABLE:      .BLKW   8.
10312  007764                     TABEND:
10313
10314  007764  000000             CHMASK: .WORD 0         ;BITS 3-0 INDICATE WHICH CH'S ARE ACTIVE FOR THIS TEST
10315  007766  000000             CHCNT:  .WORD 0         ;CHANNEL COUNTER
10316  007770  000000             PRIERR: .WORD 0         ;ERROR WORD, CONTAINS NO. OF INTR OUT OF ORDER
10317  007772  000000             TCNT:   .WORD 0         ;TABLE PASS CTR
10318  007774  000000             TEMP:   .WORD 0         ;FOR SERVICE ROUTINES
10319
```

```
10321
10322
10338                                   ;;*****************************************************************
  (3)                                   ;*TEST 22       TEST DATA TRANSFERS WITH ALL ACTIVE LINES INTERRUPTING
  (4)                                   ;*             IN THIS WE'LL ENABLE INTERRUP  ON ALL CHANNELS
  (4)                                   ;*             FOR THIS DLV11-J. THEN WE'LL X...  AN INCREMENTING
  (4)                                   ;*             DATA PATERN VIA INTERRUPTS AND RECORD THE RECEIVER
  (4)                                   ;*             INTR. IN THE RECEIVER STATUS TABLE.
  (4)                                   ;*       NOTE: DOUBLE BUFFERING CANNOT BE TESTED AT ITS MAX SPEED
  (4)                                   ;*             BECAUSE OF APT CONSIDERATIONS. I.E. APT SENDS
  (4)                                   ;*             'BREAKS' WHICH CAUSE OVERRUN ERRORS. THEREFORE
  (4)                                   ;*             THE XMIT IE IS NOT ENABLED AGAIN UNTIL THE RECVR
  (4)                                   ;*             HAS OBTAINED THE PREVIOUS WORD.
  (3)                                   ;*****************************************************************
  (2)  007776  000004                   TST22:  SCOPE
  (2)
  (1)  010000  012737  000001  001160           MOV     #1,$TIMES       ;;DO 1 ITERATION
  (2)  010006  012737  000022  001200           MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
10343                                                                       ;CLR ERROR WORD,
10344  010014                                                           LET ERWRD := #0
  (4)  010014  005037  012366                   CLR     ERWRD
10345
10346
10347                                                                       ;CLEAR OUT RECEIVER TABLES
10348  010020                                                           LET R0 := #CHOTAB
  (4)  010020  012700  011360                   MOV     #CHOTAB,R0
10349  010024                                                           REPEAT
  (3)  010024                           $176:
10350  010024                                                               LET (R0)+ := #0
  (4)  010024  005020                           CLR     (R0)+
10351  010026                                                           UNTIL R0 EQ #STATEND
  (3)  010026  020027  012360                   CMP     R0,#STATEND
  (6)  010032  001374                           BNE     $176
10352
10353                                                                       ;SET UP ALL VECTORS
10354  010034                                                           LET R1 := #R0SRV
  (4)  010034  012701  010660                   MOV     #R0SRV,R1
10355  010040                                                           LET R0 := DLVEC
  (4)  010040  013700  001256                   MOV     DLVEC,R0
10356  010044                                                           REPEAT
  (3)  010044                           $177:
10357  010044                                                               LET (R0)+ := R1
  (4)  010044  010120                           MOV     R1,(R0)+
10358  010046                                                               LET (R0)+ := #PR6
  (4)  010046  012720  000300                   MOV     #PR6,(R0)+
10359  010052                                                               LET R1 := R1 + #54
  (7)  010052  062701  000054                   ADD     #54,R1
10360  010056                                                               LET (R0)+ := R1
  (4)  010056  010120                           MOV     R1,(R0)+
10361  010060                                                               LET (R0)+ := #PR6
  (4)  010060  012720  000300                   MOV     #PR6,(R0)+
10362  010064                                                               LET R1 := R1 + #44
  (7)  010064  062701  000044                   ADD     #44,R1
10363  010070                                                           UNTIL R1 EQ #SEREND
  (3)  010070  020127  011360                   CMP     R1,#SEREND
  (6)  010074  001363                           BNE     $177
```

```
10364
10365  010076                                  SETPRI #PR6
10366
10367
10368
10369  010110                                                    ;SET PRIORTY TO 6
  (4)  010110  013700  001254                  MOV     DLADD,R0
10370                                                           ;ENABLE INTR'S ON ALL ACTIVE LINES
10371  010114                                                    ;THIS MODULE
  (4)  010114  013737  014162  012360          MOV     MASK,CHMSK
10372  010122                                                   LET R0 := DLADD
  (4)  010122  005037  012362                  CLR     CHCTR
10373                                                           ;INITIALIZE
10374  010126                                                   LET CHMSK := MASK
  (4)  010126  005037  012364                  CLR     ACTCH
10375  010132                                                   LET CHCTR := #0
  (3)  010132                        $200:
10376  010132                                                   ;ACTIVE CHANNEL CTR
  (6)  010132  032737  000001  012360          BIT     #BIT0,CHMSK
  (9)  010140  001413                          BEQ     $201
10377                                                          LET ACTCH := #0
10378  010142
  (7)  010142  005237  012364                  INC     ACTCH
10379                                                          REPEAT
10380  010146
  (7)  010146  052710  000100                  BIS     #BIT6,(R0)
10381  010152                                                      IF #BIT0 SETIN CHMSK THEN
  (7)  010152  062700  000004                  ADD     #4,R0
10382
10383  010156                                                          ;# OF ACTIVE CH
  (7)  010156  052710  000100                  BIS     #BIT6,(R0)     LET ACTCH := ACTCH + #1
10384  010162
  (7)  010162  062700  000004                  ADD     #4,R0          ;SET RCSR IE
10385  010166                                                         LET (R0) := (R0) SET.BY #BIT6
  (4)  010166  000402                          BR      $202
  (3)  010170                        $201:                           LET R0 := R0 + #4
10386  010170
  (7)  010170  062700  000010                  ADD     #10,R0         ;SET XCSR IE
10387  010174                                                         LET (R0) := (R0) SET.BY #BIT6
  (4)  010174                        $202:
10388  010174                                                         LET R0 := R0 + #4
  (8)  010174  006237  012360                  ASR     CHMSK
10389                                                         ELSE
10390
10391  010200                                                         LET R0 := R0 + #10
  (7)  010200  005237  012362                  INC     CHCTR
10392  010204                                                  ENDIF
  (6)  010204  023727  012362  000003          CMP     CHCTR,#3
  (9)  010212  001007                          BNE     $203       LET CHMSK := CHMSK SHIFT -1
10393  010214
  (3)  010214  004737  013530                  JSR     PC,TSTCON
10394  010220                                                     LET CHCTR := CHCTR + #1
  (6)  010220  005737  013604                  TST     CONSOLE
  (9)  010224  001402                          BEQ     $204       IF CHCTR EQ #3 THEN
10395  010226
  (7)  010226  005237  012362                  INC     CHCTR
                                                                     CALL TSTCON
                                                                     IF CONSOLE NE #0 THEN

                                                                       LET CHCTR := CHCTR + #1
```

```
10396  010232                                                                  ENDIF
  (4)  010232                          $204:
10397  010232                                                              ENDIF
  (4)  010232                          $203:
10398  010232                                                          UNTIL CHCTR EQ #4
  (3)  010232  023727  012362  000004          CMP     CHCTR,#4
  (6)  010240  001334                          BNE     $200
10399
10400                                                                   ;INIT BUFFER POINTERS FOR SERVICE ROUTINES
10401  010242                                                           LET R0 := #CH0TAB
  (4)  010242  012700  011360                  MOV     #CH0TAB,R0
10402  010246                                                           LET R1 := #CH1TAB
  (4)  010246  012701  011560                  MOV     #CH1TAB,R1
10403  010252                                                           LET R2 := #CH2TAB
  (4)  010252  012702  011760                  MOV     #CH2TAB,R2
10404  010256                                                           LET R3 := #CH3TAB
  (4)  010256  012703  012160                  MOV     #CH3TAB,R3
10405
10406                                                                   ;FILL XBUF WORDS
10407                                                                   ;FROM TMP0 THRU TMP3E
10408                                                                   ;START WORD CH 0 ALWAYS 0
10409  010262                                                           LET TMP0 := #0
  (4)  010262  005037  012372                  CLR     TMP0
10410                                                                   ;OFFSET TO INDFX
10411  010266                                                           LET R4 := #0
  (4)  010266  005004                          CLR     R4
10412                                                                   ;PTR TO TABLE TO BE FILLED
10413  010270                                                           LET R5 := #TMP0E
  (4)  010270  012705  012374                  MOV     #TMP0E,R5
10414  010274                                                           REPEAT
  (3)  010274                          $205:
10415  010274                                                             IF #BIT0 NOTSETIN $USWR THEN
  (6)  010274  032737  000001  001220          BIT     #BIT0,$USWR
  (9)  010302  001005                          BNE     $206
10416  010304                                                                LET (R5)+ := TABL7(R4)
  (4)  010304  016425  012426                  MOV     TABL7(R4),(R5)+
10417  010310                                                                LET (R5)+ := TABL7(R4)
  (4)  010310  016425  012426                  MOV     TABL7(R4),(R5)+
10418  010314                                                             ELSE
  (4)  010314  000404                          BR      $207
  (3)  010316                          $206:
10419  010316                                                                LET (R5)+ := TABL8(R4)
  (4)  010316  016425  012436                  MOV     TABL8(R4),(R5)+
10420  010322                                                                LET (R5)+ := TABL8(R4)
  (4)  010322  016425  012436                  MOV     TABL8(R4),(R5)+
10421  010326                                                             ENDIF
  (4)  010326                          $207:
10422  010326                                                                LET R4 := R4 + #2
  (7)  010326  062704  000002                  ADD     #2,R4
10423  010332                                                           UNTIL R4 EQ #10
  (3)  010332  020427  000010                  CMP     R4,#10
  (6)  010336  001356                          BNE     $205
10424
10425                                                                   ;INIT TMP4
10426  010340                                                           ;WILL BE DECR BY XMIT SERVICE ROUTINES TILL 0
  (4)  010340  013737  012364  012414          MOV     ACTCH,TMP4      LET TMP4 := ACTCH
```

```
10427
10428  010346                                  SETPRI #PR0                     ;SET PRIORITY TO 0
10429
10430  010360                                                                  REPEAT
  (3)  010360                          $210:
10431  010360                                                                  UNTIL TMP4 EQ #0
  (3)  010360  005737  012414                  TST     TMP4
  (6)  010364  001375                          BNE     $210
10432
10433                                                                         ;WAIT FOR RECV'S TO CATCH UP
10434  010366                                                                  LET TMP4 := #100000
  (4)  010366  012737  100000  012414          MOV     #100000,TMP4
10435  010374                                                                  REPEAT
  (3)  010374                          $211:
10436  010374                                                                     LET TMP4 := TMP4 - #1
  (7)  010374  005337  012414                  DEC     TMP4
10437  010400                                                                  UNTIL TMP4 EQ #0
  (3)  010400  005737  012414                  TST     TMP4
  (6)  010404  001373                          BNE     $211
10438
10439                                                                         ;DISABLE ALL INTR'S
10440  010406                                                                  BRESET
  (1)  010406  000005                          RESET
10441
10442                                                                         ;CHECK RECEIVER TABLE
10443                                                                         ;RESET TMP0,1,2,3
10444  010410                                                                  LET TMP0 := #0
  (4)  010410  005037  012372                  CLR     TMP0
10445  010414                                                                  LET TMP1 := TMP0E
  (4)  010414  013737  012374  012376          MOV     TMP0E,TMP1
10446  010422                                                                  LET TMP2 := TMP1E
  (4)  010422  013737  012400  012402          MOV     TMP1E,TMP2
10447  010430                                                                  LET TMP3 := TMP2E
  (4)  010430  013737  012404  012406          MOV     TMP2E,TMP3
10448
10449  010436                                                                  ;INITIALIZE
  (4)  010436  005037  012362                  CLR     CHCTR                   LET CHCTR := #0
10450  010442                                                                  LET CHMSK := MASK
  (4)  010442  013737  014162  012360          MOV     MASK,CHMSK
10451  010450                                                                REPEAT
  (3)  010450                          $212:
10452  010450                                                                  IF #BIT0 SETIN CHMSK THEN
  (6)  010450  032737  000001  012360          BIT     #BIT0,CHMSK
  (9)  010456  001436                          BEQ     $213
10453  010460                                                                     LET R0 := CHCTR
  (4)  010460  013700  012362                  MOV     CHCTR,R0
10454                                                                           ;OFFSET TO INDEX (2X)
10455  010464                                                                   LET R0 := R0 SHIFT +1
  (8)  010464  006300                          ASL     R0
10456                                                                           ;GET TABLE LOC FOR THAT CH.
10457  010466                                                                   LET R2 := TABL6(R0)
  (4)  010466  016002  012416                  MOV     TABL6(R0),R^
10458                                                                           ;OFFSET TO INDEX (4X)
10459  010472                                                                   LET R0 := R0 SHIFT +1
  (8)  010472  006300                          ASL     R0
10460                                                                           ;GET EXPECTED STARTING WORD
```

```
10461  010474
  (4)  010474  016001  012372              MOV     TMP0(R0),R1
10462  010500
  (7)  010500  062700  000002              ADD     #2,R0
10463
10464  010504
  (4)  010504  016003  012372              MOV     TMP0(R0),R3
10465
10466  010510
  (3)  010510                      $214:
10467  010510
  (6)  010510  032712  100000              BIT     #BIT15,(R2)
  (9)  010514  001403                      BEQ     $215
10468  010516
  (7)  010516  052737  000400  012366      BIS     #BIT8,ERWRD
10469  010524
  (4)  010524                      $215:
10470
10471  010524
  (6)  010524  121201                      CMPB    (R2),R1
  (9)  010526  001402                      BEQ     $216
10472  010530
  (7)  010530  005237  012366              INC     ERWRD
10473  010534
  (4)  010534                      $216:
10474
10475  010534
  (7)  010534  005201                      INC     R1
10476
10477  010536
  (7)  010536  062702  000002              ADD     #2,R2
10478
10479  010542
  (4)  010542  020103                      CMP     R1,R3
  (6)  010544  001403                      BEQ     $217
  (4)  010546  005737  012366              TST     ERWRD
  (7)  010552  001756                      BEQ     $214
  (4)  010554                      $217:
10480  010554
  (4)  010554                      $213:
10481  010554
  (7)  010554  005237  012362              INC     CHCTR
10482  010560
  (8)  010560  006237  012360              ASR     CHMSK
10483  010564
  (6)  010564  023727  012362  000003      CMP     CHCTR,#3
  (9)  010572  001010                      BNE     $220
10484  010574
  (3)  010574  004737  013530              JSR     PC,TSTCON
10485  010600
  (6)  010600  023727  013604  000001      CMP     CONSOLE,#TRUE
  (9)  010606  001002                      BNE     $221
10486
10487  010610
  (7)  010610  005237  012362              INC     CHCTR
10488  010614
```

```
LET R1 := TMP0(R0)

LET R0 := R0 + #2

;GET MAX WORD FOR THAT CH.
LET R3 := TMP0(R0)

REPEAT

    IF #BIT15 SETIN (R2) THEN

        LET ERWRD := ERWRD SET.BY #BIT8

    ENDIF

        ;CHECK FOR DATA COMPARE ERROR
    IFB (R2) NE R1 THEN

        LET ERWRD := ERWRD + #1

    ENDIF

        ;BUMP EXPECTED DATA WORD
        LET R1 := R1 + #1

        ;BUMP TABLE PTR
        LET R2 := R2 + #2

UNTIL R1 EQ R3 OR ERWRD NE #0

ENDIF

 LET CHCTR := CHCTR + #1

 LET CHMSK := CHMSK SHIFT -1

      IF CHCTR EQ #3 THEN

          CALL TSTCON

         IF CONSOLE EQ #TRUE THEN

             ;EXIT
             LET CHCTR := CHCTR + #1

         ENDIF
```

```
  (4)  010614                          $221:                                                  ENDIF
10489  010614
  (4)  010614                          $220:                                        UNTIL CHCTR EQ #4 OR ERWRD NE #0
10490  010614
  (4)  010614  023727  012362  000004          CMP     CHCTR,#4
  (6)  010622  001403                          BEQ     $222
  (4)  010624  005737  012366                  TST     ERWRD
  (7)  010630  001707                          BEQ     $212
  (4)  010632                          $222:
10491
10492  010632                                                      IF #BIT8 SETIN ERWRD THEN
  (6)  010632  032737  000400  012366          BIT     #BIT8,ERWRD
  (9)  010640  001401                          BEQ     $223
10493                                                                   ;ERROR FLAG UP AFTER TRANSFER
10494  010642                                                       ERRHRD  270
  (1)  010642  104270                          ERROR   270
10495  010644                                                      ENDIF
  (4)  010644                          $223:
10496  010644                                                      IFB ERWRD NE #0 THEN
  (6)  010644  105737  012366                  TSTB    ERWRD
  (9)  010650  001401                          BEQ     $224
10497                                                                   ;DATA COMPARE ERROR
10498  010652                                                       ERRHRD  271
  (1)  010652  104271                          ERROR   271
10499  010654                                                      ENDIF
  (4)  010654                          $224:
10500
10501
10502  010654  000137  006502                  JMP     TST20                  ;EXIT TEST
```

```
10504
10505
10510                                   ;;*****************************************************************
10511                                   ;*                START OF SERVICE ROUTINES
10512                                   ;;*****************************************************************
10517
10518  010660                                                          BGNSRV ROSRV
  (1)  010660                           ROSRV:
10519  010660                                                              LET TMP := RCSR
  (4)  010660  013737  001260  012370           MOV     RCSR,TMP
10520  010666                                                              LET TMP := TMP + #2
  (7)  010666  062737  000002  012370           ADD     #2,TMP
10521  010674                                                              LET (R0)+ := @TMP
  (4)  010674  017720  001470                   MOV     @TMP,(R0)+
10522  010700                                                              IF TMP0 NE TMP0E THEN
  (6)  010700  023737  012372  012374           CMP     TMP0,TMP0E
  (9)  010706  001407                           BEQ     $225
10523                                                                           ;GO TO XCSR
10524  010710                                                                   LET TMP := TMP + #2
  (7)  010710  062737  000002  012370           ADD     #2,TMP
10525                                                                           ;ENABLE XMIT INTERR
10526  010716                                                                   LET @TMP := @TMP SET.BY #BIT6
  (7)  010716  052777  000100  001444           BIS     #BIT6,@TMP
10527  010724                                                              ELSE
  (4)  010724  000402                           BR      $226
  (3)  010726                           $225:
10528                                                                           ;ALL DONE
10529  010726                                                                   LET TMP4 := TMP4 - #1
  (7)  010726  005337  012414                   DEC     TMP4
10530  010732                                                              ENDIF
  (4)  010732                           $226:
10531  010732                                                          ENDSRV
  (1)  010732  000002                           RTI
10532
10533  010734                                                          BGNSRV XOSRV
  (1)  010734                           XOSRV:
10534  010734                                                              LET TMP := RCSR
  (4)  010734  013737  001260  012370           MOV     RCSR,TMP
10535  010742                                                              LET TMP := TMP + #6
  (7)  010742  062737  000006  012370           ADD     #6,TMP
10536  010750                                                              LET @TMP := TMP0
  (4)  010750  013777  012372  001412           MOV     TMP0,@TMP
10537  010756                                                              LET TMP0 := TMP0 + #1
  (7)  010756  005237  012372                   INC     TMP0
10538                                                                           ;GO BACK TO XCSR
10539  010762                                                                   LET TMP := TMP - #2
  (7)  010762  162737  000002  012370           SUB     #2,TMP
10540                                                                           ;DISABLE XMIT INTERRUPTS,
10541                                                                           ;NOT EXER DOUBLE BUFFERING
10542  010770                                                                   LET @TMP := @TMP CLR.BY #BIT6
  (7)  010770  042777  000100  001372           BIC     #BIT6,@TMP
10543  010776                                                          ENDSRV
  (1)  010776  000002                           RTI
10544
10545  011000                                                          BGNSRV R1SRV
  (1)  011000                           R1SRV:
```

```
10546  011000
  (4)  011000  013737  001260  012370          MOV     RCSR,TMP
10547  011006
  (7)  011006  062737  000012  012370          ADD     #12,TMP
10548  011014
  (4)  011014  017721  001350                  MOV     @TMP,(R1)+
10549  011020
  (6)  011020  023737  012376  012400          CMP     TMP1,TMP1E
  (9)  011026  001407                          BEQ     $227
10550
10551  011030
  (7)  011030  062737  000002  012370          ADD     #2,TMP
10552
10553  011036
  (7)  011036  052777  000100  001324          BIS     #BIT6,@TMP
10554  011044
  (4)  011044  000402                          BR      $230
  (3)  011046                          $227:
10555
10556  011046
  (7)  011046  005337  012414                  DEC     TMP4
10557  011052
  (4)  011052                          $230:
10558  011052
  (1)  011052  000002                          RTI
10559
10560  011054
  (1)  011054                          X1SRV:
10561  011054
  (4)  011054  013737  001260  012370          MOV     RCSR,TMP
10562  011062
  (7)  011062  062737  000016  012370          ADD     #16,TMP
10563  011070
  (4)  011070  013777  012376  001272          MOV     TMP1,@TMP
10564  011076
  (7)  011076  005237  012376                  INC     TMP1
10565
10566  011102
  (7)  011102  162737  000002  012370          SUB     #2,TMP
10567
10568
10569  011110
  (7)  011110  042777  000100  001252          BIC     #BIT6,@TMP
10570  011116
  (1)  011116  000002                          RTI
10571
10572  011120
  (1)  011120                          R2SRV:
10573  011120
  (4)  011120  013737  001260  012370          MOV     RCSR,TMP
10574  011126
  (7)  011126  062737  000022  012370          ADD     #22,TMP
10575  011134
  (4)  011134  017722  001230                  MOV     @TMP,(R2)+
10576  011140
  (6)  011140  023737  012402  012404          CMP     TMP2,TMP2E
```

```
        LET TMP := RCSR

        LET TMP := TMP + #12

        LET (R1)+ := @TMP

        IF TMP1 NE TMP1E THEN

            ;GO TO XCSR
            LET TMP := TMP + #2

            ;ENABLE XMIT INTERR
            LET @TMP := @TMP SET.BY #BIT6

        ELSE

            ;ALL DONE
            LET TMP4 := TMP4 - #1

        ENDIF

    ENDSRV

    BGNSRV X1SRV

        LET TMP := RCSR

        LET TMP := TMP + #16

        LET @TMP := TMP1

        LET TMP1 := TMP1 + #1

            ;GO BACK TO XCSR
        LET TMP := TMP - #2

            ;DISABLE XMIT INTERRUPTS,
            ;NOT EXER DOUBLE BUFF
        LET @TMP := @TMP CLR.BY #BIT6

    ENDSRV

    BGNSRV R2SRV

        LET TMP := RCSR

        LET TMP := TMP + #22

        LET (R2)+ := @TMP

        IF TMP2 NE TMP2E THEN
```

```
  (9)  011146  001407                    BEQ     $231
10577
10578  011150
  (7)  011150  062737  000002  012370    ADD     #2,TMP
10579
10580  011156
  (7)  011156  052777  000100  001204    BIS     #BIT6,@TMP
10581  011164
  (4)  011164  000402                    BR      $232
  (3)  011166                    $231:
10582
10583  011166
  (7)  011166  005337  012414            DEC     TMP4
10584  011172
  (4)  011172                    $232:
10585  011172
  (1)  011172  000002                    RTI
10586
10587  011174
  (1)  011174                    X2SRV:
10588  011174
  (4)  011174  013737  001260  012370    MOV     RCSR,TMP
10589  011202
  (7)  011202  062737  000026  012370    ADD     #26,TMP
10590  011210
  (4)  011210  013777  012402  001152    MOV     TMP2,@TMP
10591  011216
  (7)  011216  005237  012402            INC     TMP2
10592
10593  011222
  (7)  011222  162737  000002  012370    SUB     #2,TMP
10594
10595
10596  011230
  (7)  011230  042777  000100  001132    BIC     #BIT6,@TMP
10597  011236
  (1)  011236  000002                    RTI
10598
10599  011240
  (1)  011240                    R3SRV:
10600  011240
  (4)  011240  013737  001260  012370    MOV     RCSR,TMP
10601  011246
  (7)  011246  062737  000032  012370    ADD     #32,TMP
10602  011254
  (4)  011254  017723  001110            MOV     @TMP,(R3)+
10603  011260
  (6)  011260  023737  012406  012410    CMP     TMP3,TMP3E
  (9)  011266  001407                    BEQ     $233
10604
10605  011270
  (7)  011270  062737  000002  012370    ADD     #2,TMP
10606
10607  011276
  (7)  011276  052777  000100  001064    BIS     #BIT6,@TMP
10608  011304
```

```
                    ;GO TO XCSR
                    LET TMP := TMP + #2

                    ;ENABLE XMIT INTERR
                    LET @TMP := @TMP SET.BY #BIT6
                ELSE

                    ;ALL DONE
                    LET TMP4 := TMP4 - #1
                ENDIF
            ENDSRV

            BGNSRV X2SRV

                LET TMP := RCSR

                LET TMP := TMP + #26

                LET @TMP := TMP2

                LET TMP2 := TMP2 + #1

                    ;GO BACK TO XCSR
                LET TMP := TMP - #2

                    ;DISABLE XMIT INTERRUPTS,
                    ;NOT EXER DOUBLE BUFF.
                LET @TMP := @TMP CLR.BY #BIT6
            ENDSRV

            BGNSRV R3SRV

                LET TMP := RCSR

                LET TMP := TMP + #32

                LET (R3)+ := @TMP

                IF TMP3 NE TMP3E THEN

                    ;GO TO XCSR
                    LET TMP := TMP + #2

                    ;ENABLE XMIT INTERR
                    LET @TMP := @TMP SET.BY #BIT6
                ELSE
```

```
   (4)  011304  000402                        BR      $234
   (3)  011306                         $233:
10609
10610  011306                                                    ;ALL DONE
   (7)  011306  005337  012414                DEC     TMP4      LET TMP4 := TMP4 - #1
10611  011312                                                ENDIF
   (4)  011312                         $234:
10612  011312                                                ENDSRV
   (1)  011312  000002                        RTI
10613
10614  011314                                                BGNSRV X3SRV
   (1)  011314                         X3SRV:
10615  011314                                                    LET TMP := RCSR
   (4)  011314  013737  001260  012370        MOV     RCSR,TMP
10616  011322                                                    LET TMP := TMP + #36
   (7)  011322  062737  000036  012370        ADD     #36,TMP
10617  011330                                                    LET @TMP := TMP3
   (4)  011330  013777  012406  001032        MOV     TMP3,@TMP
10618  011336                                                    LET TMP3 := TMP3 + #1
   (7)  011336  005237  012406                INC     TMP3
10619                                                         ;GO BACK TO XCSR
10620  011342                                                LET TMP := TMP - #2
   (7)  011342  162737  000002  012370        SUB     #2,TMP
10621                                                         ;DISABLE XMIT INTERRUPTS,
10622                                                         ;NOT EXER DOUBLE BUFF
10623  011350                                                LET @TMP := @TMP CLR.BY #BIT6
   (7)  011350  042777  000100  001012        BIC     #BIT6,@TMP
10624  011356                                                ENDSRV
   (1)  011356  000002                        RTI
10625  011360                         SEREND:
```

```
10627
10628
10629                                   ;** RECEIVER STATUS TABLES **
10630                                                                   ;7 BIT WDS      8 BIT WDS
10631  011360  000100                  CH0TAB: .BLKW 100               ;  0- 37          0- 77
10632  011560  000100                  CH1TAB: .BLKW 100               ; 40- 77        100-177
10633  011760  000100                  CH2TAB: .BLKW 100               ;100-137        200-277
10634  012160  000100                  CH3TAB: .BLKW 100               ;140-177        300-377
10635  012360                          STATEND:
10636  012360  000000                  CHMSK:  .WORD 0                 ;MASK OF ACTIVE CHANNELS
10637  012362  000000                  CHCTR:  .WORD 0                 ;CHANNEL CTR
10638  012364  000000                  ACTCH:  .WORD 0                 ;ACTIVE CHANNEL CTR
10639  012366  000000                  ERWRD:  .WORD 0                 ;ERROR WORD, LOW BYTE IS NO. OF DATA ERRORS
10640                                                                  ;HIGH BYTE INDICATES ERROR FLAG DETECTED
10641
10642  012370  000000                  TMP:    .WORD 0                 ;TEMP STORAGE FOR SERVICE ROUTINES
10643
10644  012372  000000                  TMP0:   0                       ;START WORD FOR CH 0
10645  012374  000000                  TMP0E:  0                       ;END   WORD FOR CH 0
10646  012376  000000                  TMP1:   0                       ;START              1
10647  012400  000000                  TMP1E:  0                       ;END                1
10648  012402  000000                  TMP2:   0                       ;START              2
10649  012404  000000                  TMP2E:  0                       ;END                2
10650  012406  000000                  TMP3:   0                       ;START              3
10651  012410  000002                  TMP3E:  .BLKW 2                 ;END                3
10652
10653  012414  000000                  TMP4:   .WORD
10654
10655  012416  011360                  TABL6:  CH0TAB
10656  012420  011560                          CH1TAB
10657  012422  011760                          CH2TAB
10658  012424  012160                          CH3TAB
10659
10660  012426  000040                  TABL7:  40                      ;7 BIT WORD TABLE
10661  012430  000100                          100
10662  012432  000140                          140
10663  012434  000200                          200
10664
10665  012436  000100                  TABL8:  100                     ;8 BIT WORD TABLE
10666  012440  000200                          200
10667  012442  000300                          300
10668  012444  000400                          400
10669
10670
```

```
10672
10673
10678                                   ;;*****************************************************************
10679                                   ;* ROUTINE:TIMER
10680                                   ;*       THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
10681                                   ;*       IN ANY REGISTER.
10682                                   ;*       INPUTS:
10683                                   ;*            HOWLONG    THE MAXIMUM AMOUNT OF TIME TO SPEND IN
10684                                   ;*                       THIS ROUTINE.
10685                                   ;*            WHICHBIT     A MASK WITH THE BIT(S) SET THAT ARE
10686                                   ;*                         TO BE CHECKED.
10687                                   ;*            REG     A POINTER TO THE REGISTER TO BE CHECKED
10688                                   ;*            SETCLR  THE DESIRED RESULTS
10689                                   ;*                    EITHER #SET OR #CLEAR
10690                                   ;*       OUTPUT:
10691                                   ;*            THE 'C' BIT IS SET TO INDICATE AN ERROR
10692                                   ;*            BUT IT IS TESTED BY THE IF.ERROR STATEMENT
10693                                   ;*
10694                                   ;*
10695                                   ;*            NOTE::  THE USE OF (R5) IS PART OF THE LINKAGE
10696                                   ;*                    MECHANISM BETWEEN THE CALLER AND THE CALLED
10697                                   ;*****************************************************************
10698
10699
10704  012446                                                ROUTINE TIMER <HOWLONG,WHICHBIT,REG,SETCLR>
  (2)  012446                           TIMER:
10705  012446                                                      LET     REGSAV := REG(R5) ; GET POINTER TO REGIST
  (4)  012446  016537  000004  012606         MOV     REG(R5),REGSAV
10706  012454                                                      LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
  (4)  012454  016537  000000  012610         MOV     HOWLONG(R5),TIMSAV
10707  012462                                                      LET     FLAG :B= #0 ; INITIALIZE THE EXIT FLAG
  (4)  012462  105037  012612                 CLRB    FLAG
10708
10709
10710                                                       ;       START OF AN INFINITE LOOP
10711
10712  012466                                             LOOP
  (3)  012466                           $237:
10713                                                             ; TEST TO SEE IF WHICHBIT IS SET
10714  012466                                                   IF      WHICHBIT(R5) NOTSETIN @REGSAV THEN
  (6)  012466  036577  000002  000112         BIT     WHICHBIT(R5),@REGSAV
  (9)  012474  001003                         BNE     $241
10715  012476                                                      LET     HOLDSC :B= #0
  (4)  012476  105037  012613                 CLRB    HOLDSC
10716  012502                                                   ELSE
  (4)  012502  000403                         BR      $242
  (3)  012504                           $241:
10717  012504                                                      LET     HOLDSC :B= #SET ; REMEMBER THIS
  (4)  012504  112737  177777  012613         MOVB    #SET,HOLDSC
10718  012512                                                   ENDIF
  (4)  012512                           $242:
10719
10720                                                       ; NOW SEE IF THAT WAS WHAT WE WANTED
10721  012512                                               IFB     HOLDSC EQ SETCLR(R5) THEN
  (6)  012512  123765  012613  000006         CMPB    HOLDSC,SETCLR(R5)
  (9)  012520  001003                         BNE     $243
```

```
10722                                                          ; JUST THE THING WE NEEDED
10723  012522                                                        LET     FLAG :B= #TRUE
  (4)  012522  112737  000001  012612          MOVB    #TRUE,FLAG
10724  012530                                                     ENDIF
  (4)  012530                          $243:
10725
10726  012530                                                   EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0
  (4)  012530  123727  012612  000001          CMPB    FLAG,#TRUE
  (6)  012536  001414                          BEQ     $240
  (4)  012540  005737  012610                  TST     TIMSAV
  (6)  012544  003411                          BLE     $240
10727                                                          ; ONE WAY OR THE OTHER, WE ARE DONE
10728                                                        ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
10729
10730  012546                                                     WAITMS  1       ;WAIT FOR 1 MILLI-SECONDS
  (4)  012546  010546                          MOV     R5,-(SP)
  (5)  012550  012745  000001                  MOV     #1,-(R5)
  (4)  012554  004737  012620                  JSR     PC,WAIT
  (4)  012560  012605                          MOV     (SP)+,R5
10731  012562                                                        LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN
  (7)  012562  005337  012610                  DEC     TIMSAV
10732  012566                                                   ENDLOOP                 ; CONTINUED AT THE TOP
  (4)  012566  000737                          BR      $237
  (3)  012570                          $240:
10733
10734                                        ; ONLY 2 WAYS TO GET HERE
10735                                        ;       1). WE RAN OUT OF TIME---ERROR ! !
10736                                        ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
10737
10738  012570                                                        IFB     FLAG EQ #TRUE THEN
  (6)  012570  123727  012612  000001          CMPB    FLAG,#TRUE
  (9)  012576  001001                          BNE     $244
10739  012600                                                        RETURN  NO.ERROR    ; GOOD
  (4)  012600  000405                          BR      $235
10740  012602                                                     ENDIF
  (4)  012602                          $244:
10741  012602                                                     RETURN  ERROR           ; BAD
  (2)  012602  000261                          SEC
  (4)  012604  000404                          BR      $236
10742
10743  012606  000000                REGSAV: .WORD 0
10744  012610  000000                TIMSAV: .WORD 0
10745  012612     000                 FLAG:   .BYTE 0
10746  012613     000                 HOLDSC: .BYTE 0
10747                                                          ; WE ARE DONE  GO BACK HOME
10748  012614                                                        ENDRTN
  (3)  012614                          $235:
  (2)  012614  000241                          CLC
  (3)  012616                          $236:
  (2)  012616  000207                          RTS     PC
```

```
 10750
 10751
 10756                          ;;*****************************************************************
 10757                          ;* ROUTINE:WAIT
 10758                          ;*       THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
 10759                          ;*       MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME.
 10760                          ;*       THIS   IS   ACCOMPLISHED   BY  INCREMENTING  A
 10761                          ;*       REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
 10762                          ;*       TO APPROXIMATE 1 MILLI SEC.
 10763                          ;*****************************************************************
 10768  012620                                  ROUTINE WAIT    <TIME>
   (2)  012620                  WAIT:
 10769  012620                                          PUSH    <R1,R2,R3>
 10770  012626                                          LET R1 := TIME(R5)
   (4)  012626  016501  000000                  MOV     TIME(R5),R1
 10771  012632                                          INCRU R2 FROM #1 TO R1 BY #1
   (4)  012632  012702  000001                  MOV     #1,R2
   (5)  012636  000402                          BR      $247
   (4)  012640                  $250:
   (8)  012640  062702  000001                  ADD     #01,R2
   (5)  012644                  $247:
   (5)  012644  020201                          CMP     R2,R1
   (7)  012646  101010                          BHI     $251
 10772  012650                                                  INCR R3 FROM #0 TO #100 BY #1
   (4)  012650  005003                          CLR     R3
   (5)  012652  000401                          BR      $252
   (4)  012654                  $253:
   (8)  012654  005203                          INC     R3
   (5)  012656                  $252:
   (5)  012656  020327  000100                  CMP     R3,#100
   (7)  012662  003001                          BGT     $254
 10773  012664                                                  ENDINC
   (4)  012664  000773                          BR      $253
   (3)  012666                  $254:
 10774  012666                                          ENDINC
   (4)  012666  000764                          BR      $250
   (3)  012670                  $251:
 10775  012670                                          POP     <R3,R2,R1>
 10776  012676                                  ENDRTN
   (3)  012676                  $245:
   (3)  012676                  $246:
   (2)  012676  000207                          RTS     PC
```

```
10778
10779
10780
10781                                   .SBTTL  ROUTINE TO SIZE THE BUS
10782
10783  012700  005737  001252          SIZE:   TST     $DEVM           ;BR IF INITIALLY SET
10784  012704  001060                          BNE     4$              ;BYPASS AUTO-SIZING
10785
10786  012706  012737  013610  000004  1$:     MOV     #INTSRV,ERRVEC
10787  012714  012737  000300  000006          MOV     #PR6,ERRVEC+2
10788
10789  012722  013737  001250  013514          MOV     $BASE,TMP6
10790  012730  012737  000001  013512          MOV     #1,SFTREG
10791
10792  012736  005037  013616          2$:     CLR     INTFLAG
10793  012742  005777  000546                  TST     @TMP6           ;SEE IF THERE
10794  012746  005737  013616                  TST     INTFLAG         ;GOT INTERRUPT?
10795  012752  001003                          BNE     3$              ;BR IF YES  (NO DEVICE)
10796  012754  053737  013512  001252          BIS     SFTREG,$DEVM    ;ELSE THERE, SET MAP
10797
10798  012762  062737  000010  013514  3$:     ADD     #10,TMP6        ;GO TO NEXT CHANNEL
10799  012770  006337  013512                  ASL     SFTREG
10800  012774  001360                          BNE     2$              ;BR IF MORE TO GO
10801
10802  012776  012737  000006  000004          MOV     #6,ERRVEC       ;RESTORE
10803  013004  005037  000006                  CLR     ERRVEC+2
10804
10805  013010  005737  001252                  TST     $DEVM           ;ANYONE THERE?
10806  013014  001014                          BNE     4$              ;BR IF YES
10807  013016                                  TYPTXT  <<CRLF>!NO DEV TO TEST!>
10808  013044  000777                          BR      .               ;LOOP HERE TILL PROBLEM FIXED
10809                                                                  ;& RESTART AT 200
10810  013046  105737  001221          4$:     TSTB    $USWR+1         ;SEE IF ANY CONSOLE
10811  013052  001434                          BEQ     6$              ;BR IF NONE
10812
10813  013054  010046                          MOV     R0,-(SP)        ;SAVE R0
10814  013056  113700  001221                  MOVB    $USWR+1,R0      ;GET CONSOLE DEVICE X2
10815  013062  020027  000010                  CMP     R0,#10
10816  013066  003005                          BGT     5$              ;BR IF TOO HI
10817  013070  056037  013516  001252          BIS     TABL1-2(R0),$DEVM       ;SET CONSOLE BIT ON MAP
10818  013076  012600                          MOV     (SP)+,R0        ;RESTORE R0
10819  013100  000421                          BR      6$
10820
10821  013102                          5$:     TYPTXT  <<CRLF>!CONSOLE IN $USWR WRONG!>
10822  013140  000137  014362                  JMP     $EOP                    ;START ALL OVER
10823                                                                  ;STACK WILL RESET IN 'LOOP'
10824  013144  013737  001252  013502  6$:     MOV     $DEVM,MASK1
10825  013152  013737  001252  013504          MOV     $DEVM,MASK2
10826  013160  013737  001252  013506          MOV     $DEVM,MASK3
10827  013166  013737  001252  013510          MOV     $DEVM,MASK4
10828
10829  013174  042737  177760  013502          BIC     #^C<17>,MASK1           ;SAVE ONLY MODULE 1 CHANNELS
10830  013202  042737  177417  013504          BIC     #^C<360>,MASK2          ;                 2
10831  013210  042737  170377  013506          BIC     #^C<7400>,MASK3         ;                 3
10832  013216  042737  007777  013510          BIC     #^C<170000>,MASK4       ;                 4
10833
```

```
10834  013224  006237  013504          ASR     MASK2
10835  013230  006237  013504          ASR     MASK2
10836  013234  006237  013504          ASR     MASK2
10837  013240  006237  013504          ASR     MASK2                   ;RIGHT JUSTIFY MODULE 2 CHANNELS
10838
10839  013244  000337  013506          SWAB    MASK3                   ;                       3
10840
10841  013250  000337  013510          SWAB    MASK4
10842  013254  006237  013510          ASR     MASK4
10843  013260  006237  013510          ASR     MASK4
10844  013264  006237  013510          ASR     MASK4
10845  013270  006237  013510          ASR     MASK4                   ;                       4
10846
10847  013274                          TYPTXT  <<CRLF>!WILL TEST:!<CRLF>>
10848
10849  013320  012700  013502          MOV     #MASK1,R0
10850  013324  012737  000061  013474  MOV     #61,MODNUM              ;INIT MOD #
10851  013332  012737  000060  013500  MOV     #60,CHNUM+2             ;INIT CHANNEL #
10852
10853  013340  012037  013516  7$:     MOV     (R0)+,MSK
10854  013344  001001                  BNE     8$                      ;BR IF ANY THERE
10855  013346  000441                  BR      11$                     ;ELSE TRY NEXT MODULE
10856
10857  013350                  8$:     TYPTXT  <<CRLF>!MODULE !>
10858  013370  104401  013474          TYPE    ,MODNUM
10859  013374                          TYPTXT  <!  CHANNEL !>
10860
10861  013416  032737  000001  013516  9$:     BIT     #BIT0,MSK
10862  013424  001402                  BEQ     10$                     ;BR IF NOT TESTING THAT CHANNEL
10863  013426  104401  013476          TYPE    ,CHNUM                  ;ELSE TYPE IT
10864
10865  013432  005237  013500  10$:     INC     CHNUM+2                 ;BUMP CHANNEL CTR
10866  013436  006237  013516          ASR     MSK
10867  013442  001365                  BNE     9$                      ;BR IF MORE TO CHECK
10868  013444  012737  000060  013500  MOV     #60,CHNUM+2             ;ELSE RESET CH #
10869
10870  013452  005237  013474  11$:     INC     MODNUM                  ;BUMP MOD #
10871  013456  023727  013474  000065  CMP     MODNUM,#65              ;ALL DONE?
10872  013464  001325                  BNE     7$                      ;BR IF NO
10873  013466  104401  001171          TYPE    ,$CRLF
10874  013472  000207                  RTS     PC                      ;ELSE EXIT
10875
10876  013474  000061          MODNUM: .ASCIZ  /1/
10877  013476  020040  000060  CHNUM:  .ASCIZ  /  0/
10878
10879  013502  000000          MASK1:  0                       ;CONTAINS MODULE 1 CHANNELS
10880  013504  000000          MASK2:  0                       ;                 2
10881  013506  000000          MASK3:  0                       ;                 3
10882  013510  000000          MASK4:  0                       ;                 4
10883  013512  000000          SFTREG: 0
10884  013514  000000          TMP6:   0
10885  013516  000000          MSK:    0
10886  013520  000010          TABL1:  10                      ;SET $DEVM IF CONSOLE ON MODULE 1
10887  013522  000200                  200                     ;                             2
10888  013524  004000                  4000                    ;                             3
10889  013526  100000                  100000                  ;                             4
```

```
10890
10891
10892
10893
10894                                   .SBTTL  ROUTINE TO TEST FOR CONSOLE DEVICE
10895
10896  013530  113737  001221  013606  TSTCON: MOVB    $USWR+1,TMP7            ;GET CONSOLE INFO ONLY
10897  013536  006237  013606                  ASR     TMP7                    ;RIGHT JUSTIFY
10898  013542  023737  013606  001206          CMP     TMP7,$UNIT              ;SEE IF UNIT UNDER TEST HAS CONSOLE
10899  013550  001403                          BEQ     1$                      ;BR IF YES
10900  013552  005037  013604                  CLR     CONSOLE                 ;ELSE CLEAR STALE INFO
10901  013556  000207                          RTS     PC
10902
10903  013560  013737  001144  014156  1$:     MOV     $TKS,ADDRESS            ;FORCE TTY ADDRESS
10904  013566  012737  000060  014160          MOV     #TKVEC,VECTOR           ;       VECTOR
10905  013574  012737  000001  013604          MOV     #TRUE,CONSOLE           ;SET FLAG
10906  013602  000207                          RTS     PC
10907
10908  013604  000000                  CONSOLE:        0                       ;FLAG, 1= TESTING CONSOLE DEVICE
10909  013606  000000                  TMP7:           0                       ;HOLD CONSOLE #
10910
10911
10912
10913                                   .SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE
10914
10919                                   ;;*****************************************************************
10920                                   ;*      THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
10921                                   ;*      'INTFLAG'  EACH TIME IT IS CALLED.  IT ASSUMES
10922                                   ;*      THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
10923                                   ;*      TO LOOK FOR.
10924                                   ;;*****************************************************************
10929  013610                           BGNSRV INTSRV                   ;GLOBAL INTERRUPT SERVICE ROUTINE
  (1)  013610                           INTSRV:
10930                                                           ;ADD 1 TO 'INTERRUPT OCCURED' FLAG
10931  013610                                                           LET INTFLAG := INTFLAG + #1
  (7)  013610  005237  013616                   INC     INTFLAG
10932  013614                                   ENDSRV                  ;THAT'S ALL
  (1)  013614  000002                           RTI
10933  013616  000000                  INTFLAG:  0
10934
```

```
10936
10937
10938                                   .SBTTL  ROUTINE TO SET UP FOR NEXT MODULE/CHANNEL ADDRESS
10939
10940
10945                                   ;;*****************************************************************
10946                                   ;*      THIS  ROUTINE  CAUSES  ADRS   TO POINT TO  THE
10947                                   ;*      ADDRESS  OF  DLV11-J  UNDER  TEST,  ADRS +2 TO
10948                                   ;*      POINT TO THE VECTOR OF THE DLV11-J UNDER TEST.
10949                                   ;;*****************************************************************
10954
10955  013620  013737  001250  014156  CYCLE:  MOV     $BASE,ADDRESS
10956  013626  013737  001244  014160          MOV     $VECT1,VECTOR
10957
10958  013634  023727  001206  000005          CMP     $UNIT,#5                ;DONE ALL MODULES?
10959  013642  001013                          BNE     1$                      ;BR IF NO
10960  013644  005037  014166                  CLR     PHASE2                  ;RE-INIT
10961  013650  005037  014164                  CLR     CHAN
10962  013654  012737  000001  001206          MOV     #1,$UNIT
10963  013662  005037  014170                  CLR     P1CNT
10964  013666  000137  014362                  JMP     $EOP                    ;& EXIT BACK TO 'LOOP'
10965                                                                          ;STACK WILL GET RESET THERE.
10966  013672  013700  001206          1$:     MOV     $UNIT,R0
10967  013676  006300                          ASL     R0                      ;MULT BY 2 TO INDEX INTO TABLES
10968  013700  066037  014114  014156          ADD     TABL2-2(R0),ADDRESS     ;ADD MODULE OFFSET
10969  013706  066037  014114  014160          ADD     TABL2-2(R0),VECTOR      ;ADD VECTOR OFFSET
10970  013714  017037  014124  014162          MOV     @TABL3-2(R0),MASK       ;STORE CHANNELS IN 'MASK'
10971
10972  013722  005037  013604                  CLR     CONSOLE
10973  013726  005737  014162                  TST     MASK                    ;ANY CHANNELS TO BE TESTED?
10974  013732  001004                          BNE     2$                      ;BR IF YES
10975  013734  005237  001206                  INC     $UNIT                   ;ELSE BUMP UNIT #
10976  013740  000137  013620                  JMP     CYCLE                   ;TRY AGAIN
10977
10978  013744  023727  014166  000001  2$:     CMP     PHASE2,#TRUE            ;DOING PHASE 2?
10979  013752  001452                          BEQ     6$                      ;BR IF YES
10980  013754  023727  014164  000004          CMP     CHAN,#4                 ;ELSE, DONE ALL CHANNELS?
10981  013762  001021                          BNE     4$                      ;BR IF NO
10982
10983  013764  005037  014164                  CLR     CHAN                    ;ELSE BACK TO 0
10984  013770  032737  000020  001220          BIT     #WRAP,$USWR             ;DOING DATA WRAP TESTS?
10985  013776  001004                          BNE     3$                      ;BR IF YES
10986  014000  005237  001206                  INC     $UNIT                   ;ELSE GO TO NEXT UNIT, NO PHASE 2
10987  014004  000137  013620                  JMP     CYCLE                   ;& DO IT
10988
10989  014010  012737  000001  014166  3$:     MOV     #TRUE,PHASE2
10990  014016  005037  014170                  CLR     P1CNT
10991  014022  000137  013620                  JMP     CYCLE                   ;GO TO PHASE 2 TESTING
10992
10993  014026  013700  014164          4$:     MOV     CHAN,R0
10994  014032  006300                          ASL     R0                      ;MULT BY 2 TO INDEX TABLE
10995  014034  066037  014136  014156          ADD     TABL4(R0),ADDRESS       ;ADD CHANNEL OFFSET
10996  014042  066037  014136  014160          ADD     TABL4(R0),VECTOR        ;ADD VECTOR OFFSET
10997  014050  036037  014146  014162          BIT     TABL5(R0),MASK          ;IS THAT CHANNEL TO BE TESTED?
10998  014056  001413                          BEQ     7$                      ;BR IF NO
10999
```

```
11000  014060  023727  014164  000003          CMP     CHAN,#3                 ;IS IT CHAN 3?
11001  014066  001002                          BNE     5$                      ;BR IF NO
11002  014070  004737  013530                  JSR     PC,TSTCON               ;ELSE SEE IF ITS THE CONSOLE
11003
11004  014074  005237  014164          5$:     INC     CHAN                    ;BUMP CHANNEL CTR FOR NEXT TIME
11005  014100  012701  014156          6$:     MOV     #ADDRESS,ADRS
11006  014104  000207                          RTS     PC                      ;GO TO PHASE 1 TESTS
11007
11008  014106  005237  014164          7$:     INC     CHAN
11009  014112  000137  013620                  JMP     CYCLE                   ;TRY AGAIN
11010
11011  014116  000000                  TABL2:  0               ;OFFSET FOR MODULE 1
11012  014120  000040                          40              ;               2
11013  014122  000100                          100             ;               3
11014  014124  000140                          140             ;               4
11015
11016  014126  013502                  TABL3:  MASK1           ;MODULE ? CHANNEL DATA
11017  014130  013504                          MASK2           ;       2
11018  014132  013506                          MASK3           ;       3
11019  014134  013510                          MASK4           ;       4
11020
11021  014136  000000                  TABL4:  0               ;CHANNEL 0 OFFSET
11022  014140  000010                          10              ;         1
11023  014142  000020                          20              ;         2
11024  014144  000030                          30              ;         3
11025
11026  014146  000001                  TABL5:  BIT0            ;BIT TO TEST CHANNEL 0 PRESENT
11027  014150  000002                          BIT1            ;                     1
11028  014152  000004                          BIT2            ;                     2
11029  014154  000010                          BIT3            ;                     3
11030
11031  014156  000000                  ADDRESS: 0
11032  014160  000000                  VECTOR: 0
11033  014162  000000                  MASK:   0
11034  014164  000000                  CHAN:   0
11035  014166  000000                  PHASE2: 0
11036  014170  000000                  P1CNT:  0               ;CTR TO PRINT 'PHASE 1' ONLY ONCE
11037
11038
```

```
11040
11041
11042  014172                                                    ROUTINE MYTYPE
  (2)  014172                          MYTYPE:
11043  014172                                  TYPTXT  <<CRLF>*TEST # *>
11044  014212                                  TYPOCT  $TESTN
11045  014220                                  TYPTXT  <*,ERROR # *>
11046  014240  113737  001114  001176          MOVB    $ITEMB,$FATAL   ; APT FATAL ERROR NUMBER
11047  014246                                  TYPOCS  $FATAL
11048  014256                                  TYPTXT  <*,PC = *>
11049  014274                                  TYPOCT  $ERRPC
11050  014302                                  TYPTXT  <*,CSR: *>
11051  014320                                  TYPOCT  DLADD
11052  014326                                  TYPTXT  <*,VECTOR: *>
11053  014346                                  TYPOCT  DLVEC
11054  014354  104401  001171                  TYPE    ,$CRLF
11055  014360                                                    ENDRTN
  (3)  014360                          $255:
  (3)  014360                          $256:
  (2)  014360  000207                     RTS     PC
11060
```

```
11062
11063          -                        .SBTTL  END OF PASS ROUTINE
  (1)
  (2)                                   ;;*********************************************************************
  (1)                                   ;*INCREMENT THE PASS NUMBER ($PASS)
  (1)                                   ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
  (1)                                   ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
  (1)                                   ;*IF THERES A MONITOR GO TO IT
  (1)                                   ;*IF THERE ISN'T JUMP TO LOOP
  (1)
  (1)  014362                           $EOP:
  (1)  014362  000004                           SCOPE
  (1)  014364  005037  001102                   CLR     $TSTNM          ;;ZERO THE TEST NUMBER
  (1)  014370  005037  001160                   CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
  (1)  014374  005237  001202                   INC     $PASS           ;;INCREMENT THE PASS NUMBER
  (1)  014400  042737  100000  001202          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
  (1)  014406  005327                           DEC     (PC)+           ;;LOOP?
  (1)  014410  000001                   $EOPCT: .WORD   1
  (1)  014412  003022                           BGT     $DOAGN          ;;YES
  (1)  014414  012737                           MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
  (1)  014416  000001                   $ENDCT: .WORD   1
  (1)  014420  014410                           $EOPCT
  (1)  014422  104401  014467                   TYPE    ,$ENDMG         ;;TYPE "END PASS #"
  (2)  014426  013746  001202                   MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
  (2)  014432  104405                           TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
  (1)  014434  104401  014464                   TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
  (1)  014440  013700  000042           $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
  (1)  014444  001405                           BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
  (1)  014446  000005                           RESET                   ;;CLEAR THE WORLD
  (1)  014450  004710                   $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
  (1)  014452  000240                           NOP                     ;;SAVE ROOM
  (1)  014454  000240                           NOP                     ;;FOR
  (1)  014456  000240                           NOP                     ;;ACT11
  (1)  014460                           $DOAGN:
  (1)  014460  000137                           JMP     @(PC)+          ;;RETURN
  (1)  014462  001732                   $RTNAD: .WORD   LOOP
  (1)
  (1)
  (1)  014464     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
  (1)  014467     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
  (1)  014474  050040  051501  020123
  (1)  014502  000043
```

```
 11065
 11066                                  .SBTTL  TYPE ROUTINE
   (1)
   (2)                                  ;;*****************************************************************
   (1)                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
   (1)                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
   (1)                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
   (1)                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
   (1)                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
   (1)                                  ;*
   (1)                                  ;*CALL:
   (1)                                  ;*1) USING A TRAP INSTRUCTION
   (1)                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
   (1)                                  ;*OR
   (1)                                  ;*      TYPE
   (1)                                  ;*      MESADR
   (1)                                  ;*
   (1)
   (1)  014504  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
   (1)  014510  100002                          BPL     1$              ;;BR IF YES
   (1)  014512  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
   (1)  014514  000430                          BR      3$              ;;LEAVE
   (1)  014516  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
   (1)  014520  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
   (1)  014524  122737  000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
   (1)  014532  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
   (1)  014534  132737  000100  001215          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
   (1)  014542  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
   (1)  014544  010037  014554                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
   (1)  014550  004737  015550                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
   (1)  014554  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
   (1)  014556  132737  000040  001215  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
   (1)  014564  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
   (1)  014566  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
   (1)  014570  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
   (1)  014572  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
   (1)  014574  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
   (1)  014576  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
   (1)  014602  000002                          RTI                     ;;RETURN
   (1)  014604  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
   (1)  014610  001430                          BEQ     8$
   (1)  014612  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
   (1)  014616  001006                          BNE     5$
   (1)  014620  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
   (1)  014622  104401                          TYPE                    ;;TYPE A CR AND LF
   (1)  014624  001171                          $CRLF
   (1)  014626  105037  014762                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
   (1)  014632  000755                          BR      2$              ;;GET NEXT CHARACTER
   (1)  014634  004737  014716          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
   (1)  014640  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
   (1)  014644  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
   (1)  014646  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
   (1)                                                                  ;;AND THE NULL CHAR.
   (1)  014652  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
   (1)  014656  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
   (1)  014660  004737  014716                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
```

```
(1)  014664  105337  014762              DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
(1)  014670  000770                      BR      7$              ;;LOOP
(1)
(1)                                    ;HORIZONTAL TAB PROCESSOR
(1)
(1)  014672  112716  000040            8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
(1)  014676  004737  014716            9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
(1)  014702  132737  000007  014762            BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
(1)  014710  001372                            BNE     9$              ;;TAB STOP
(1)  014712  005726                            TST     (SP)+           ;;POP SPACE OFF STACK
(1)  014714  000724                            BR      2$              ;;GET NEXT CHARACTER
(1)  014716  105777  164226            $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
(1)  014722  100375                            BPL     $TYPEC
(1)  014724  116677  000002  164220            MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)
(1)  014732  122766  000015  000002  CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
(1)  014740  001003                            BNE     1$              ;;BRANCH IF NO
(1)  014742  105037  014762                    CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
(1)  014746  000406                            BR      $TYPEX          ;;EXIT
(1)  014750  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
(1)  014756  001402                            BEQ     $TYPEX          ;;BRANCH IF YES
(1)  014760  105227                            INCB    (PC)+           ;;COUNT THE CHARACTER
(1)  014762  000000                    $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
(1)  014764  000207                    $TYPEX: RTS     PC
(1)
(1)
(1)
```

```
11068
11069                                   .SBTTL  TTY INPUT ROUTINE
  (1)
  (2)                                   ;;*****************************************************************
  (1)
  (1)
  (1)                                   .ENABL  LSB
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
  (1)                                   ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
  (1)                                   ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
  (1)                                   ;*WHEN OPERATING IN TTY FLAG MODE.
  (1)  014766  022737  000176  001140   $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
  (1)  014774  001074                          BNE     15$             ;;BRANCH IF NO
  (1)  014776  105777  164142                  TSTB    @$TKS           ;;CHAR THERE?
  (1)  015002  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
  (1)  015004  117746  164136                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
  (1)  015010  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
  (1)  015014  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
  (1)  015020  001062                          BNE     15$             ;;NO, RETURN TO USER
  (1)  015022  123727  001134  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
  (1)  015030  001456                          BEQ     15$             ;;BRANCH IF YES
  (1)
  (1)  015032  104401  015513                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
  (1)  015036  104401  015520          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
  (2)  015042  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
  (2)  015046  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  (1)  015050  104401  015531                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
  (1)  015054  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
  (1)  015056  005046                          CLR     -(SP)
  (1)                                          ;;THE NEW SWR
  (1)  015060  105777  164060          7$:     TSTB    @$TKS           ;;CHAR THERE?
  (1)  015064  100375                          BPL     7$              ;;IF NOT TRY AGAIN
  (1)
  (1)  015066  117746  164054                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
  (1)  015072  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
  (1)
  (1)
  (1)
  (1)  015076  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
  (1)  015102  001005                          BNE     10$             ;;BRANCH IF NOT
  (1)  015104  104401  015506                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
  (1)  015110  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
  (1)  015114  000757                          BR      19$             ;;LET'S TRY IT AGAIN
  (1)
  (1)
  (1)  015116  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
  (1)  015122  001022                          BNE     16$             ;;BRANCH IF NO
  (1)  015124  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
  (1)  015130  001403                          BEQ     11$             ;;BRANCH IF YES
  (1)  015132  016677  000002  164000          MOV     2(SP),@SWR      ;;SAVE NEW SWR
  (1)  015140  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
  (1)  015144  104401  001171          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
  (1)  015150  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
  (1)  015156  001003                          BNE     15$             ;;BRANCH IF NOT
```

```
(1)  015160  012777  000100  163756          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
(1)  015166  000002                  15$:     RTI                     ;;RETURN
(1)  015170  004737  014716          16$:     JSR     PC,$TYPEC       ;;ECHO CHAR
(1)  015174  021627  000060                   CMP     (SP),#60        ;;CHAR < 0?
(1)  015200  002420                           BLT     18$             ;;BRANCH IF YES
(1)  015202  021627  000067                   CMP     (SP),#67        ;;CHAR > 7?
(1)  015206  003015                           BGT     18$             ;;BRANCH IF YES
(1)  015210  042726  000060                   BIC     #60,(SP)+       ;;STRIP-OFF ASCII
(1)  015214  005766  000002                   TST     2(SP)           ;;IS THIS THE FIRST CHAR
(1)  015220  001403                           BEQ     17$             ;;BRANCH IF YES
(1)  015222  006316                           ASL     (SP)            ;;NO, SHIFT PRESENT
(1)  015224  006316                           ASL     (SP)            ;;   CHAR OVER TO MAKE
(1)  015226  006316                           ASL     (SP)            ;;   ROOM FOR NEW ONE.
(1)  015230  005266  000002          17$:     INC     2(SP)           ;;KEEP COUNT OF CHAR
(1)  015234  056616  177776                   BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
(1)  015240  000707                           BR      7$              ;;GET THE NEXT ONE
(1)  015242  104401  001170          18$:     TYPE    ,$QUES          ;;TYPE ?<CR><LF>
(1)  015246  000720                           BR      20$             ;;SIMULATE CONTROL-U
(1)                                 .DSABL   LSB
(1)
(1)
(2)                                 ;;**********************************************************************
(1)                                 ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                 ;*CALL:
(1)                                 ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                 ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
(1)                                 ;*                              ;;WITH PARITY BIT STRIPPED OFF
(1)                                 ;
(1)
(1)  015250  011646                 $RDCHR:  MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
(1)  015252  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
(1)  015260  105777  163660          1$:      TSTB    @$TKS           ;;WAIT FOR
(1)  015264  100375                           BPL     1$              ;;A CHARACTER
(1)  015266  117766  163654  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
(1)  015274  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
(1)  015302  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
(1)  015310  001013                           BNE     3$              ;;BRANCH IF NO
(1)  015312  105777  163626          2$:      TSTB    @$TKS           ;;WAIT FOR A CHARACTER
(1)  015316  100375                           BPL     2$              ;;LOOP UNTIL ITS THERE
(1)  015320  117746  163622                   MOVB    @$TKB,-(SP)     ;;GET CHARACTER
(1)  015324  042716  177600                   BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
(1)  015330  022627  000021                   CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
(1)  015334  001366                           BNE     2$              ;;IF NOT DISCARD IT
(1)  015336  000750                           BR      1$              ;;YES, RESUME
(1)  015340  026627  000004  000140  3$:      CMP     4(SP),#140      ;;IS IT UPPER CASE?
(1)  015346  002407                           BLT     4$              ;;BRANCH IF YES
(1)  015350  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
(1)  015356  003003                           BGT     4$              ;;BRANCH IF YES
(1)  015360  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
(1)  015366  000002                  4$:      RTI                     ;;GO BACK TO USER
(2)                                 ;;**********************************************************************
(1)                                 ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                 ;*CALL:
(1)                                 ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
(1)                                 ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                 ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
```

```
(1)
(1)  015370  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
(1)  015372  012703  015476          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
(1)  015376  022703  015506          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
(1)  015402  101405                          BLOS    4$              ;;BR IF YES
(1)  015404  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
(1)  015406  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
(1)  015410  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
(1)  015414  001003                          BNE     3$              ;;SKIP IF NOT
(1)  015416  104401  001170          4$:     TYPE    ,$QUES          ;;TYPE A '?'
(1)  015422  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
(1)  015424  111337  015474          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
(1)  015430  104401  015474                  TYPE    ,9$
(1)  015434  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1)  015440  001356                          BNE     2$              ;;LOOP IF NOT RETURN
(1)  015442  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
(1)  015446  104401  001172                  TYPE    ,$LF            ;;TYPE A LINE FEED
(1)  015452  012603                          MOV     (SP)+,R3        ;;RESTORE R3
(1)  015454  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1)  015456  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
(1)  015464  012766  015476  000004          MOV     #$TTYIN,4(SP)
(1)  015472  000002                          RTI                     ;;RETURN
(1)  015474     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  015475     000                          .BYTE   0               ;;TERMINATOR
(1)  015476  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
(1)  015506  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
(1)  015513     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
(1)  015520  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
(1)  015526  020075     000
(1)  015531     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
(1)  015536  036440  000040
```

```
 11071
 11072                                  .SBTTL  APT COMMUNICATIONS ROUTINE
   (1)
   (2)                                  ;;*****************************************************************
   (1)  015542  112737  000001  016006  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
   (1)  015550  112737  000001  016004  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
   (1)  015556  000403                          BR      $ATYC
   (1)  015560  112737  000001  016006  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
   (1)  015566                          $ATYC:
   (3)  015566  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
   (3)  015570  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   (1)  015572  105737  016004                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
   (1)  015576  001450                          BEQ     5$              ;;IF NOT:  BR
   (1)  015600  122737  000001  001214          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
   (1)  015606  001031                          BNE     3$              ;;IF NOT:  BR
   (1)  015610  132737  000100  001215          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
   (1)  015616  001425                          BEQ     3$              ;;IF NOT:  BR
   (1)  015620  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
   (1)  015624  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
   (1)  015632  005737  001174          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
   (1)  015636  001375                          BNE     1$              ;;IF NOT:  WAIT
   (1)  015640  010037  001210                  MOV     R0,$MSGAD
   (1)                                  ;;PUT ADDR IN MAILBOX
   (1)  015644  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
   (1)  015646  001376                          BNE     2$
   (1)  015650  163700  001210                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
   (1)  015654  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
   (1)  015656  010037  001212                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
   (1)  015662  012737  000004  001174          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
   (1)  015670  000413                          BR      5$
   (1)  015672  017637  000004  015716  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
   (1)  015700  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
   (3)  015706  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
   (1)  015712  004737  014504                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
   (1)  015716  000000                  4$:     .WORD   0
   (1)  015720                          5$:
   (1)  015720  105737  016006          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
   (1)  015724  001416                          BEQ     12$             ;;IF NOT:  BR
   (1)  015726  005737  001214                  TST     $ENV            ;;RUNNING UNDER APT?
   (1)  015732  001413                          BEQ     12$             ;;IF NOT:  BR
   (1)  015734  005737  001174          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
   (1)  015740  001375                          BNE     11$             ;;IF NOT:  WAIT
   (1)  015742  017637  000004  001176          MOV     @4(SP),$FATAL   ;;GET ERROR #
   (1)  015750  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
   (1)  015756  005237  001174                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
   (1)  015762  105037  016006          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
   (1)  015766  105037  016005                  CLRB    $LFLG           ;;CLEAR LOG FLAG
   (1)  015772  105037  016004                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
   (3)  015776  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
   (3)  016000  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
   (1)  016002  000207                          RTS     PC              ;;RETURN
   (1)  016004     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
   (1)  016005     000          $LFLG:  .BYTE   0
   (1)                                  ;;LOG FLAG
   (1)  016006     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
   (1)          016010                  .EVEN
```

```
    (1)          000200                  APTSIZE=200
    (1)          000001                  APTENV=001
    (1)          000100                  APTSPOOL=100
    (1)          000040                  APTCSUP=040
```

```
 11074
 11075                                  .SBTTL  ERROR HANDLER ROUTINE
   (1)
   (2)                                  ;;**********************************************************************
   (1)                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
   (1)                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
   (1)                                  ;*AND GO TO MYTYPE ON ERROR
   (1)                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
   (1)                                  ;*SW15=1        HALT ON ERROR
   (1)                                  ;*SW13=1        INHIBIT ERROR TYPEOUTS
   (1)                                  ;*SW10=1        BELL ON ERROR
   (1)                                  ;*SW09=1        LOOP ON ERROR
   (1)                                  ;*CALL
   (1)                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
   (1)
   (1)  016010                          $ERROR:
   (1)  016010  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
   (1)  016012  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
   (1)  016016  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
   (1)  016020  013777  001102  163114          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
   (1)  016026  032777  002000  163104          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
   (1)  016034  001402                          BEQ     1$              ;;NO - SKIP
   (1)  016036  104401  001164                  TYPE    ,$BELL          ;;RING BELL
   (1)  016042  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
   (1)  016046  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
   (1)  016052  162737  000002  001116          SUB     #2,$ERRPC
   (1)  016060  117737  163032  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
   (1)  016066  032777  020000  163044          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
   (1)  016074  001004                          BNE     20$             ;;SKIP TYPEOUTS
   (1)  016076  004737  014172                  JSR     PC,MYTYPE       ;;GO TO USER ERROR ROUTINE
   (1)  016102  104401  001171                  TYPE    ,$CRLF
   (1)  016106                          20$:
   (1)  016106  122737  000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
   (1)  016114  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
   (1)  016116  113737  001114  016130          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
   (1)  016124  004737  015560                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
   (1)  016130     000                  21$:    .BYTE   0
   (1)  016131     000                          .BYTE   0
   (1)  016132  000777                  22$:    BR      22$             ;;APT ERROR LOOP
   (1)  016134  005777  163000          2$:     TST     @SWR            ;;HALT ON ERROR
   (1)  016140  100002                          BPL     3$              ;;SKIP IF CONTINUE
   (1)  016142  000000                          HALT                    ;;HALT ON ERROR!
   (1)  016144  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
   (1)  016146  032777  001000  162764  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
   (1)  016154  001402                          BEQ     4$              ;;BR IF NO
   (1)  016156  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
   (1)  016162  005737  001162          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
   (1)  016166  001402                          BEQ     5$              ;;BR IF NONE
   (1)  016170  013716  001162                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
   (1)  016174                          5$:
   (1)  016174  022737  014450  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
   (1)  016202  001001                          BNE     6$              ;;BRANCH IF NO
   (1)  016204  000000                          HALT                    ;;YES
   (1)  016206                          6$:
   (1)  016206  000002                          RTI                     ;;RETURN
```

```
11077
11078                                   .SBTTL  SCOPE HANDLER ROUTINESTARS
  (1)                                   ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
  (1)                                   ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
  (1)                                   ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
  (1)                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
  (1)                                   ;*SW14=1        LOOP ON TEST
  (1)                                   ;*SW11=1        INHIBIT ITERATIONS
  (1)                                   ;*SW09=1        LOOP ON ERROR
  (1)                                   ;*SW08=1        LOOP ON TEST IN SWR<7:0>
  (1)                                   ;*CALL
  (1)                                   ;*      SCOPE           ;;SCOPE=IOT
  (1)
  (1)  016210                           $SCOPE:
  (1)  016210  104407                           CKSWR                           ;;TEST  FOR CHANGE IN SOFT-SWR
  (1)  016212  032777  040000  162720   1$:     BIT     #BIT14,@SWR             ;;LOOP ON PRESENT TEST?
  (1)  016220  001114                           BNE     $OVER                   ;;YES IF SW14=1
  (1)                                   ;#####START OF CODE FOR THE XOR TESTER#####
  (1)  016222  000416                   $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
  (1)                                                                           ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
  (1)  016224  013746  000004                   MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
  (1)  016230  012737  016250  000004           MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
  (1)  016236  005737  177060                   TST     @#177060                ;;TIME OUT ON XOR?
  (1)  016242  012637  000004                   MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
  (1)  016246  000463                           BR      $SVLAD                  ;;GO TO THE NEXT TEST
  (1)  016250  022626                   5$:     CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
  (1)  016252  012637  000004                   MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
  (1)  016256  000423                           BR      7$                      ;;LOOP ON THE PRESENT TEST
  (1)  016260                           6$:;#####END OF CODE FOR THE XOR TESTER#####
  (1)  016260  032777  000400  162652           BIT     #BIT08,@SWR             ;;LOOP ON SPEC. TEST?
  (1)  016266  001404                           BEQ     2$                      ;;BR IF NO
  (1)  016270  127737  162644  001102           CMPB    @SWR,$TSTNM             ;;ON THE RIGHT TEST?    SWR<7:0>
  (1)  016276  001465                           BEQ     $OVER                   ;;BR IF YES
  (1)  016300  105737  001103           2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
  (1)  016304  001421                           BEQ     3$                      ;;BR IF NO
  (1)  016306  123737  001115  001103           CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
  (1)  016314  101015                           BHI     3$                      ;;BR IF NO
  (1)  016316  032777  001000  162614           BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
  (1)  016324  001404                           BEQ     4$                      ;;BR IF NO
  (1)  016326  013737  001110  001106   7$:     MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
  (1)  016334  000446                           BR      $OVER
  (1)  016336  105037  001103           4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
  (1)  016342  005037  001160                   CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
  (1)  016346  000415                           BR      1$                      ;;ESCAPE TO THE NEXT TEST
  (1)  016350  032777  004000  162562   3$:     BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
  (1)  016356  001011                           BNE     1$                      ;;BR IF YES
  (1)  016360  005737  001202                   TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
  (1)  016364  001406                           BEQ     1$                      ;;      INHIBIT ITERATIONS
  (1)  016366  005237  001104                   INC     $ICNT                   ;;INCREMENT ITERATION COUNT
  (1)  016372  023737  001160  001104           CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
  (1)  016400  002024                           BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
  (1)  016402  012737  000001  001104   1$:     MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
  (1)  016410  013737  016466  001160           MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
  (1)  016416  105237  001102           $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
  (1)  016422  113737  001102  001200           MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
  (1)  016430  011637  001106                   MOV     (SP),$LPADR             ;;SAVE SCOPE LOOP ADDRESS
```

```
(1)  016434  011637  001110                MOV     (SP),$LPERR      ;;SAVE ERROR LOOP ADDRESS
(1)  016440  005037  001162                CLR     $ESCAPE          ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1)  016444  112737  000001  001115        MOVB    #1,$ERMAX        ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1)  016452  013777  001102  162462  $OVER: MOV     $TSTNM,@DISPLAY  ;;DISPLAY TEST NUMBER
(1)  016460  013716  001106                MOV     $LPADR,(SP)      ;;FUDGE RETURN ADDRESS
(1)  016464  000002                        RTI                      ;;FIXES PS
(1)  016466  003720                $MXCNT: 2000.                    ;;MAX. NUMBER OF ITERATIONS
```

```
11080
11081                                   .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
  (1)                                   ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
  (1)                                   ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
  (1)                                   ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
  (1)                                   ;*REPLACED WITH SPACES.
  (1)                                   ;*CALL:
  (1)                                   ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
  (1)                                   ;*      TYPDS                   ;;GO TO THE ROUTINE
  (1)
  (1)  016470                           $TYPDS:
  (3)  016470  010046                           MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  016472  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (3)  016474  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
  (3)  016476  010346                           MOV     R3,-(SP)        ;;PUSH R3 ON STACK
  (3)  016500  010546                           MOV     R5,-(SP)        ;;PUSH R5 ON STACK
  (1)  016502  012746  020200                   MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
  (1)  016506  016605  000020                   MOV     20(SP),R5       ;;GET THE INPUT NUMBER
  (1)  016512  100004                           BPL     1$              ;;BR IF INPUT IS POS.
  (1)  016514  005405                           NEG     R5              ;;MAKE THE BINARY NUMBER POS.
  (1)  016516  112766  000055  000001           MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
  (1)  016524  005000                   1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
  (1)  016526  012703  016704                   MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
  (1)  016532  112723  000040                   MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
  (1)  016536  005002                   2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
  (1)  016540  016001  016674                   MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
  (1)  016544  160105                   3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
  (1)  016546  002402                           BLT     4$              ;;BR IF DONE
  (1)  016550  005202                           INC     R2              ;;INCREASE THE BCD DIGIT BY 1
  (1)  016552  000774                           BR      3$
  (1)  016554  060105                   4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
  (1)  016556  005702                           TST     R2              ;;CHECK IF BCD DIGIT=0
  (1)  016560  001002                           BNE     5$              ;;FALL THROUGH IF 0
  (1)  016562  105716                           TSTB    (SP)            ;;STILL DOING LEADING 0'S?
  (1)  016564  100407                           BMI     7$              ;;BR IF YES
  (1)  016566  106316                   5$:     ASLB    (SP)            ;;MSD?
  (1)  016570  103003                           BCC     6$              ;;BR IF NO
  (1)  016572  116663  000001  177777           MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
  (1)  016600  052702  000060           6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
  (1)  016604  052702  000040           7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
  (1)  016610  110223                           MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
  (1)  016612  005720                           TST     (R0)+           ;;JUST INCREMENTING
  (1)  016614  020027  000010                   CMP     R0,#10          ;;CHECK THE TABLE INDEX
  (1)  016620  002746                           BLT     2$              ;;GO DO THE NEXT DIGIT
  (1)  016622  003002                           BGT     8$              ;;GO TO EXIT
  (1)  016624  010502                           MOV     R5,R2           ;;GET THE LSD
  (1)  016626  000764                           BR      6$              ;;GO CHANGE TO ASCII
  (1)  016630  105726                   8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
  (1)  016632  100003                           BPL     9$              ;;BR IF NO
  (1)  016634  116663  177777  177776           MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
  (1)  016642  105013                   9$:     CLRB    (R3)            ;;SET THE TERMINATOR
  (3)  016644  012605                           MOV     (SP)+,R5        ;;POP STACK INTO R5
  (3)  016646  012603                           MOV     (SP)+,R3        ;;POP STACK INTO R3
```

```
(3)  016650  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  016652  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  016654  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  016656  104401  016704                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
(1)  016662  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
(1)  016670  012616                        MOV     (SP)+,(SP)
(1)  016672  000002                        RTI                     ;;RETURN TO USER
(1)  016674  023420                $DTBL:  10000.
(1)  016676  001750                        1000.
(1)  016700  000144                        100.
(1)  016702  000012                        10.
(1)  016704  000004                $DBLK:  .BLKW   4
```

```
11083
11084                                   .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
  (1)                                   ;*OCTAL (ASCII) NUMBER AND TYPE IT.
  (1)                                   ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
  (1)                                   ;*CALL:
  (1)                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
  (1)                                   ;*      TYPOS                   ;;CALL FOR TYPEOUT
  (1)                                   ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
  (1)                                   ;*      .BYTE   M               ;;M=1 OR 0
  (1)                                   ;*                                      ;;1=TYPE LEADING ZEROS
  (1)                                   ;*                                      ;;0=SUPPRESS LEADING ZEROS
  (1)                                   ;*
  (1)                                   ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
  (1)                                   ;*$TYPOS OR $TYPOC
  (1)                                   ;*CALL:
  (1)                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
  (1)                                   ;*      TYPON                   ;;CALL FOR TYPEOUT
  (1)                                   ;*
  (1)                                   ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
  (1)                                   ;*CALL:
  (1)                                   ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
  (1)                                   ;*      TYPOC                   ;;CALL FOR TYPEOUT
  (1)
  (1)  016714  017646  000000           $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
  (1)  016720  116637  000001  017137           MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
  (1)  016726  112637  017141                   MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
  (1)  016732  062716  000002                   ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
  (1)  016736  000406                           BR      $TYPON
  (1)  016740  112737  000001  017137   $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
  (1)  016746  112737  000006  017141           MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
  (1)  016754  112737  000005  017136   $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
  (1)  016762  010346                           MOV     R3,-(SP)        ;;SAVE R3
  (1)  016764  010446                           MOV     R4,-(SP)        ;;SAVE R4
  (1)  016766  010546                           MOV     R5,-(SP)        ;;SAVE R5
  (1)  016770  113704  017141                   MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
  (1)  016774  005404                           NEG     R4
  (1)  016776  062704  000006                   ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
  (1)  017002  110437  017140                   MOVB    R4,$OMODE       ;;SAVE IT FOR USE
  (1)  017006  113704  017137                   MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
  (1)  017012  016605  000012                   MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
  (1)  017016  005003                           CLR     R3              ;;CLEAR THE OUTPUT WORD
  (1)  017020  006105                   1$:     ROL     R5              ;;ROTATE MSB INTO "C"
  (1)  017022  000404                           BR      3$              ;;GO DO MSB
  (1)  017024  006105                   2$:     ROL     R5              ;;FORM THIS DIGIT
  (1)  017026  006105                           ROL     R5
  (1)  017030  006105                           ROL     R5
  (1)  017032  010503                           MOV     R5,R3
  (1)  017034  006103                   3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
  (1)  017036  105337  017140                   DECB    $OMODE          ;;TYPE THIS DIGIT?
  (1)  017042  100016                           BPL     7$              ;;BR IF NO
  (1)  017044  042703  177770                   BIC     #177770,R3      ;;GET RID OF JUNK
  (1)  017050  001002                           BNE     4$              ;;TEST FOR 0
  (1)  017052  005704                           TST     R4              ;;SUPPRESS THIS 0?
```

```
(1)  017054  001403                    BEQ     5$              ;;BR IF YES
(1)  017056  005204              4$:   INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)  017060  052703  000060            BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
(1)  017064  052703  000040      5$:   BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
(1)  017070  110337  017134            MOVB    R3,8$           ;;SAVE FOR TYPING
(1)  017074  104401  017134            1,PE    ,8$             ;;GO TYPE THIS DIGIT
(1)  017100  105337  017136      7$:   DECB    $OCNT           ;;COUNT BY 1
(1)  017104  003347                    BGT     2$              ;;BR IF MORE TO DO
(1)  017106  002402                    BLT     6$              ;;BR IF DONE
(1)  017110  005204                    INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)  017112  000744                    BR      2$              ;;GO DO THE LAST DIGIT
(1)  017114  012605              6$:   MOV     (SP)+,R5        ;;RESTORE R5
(1)  017116  012604                    MOV     (SP)+,R4        ;;RESTORE R4
(1)  017120  012603                    MOV     (SP)+,R3        ;;RESTORE R3
(1)  017122  016666  000002  000004    MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)  017130  012616                    MOV     (SP)+,(SP)
(1)  017132  000002                    RTI                     ;;RETURN
(1)  017134     000              8$:   .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)  017135     000                    .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)  017136     000              $OCNT: .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)  017137     000              $OFILL:.BYTE   0               ;;ZERO FILL SWITCH
(1)  017140  000000              $OMODE:.WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
11086
11087                                   .SBTTL  TRAP DECODER
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
  (1)                                   ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
  (1)                                   ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
  (1)                                   ;*GO TO THAT ROUTINE.
  (1)
  (1)  017142  010046                   $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
  (1)  017144  016600  000002                   MOV     2(SP),R0        ;;GET TRAP ADDRESS
  (1)  017150  005740                           TST     -(R0)           ;;BACKUP BY 2
  (1)  017152  111000                           MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
  (1)  017154  006300                           ASL     R0              ;;POSITION FOR INDEXING
  (1)  017156  016000  017176                   MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
  (1)  017162  000200                           RTS     R0              ;;GO TO ROUTINE
  (1)
  (1)
  (1)                                   ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
  (1)
  (1)  017164  011646                   $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
  (1)  017166  016666  000004  000002           MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
  (1)  017174  000002                           RTI                     ;;RESTORE THE PSW
  (1)
  (3)                                   .SBTTL  TRAP TABLE
  (3)
  (3)                                   ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
  (3)                                   ;*BY THE "TRAP" INSTRUCTION.
  (3)
  (3)                                   ;       ROUTINE
  (3)                                   ;       -------
  (3)  017176  017164                   $TRPAD: .WORD   $TRAP2
  (3)  017200  014504                           $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
  (3)  017202  016740                           $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
  (3)  017204  016714                           $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
  (3)  017206  016754                           $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
  (3)  017210  016470                           $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
  (1)
  (3)  017212  015036                           $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
  (1)
  (3)  017214  014766                           $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
  (3)  017216  015250                           $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
  (3)  017220  015370                           $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
11088
11089                                           ;MAXMEM FOR APT = 17400
11090                                           ;MAXMEM NON APT = 17500  (ABS LOADER)
11091                                   ;THE FOLLOWING CALL TO CNMAC2.SML WAS ADDED TO INIT SBC 11/21
11092                                   ;SPECIFICS
11093           017222                  POINT=.                         ;SAVE POINTER
  (1)       000100                    .=100
  (1)  000100  017222                   $CLKVEC                         ;LKVEC  HANDLER
  (1)  000102  000300                   300                             ;INTERRUPT HANDLER PRI
  (1)       000140                    .=140                           ;BRKVEC
  (1)  000140  170000                   170000                          ;ODT START ADDRESS
  (1)  000142  000300                   300                             ;PRIORITY
  (1)       017222                    .=POINT                         ;RESTORE POINTER
```

```
   (1)  017222  104401  017230          $CLKVEC:        TYPE,CLKMES
   (1)  017226  000000                                  HALT
   (1)  017230  005015  045514  042526  CLKMES:  .ASCIZ <15><12>/LKVEC INTERRUPT - DISCONNECT LTC /
   (1)  017236  020103  047111  042524
   (1)  017244  051122  050125  020124
   (1)  017252  020055  044504  041523
   (1)  017260  047117  042516  052103
   (1)  017266  046040  041524  000040
 11094          000001                  .END
```

```
ABASE = 176500          8619#   8857
ACDW1 = 000000          8857
ACDW2 = 000000          8857
ACPUOP= 000000          8857
ACTCH   012364         10374*  10378*  10426   10638#
ADDRES  014156         10903*  10955*  10968*  10995*  11005   11031#
ADDW0 = 000000          8857
ADDW1 = 000000          8857
ADDW10= 000000          8857
ADDW11= 000000          8857
ADDW12= 000000          8857
ADDW13= 000000          8857
ADDW14= 000000          8857
ADDW15= 000000          8857
ADDW2 = 000000          8857
ADDW3 = 000000          8857
ADDW4 = 000000          8857
ADDW5 = 000000          8857
ADDW6 = 000000          8857
ADDW7 = 000000          8857
ADDW8 = 000000          8857
ADDW9 = 000000          8857
ADEVCT= 000000          8857
ADEVM = 000017          8620#   8857
AENV  = 000000          8857
AENVM = 000000          8857
AFATAL= 000000          8857
AMADR1= 000000          8857
AMADR2= 000000          8857
AMADR3= 000000          8857
AMADR4= 000000          8857
AMAMS1= 000000          8857
AMAMS2= 000000          8857
AMAMS3= 000000          8857
AMAMS4= 000000          8857
AMSGAD= 000000          8857
AMSGLG= 000000          8857
AMSGTY= 000000          8857
AMTYP1= 000000          8857
AMTYP2= 000000          8857
AMTYP3= 000000          8857
AMTYP4= 000000          8857
APASS = 000000          8857
APRIOR= 000000          8857
APTCSU= 000040         11066   11072#
APTENV= 000001         11066   11072#  11075
APTSIZ= 000200          8871   11072#
APTSPO= 000100         11066   11072#
ASWREG= 000000          8857
ATESTN= 000000          8857
AUNIT = 000000          8857
AUSWR = 000031          8622#   8857
AVECT1= 000300          8621#   8857
AVECT2= 000000          8857
BIT0  = 000001          8724#   9728    9739    9792    9942    9968    10103   10376   10415   10452   10861   11026
BIT00 = 000001          8724#   8757    8777    8799    8820
```

```
BIT01 = 000002          8724#   8776    8819
BIT02 = 000004          8724#   8775    8818
BIT03 = 000010          8724#   8774    8817
BIT04 = 000020          8724#   8773    8816
BIT05 = 000040          8724#   8772    8815
BIT06 = 000100          8724#   8751    8771    8793    8814
BIT07 = 000200          8724#   8750    8770    8792    8813
BIT08 = 000400          8724#  11078
BIT09 = 001000          8724#  11075   11078
BIT1  = 000002          8724#   9945    9971   11027
BIT10 = 002000          8724#  11075
BIT11 = 004000          8724#  11078
BIT12 = 010000          8724#   8765   10034
BIT13 = 020000          8724#   8764   11075
BIT14 = 040000          8724#   8763   11078
BIT15 = 100000          8724#   8762   10467
BIT2  = 000004          8724#   9953    9974   11028
BIT3  = 000010          8724#   9957    9979   11029
BIT4  = 000020          8724#
BIT5  = 000040          8724#
BIT6  = 000100          8724#  10105   10108   10213   10224   10234   10244   10254   10264   10274   10284   10380   10383
                       10526   10542   10553   10569   10580   10596   10607   10623
BIT7  = 000200          8724#
BIT8  = 000400          8724#  10468   10492
BIT9  = 001000          8724#
BPTVEC= 000014          8724#
BREAK = 000001          8799#   8999    9007    9009    9018    9020    9029    9032    9930    9988    9990    9999
BRK   = 000010          8828#   8992    9918
BRKVEC= 000140          8724#
CHAN    014164          8881*  10961*  10980   10983*  10993   11000   11004*  11008*  11034#
CHCNT   007766         10100*  10125*  10126   10129*  10132   10315#
CHCTR   012362         10372*  10391*  10392   10395*  10398   10449*  10453   10481*  10483   10487*  10490   10637#
CHMASK  007764         10099*  10103   10124*  10158*  10160*  10163   10314#
CHMSK   012360         10371*  10376   10388*  10450*  10452   10482*  10636#
CHNUM   013476         10851*  10863   10865*  10868*  10877#
CHOTAB  011360         10348   10401   10631#  10655
CH1TAB  011560         10402   10632#  10656
CH2TAB  011760         10403   10633#  10657
CH3TAB  012160         10404   10634#  10658
CKSWR = 104407         11075   11078   11087#
CLKMES  017230         11093#
CLR   = 000000          8733#   9665
CONSOL  013604          9192    9281    9325    9368    9413    9606    9689    9758    9921   10128   10167   10394   10485
                       10900*  10905*  10908#  10972*
CR    = 000015          8724#  11066
CRLF  = 000200          8724#   8872   10036   10039   10807   10821   10847   10857   11043   1106o
CYCLE   013620          8885   10955#  10976   10987   10991   11009
DATA    005662          9791*   9808    9854    9856    9862#   9879    9881*   9885*   9886
DDISP = 177570          8724#   8857    8871
DISPLA  001142          8857#   8871*  11075*  11078*
DISPRE  000174          8834#   8871
DLADD   001254          8859#   8892*   8896    8897    8898    8899    8928    8947   10044   10101   10369   11051
DLADDR= 176500          8729#   8859    8861    8862    8863    8864
DLVEC   001256          8860#   8894*   9537    9612    9780   10046   10087   10355   11053
DSWR  = 177570          8724#   8857    8871
EMTVEC= 000030          8724#   8871*
```

```
ERRCHK  006500          9928*  9942*  9945*  9953*  9968   9971   9974   9979  10021#
ERRCNT  005622          9790*  9808   9814   9837#  9854   9883*  9957*
ERROR = 100000          8762#  9460   9507   9720   9815   9879
ERRORF  003304          9199*  9234*  9247   9262#
ERRVEC= 000004          8724#  8871*  8930*  8931*  8968*  8969*  10786* 10787* 10802* 10803* 11078*
ERWRD   012366         10344*  10468* 10472*  10479  10490  10492  10496  10639#
EVENOD= 000004          8827#  9949
EXITFL  003306          9200*  9253*  9256*  9258   9263#
FLAG    012612         10707*  10723* 10726   10738  10745#
FRERR = 020000          8764#  9819   9944
GNS   = ****** U        8834   8872   10036  10039  10043  10045  10047  10807  10821  10847  10857  10859  11043
                       11045  11048  11050  11052  11087
GTSWR = 104406          8872   11087#
HOLDSC  012613         10715*  10717* 10721  10746#
HOWLON= 000000         10704#  10706
HT    = 000011          8724#  11066
I       001270          8865#  8933*  8946*  8947   8948
INTFLA  013616          8937*  8941   9534*  9556   9568   9578*  9583   9615*  9632   9643   9660*  9665   10792*
                       10794  10931* 10933#
INTRTA  007744         10078   10144  10155  10157  10311#
INTSRV  013610          8844   8930   9541   9612   10786  10929#
IOTVEC= 000020          8724#  8871*
LF    = 000012          8724#  11066
LKVEC = 000100          8724#
LOOP    001732          8884#  10056  11063
MASK    014162         10099   10158  10371  10450  10970* 10973  10997  11033#
MASK1   013502         10824*  10829* 10849  10879#  11016
MASK2   013504         10825*  10830* 10834* 10835* 10836* 10837* 10880#  11017
MASK3   013506         10826*  10831* 10839* 10881#  11018
MASK4   013510         10827*  10832* 10841* 10842* 10843* 10844* 10845* 10882#  11019
MODNUM  013474         10850*  10858  10870* 10871  10876#
MODTST  007000          8907   10060#
MSK     013516         10853*  10861  10866* 10885#
MYTYPE  014172         11042#  11075
NUMBER  005624          9793*  9795*  9808   9838#  9854   9881   9886
ODTST = 170000          8724#
ORERR = 040000          8763#  9449   9474   9498   9816
PARITY= 000002          8826#  9947
PASS    003302          9196*  9248*  9249   9261#
PERR  = 010000          8765#  9822   9951   9956
PHASE2  014166          8879*  8906   10035  10052  10053* 10960* 10978  10989* 11035#
PIRQ  = 177772          8724#
PIRQVE= 000240          8724#
POINT = 017222         11093#
PRIERR  007770         10154*  10178* 10181  10316#
PR0   = 000000          8724#  9551   9620   9778   10147  10428
PR1   = 000040          8724#
PR2   = 000100          8724#
PR3   = 000140          8724#
PR4   = 000200          8724#
PR5   = 000240          8724#
PR6   = 000300          8724#  8845   8871   8931   9050   9542   9612   9672   9783   9787   10084  10091  10358
                       10361  10365  10787
PR7   = 000340          8724#
PS    = 177776          8724#
PSW   = 177776          8724#
```

```
PWRVEC= 000024           8724#
P1CNT   014170           8882*  10038  10040*  10963*  10990*  11036#
RBUF    001262           8862#   8897*   8756    9393    9446    9474    9498    9507    9718    9877    9939    9963   10004
                        10015
RCH0    007724          10208  10297#
RCH1    007726          10229  10298#
RCH2    007730          10249  10299#
RCH3    007732          10269  10300#
RCSR    001260           8861#  8896*   8952    9112    9120*   9122    9131*   9133    9142*   9145    9293    9307    9337
                         9349*  9350    9380    9395    9432    9490    9617*   9629*   9658*   9710    9803*   9813*   9882*
                         9887*  9934   10008   10210   10221   10231   10241   10251   10261   10271   10281   10519   10534
                        10546  10561   10573   10588   10600   10615
RCVRDO= 000200           8750#  9293    9307    9337    9350    9380    9395    9432    9490    9627    9710    9934   10008
                        10114
RCVRIE= 000100           8751#  9112    9120    9122    9131    9133    9142    9145    9617    9629    9658    9803    9813
                         9882   9887
RCVROS  007424          10088  10206#
RCVR1S  007504          10227#
RCVR2S  007564          10247#
RCVR3S  007644          10267#
RDATA0= 000001           8777#
RDATA1= 000002           8776#
RDATA2= 000004           8775#
RDATA3= 000010           8774#
RDATA4= 000020           8773#
RDATA5= 000040           8772#
RDATA6= 000100           8771#
RDATA7= 000200           8770#
RDCHR = 104410          11069  11087#
RDLIN = 104411          11087#
RDRRUN= 000001           8757#  9349
REC     005664           9782   9875#
REG   = 000004          10704#  10705
REGSAV  012606          10705*  10714  10743#
RESVEC= 000010           8724#
RHLD    005770           9815   9816   9819   9822   9877*  9879   9895#
ROSRV   010660          10354  10518#
R1SRV   011000          10545#
R2SRV   011120          10572#
R3SRV   011240          10599#
R5STAC  001332           8867#  8900
SEREND  011360          10363  10625#
SET   = 177777           8732#  9211   9234   9241   9247   9253   9256   9258   9293   9337   9380   9432   9490
                         9545   9556   9583   9627   9632   9700   9710   9934   9990  10008  10114  10717
SETCLR= 000006          10704#  10721
SFTREG  013512          10790*  10796  10799*  10883#
SIZE    012700           8877   10783#
SRVEND  007724          10093  10286#
STACK = 001100           8724#  8871   8884
START   001334           8834   8850   8871#
STATEN  012360          10351  10635#
STKLMT= 177774           8724#
SWR     001140           8857#  8871*  8872   10034  11069*  11075  11078
SWREG   000176           8834#  8871   8872   11069
SW0   = 000001           8724#
SW00  = 000001           8724#
```

```
SW01   = 000002          8724#
SW02   = 000004          8724#
SW03   = 000010          8724#
SW04   = 000020          8724#
SW05   = 000040          8724#
SW06   = 000100          8724#
SW07   = 000200          8724#
SW08   = 000400          8724#
SW09   = 001000          8724#
SW1    = 000002          8724#
SW10   = 002000          8724#
SW11   = 004000          8724#
SW12   = 010000          8724#
SW13   = 020000          8724#
SW14   = 040000          8724#
SW15   = 100000          8724#
SW2    = 000004          8724#
SW3    = 000010          8724#
SW4    = 000020          8724#
SW5    = 000040          8724#
SW6    = 000100          8724#
SW7    = 000200          8724#
SW8    = 000400          8724#
SW9    = 001000          8724#
TABEND   007764         10081   10312#
TABL1    013520         10817   10886#
TABL2    014116         10968   10969   11011#
TABL3    014126         10970   11016#
TABL4    014136         10995   10996   11021#
TABL5    014146         10997   11026#
TABL6    012416         10457   10655#
TABL7    012426         10416   10417   10660#
TABL8    012436         10419   10420   10665#
TBITVE=  000014          8724#
TBUF     001266          8864#   8899*   8964    9207*   9219*   9227*   9288*   9333*   9376*   9430*   9441*   9489*   9581*
                         9623*   9662*   9708*   9856*   9932*  10006*
TCH0     007734         10219   10301#
TCH1     007736         10239   10302#
TCH2     007740         10259   10303#
TCH3     007742         10279   10304#
TCNT     007772         10152*  10162*  10169*  10170*  10172*  10174   10317#
TCSR     001264          8863#   8898*   8960    8999    9007*   9009    9018*   9020    9029*   9032    9055    9063*   9065
                         9074*   9076    9085*   9088    9165    9211    9220    9231    9241    9545    9548*   9554*   9576*
                         9627    9700    9801*   9812*   9859*   9890*   9930*   9988*   9990    9999*
TDATA0=  000001          8820#
TDATA1=  000002          8819#
TDATA2=  000004          8818#
TDATA3=  000010          8817#
TDATA4=  000020          8816#
TDATA5=  000040          8815#
TDATA6=  000100          8814#
TDATA7=  000200          8813#
TEMP     007774         10210*  10211*  10213*  10221*  10222*  10224*  10231*  10232*  10234*  10241*  10242*  10244*  10251*
                        10252*  10254*  10261*  10262*  10264*  10271*  10272*  10274*  10281*  10282*  10284*  10318#
TIME   = 000000         10768#  10770
TIMER    012446          9211    9241    9293    9337    9380    9432    9490    9545    9556    9583    9627    9632    9665
```

```
                    9700   9710   9934   9990  10008  10114  10704#
TIMSAV  012610     10706*  10726  10731* 10744#
TKVEC = 000060      8724#  10904
TMP     012370     10519* 10520* 10521  10524* 10526* 10534* 10535* 10536* 10539* 10542* 10546* 10547* 10548
                   10551* 10553* 10561* 10562* 10563* 10566* 10569* 10573* 10574* 10575  10578* 10580* 10588*
                   10589* 10590* 10593* 10596* 10600* 10601* 10602  10605* 10607* 10615* 10616* 10617* 10620*
                   10623* 10642#
TMP0    012372     10409* 10444* 10461  10464  10522  10536  10537* 10644#
TMP0E   012374     10413  10445  10522  10645#
TMP1    012376     10445* 10549  10563  10564* 10646#
TMP1E   012400     10446  10549  10647#
TMP2    012402     10446* 10576  10590  10591* 10648#
TMP2E   012404     10447  10576  10649#
TMP3    012406     10447* 10603  10617  10618* 10650#
TMP3E   012410     10603  10651#
TMP4    012414     10426* 10431  10434* 10436* 10437  10529* 10556* 10583* 10610* 10653#
TMP6    013514     10789* 10793  10798* 10884#
TMP7    013606     10896* 10897* 10898  10909#
TPVEC = 000064      8724#
TRAN    005626      9786   9853#
TRAPVE= 000034      8724#  8871*
TRTVEC= 000014      8724#
TRUE  = 000001      8734#  8906   9192   9281   9325   9368   9606   9689   9758   9921   10035  10052  10485
                   10723  10726  10738  10905  10978  10989
TSTCON  013530     10127  10166  10393  10484  10896# 11002
TST1    002066      8923#
TST10   003440      9282   9319#
TST11   003566      9326   9362#
TST12   003712      9369   9407#
TST13   004222      9417   9420   9454   9467   9479   9504   9513   9528#
TST14   004512      9601#
TST15   005046      9607   9684#
TST16   005322      9691   9704   9714   9732   9737   9743   9753#
TST17   005772      9759   9834   9910#
TST2    002264      8986#
TST20   006502      9919   9923   10020  10029# 10502
TST21   007000     10070#
TST22   007776     10190  10338#
TST3    002440      8993   9044#
TST4    002614      9104#
TST5    002756      9157#
TST6    003016      9186#
TST7    003310      9193   9217   9260   9275#
TYPDS = 104405     10048  11063  11087#
TYPE  = 104401      8872  10036  10039  10043  10045  10047  10049  10807  10821  10847  10857  10858  10859
                   10863  10873  11043  11045  11048  11050  11052  11054  11063  11066  11069  11075  11081
                   11084  11087# 11093
TYPOC = 104402     10044  10046  11044  11049  11051  11053  11069  11087#
TYPON = 104404     11087#
TYPOS = 104403     11047  11087#
VECTOR  014160     10904* 10956* 10969* 10996* 11032#
WAIT    012620      9443   9565   9640  10000  10730  10768#
WHICHB= 000002     10704# 10714
WRAP  = 000020      8829#  9281   9325   9368   9414   9606   9689   9758   9918  10984
XMITIE= 000100      8793#  9055   9063   9065   9074   9076   9085   9088   9548   9554   9576   9801   9812
                    9859   9890
```

```
XMITRD= 000200           8792#   9165    9211    9231    9241    9545    9700
XMITOS  007454          10217#
XMIT1S  007534          10237#
XMIT2S  007614          10257#
XMIT3S  007674          10277#
XOSRV   010734          10533#
X1SRV   011054          10560#
X2SRV   011174          10587#
X3SRV   011314          10614#
$APTHD  001000           8856#
$ASTAT= ****** U        11072
$ATYC   015566          11072#
$ATY1   015542          11072#
$ATY3   015550          11066   11072#
$ATY4   015560          11072#  11075
$AUTOB  001134           8857#   8872*  11069
$BASE   001250           8857#  10789   10955
$BDADR  001122           8857#
$BDDAT  001126           8857#
$BELL   001164           8857#  11075
$BGNLE= 177777           8612#
$CHARC  014762          11066#*
$CKSWR  014766          11069#  11087
$CLKVE  017222          11093#
$CMTAG  001100           8857#   8871
$CM3  = 000000           8857#
$CNTLG  015513          11069#
$CNTLU  015506          11069#
$CPUOP  001222           8857#
$CRLF   001171           8857#  10049   10873   11054   11066   11069   11075
$DBLK   016704          11081#
$DEVCT  001204           8857#   8905*
$DEVM   001252           8857#  10783   10796*  10805   10817*  10824   10825   10826   10827
$DOAGN  014460          11063#
$DTBL   016674          11081#
$ENDAD  014450           8854    8872   11063#  11075
$ENDCT  014416           8871   11063#
$ENDMG  014467          11063#
$ENULL  014464          11063#
$ENV    001214           8857#   8872   11066   11072   11075
$ENVM   001215           8857#   8871   11066   11072
$EOP    014362          10822   10964   11063#
$EOPCT  014410           8871*  11063#
$ERFLG  001103           8857#  11075*  11078*
$ERMAX  001115           8857#   8871*  11078*
$ERRFL= 000000           8892#   8894#   8896#   8897#   8898#   8899#   8900#   8905#   8909#   8928#   8933#   8935#   8937#
                         8946#   8947#   8997#   9006#   9007#   9016#   9018#   9027#   9029#   9053#   9062#   9063#   9072#
                         9074#   9083#   9085#   9110#   9119#   9120#   9129#   9131#   9140#   9142#   9163#   9196#   9199#
                         9200#   9207#   9211#   9227#   9234#   9241#   9248#   9253#   9256#   9285#   9288#   9293#   9303#
                         9329#   9333#   9337#   9349#   9372#   9376#   9380#   9393#   9425#   9430#   9432#   9441#   9443#
                         9446#   9459#   9471#   9483#   9489#   9490#   9534#   9537#   9539#   9543#   9545#   9548#   9554#
                         9556#   9565#   9574#   9576#   9578#   9581#   9583#   9612#   9614#   9615#   9617#   9623#   9627#
                         9629#   9632#   9640#   9655#   9658#   9660#   9662#   9665#   9695#   9700#   9708#   9710#   9718#
                         9780#   9782#   9783#   9786#   9787#   9790#   9791#   9793#   9795#   9801#   9803#   9812#   9813#
                         9856#   9859#   9877#   9881#   9882#   9883#   9885#   9887#   9890#   9926#   9928#   9930#   9932#
                         9934#   9939#   9942#   9945#   9953#   9957#   9986#   9988#   9990#   9999#  10000#  10004#  10006#
```

```
                    10008#  10040#  10053#  10054#  10078#  10080#  10087#  10088#  10090#  10091#  10092#  10099#  10100#
                    10101#  10105#  10106#  10108#  10110#  10112#  10114#  10122#  10124#  10125#  10127#  10129#  10144#
                    10152#  10154#  10155#  10157#  10208#  10210#  10211#  10213#  10219#  10221#  10222#  10224#  10229#
                    10231#  10232#  10234#  10239#  10241#  10242#  10244#  10249#  10251#  10252#  10254#  10259#  10261#
                    10262#  10264#  10269#  10271#  10272#  10274#  10279#  10281#  10282#  10284#  10344#  10348#  10350#
                    10354#  10355#  10357#  10358#  10359#  10360#  10361#  10362#  10369#  10371#  10372#  10374#  10378#
                    10380#  10381#  10383#  10384#  10386#  10388#  10391#  10393#  10395#  10401#  10402#  10403#  10404#
                    10409#  10411#  10413#  10416#  10417#  10419#  10420#  10422#  10426#  10434#  10436#  10444#  10445#
                    10446#  10447#  10449#  10450#  10453#  10455#  10457#  10459#  10461#  10462#  10464#  10468#  10472#
                    10475#  10477#  10481#  10482#  10484#  10487#  10519#  10520#  10521#  10524#  10526#  10529#  10534#
                    10535#  10536#  10537#  10539#  10542#  10546#  10547#  10548#  10551#  10553#  10556#  10561#  10562#
                    10563#  10564#  10566#  10569#  10573#  10574#  10575#  10578#  10580#  10583#  10588#  10589#  10590#
                    10591#  10593#  10596#  10600#  10601#  10602#  10605#  10607#  10610#  10615#  10616#  10617#  10618#
                    10620#  10623#  10705#  10706#  10707#  10715#  10717#  10723#  10730#  10731#  10739#  10741#  10770#
                    10771#  10772#  10931#
$ERROR  016010       8871    11075#
$ERRPC  001116       8857#   11049   11075*
$ERRTB  001254       8857#
$ERTTL  001112       8857#   10048   10051*  11075*
$ESCAP  001162       8857#    8871*  11075   11078*
$ETABL  001214       8857#
$ETEND  001254       8856    8857#
$FATAL  001176       8857#   11046*  11047   11072*
$FFLG   016006      11072#*
$FILLC  001156       8857#   11066
$FILLS  001155       8857#   11066
$FSAND= 000310       8612#    8906    8941    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112
                     9122     9133    9145    9165    9192    9231    9247    9249    9281    9307    9325    9350    9368
                     9395     9413    9414    9449    9460    9474    9498    9507    9568    9606    9643    9689    9720
                     9727     9728    9729    9739    9758    9792    9814    9815    9816    9819    9822    9854    9879
                     9886     9918    9921    9940    9944    9947    9949    9951    9956    9968    9971    9974    9979
                    10015    10034   10035   10038   10052   10103   10126   10128   10181   10376   10392   10394   10415
                    10452    10467   10471   10483   10485   10492   10496   10522   10549   10576   10603   10714   10721
                    10738
$FSBAD= 000401       8612#    8906    8941    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112
                     9122     9133    9145    9165    9192    9231    9247    9249    9281    9307    9325    9350    9368
                     9395     9413    9414    9449    9460    9474    9498    9507    9568    9606    9643    9689    9720
                     9727     9728    9729    9739    9758    9792    9814    9815    9816    9819    9822    9854    9879
                     9886     9918    9921    9940    9944    9947    9949    9951    9956    9968    9971    9974    9979
                    10015    10034   10035   10038   10052   10103   10126   10128   10181   10376   10392   10394   10415
                    10452    10467   10471   10483   10485   10492   10496   10522   10549   10576   10603   10714   10721
                    10738
$FSBLA= 000170       8612#
$FSCAS= 000150       8612#
$FSDEC= 000220       8612#
$FSGOO= 000400       8612#    8906    8941    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112
                     9122     9133    9145    9165    9192    9214    9231    9242    9247    9249    9281    9296    9307
                     9325     9339    9350    9368    9382    9395    9413    9414    9435    9449    9460    9474    9493
                     9498     9507    9559    9568    9606    9635    9643    9667    9689    9701    9711    9720    9727
                     9728     9729    9739    9758    9792    9814    9815    9816    9819    9822    9854    9879    9886
                     9918     9921    9935    9940    9944    9947    9949    9951    9956    9968    9971    9974    9979
                     9993    10009   10015   10034   10035   10038   10052   10103   10116   10126   10128   10181   10376
                    10392    10394   10415   10452   10467   10471   10483   10485   10492   10496   10522   10549   10576
                    10603    10714   10721   10738
$FSIF = 000110       8612#    8906    8908    8910    8941    8944    8992    8994    8999    9002    9009    9012    9020
                     9023     9032    9035    9055    9058    9065    9068    9076    9079    9088    9091    9112    9115
```

```
                    9122    9125    9133    9136    9145    9148    9165    9172    9192    9194    9214    9218    9231
                    9237    9242    9245    9246    9247    9249    9254    9255    9257    9281    9283    9296    9299
                    9307    9310    9325    9327    9339    9343    9350    9354    9368    9370    9382    9386    9395
                    9399    9413    9414    9416    9418    9419    9421    9435    9438    9449    9455    9460    9468
                    9474    9480    9493    9496    9498    9505    9507    9511    9559    9562    9568    9571    9606
                    9608    9635    9638    9643    9646    9667    9670    9689    9692    9701    9705    9711    9715
                    9720    9723    9727    9728    9729    9733    9734    9738    9739    9744    9758    9760    9792
                    9794    9796    9814    9815    9816    9819    9822    9825    9828    9829    9832    9833    9854
                    9857    9879    9884    9886    9888    9891    9892    9918    9920    9921    9924    9925    9935
                    9938    9940    9943    9944    9946    9947    9949    9951    9954    9955    9956    9958    9959
                    9960    9968    9970    9971    9973    9974    9978    9979    9984    9993    9996   10009   10012
                   10015   10018   10034   10035   10037   10038   10041   10042   10050   10052   10055   10103   10116
                   10119   10120   10126   10128   10130   10131   10181   10188   10376   10385   10387   10392   10394
                   10396   10397   10415   10418   10421   10452   10467   10469   10471   10473   10480   10483   10485
                   10488   10489   10492   10495   10496   10499   10522   10527   10530   10549   10554   10557   10576
                   10581   10584   10603   10608   10611   10714   10716   10718   10721   10724   10738   10740
$F$INC= 000210      8612#   9695    9745   10771   10772   10773   10774
$F$LOO= 000200      8612#   9197    9258    9259   10712   10726   10732
$F$NAM= 000160      8612#
$F$NO = 000403      8612#   8941    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112    9122
                    9133    9145    9165    9192    9231    9247    9249    9281    9307    9325    9350    9368    9395
                    9413    9414    9449    9460    9474    9498    9507    9568    9606    9643    9689    9720    9727
                    9728    9729    9739    9758    9792    9814    9815    9816    9819    9822    9854    9879    9886
                    9918    9921    9944    9947    9949    9951    9956    9968    9971    9974    9979   10034   10035
                   10038   10052   10103   10126   10128   10181   10376   10392   10394   10415   10452   10467   10483
                   10485   10492   10522   10549   10576   10603   10714
$F$OR = 000320      8612#   8906    8941    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112
                    9122    9133    9145    9165    9192    9231    9247    9249    9281    9307    9325    9350    9368
                    9395    9413    9414    9449    9460    9474    9498    9507    9568    9606    9643    9689    9720
                    9727    9728    9729    9739    9758    9792    9814    9815    9816    9819    9822    9854    9879
                    9886    9918    9921    9940    9944    9947    9949    9951    9956    9968    9971    9974    9979
                   10015   10034   10035   10038   10052   10103   10126   10128   10181   10376   10392   10394   10415
                   10452   10467   10471   10483   10485   10492   10496   10522   10549   10576   10603   10714   10721
                   10738
$F$RTN= 000300      8612#   10704   10748   10768   10776   11042   11055
$F$SEL= 000140      8612#
$F$UNT= 000130      8612#   8934    8948    9807    9808   10079   10081   10089   10093   10102   10132   10349   10351
                   10356   10363   10375   10398   10414   10423   10430   10431   10435   10437   10451   10466   10479
                   10490
$F$WHI= 000120      8612#   9281    9325    9368    9606    9689    9739    9758    9854    9879    9918
$F$YES= 000402      8612#   8906    8941    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112
                    9122    9133    9145    9165    9192    9231    9247    9249    9281    9307    9325    9350    9368
                    9395    9413    9414    9449    9460    9474    9498    9507    9568    9606    9643    9689    9720
                    9727    9728    9729    9739    9758    9792    9814    9815    9816    9819    9822    9854    9879
                    9886    9918    9921    9940    9944    9947    9949    9951    9956    9968    9971    9974    9979
                   10015   10034   10035   10038   10052   10103   10126   10128   10181   10376   10392   10394   10415
                   10452   10467   10471   10483   10485   10492   10496   10522   10549   10576   10603   10714   10721
                   10738
$GDADR  001120      8857#
$GDDAT  001124      8857#
$GET42  014440     11063#
$GTSWR  015036     11069#  11087
$HD     000000      8721
$HIBTS  001000      8856#
$ICNT   001104      8857#  11078*
$IFLEV= 177777      8612#   8906#   8910#   8941#   8944#   8992#   8994#   8999#   9002#   9009#   9012#   9020#   9023#
```

```
                    9032#  9035#  9055#  9058#  9065#  9068#  9076#  9079#  9088#  9091#  9112#  9115#  9122#
                    9125#  9133#  9136#  9145#  9148#  9165#  9172#  9192#  9194#  9214#  9218#  9231#  9242#
                    9245#  9246#  9247#  9249#  9254#  9257#  9281#  9283#  9296#  9299#  9307#  9310#  9325#
                    9327#  9339#  9343#  9350#  9354#  9368#  9370#  9382#  9386#  9395#  9399#  9413#  9414#
                    9418#  9421#  9435#  9438#  9449#  9455#  9460#  9468#  9474#  9480#  9493#  9496#  9498#
                    9505#  9507#  9511#  9559#  9562#  9568#  9571#  9606#  9608#  9635#  9638#  9643#  9646#
                    9667#  9670#  9689#  9692#  9701#  9705#  9711#  9715#  9720#  9723#  9727#  9728#  9729#
                    9733#  9738#  9739    9744#  9758#  9760#  9792#  9796#  9814#  9815#  9816#  9819    9822
                    9828#  9832#  9833#  9854#  9857#  9879#  9886#  9891#  9892#  9918#  9921#  9924#  9925#
                    9935#  9938#  9940#  9943#  9944#  9946#  9947#  9949#  9951#  9954#  9956#  9958#  9959#
                    9960#  9968#  9970#  9971#  9973#  9974#  9978#  9979#  9984#  9993#  9996#  10009#  10012#
                   10015#  10018#  10034#  10035#  10038#  10041#  10042#  10050#  10052#  10055#  10103#  10116#  10119#
                   10120#  10126#  10128#  10130#  10131#  10181#  10188#  10376#  10387#  10392#  10394#  10396#  10397#
                   10415#  10421#  10452#  10467#  10469#  10471#  10473#  10480#  10483#  10485#  10488#  10489#  10492#
                   10495#  10496#  10499#  10522#  10530#  10549#  10557#  10576#  10584#  10603#  10611#  10714#  10718#
                   10721#  10724#  10738#  10740#
$INTAG  001135      8857#  11069
$ISKO = 000001      8906#  8910   8941#  8944   8992#  8994   8999#  9002   9009#  9012   9020#  9023   9032#
                    9035   9055#  9058   9065#  9068   9076#  9079   9088#  9091   9112#  9115   9122#  9125
                    9133#  9136   9145#  9148   9165#  9172   9192#  9194   9214#  9218   9231#  9246   9247#
                    9257   9281#  9283   9296#  9299   9307#  9310   9325#  9327   9339#  9343   9350#  9354
                    9368#  9370   9382#  9386   9395#  9399   9413#  9421   9435#  9438   9449#  9455   9460#
                    9468   9474#  9480   9493#  9496   9498#  9505   9507#  9511   9559#  9562   9568#  9571
                    9606#  9608   9635#  9638   9643#  9646   9667#  9670   9689#  9692   9701#  9705   9711#
                    9715   9720#  9723   9727#  9739#  9744   9758#  9760   9792#  9796   9814#  9833   9854#
                    9857   9879#  9892   9918#  9925   9935#  9938   9940#  9943   9944#  9946   9947#  9960
                    9968#  9970   9971#  9973   9974#  9978   9979#  9984   9993#  9996   10009#  10012  10015#
                   10018   10034#  10050  10052#  10055  10103#  10120  10126#  10131  10181#  10188  10376#  10387
                   10392#  10397   10415#  10421   10452#  10480   10483#  10489   10492#  10495   10496#  10499   10522#
                   10530   10549#  10557   10576#  10584   10603#  10611   10714#  10718   10721#  10724   10738#  10740
$ISK1 = 000001      9242#  9245   9249#  9254   9414#  9418   9728#  9738   9815#  9832   9886#  9891   9921#
                    9924   9949#  9959   10035#  10042  10116#  10119  10128#  10130  10394#  10396  10467#  10469
                   10471#  10473   10485#  10488
$ISK2 = 000001      9729#  9733   9816#  9819#  9822#  9828   9951#  9954   9956#  9958   10038#  10041
$ITEMB  001114      8857#  11046  11075*
$LF     001172      8857#  11066  11069   11075
$LFLG   016005     11072#*
$LOCTA= 177777      8612#  8906   8908   8910   8934   8941   8944   8948   8992   8994   8999   9002   9009
                    9012   9020   9023   9032   9035   9055   9058   9065   9068   9076   9079   9088   9091
                    9112   9115   9122   9125   9133   9136   9145   9148   9165   9172   9192   9194   9197
                    9214   9218   9231   9237   9242   9245   9246   9247   9249   9254   9255   9257   9258
                    9259   9281   9283   9296   9299   9307   9310   9325   9327   9339   9343   9350   9354
                    9368   9370   9382   9386   9395   9399   9413   9414   9416   9418   9419   9421   9435
                    9438   9449   9455   9460   9468   9474   9480   9493   9496   9498   9505   9507   9511
                    9559   9562   9568   9571   9606   9608   9635   9638   9643   9646   9667   9670   9689
                    9692   9695   9701   9705   9711   9715   9720   9723   9727   9728   9729   9733   9734
                    9738   9739   9744   9745   9758   9760   9792   9794   9796   9807   9808   9814   9815
                    9816   9819   9822   9825   9828   9829   9832   9833   9854   9857   9879   9884   9886
                    9888   9891   9892   9918   9920   9921   9924   9925   9935   9938   9940   9943   9944
                    9946   9947   9949   9951   9954   9955   9956   9958   9959   9960   9968   9970   9971
                    9973   9974   9978   9979   9984   9993   9996   10009  10012  10015  10018  10034  10035
                   10037   10038  10041  10042  10050  10052  10055  10079  10081  10089  10093  10102  10103
                   10116   10119  10120  10126  10128  10130  10131  10132  10181  10188  10349  10351  10356
                   10363   10375  10376  10385  10387  10392  10394  10396  10397  10398  10414  10415  10418
                   10421   10423  10430  10431  10435  10437  10451  10452  10466  10467  10469  10471  10473
                   10479   10480  10483  10485  10488  10489  10490  10492  10495  10496  10499  10522  10527
```

		10402 10403 10404 10409 10411 10413 10415 10416 10417 10418 10419 10420 10422
		10423 10426 10431 10434 10436 10437 10444 10445 10446 10447 10449 10450 10452
		10453 10455 10457 10459 10461 10462 10464 10467 10468 10471 10472 10475 10477
		10479 10481 10482 10483 10484 10485 10487 10490 10492 10496 10519 10520 10521
		10522 10524 10526 10527 10529 10534 10535 10536 10537 10539 10542 10546 10547
		10548 10549 10551 10553 10554 10556 10561 10562 10563 10564 10566 10569 10573
		10574 10575 10576 10578 10580 10581 10583 10588 10589 10590 10591 10593 10596
		10600 10601 10602 10603 10605 10607 10608 10610 10615 10616 10617 10618 10620
		10623 10705 10706 10707 10714 10715 10716 10717 10721 10723 10726 10730 10731
		10732 10738 10739 10741 10748 10770 10771 10772 10773 10774 10776 10931 11055
$LSTST=	177777	8612# 8616# 8906 8908 8910 8934 8941 8944 8948 8992 8994 8999 9002
		9009 9012 9020 9023 9032 9035 9055 9058 9065 9068 9076 9079 9088
		9091 9112 9115 9122 9125 9133 9136 9145 9148 9165 9172 9192 9194
		9197 9214 9218 9231 9237 9242 9245 9246 9247 9249 9254 9255 9257
		9258 9259 9281 9283 9296 9299 9307 9310 9325 9327 9339 9343 9350
		9354 9368 9370 9382 9386 9395 9399 9413 9414 9416 9418 9419 9421
		9435 9438 9449 9455 9460 9468 9474 9480 9493 9496 9498 9505 9507
		9511 9559 9562 9568 9571 9606 9608 9635 9638 9643 9646 9667 9670
		9689 9692 9695 9701 9705 9711 9715 9720 9723 9727 9728 9729 9733
		9734 9738 9739 9744 9745 9758 9760 9792 9794 9796 9807 9808 9814
		9815 9816 9819 9822 9825 9828 9829 9832 9833 9854 9857 9879 9884
		9886 9888 9891 9892 9918 9920 9921 9924 9925 9935 9938 9940 9943
		9944 9946 9947 9949 9951 9954 9955 9956 9958 9959 9960 9968 9970
		9971 9973 9974 9978 9979 9984 9993 9996 10009 10012 10015 10018 10034
		10035 10037 10038 10041 10042 10050 10052 10055 10079 10081 10089 10093 10102
		10103 10116 10119 10120 10126 10128 10130 10131 10132 10181 10188 10349 10351
		10356 10363 10375 10376 10385 10387 10392 10394 10396 10397 10398 10414 10415
		10418 10421 10423 10430 10431 10435 10437 10451 10452 10466 10467 10469 10471
		10473 10479 10480 10483 10485 10488 10489 10490 10492 10495 10496 10499 10522
		10527 10530 10549 10554 10557 10576 10581 10584 10603 10608 10611 10704 10712
		10714 10716 10718 10721 10724 10726 10732 10738 10740 10748 10768 10771 10772
		10773 10774 10776 11042 11055
$LSTTA=	000000	8612# 8614# 8908 8910 8934 8944 8994 9002 9012 9023 9035 9058 9068
		9079 9091 9115 9125 9136 9148 9172 9194 9197 9218 9237 9245 9246
		9254 9255 9257 9259 9281 9283 9299 9310 9325 9327 9343 9354 9368
		9370 9386 9399 9416 9418 9419 9421 9438 9455 9468 9480 9496 9505
		9511 9562 9571 9606 9608 9638 9646 9670 9689 9692 9695 9705 9715
		9723 9733 9734 9738 9739 9744 9745 9758 9760 9794 9796 9807 9808
		9819 9822 9825 9828 9829 9832 9833 9857 9879 9884 9888 9891 9892
		9918 9920 9924 9925 9938 9943 9946 9954 9955 9958 9959 9960 9970
		9973 9978 9984 9996 10012 10018 10037 10041 10042 10050 10055 10079 10089
		10102 10119 10120 10130 10131 10188 10349 10356 10375 10385 10387 10396 10397
		10414 10418 10421 10430 10435 10451 10466 10469 10473 10479 10480 10488 10489
		10490 10495 10499 10527 10530 10554 10557 10581 10584 10608 10611 10704 10712
		10716 10718 10724 10732 10740 10748 10768 10771 10772 10773 10774 10776 11042
		11055
$MADR1	001226	8857#
$MADR2	001232	8857#
$MADR3	001236	8857#
$MADR4	001242	8857#
$MAIL	001174	8856 8857# 8871 8872 8923 8986 9044 9104 9157 9186 9275 9319 9362
		9407 9528 9601 9684 9753 9910 10070 10338 11066 11075 11078
$MAMS1	001224	8857#
$MAMS2	001230	8857#
$MAMS3	001234	8857#
$MAMS4	001240	8857#

```
$MBADR  001002          8856#
$MCALL= 000000          5717#   8612
$MFLG   016004         11072#*
$MNEW   015531         11069#
$MSGAD  001210          8857#  11072*
$MSGLG  001212          8857#  11072*
$MSGTY  001174          8857#  11072*
$MSWR   015520         11069#
$MTYP1  001225          8857#
$MTYP2  001231          8857#
$MTYP3  001235          8857#
$MTYP4  001241          8857#
$MXCNT  016466         11078#
$NESTL= 177777          8612#   8906#   8908    8910#   8934#   8941#   8944#   8948#   8992#   8994#   8999#   9002#   9009#
                        9012#   9020#   9023#   9032#   9035#   9055#   9058#   9065#   9068#   9076#   9079#   9088#   9091#
                        9112#   9115#   9122#   9125#   9133#   9136#   9145#   9148#   9165#   9172#   9192#   9194#   9197#
                        9214#   9218#   9231#   9237    9242#   9245#   9246#   9247#   9249#   9254#   9255    9257#   9258
                        9259#   9281#   9283#   9296#   9299#   9307#   9310#   9325#   9327#   9339#   9343#   9350#   9354#
                        9368#   9370#   9382#   9386#   9395#   9399#   9413#   9414#   9416    9418#   9419    9421#   9435#
                        9438#   9449#   9455#   9460#   9468#   9474#   9480#   9493#   9496#   9498#   9505#   9507#   9511#
                        9559#   9562#   9568#   9571#   9606#   9608#   9635#   9638#   9643#   9646#   9667#   9670#   9689#
                        9692#   9695#   9701#   9705#   9711#   9715#   9720#   9723#   9727#   9728#   9729#   9733#   9734
                        9738#   9739    9744#   9745#   9758#   9760#   9792#   9794    9796#   9807#   9808#   9814#   9815#
                        9816#   9819    9822    9825    9828#   9829    9832#   9833#   9854#   9857#   9879#   9884    9886#
                        9888    9891#   9892#   9918#   9920    9921#   9924#   9925#   9935#   9938#   9940#   9943#   9944#
                        9946#   9947#   9949#   9951#   9954#   9955    9956#   9958#   9959#   9960#   9968#   9970#   9971#
                        9973#   9974#   9978#   9979#   9984#   9993#   9996#  10009#  10012#  10015#  10018#  10034#  10035#
                       10037   10038#  10041#  10042#  10050#  10052#  10055#  10079#  10081#  10089#  10093#  10102#  10103#
                       10116#  10119#  10120#  10126#  10128#  10130#  10131#  10132#  10181#  10188#  10349#  10351#  10356#
                       10363#  10375#  10376#  10385   10387#  10392#  10394#  10396#  10397#  10398#  10414#  10415#  10418
                       10421#  10423#  10430#  10431#  10435#  10437#  10451#  10452#  10466#  10467#  10469#  10471#  10473#
                       10479#  10480#  10483#  10485#  10488#  10489#  10490#  10492#  10495#  10496#  10499#  10522#  10527
                       10530#  10549#  10554   10557#  10576#  10581   10584#  10603#  10608   10611#  10704#  10712#  10714#
                       10716   10718#  10721#  10724#  10726   10732#  10738#  10740#  10748#  10768#  10771#  10772#  10773#
                       10774#  10776#  11042#  11055#
$NSK0 = 000300          8906#   8908    8910    8934#   8948    8992#   8994    8999#   9002    9009#   9012    9020#   9023
                        9032#   9035    9055#   9058    9065#   9068    9076#   9079    9088#   9091    9112#   9115    9122#
                        9125    9133#   9136    9145#   9148    9165#   9172    9192#   9194    9197#   9258    9259    9281#
                        9283    9296#   9299    9307#   9310    9325#   9327    9339#   9343    9350#   9354    9368#   9370
                        9382#   9386    9395#   9399    9413#   9419    9421    9435#   9438    9449#   9455    9460#   9468
                        9474#   9480    9493#   9496    9498#   9505    9507#   9511    9559#   9562    9568#   9571    9606#
                        9608    9635#   9638    9643#   9646    9667#   9670    9689#   9692    9695#   9745    9758#   9760
                        9792#   9794    9796    9807#   9808    9814#   9833    9854#   9857    9879#   9884    9892    9918#
                        9920    9925    9935#   9938    9940#   9943    9944#   9946    9947#   9960    9968#   9970    9971#
                        9973    9974#   9978    9979#   9984    9993#   9996   10009#  10012   10015#  10018   10034#  10050
                       10052#  10055   10079#  10081   10089#  10093   10102#  10132   10181#  10188   10349#  10351   10356#
                       10363   10375#  10398   10414#  10423   10430#  10431   10435#  10437   10451#  10490   10492#  10495
                       10496#  10499   10522#  10527   10530   10549#  10554   10557   10576#  10581   10584   10603#  10608
                       10611   10704#  10748   10768#  10776   11042#  11055
$NSK1 = 000210          8941#   8944    9214#   9218    9231#   9237    9246    9247#   9255    9257    9414#   9416    9418
                        9701#   9705    9711#   9715    9720#   9723    9727#   9739    9744    9815#   9829    9832    9886#
                        9888    9891    9921#   9924    9949#   9955    9959   10035#  10037   10042   10103#  10120   10126#
                       10131   10376#  10385   10387   10392#  10397   10415#  10418   10421   10452#  10480   10483#  10489
                       10712#  10726   10732   10738#  10740   10771#  10774
$NSK2 = 000210          9242#   9245    9249#   9254    9728#   9734    9738    9816#   9819    9822    9825    9828    9951#
                        9954    9956#   9958   10038#  10041   10116#  10119   10128#  10130   10394#  10396   10466#  10479
```

```
                    10485#  10488   10714#  10716   10718   10721#  10724   10772#  10773
$NSK3 = 000110       9729#   9733   10467#  10469   10471#  10473
$NULL   001154       8857#  11066
$NWTST= 000001       8923#   8986#   9044#   9104#   9157#   9186#   9275#   9319#   9362#   9407#   9528#   9601#   9684#
                     9753#   9910#  10029#  10070#  10338#
$OCNT   017136      11084#*
$OMODE  017140      11084#*
$OVER   016452      11078#
$PASS   001202       8857#   8871*  11063*  11078
$PASTM  001006       8856#
$QUES   001170       8857#  11066   11069   11075
$RDCHR  015250      11069#  11087
$RDDEC= ****** U    11087
$RDLIN  015370      11069#  11087
$RDOCT= ****** U    11087
$RDSZ = 000010      11069#
$RTNAD  014462      11063#
$R2A  = ****** U    11087
$SAVLE= 177777       8612#   9259#   9695#  10732#  10771#  10772#
$SAVRE= ****** U    11087
$SCOPE  016210       8871   11078#
$SETUP= 000127       8868#   8871    8872   11063   11069   11075   11078
$SSK0 = 000253       9259#   9695#  10732#  10771#  10772#
$STUP = 177777       8868#
$SVLAD  016416      11078#
$SVPC = 000430       8854#
$SWR  = 167400       8617#   8721    8723    8857    8871    8923    8986    9044    9104    9157    9186    9275    9319
                     9362    9407    9528    9601    9684    9753    9910   10029   10070   10338   11063   11075   11078
$SWREG  001216       8857#   8871
$SWRMK= 000000       8723   11078
$TAGLE= 177777       8612#   8906#   8908#   8910#   8934#   8941#   8944#   8948#   8992#   8994#   8999#   9002#   9009#
                     9012#   9020#   9023#   9032#   9035#   9055#   9058#   9065#   9068#   9076#   9079#   9088#   9091#
                     9112#   9115#   9122#   9125#   9133#   9136#   9145#   9148#   9165#   9172#   9192#   9194#   9197#
                     9214#   9218#   9231#   9237#   9242#   9245#   9246#   9247#   9249#   9254#   9255#   9257#   9258
                     9259#   9281#   9283#   9296#   9299#   9307#   9310#   9325#   9327#   9339#   9343#   9350#   9354#
                     9368#   9370#   9382#   9386#   9395#   9399#   9413#   9414#   9416#   9418#   9419#   9421#   9435#
                     9438#   9449#   9455#   9460#   9468#   9474#   9480#   9493#   9496#   9498#   9505#   9507#   9511#
                     9559#   9562#   9568#   9571#   9606#   9608#   9635#   9638#   9643#   9646#   9667#   9670#   9689#
                     9692#   9695#   9701#   9705#   9711#   9715#   9720#   9723#   9727#   9728#   9729#   9733#   9734#
                     9738#   9739#   9744#   9745#   9758#   9760#   9792#   9794#   9796#   9807#   9808#   9814#   9815#
                     9816#   9819#   9822#   9825#   9828#   9829#   9832#   9833#   9854#   9857#   9879#   9884#   9886#
                     9888#   9891#   9892#   9918#   9920#   9921#   9924#   9925#   9935#   9938#   9940#   9943#   9944#
                     9946#   9947#   9949#   9951#   9954#   9955#   9956#   9958#   9959#   9960#   9968#   9970#   9971#
                     9973#   9974#   9978#   9979#   9984#   9993#   9996#  10009#  10012#  10015#  10018#  10034#  10035#
                    10037#  10038#  10041#  10042#  10050#  10052#  10055#  10079#  10081#  10089#  10093#  10102#  10103#
                    10116#  10119#  10120#  10126#  10128#  10130#  10131#  10132#  10181#  10188#  10349#  10351#  10356#
                    10363#  10375#  10376#  10385#  10387#  10392#  10394#  10396#  10397#  10398#  10414#  10415#  10418#
                    10421#  10423#  10430#  10431#  10435#  10437#  10451#  10452#  10466#  10467#  10469#  10471#  10473#
                    10479#  10480#  10483#  10485#  10488#  10489#  10490#  10492#  10495#  10496#  10499#  10522#  10527#
                    10530#  10549#  10554#  10557#  10576#  10581#  10584#  10603#  10608#  10611#  10712#  10714#  10716#
                    10718#  10721#  10724#  10726   10732#  10738#  10740#  10771#  10772#  10773#  10774#
$TAGNU= 000257       8612#   8906#   8908#   8934#   8941#   8992#   8999#   9009#   9020#   9032#   9055#   9065#   9076#
                     9088#   9112#   9122#   9133#   9145#   9165#   9192#   9197#   9214#   9231#   9237#   9242#   9247#
                     9249#   9255#   9281#   9296#   9307#   9325#   9339#   9350#   9368#   9382#   9395#   9413#   9414#
                     9416#   9419#   9435#   9449#   9460#   9474#   9493#   9498#   9507#   9559#   9568#   9606#   9635#
                     9643#   9667#   9689#   9695#   9701#   9711#   9720#   9727#   9728#   9729#   9734#   9739#   9758#
```

```
                    9792#  9794#  9807#  9808#  9814#  9815#  9816#  9819#  9822#  9825#  9829#  9854#  9879#
                    9884#  9886#  9888#  9918#  9920#  9921#  9935#  9940#  9944#  9947#  9949#  9951#  9955#
                    9956#  9968#  9971#  9974#  9979#  9993# 10009# 10015# 10034# 10035# 10037# 10038# 10052#
                   10079# 10089# 10102# 10103# 10116# 10126# 10128# 10181# 10349# 10356# 10375# 10376# 10385#
                   10392# 10394# 10414# 10415# 10418# 10430# 10435# 10451# 10452# 10466# 10467# 10471# 10479#
                   10483# 10485# 10490# 10492# 10496# 10522# 10527# 10549# 10554# 10576# 10581# 10603# 10608#
                   10704# 10712# 10714# 10716# 10721# 10738# 10768# 10771# 10772# 11042#
$TEMP = 000300      8892#  8894#  8896#  8897#  8898#  8899#  8900#  8905#  8908#  8909#  8910#  8928#  8933#
                    8935#  8937#  8944#  8946#  8947#  8948#  8994#  8997#  9002#  9006#  9007#  9012#  9016#
                    9018#  9023#  9027#  9029#  9035#  9053#  9058#  9062#  9063#  9068#  9072#  9074#  9079#
                    9083#  9085#  9091#  9110#  9115#  9119#  9120#  9125#  9129#  9131#  9136#  9140#  9142#
                    9148#  9163#  9172#  9194#  9196#  9199#  9200#  9207#  9218#  9227#  9234#  9237#  9245#
                    9246#  9248#  9253#  9254#  9255#  9256#  9257#  9258#  9259#  9283#  9285#  9288#  9299#
                    9303#  9310#  9327#  9329#  9333#  9343#  9349#  9354#  9370#  9372#  9376#  9386#  9393#
                    9399#  9416#  9418#  9419#  9421#  9425#  9430#  9438#  9441#  9446#  9455#  9459#  9468#
                    9471#  9480#  9483#  9489#  9496#  9505#  9511#  9534#  9537#  9539#  9543#  9548#  9554#
                    9562#  9571#  9574#  9576#  9578#  9581#  9608#  9612#  9614#  9615#  9617#  9623#  9629#
                    9638#  9646#  9655#  9658#  9660#  9662#  9670#  9692#  9695#  9705#  9708#  9715#  9718#
                    9723#  9733#  9734#  9738#  9739#  9744#  9745#  9760#  9780#  9782#  9783#  9786#  9787#
                    9790#  9791#  9793#  9794#  9795#  9796#  9801#  9803#  9808#  9812#  9813#  9819#  9822#
                    9825#  9828#  9829#  9832#  9833#  9856#  9857#  9859#  9877#  9881#  9882#  9883#  9884#
                    9885#  9887#  9888#  9890#  9891#  9892#  9920#  9924#  9925#  9926#  9928#  9930#  9932#
                    9938#  9939#  9942#  9943#  9945#  9946#  9953#  9954#  9955#  9957#  9958#  9959#  9960#
                    9970#  9973#  9978#  9984#  9986#  9988#  9996#  9999# 10004# 10006# 10012# 10018# 10037#
                   10040# 10041# 10042# 10050# 10053# 10054# 10055# 10078# 10080# 10081# 10087# 10088# 10090#
                   10091# 10092# 10093# 10099# 10100# 10101# 10105# 10106# 10108# 10110# 10112# 10119# 10120#
                   10122# 10124# 10125# 10129# 10130# 10131# 10132# 10144# 10152# 10154# 10155# 10157# 10188#
                   10208# 10210# 10211# 10213# 10219# 10221# 10222# 10224# 10229# 10231# 10232# 10234# 10239#
                   10241# 10242# 10244# 10249# 10251# 10252# 10254# 10259# 10261# 10262# 10264# 10269# 10271#
                   10272# 10274# 10279# 10281# 10282# 10284# 10344# 10348# 10350# 10351# 10354# 10355# 10357#
                   10358# 10359# 10360# 10361# 10362# 10363# 10369# 10371# 10372# 10374# 10378# 10380# 10381#
                   10383# 10384# 10385# 10386# 10387# 10388# 10391# 10395# 10396# 10397# 10398# 10401# 10402#
                   10403# 10404# 10409# 10411# 10413# 10416# 10417# 10418# 10419# 10420# 10421# 10422# 10423#
                   10426# 10431# 10434# 10436# 10437# 10444# 10445# 10446# 10447# 10449# 10450# 10453# 10455#
                   10457# 10459# 10461# 10462# 10464# 10468# 10469# 10472# 10473# 10475# 10477# 10479# 10480#
                   10481# 10482# 10487# 10488# 10489# 10490# 10495# 10499# 10519# 10520# 10521# 10524# 10526#
                   10527# 10529# 10530# 10534# 10535# 10536# 10537# 10539# 10542# 10546# 10547# 10548# 10551#
                   10553# 10554# 10556# 10557# 10561# 10562# 10563# 10564# 10566# 10569# 10573# 10574# 10575#
                   10578# 10580# 10581# 10583# 10584# 10588# 10589# 10590# 10591# 10593# 10596# 10600# 10601#
                   10602# 10605# 10607# 10608# 10610# 10611# 10615# 10616# 10617# 10618# 10620# 10623# 10705#
                   10706# 10707# 10715# 10716# 10717# 10718# 10723# 10724# 10726# 10731# 10732# 10740# 10748#
                   10770# 10771# 10772# 10773# 10774# 10776# 10931# 11055#
$TESTN  001200      8857#  8923*  8986*  9044*  9104*  9157*  9186*  9275*  9319*  9362*  9407*  9528*  9601*
                    9684*  9753*  9910* 10070* 10338* 11044  11078*
$TIMES  001160      8857#  8871*  8923*  8986*  9044*  9104*  9157*  9186*  9275*  9319*  9362*  9407*  9528*
                    9601*  9684*  9753*  9910* 10029* 10070* 10338* 11063* 11078*
$TKB    001146      8857# 11069
$TKS    001144      8857# 10903  11069*
$TN  =  000023      8618#  8721   8923#  8986#  8993   9044#  9104#  9157#  9186#  9193   9260   9275#  9282
                    9319#  9326   9362#  9369   9407#  9417   9420   9454   9467   9479   9504   9513   9528#
                    9601#  9684#  9691   9704   9714   9732   9737   9743   9753#  9834   9910#  10020  10029#
                   10070# 10190  10338#
$TPB    001152      8857# 11066*
$TPFLG  001157      8857# 11066
$TPS    001150      8857# 11066
$TRAP   017142      8871  11087#
```

```
$TRAP2  017164          11087#
$TRP  = 000012          11087#
$TRPAD  017176          11087#
$TSK0 = 000251           8906#  8908#  8910   8934#  8948   8992#  8994   8999#  9002   9009#  9012   9020#  9023
                         9032#  9035   9055#  9058   9065#  9068   9076#  9079   9088#  9091   9112#  9115   9122#
                         9125   9133#  9136   9145#  9148   9165#  9172   9192#  9194   9197#  9259   9281#  9283
                         9296#  9299   9307#  9310   9325#  9327   9339#  9343   9350#  9354   9368#  9370   9382#
                         9386   9395#  9399   9413#  9419#  9421   9435#  9438   9449#  9455   9460#  9468   9474#
                         9480   9493#  9496   9498#  9505   9507#  9511   9559#  9562   9568#  9571   9606#  9608
                         9635#  9638   9643#  9646   9667#  9670   9689#  9692   9695#  9745   9758#  9760   9792#
                         9794#  9796   9807#  9808   9814#  9833   9854#  9857   9879#  9884#  9892   9918#  9920#
                         9925   9935#  9938   9940#  9943   9944#  9946   9947#  9960   9968#  9970   9971#  9973
                         9974#  9978   9979#  9984   9993#  9996   10009# 10012  10015# 10018  10034# 10050  10052#
                        10055  10079# 10081  10089# 10093  10102# 10132  10181# 10188  10349# 10351  10356# 10363
                        10375# 10398  10414# 10423  10430# 10431  10435# 10437  10451# 10490  10492# 10495  10496#
                        10499  10522# 10527# 10530  10549# 10554# 10557  10576# 10581# 10584  10603# 10608# 10611
                        10712# 10732  10738# 10740  10771# 10774
$TSK1 = 000250           8941#  8944   9197#  9258   9259   9414#  9416#  9418   9695#  9745   9815#  9829#  9832
                         9886#  9888#  9891   9921#  9924   9949#  9955#  9959   10035# 10037# 10042  10103# 10120
                        10126# 10131  10376# 10385# 10387  10392# 10397  10415# 10418# 10421  10452# 10480  10483#
                        10489  10712# 10726  10732  10771# 10774
$TSK2 = 000254           9214#  9218   9231#  9237#  9246   9247#  9255#  9257   9701#  9705   9711#  9715   9720#
                         9723   9727#  9739#  9744   9816#  9819#  9828   9951#  9954   9956#  9958   10038# 10041
                        10116# 10119  10128# 10130  10394# 10396  10466# 10479  10485# 10488  10714# 10716# 10718
                        10721# 10724  10772# 10773
$TSK3 = 000253           9242#  9245   9249#  9254   9728#  9734#  9738   9739#  9744   9819#  9822#  9828   10467#
                        10469  10471# 10473  10772# 10773
$TSK4 = 000126           9729#  9733   9822#  9825#  9828
$TSTM   001004           8856#
$TSTNM  001102           8857#  8909*  11063* 11075  11078*
$TTYIN  015476          11069#
$TYPBN= ****** U        11087
$TYPDS  016470          11081#  11087
$TYPE   014504          11066#  11072  11087
$TYPEC  014716          11066#  11069
$TYPEX  014764          11066#
$TYPOC  016740          11084#  11087
$TYPON  016754          11084#  11087
$TYPOS  016714          11084#  11087
$UNIT   001206           8857#  8880*  10054* 10898  10958  10962* 10966  10975* 10986*
$UNITM  001010           8856#
$USWR   001220           8857#  8992   9281   9325   9368   9414   9606   9689   9728   9739   9758   9792   9918
                         9947   9949   10415  10810  10814  10896  10984
$VECT1  001244           8857#  10956
$VECT2  001246           8857#
$XTSTR  016222          11078#
$YESNO= 000001           8897#  8898#  8899#  8905#  8946#  8947#  9007#  9018#  9029#  9063#  9074#  9085#  9120#
                         9131#  9142#  9248#  9349#  9539#  9548#  9554#  9576#  9617#  9629#  9658#  9695#  9801#
                         9803#  9812#  9813#  9859#  9882#  9883#  9885#  9887#  9890#  9930#  9942#  9945#  9953#
                         9957#  9988#  9999#  10040# 10054# 10092# 10105# 10106# 10108# 10112# 10122# 10124# 10125#
                        10129# 10211# 10213# 10222# 10224# 10232# 10234# 10242# 10244# 10252# 10254# 10262# 10264#
                        10272# 10274# 10282# 10284# 10359# 10362# 10378# 10380# 10381# 10383# 10384# 10386# 10388#
                        10391# 10395# 10422# 10436# 10455# 10459# 10462# 10468# 10472# 10475# 10477# 10481# 10482#
                        10487# 10520# 10524# 10526# 10529# 10535# 10537# 10539# 10542# 10547# 10551# 10553# 10556#
                        10562# 10564# 10566# 10569# 10574# 10578# 10580# 10583# 10589# 10591# 10593# 10596# 10601#
                        10605# 10607# 10610# 10616# 10618# 10620# 10623# 10731# 10771# 10772# 10931#
```

```
$$ARGC= 000000  10704#  10768#  11042#
$$BYTE= 000402   8906#   8941#   8992#   8999#   9009#   9020#   9032#   9055#   9065#   9076#   9088#   9112#   9122#
                 9133#   9145#   9165#   9192#   9231#   9247#   9249#   9281#   9307#   9325#   9350#   9368#   9395#
                 9413#   9414#   9449#   9460#   9474#   9498#   9507#   9568#   9606#   9643#   9689#   9720#   9727#
                 9728#   9729#   9739#   9758#   9792#   9814#   9815#   9816#   9819#   9822#   9854#   9879#   9886#
                 9918#   9921#   9940#   9944#   9947#   9949#   9951#   9956#   9968#   9971#   9974#   9979#  10015#
                10034#  10035#  10038#  10052#  10103#  10126#  10128#  10181#  10376#  10392#  10394#  10415#  10452#
                10467#  10471#  10483#  10485#  10492#  10496#  10522#  10549#  10576#  10603#  10714#  10721#  10738#
$$FLAG= 000001   8906#   8910#   8941#   8944#   8992#   8994#   8999#   9002#   9009#   9012#   9020#   9023#   9032#
                 9035#   9055#   9058#   9065#   9068#   9076#   9079#   9088#   9091#   9112#   9115#   9122#   9125#
                 9133#   9136#   9145#   9148#   9165#   9172#   9192#   9194#   9214#   9218#   9231#   9242#   9245#
                 9246#   9247#   9249#   9254#   9257#   9281#   9283#   9296#   9299#   9307#   9310#   9325#   9327#
                 9339#   9343#   9350#   9354#   9368#   9370#   9382#   9386#   9395#   9399#   9413#   9414#   9418#
                 9421#   9435#   9438#   9449#   9455#   9460#   9468#   9474#   9480#   9493#   9496#   9498#   9505#
                 9507#   9511#   9559#   9562#   9568#   9571#   9606#   9608#   9635#   9638#   9643#   9646#   9667#
                 9670#   9689#   9692#   9701#   9705#   9711#   9715#   9720#   9723#   9727#   9728#   9729#   9733#
                 9738#   9739#   9744#   9758#   9760#   9792#   9796#   9814#   9815#   9816#   9819#   9822#   9828#
                 9832#   9833#   9854#   9857#   9879#   9886#   9891#   9892#   9918#   9921#   9924#   9925#   9935#
                 9938#   9940#   9943#   9944#   9946#   9947#   9949#   9951#   9954#   9956#   9958#   9959#   9960#
                 9968#   9970#   9971#   9973#   9974#   9978#   9979#   9984#   9993#   9996#  10009#  10012#  10015#
                10018#  10034#  10035#  10038#  10041#  10042#  10050#  10052#  10055#  10103#  10116#  10119#  10120#
                10126#  10128#  10130#  10131#  10181#  10188#  10376#  10387#  10392#  10394#  10396#  10397#  10415#
                10421#  10452#  10467#  10469#  10471#  10473#  10480#  10483#  10485#  10488#  10489#  10492#  10495#
                10496#  10499#  10522#  10530#  10549#  10557#  10576#  10584#  10603#  10611#  10714#  10718#  10721#
                10724#  10738#  10740#
$$FROM= 000000   9211#   9241#   9293#   9337#   9380#   9432#   9443#   9490#   9545#   9556#   9565#   9583#   9627#
                 9632#   9640#   9665#   9700#   9710#   9934#   9990#  10000#  10008#  10114#  10127#  10393#  10484#
                10730#
$$GET4= 000000  11063#
$$LOC = 012576   8906#   8941#   8948#   8992#   8999#   9009#   9020#   9032#   9055#   9065#   9076#   9088#   9112#
                 9122#   9133#   9145#   9165#   9192#   9214#   9231#   9242#   9247#   9249#   9281#   9296#   9307#
                 9325#   9339#   9350#   9368#   9382#   9395#   9413#   9414#   9435#   9449#   9460#   9474#   9493#
                 9498#   9507#   9559#   9568#   9606#   9635#   9643#   9667#   9689#   9701#   9711#   9720#   9727#
                 9728#   9729#   9739#   9758#   9792#   9808#   9814#   9815#   9816#   9819#   9822#   9854#   9879#
                 9886#   9918#   9921#   9935#   9940#   9944#   9947#   9949#   9951#   9956#   9968#   9971#   9974#
                 9979#   9993#  10009#  10015#  10034#  10035#  10038#  10052#  10081#  10093#  10103#  10116#  10126#
                10128#  10132#  10181#  10351#  10363#  10376#  10392#  10394#  10398#  10415#  10423#  10431#  10437#
                10452#  10467#  10471#  10479#  10483#  10485#  10490#  10492#  10496#  10522#  10549#  10576#  10603#
                10714#  10721#  10738#
$$LOCN= 000000  10704#  107 8   10763#  10776   11042#  11055
$$RETU= 000000  10704#  10739#  10741#  10748   10768#  10776   11042#  11055
$$RTN1= 000255  10704#  10739   10748   10768#  10776   11042#  11055
$$RTN2= 000256  10704#  10741   10748   10768#  10776   11042#  11055
$$TO  = 000001   9211#   9241#   9293#   9337#   9380#   9432#   9443#   9490#   9545#   9556#   9565#   9583#   9627#
                 9632#   9640#   9665#   9700#   9710#   9934#   9990#  10000#  10008#  10114#  10127#  10393#  10484#
                10730#
$OFILL  017137  11084#*
$1      002062   8906    8908#
$10     002410   9020    9023#
$100    005162   9701    9705#
$101    005224   9711    9715#
$102    005240   9720    9723#
$103    005276   9727    9739#
$104    005270   9728    9734#
$105    005266   9729    9733#
$106    005274   9734    9738#
```

$107	005320	9739	9744#
$11	002440	9032	9035#
$110	005320	9739	9744#
$111	005360	9758#	
$112	005364	9758	9760#
$113	005452	9792	9794#
$114	005460	9794	9796#
$115	005476	9807#	9808
$116	005514	9808#	
$117	005620	9814	9833#
$12	002510	9055	9058#
$120	005616	9815	9829#
$121	005562	9816	9819#
$122	005614	9819	9828#
$123	005576	9819	9822#
$124	005614	9822	9828#
$125	005612	9822	9825#
$126	005614	9825	9828#
$127	005620	9829	9832#
$13	002536	9065	9068#
$130	005652	9854	9857#
$131	005712	9879#	
$132	005734	9879	9884#
$133	005766	9884	9892#
$134	005760	9886	9888#
$135	005766	9888	9891#
$136	006030	9918#	
$137	006036	9918	9920#
$14	002564	9076	9079#
$140	006052	9920	9925#
$141	006052	9921	9924#
$142	006134	9935	9938#
$143	006152	9940	9943#
$144	006166	9944	9946#
$145	006240	9947	9960#
$146	006224	9949	9955#
$147	006222	9951	9954#
$15	002614	9088	9091#
$150	006240	9955	9959#
$151	006240	9956	9958#
$152	006266	9968	9970#
$153	006300	9971	9973#
$154	006312	9974	9978#
$155	006324	9979	9984#
$156	006374	9993	9996#
$157	006464	10009	10012#
$16	002652	9112	9115#
$160	006476	10015	10018#
$161	006750	10034	10050#
$162	006574	10035	10037#
$163	006646	10037	10042#
$164	006646	10038	10041#
$165	006774	10052	10055#
$166	007022	10079#	10081
$167	007054	10089#	10093
$17	002700	9122	9125#

$170	007112	10102#	10132
$171	007202	10103	10120#
$172	007202	10116	10119#
$173	007244	10126	10131#
$174	007244	10128	10130#
$175	007422	10181	10188#
$176	010024	10349#	10351
$177	010044	10356#	10363
$2	002066	8908	8910#
$20	002726	9133	9136#
$200	010132	10375#	10398
$201	010170	10376	10385#
$202	010174	10385	10387#
$203	010232	10392	10397#
$204	010232	10394	10396#
$205	010274	10414#	10423
$206	010316	10415	10418#
$207	010326	10418	10421#
$21	002756	9145	9148#
$210	010360	10430#	10431
$211	010374	10435#	10437
$212	010450	10451#	10490
$213	010554	10452	10480#
$214	010510	10466#	10479
$215	010524	10467	10469#
$216	010534	10471	10473#
$217	010554	10479#	
$22	003016	9165	9172#
$220	010614	10483	10489#
$221	010614	10485	10488#
$222	010632	10490#	
$223	010644	10492	10495#
$224	010654	10496	10499#
$225	010726	10522	10527#
$226	010732	10527	10530#
$227	011046	10549	10554#
$23	003046	9192	9194#
$230	011052	10554	10557#
$231	011166	10576	10581#
$232	011172	10581	10584#
$233	011306	10603	10608#
$234	011312	10608	10611#
$235	012614	10739	10748#
$236	012616	10741	10748#
$237	012466	10712#	10732
$24	003052	9197#	9259
$240	012570	10726	10732#
$241	012504	10714	10716#
$242	012512	10716	10718#
$243	012530	10721	10724#
$244	012602	10738	10740#
$245	012676	10776#	
$246	012676	10776#	
$247	012644	10771#	
$25	003300	9258	9259#
$250	012640	10771#	10774

```
$251    012670          10771   10774#
$252    012656          10772#
$253    012654          10772#  10773
$254    012666          10772   10773#
$255    014360          11055#
$256    014360          11055#
$26     003126           9214    9218#
$27     003166           9231    9237#
$3      002130           8934#   8948
$30     003222           9237    9246#
$31     003222           9242    9245#
$32     003260           9247    9255#
$33     003256           9249    9254#
$34     003266           9255    9257#
$35     003346           9281#
$36     003350           9281    9283#
$37     003416           9296    9299#
$4      002154           8941    8944#
$40     003440           9307    9310#
$40CAT= ****** U        11075   11078
$41     003476           9325#
$42     003500           9325    9327#
$43     003546           9339    9343#
$44     003566           9350    9354#
$45     003624           9368#
$46     003626           9368    9370#
$47     003674           9382    9386#
$5      002314           8992    8994#
$50     003712           9395    9399#
$51     003754           9413    9419#
$52     003750           9414    9416#
$53     003752           9416    9418#
$54     003756           9419    9421#
$55     004024           9435    9438#
$56     004062           9449    9455#
$57     004102           9460    9468#
$6      002334           8999    9002#
$60     004124           9474    9480#
$61     004172           9493    9496#
$62     004206           9498    9505#
$63     004220           9507    9511#
$64     004404           9559    9562#
$65     004432           9568    9571#
$66     004550           9606#
$67     004554           9606    9608#
$7      002362           9009    9012#
$70     004724           9635    9638#
$71     004752           9643    9646#
$72     005032           9667    9670#
$73     005104           9689#
$74     005106           9689    9692#
$75     005116           9695#
$76     005114           9695#   9745
$77     005322           9695    9745#
.     = 017274           8834#   8841#   8854#   8855#   8856#   8857#   8866#   8871    8872#   8903    8906    8941    8948
                         8992    8999    9009    9020    9032    9055    9065    9076    9088    9112    9122    9133    9145
```

```
                      9165    9192    9214    9231    9242    9247    9249    9281    9296    9307    9325    9339    9350
                      9368    9382    9395    9413    9414    9435    9449    9460    9474    9493    9498    9507    9559
                      9568    9606    9635    9643    9667    9689    9701    9711    9720    9727    9728    9729    9739
                      9758    9792    9808    9814    9815    9816    9819    9822    9854    9879    9886    9918    9921
                      9935    9940    9944    9947    9949    9951    9956    9968    9971    9974    9979    9993   10009
                     10015   10034   10035  10036#  10038  10039#   10052   10081   10093   10103   10116   10126   10128
                     10132   10181  10311#   10351   10363   10376   10392   10394   10398   10415   10423   10431   10437
                     10452   10467   10471   10479   10483   10485   10490   10492   10496   10522   10549   10576   10603
                    10631#  10632#  10633#  10634#  10651#   10714   10721   10738   10808  10847#  10857#  10859#  11043#
                    11048#  11050#   11063   11066  11069#  11072#   11075   11078  11081#  11093#
.$ASTA= ****** U    11072
.$X   = 001000       8856#
```

```
BEGIN   7390#
BGNSRV  8624#  9853  9875  10206  10217  10227  10237  10247  10257  10267  10277  10518  10533  10545  10560
       10572  10587  10599  10614  10929
BGNSUB  8634#  8935  8997  9006  9016  9027  9053  9062  9072  9083  9110  9119  9129  9140  9163
        9285  9303  9329  9372  9425  9459  9471  9483  9543  9574  9614  9655  9926  9986
BRESET  8639#  8904  9031  9087  9144  9171  9305  9674  9693  9798  9962  10150  10440
CALL    7063#  9211  9241  9293  9337  9380  9432  9443  9490  9545  9556  9565  9583  9627  9632
        9640  9665  9700  9710  9934  9990  10000  10008  10114  10127  10393  10484  10730
CASE    7329#
CLRVEC  8649#
COMMEN  1566#  8724#
DECR    6996#
DECRU   7002#
DEFAUL  7372#
ELSE    6427#  8908  9237  9255  9416  9419  9734  9739  9794  9819  9822  9825  9829  9884  9888
        9920  9955  10037  10385  10418  10527  10554  10581  10608  10716
END     7407#
ENDCOM  1578#  8724#
ENDDEC  7007#
ENDDO   6498#
ENDIF   6445#  8910  8944  8994  9002  9012  9023  9035  9058  9068  9079  9091  9115  9125  9136
        9148  9172  9194  9218  9245  9246  9254  9257  9283  9299  9310  9327  9343  9354  9370
        9386  9399  9418  9421  9438  9455  9468  9480  9496  9505  9511  9562  9571  9608  9638
        9646  9670  9692  9705  9715  9723  9733  9738  9744  9760  9796  9828  9832  9833  9857
        9891  9892  9924  9925  9938  9943  9946  9954  9958  9959  9960  9970  9973  9978  9984
        9996  10012  10018  10041  10042  10050  10055  10119  10120  10130  10131  10188  10387  10396  10397
       10421  10469  10473  10480  10488  10489  10495  10499  10530  10557  10584  10611  10718  10724  10740
ENDINC  6983#  9745  10773  10774
ENDLOO  6871#  9259  10732
ENDRTN  7225#  10748  10776  11055
ENDSEL  7316#
ENDSRV  8660#  9860  9893  10214  10225  10235  10245  10255  10265  10275  10285  10531  10543  10558  10570
       10585  10597  10612  10624  10932
ENDSUB  8670#  8945  9003  9013  9024  9036  9059  9069  9080  9092  9116  9126  9137  9149  9174
        9301  9311  9344  9387  9456  9469  9481  9512  9572  9647  9671  9985  10019
ERRDF   8674#  8943
ERRHRD  8684#  9001  9011  9022  9034  9057  9067  9078  9090  9114  9124  9135  9147  9169  9216
        9244  9252  9298  9309  9341  9352  9384  9397  9437  9451  9463  9477  9495  9500  9509
        9561  9570  9587  9637  9645  9669  9703  9712  9722  9731  9736  9742  9818  9821  9824
        9827  9831  9937  9969  9972  9975  9980  9995  10011  10017  10118  10182  10494  10498
ERROR   8724#  8943  8954  8958  8962  8966  9001  9011  9022  9034  9057  9067  9078  9090  9114
        9124  9135  9147  9169  9216  9244  9252  9298  9309  9341  9352  9384  9397  9437  9451
        9463  9477  9495  9500  9509  9561  9570  9587  9637  9645  9669  9703  9712  9722  9731
        9736  9742  9818  9821  9824  9827  9831  9937  9965  9969  9972  9975  9980  9995  10011
       10017  10118  10182  10494  10498
ESCAPE  1698#  8724#
EXIF    6829#  9258
EXIFB   6851#  10726
EXIT    8694#  8993  9193  9260  9282  9326  9369  9417  9420  9454  9467  9479  9504  9513  9691
        9704  9714  9732  9737  9743  9834  10020  10190
GETPRI  1312#  8724#
GETSWR  1771#  8724#  8872#
IF      6332#  8941  8992  8999  9009  9020  9032  9055  9065  9076  9088  9112  9122  9133  9145
        9165  9192  9231  9247  9249  9281  9307  9325  9350  9368  9395  9413  9414  9449  9460
        9474  9498  9507  9568  9606  9643  9689  9720  9727  9728  9729  9758  9792  9814  9815
        9816  9854  9879  9886  9918  9921  9944  9947  9949  9951  9956  9968  9971  9974  9979
```

```
        10034   10035   10038   10052   10103   10126   10128   10181   10376   10392   10394   10415   10452   10467   10483
        10485   10492   10522   10549   10576   10603   10714
IFB      6338#   8906    9940   10015   10471   10496   10721   10738
IFCOND   6358#
IF.ERR   6368#   9214    9242    9296    9339    9382    9435    9493    9559    9635    9667    9701    9711    9935    9993
        10009   10116
IF.NO.   6380#
INCR     6972#   9695   10772
INCRU    6978#  10771
INLINE   7946#
LEAVE    7428#
LET      7922#   8892    8894    8896    8897    8898    8899    8900    8905    8909    8928    8933    8935    8937    8946
         8947    8997    9006    9007    9016    9018    9027    9029    9053    9062    9063    9072    9074    9083    9085
         9110    9119    9120    9129    9131    9140    9142    9163    9196    9199    9200    9207    9227    9234    9248
         9253    9256    9285    9288    9303    9329    9333    9349    9372    9376    9393    9425    9430    9441    9446
         9459    9471    9483    9489    9534    9537    9539    9543    9548    9554    9574    9576    9578    9581    9612
         9614    9615    9617    9623    9629    9655    9658    9660    9662    9708    9718    9780    9782    9783    9786
         9787    9790    9791    9793    9795    9801    9803    9812    9813    9856    9859    9877    9881    9882    9883
         9885    9887    9890    9926    9928    9930    9932    9939    9942    9945    9953    9957    9986    9988    9999
        10004   10006   10040   10053   10054   10078   10080   10087   10088   10090   10091   10092   10099   10100   10101
        10105   10106   10108   10110   10112   10122   10124   10125   10129   10144   10152   10154   10155   10157   10208
        10210   10211   10213   10219   10221   10222   10224   10229   10231   10232   10234   10239   10241   10242   10244
        10249   10251   10252   10254   10259   10261   10262   10264   10269   10271   10272   10274   10279   10281   10282
        10284   10344   10348   10350   10354   10355   10357   10358   10359   10360   10361   10362   10369   10371   10372
        10374   10378   10380   10381   10383   10384   10386   10388   10391   10395   10401   10402   10403   10404   10409
        10411   10413   10416   10417   10419   10420   10422   10426   10434   10436   10444   10445   10446   10447   10449
        10450   10453   10455   10457   10459   10461   10462   10464   10468   10472   10475   10477   10481   10482   10487
        10519   10520   10521   10524   10526   10529   10534   10535   10536   10537   10539   10542   10546   10547   10548
        10551   10553   10556   10561   10562   10563   10564   10566   10569   10573   10574   10575   10578   10580   10583
        10588   10589   10590   10591   10593   10596   10600   10601   10602   10605   10607   10610   10615   10616   10617
        10618   10620   10623   10705   10706   10707   10715   10717   10723   10731   10770   10931
LOCAL    7143#
LOOP     6691#   9197   10712
MSG      8911#   8923    8982#   8986    9100#   9104    9182#   9186    9271#   9275    9521#   9528    9596#   9601    9904#
         9910   10065#  10070   10327#  10338
MULT     4484#   8724#
NEWTST   1626#   8724#   8923    8986    9044    9104    9157    9186    9275    9319    9362    9407    9528    9601    9684
         9753    9910   10029   10070   10338
NOLOCA   7183#
POP      2153#   8724#  10775   11072   11081
PUSH     2145#   8724#  10769   11072   11081
REPEAT   6516#   8934    9807   10079   10089   10102   10349   10356   10375   10414   10430   10435   10451   10466
REPORT   5463#   8610#   8724#
RETURN   7196#  10739   10741
ROUTIN   7111#  10704   10768   11042
SAVR14   7138#
SCOPE    8724#   8923    8986    9044    9104    9157    9186    9275    9319    9362    9407    9528    9601    9684    9753
         9910   10029   10070   10338   11063
SELECT   7250#
SETPRI   1279#   8724#   9050    9551    9620    9672    9778   10084   10147   10365   10428
SETTRA  11087#
SETUP    1337#   8724#   8871
SETVEC   8706#   9612
SKIP     1733#   8724#   8993    9193    9260    9282    9326    9369    9417    9420    9454    9467    9479    9504    9513
         9691    9704    9714    9732    9737    9743    9834   10020   10190
SLASH    1517#   8724#
```

```
SPACE    8724#
STARS    1485#  8724#  8739  8741  8759  8761  8781  8783  8802  8804  8823  8833  8854  8856  8857
         8923   8977   8979  8986  9044  9104  9157  9186  9275  9319  9362  9407  9528  9601  9684
         9753   9846   9848  9868  9870  9910  10029 10070 10198 10200 10338 10510 10512 10678 10697
        10756  10763  10919 10924 10945 10949 11063 11066 11069 11072 11075 11081 11084 11087
STRUCT   5720#  8611#  8612
SWRSU    1453#  8724#  8871#
TRMTRP  11087#
TYPBIN   2088#  8724#
TYPDEC   2058#  8724#  10048 11063
TYPNAM   1826#  8724#  8872
TYPNUM   2025#  8724#
TYPOCS   1978#  8724#  11047
TYPOCT   1941#  8724#  10044 10046 11044 11049 11051 11053 11069
TYPTXT   1894#  8724#  10036 10039 10043 10045 10047 10807 10821 10847 10857 10859 11043 11045 11048
        11050  11052
UNTIL    6650#  8948   9808  10081 10093 10132 10351 10363 10398 10423 10431 10437 10479 10490
UNTILB   6671#
WAITMS   8714#  9443   9565  9640  10000 10730
WHILE    6487#
WHILEB   6492#
$ADDON   5805#  8906   8908  8934  8941  8992  8999  9009  9020  9032  9055  9065  9076  9088  9112
         9122   9133   9145  9165  9192  9197  9214  9231  9237  9242  9247  9249  9255  9259  9281
         9296   9307   9325  9339  9350  9368  9382  9395  9413  9414  9416  9419  9435  9449  9460
         9474   9493   9498  9507  9559  9568  9606  9635  9643  9667  9689  9695  9701  9711  9720
         9727   9728   9729  9734  9739  9758  9792  9794  9807  9808  9814  9815  9816  9819  9822
         9825   9829   9854  9879  9884  9886  9888  9918  9920  9921  9935  9940  9944  9947  9949
         9951   9955   9956  9968  9971  9974  9979  9993  10009 10015 10034 10035 10037 10038 10052
        10079  10089  10102 10103 10116 10126 10128 10181 10349 10356 10375 10376 10385 10392 10394
        10414  10415  10418 10430 10435 10451 10452 10466 10467 10471 10479 10483 10485 10490 10492
        10496  10522  10527 10549 10554 10576 10581 10603 10608 10704 10712 10714 10716 10721 10732
        10738  10768  10771 10772 11042
$AND     6092#  9739   9854
$BRANC   5931#  8906   8908  8941  8948  8992  8999  9009  9020  9032  9055  9065  9076  9088  9112
         9122   9133   9145  9165  9192  9214  9231  9237  9242  9247  9249  9255  9258  9259  9281
         929c   9307   9325  9339  9350  9368  9382  9395  9413  9414  9416  9419  9435  9449  9460
         9474   9493   9498  9507  9559  9568  9606  9635  9643  9667  9689  9695  9701  9711  9720
         9727   9728   9729  9734  9739  9745  9758  9792  9794  9808  9814  9815  9816  9819  9822
         9825   9829   9854  9879  9884  9886  9888  9918  9920  9921  9935  9940  9944  9947  9949
         9951   9955   9956  9968  9971  9974  9979  9993  10009 10015 10034 10035 10037 10038 10052
        10081  10093  10103 10116 10126 10128 10132 10181 10351 10363 10376 10385 10392 10394 10398
        10415  10418  10423 10431 10437 10452 10467 10471 10479 10483 10485 10490 10492 10496 10522
        10527  10549  10554 10576 10581 10603 10608 10714 10716 10721 10726 10732 10738 10739 10741
        10771  10772  10773 10774
$BRCOD   6230#  9258   9281  9325  9368  9606  9689  9695  9758  9808  9879  9918  10479 10490 10726
        10771  10772
$CALL    7039#  9211   9241  9293  9337  9380  9432  9443  9490  9545  9556  9565  9583  9627  9632
         9640   9665   9700  9710  9934  9990  10000 10008 10114 10127 10393 10484 10730
$CHECK   6288#  8906   8941  8992  8999  9009  9020  9032  9055  9065  9076  9088  9112  9122  9133
         9145   9165   9192  9231  9247  9249  9281  9307  9325  9350  9368  9395  9413  9414  9449
         9460   9474   9498  9507  9568  9606  9643  9689  9720  9727  9728  9729  9739  9758  9792
         9814   9815   9816  9819  9822  9854  9879  9886  9918  9921  9940  9944  9947  9949  9951
         9956   9968   9971  9974  9979  10015 10034 10035 10038 10052 10103 10126 10128 10181 10376
        10392  10394  10415 10452 10467 10471 10483 10485 10492 10496 10522 10549 10576 10603 10714
        10721  10738
$CHK1    7725#  8892   8894  8896  8900  8909  8928  8933  8935  8937  8997  9006  9016  9027  9053
```

```
                9062    9072    9083    9110    9119    9129    9140    9163    9196    9199    9200    9207    9227    9234    9253
                9256    9285    9288    9303    9329    9333    9372    9376    9393    9425    9430    9441    9446    9459    9471
                9483    9489    9534    9537    9543    9574    9578    9581    9612    9614    9615    9623    9655    9660    9662
                9695    9708    9718    9780    9782    9783    9786    9787    9790    9791    9793    9795    9856    9877    9881
                9926    9928    9932    9939    9986   10004   10006   10053   10078   10080   10087   10088   10090   10091   10099
               10100   10101   10110   10144   10152   10154   10155   10157   10208   10210   10219   10221   10229   10231   10239
               10241   10249   10251   10259   10261   10269   10271   10279   10281   10344   10348   10350   10354   10355   10357
               10358   10360   10361   10369   10371   10372   10374   10401   10402   10403   10404   10409   10411   10413   10416
               10417   10419   10420   10426   10434   10444   10445   10446   10447   10449   10450   10453   10457   10461   10464
               10519   10521   10534   10536   10546   10548   10561   10563   10573   10575   10588   10590   10600   10602   10615
               10617   10705   10706   10707   10715   10717   10723   10770   10771   10772
$CKOP2          7826#   8897    8898    8899    8905    8946    8947    9007    9018    9029    9063    9074    9085    9120    9131
                9142    9248    9349    9539    9548    9554    9576    9617    9629    9658    9801    9803    9812    9813    9859
                9882    9883    9885    9887    9890    9930    9942    9945    9953    9957    9988    9999   10040   10054   10092
               10105   10106   10108   10112   10122   10124   10125   10129   10211   10213   10222   10224   10232   10234   10242
               10244   10252   10254   10262   10264   10272   10274   10282   10284   10359   10362   10378   10380   10381   10383
               10384   10386   10388   10391   10395   10422   10436   10455   10459   10462   10468   10472   10475   10477   10481
               10482   10487   10520   10524   10526   10529   10535   10537   10539   10542   10547   10551   10553   10556   10562
               10564   10566   10569   10574   10578   10580   10583   10589   10591   10593   10596   10601   10605   10607   10610
               10616   10618   10620   10623   10731   10931
$CKR6           7554#
$CMND           6119#   8906    8941    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112    9122    9133
                9145    9165    9192    9231    9247    9249    9281    9307    9325    9350    9368    9395    9413    9414    9449
                9460    9474    9498    9507    9568    9606    9643    9689    9720    9727    9728    9729    9739    9758    9792
                9814    9815    9816    9819    9822    9854    9879    9886    9918    9921    9940    9944    9947    9949    9951
                9956    9968    9971    9974    9979   10015   10034   10035   10038   10052   10103   10126   10128   10181   10376
               10392   10394   10415   10452   10467   10471   10483   10485   10492   10496   10522   10549   10576   10603   10714
               10721   10738
$COMPA          6280#   8906    8941    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112    9122    9133
                9145    9165    9192    9214    9231    9242    9247    9249    9281    9296    9307    9325    9339    9350    9368
                9382    9395    9413    9414    9435    9449    9460    9474    9493    9498    9507    9559    9568    9606    9635
                9643    9667    9689    9695    9701    9711    9720    9727    9728    9729    9739    9758    9792    9814    9815
                9816    9819    9822    9854    9879    9886    9918    9921    9935    9940    9944    9947    9949    9951    9956
                9968    9971    9974    9979    9993   10009   10015   10034   10035   10038   10052   10103   10116   10126   10128
               10181   10376   10392   10394   10415   10452   10467   10471   10483   10485   10492   10496   10522   10549   10576
               10603   10714   10721   10738   10771   10772
$COUNT          7032#   9211    9241    9293    9337    9380    9432    9443    9490    9545    9556    9565    9583    9627    9632
                9640    9665    9700    9710    9934    9990   10000   10008   10114   10127   10393   10484   10730
$DO             6046#
$ELSE           6387#   9739    9819    9822
$ERRMS          6002#
$EXIFA          6705#
$EXIFO          6742#
$EXIF2          6800#  10726
$EXIF3          6779#
$GENBR          5924#   8906    8908    8941    8948    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112
                9122    9133    9145    9165    9192    9214    9231    9237    9242    9247    9249    9255    9258    9259    9281
                9296    9307    9325    9339    9350    9368    9382    9395    9413    9414    9416    9419    9435    9449    9460
                9474    9493    9498    9507    9559    9568    9606    9635    9643    9667    9689    9695    9701    9711    9720
                9727    9728    9729    9734    9739    9745    9758    9792    9794    9808    9814    9815    9816    9819    9822
                9825    9829    9854    9879    9884    9886    9888    9918    9920    9921    9935    9940    9944    9947    9949
                9951    9955    9956    9968    9971    9974    9979    9993   10009   10015   10034   10035   10037   10038   10052
               10081   10093   10103   10116   10126   10128   10132   10181   10351   10363   10376   10385   10392   10394   10398
               10415   10418   10423   10431   10437   10452   10467   10471   10479   10483   10485   10490   10492   10496   10522
               10527   10549   10554   10576   10581   10603   10608   10714   10716   10721   10726   10732   10738   10739   10741
               10771   10772   10773   10774
```

$GENTA 5900# 8908 8910 8934 8944 8994 9002 9012 9023 9035 9058 9068 9079 9091 9115
9125 9136 9148 9172 9194 9197 9218 9237 9245 9246 9254 9255 9257 9259 9281
9283 9299 9310 9325 9327 9343 9354 9368 9370 9386 9399 9416 9418 9419 9421
9438 9455 9468 9480 9496 9505 9511 9562 9571 9606 9608 9638 9646 9670 9689
9692 9695 9705 9715 9723 9733 9734 9738 9739 9744 9745 9758 9760 9794 9796
9807 9808 9819 9822 9825 9828 9829 9832 9833 9857 9879 9884 9888 9891 9892
9918 9920 9924 9925 9938 9943 9946 9954 9955 9958 9959 9960 9970 9973 9978
9984 9996 10012 10018 10037 10041 10042 10050 10055 10079 10089 10102 10119 10120 10130
10131 10188 10349 10356 10375 10385 10387 10396 10397 10414 10418 10421 10430 10435 10451
10466 10469 10473 10479 10480 10488 10489 10490 10495 10499 10527 10530 10554 10557 10581
10584 10608 10611 10712 10716 10718 10724 10732 10740 10748 10771 10772 10773 10774 10776
11055

$IF 6306# 8906 8941 8992 8999 9009 9020 9032 9055 9065 9076 9088 9112 9122 9133
9145 9165 9192 9231 9247 9249 9281 9307 9325 9350 9368 9395 9413 9414 9449
9460 9474 9498 9507 9568 9606 9643 9689 9720 9727 9728 9729 9739 9758 9792
9814 9815 9816 9819 9822 9854 9879 9886 9918 9921 9940 9944 9947 9949 9951
9956 9968 9971 9974 9979 10015 10034 10035 10038 10052 10103 10126 10128 10181 10376
10392 10394 10415 10452 10467 10471 10483 10485 10492 10496 10522 10549 10576 10603 10714
10721 10738

$IFCOD 6178# 8906 8941 8948 8992 8999 9009 9020 9032 9055 9065 9076 9088 9112 9122
9133 9145 9165 9192 9231 9247 9249 9281 9307 9325 9350 9368 9395 9413 9414
9449 9460 9474 9498 9507 9568 9606 9643 9689 9720 9727 9728 9729 9739 9758
9792 9808 9814 9815 9816 9819 9822 9854 9879 9886 9918 9921 9940 9944 9947
9949 9951 9956 9968 9971 9974 9979 10015 10034 10035 10038 10052 10081 10093 10103
10126 10128 10132 10181 10351 10363 10376 10392 10394 10398 10415 10423 10431 10437 10452
10467 10471 10479 10483 10485 10490 10492 10496 10522 10549 10576 10603 10714 10721 10738

$IFCON 6345# 9214 9242 9296 9339 9382 9435 9493 9559 9635 9667 9701 9711 9935 9993
10009 10116

$IFOPR 5944# 8906 8941 8948 8992 8999 9009 9020 9032 9055 9065 9076 9088 9112 9122
9133 9145 9165 9192 9214 9231 9242 9247 9249 9281 9296 9307 9325 9339 9350
9368 9382 9395 9413 9414 9435 9449 9460 9474 9493 9498 9507 9559 9568 9606
9635 9643 9667 9689 9701 9711 9720 9727 9728 9729 9739 9758 9792 9808 9814
9815 9816 9819 9822 9854 9879 9886 9918 9921 9935 9940 9944 9947 9949 9951
9956 9968 9971 9974 9979 9993 10009 10015 10034 10035 10038 10052 10081 10093 10103
10116 10126 10128 10132 10181 10351 10363 10376 10392 10394 10398 10415 10423 10431 10437
10452 10467 10471 10479 10483 10485 10490 10492 10496 10522 10549 10576 10603 10714 10721
10738

$LET 7909# 8892 8894 8896 8897 8898 8899 8900 8905 8909 8928 8933 8935 8937 8946
8947 8997 9006 9007 9016 9018 9027 9029 9053 9062 9063 9072 9074 9083 9085
9110 9119 9120 9129 9131 9140 9142 9163 9196 9199 9200 9207 9227 9234 9248
9253 9256 9285 9288 9303 9329 9333 9349 9372 9376 9393 9425 9430 9441 9446
9459 9471 9483 9489 9534 9537 9539 9543 9548 9554 9574 9576 9578 9581 9612
9614 9615 9617 9623 9629 9655 9658 9660 9662 9708 9718 9780 9782 9783 9786
9787 9790 9791 9793 9795 9801 9803 9812 9813 9856 9859 9877 9881 9882 9883
9885 9887 9890 9926 9928 9930 9932 9939 9942 9945 9953 9957 9986 9988 9999
10004 10006 10040 10053 10054 10078 10080 10087 10088 10090 10091 10092 10099 10100 10101
10105 10106 10108 10110 10112 10122 10124 10125 10129 10144 10152 10154 10155 10157 10208
10210 10211 10213 10219 10221 10222 10224 10229 10231 10232 10234 10239 10241 10242 10244
10249 10251 10252 10254 10259 10261 10262 10264 10269 10271 10272 10274 10279 10281 10282
10284 10344 10348 10350 10354 10355 10357 10358 10359 10360 10361 10362 10369 10371 10372
10374 10378 10380 10381 10383 10384 10386 10388 10391 10395 10401 10402 10403 10404 10409
10411 10413 10416 10417 10419 10420 10422 10426 10434 10436 10444 10445 10446 10447 10449
10450 10453 10455 10457 10459 10461 10462 10464 10468 10472 10475 10477 10481 10482 10487
10519 10520 10521 10524 10526 10529 10534 10535 10536 10537 10539 10542 10546 10547 10548
10551 10553 10556 10561 10562 10563 10564 10566 10569 10573 10574 10575 10578 10580 10583
10588 10589 10590 10591 10593 10596 10600 10601 10602 10605 10607 10610 10615 10616 10617

```
        10618   10620   10623   10705   10706   10707   10715   10717   10723   10731   10770   10931
$LPCNT   6927#   9695   10771   10772
$OPADD   7566#   8897    8898    8899    8905    8946    8947    9248    9539    9695    9883    9885   10040   10054   10092
        10106   10122   10125   10129   10211   10222   10232   10242   10252   10262   10272   10282   10359   10362   10378
        10381   10384   10386   10391   10395   10422   10462   10472   10475   10477   10481   10487   10520   10524   10535
        10537   10547   10551   10562   10564   10574   10578   10589   10591   10601   10605   10616   10618   10771   10772
        10931
$OPAND   7608#
$OPCD1   7649#   8897    8898    8899    8905    8946    8947    9007    9018    9029    9063    9074    9085    9120    9131
         9142    9248    9349    9539    9548    9554    9576    9617    9629    9658    9695    9801    9803    9812    9813
         9859    9882    9883    9885    9887    9890    9930    9942    9945    9953    9957    9988    9999   10040   10054
        10092   10105   10106   10108   10112   10122   10124   10125   10129   10211   10213   10222   10224   10232   10234
        10242   10244   10252   10254   10262   10264   10272   10274   10282   10284   10359   10362   10378   10380   10381
        10383   10384   10386   10388   10391   10395   10422   10436   10455   10459   10462   10468   10472   10475   10477
        10481   10482   10487   10520   10524   10526   10529   10535   10537   10539   10542   10547   10551   10553   10556
        10562   10564   10566   10569   10574   10578   10580   10583   10589   10591   10593   10596   10601   10605   10607
        10610   10616   10618   10620   10623   10731   10771   10772   10931
$OPCD2   7524#   10124   10388   10455   10459   10482
$OPCOD   7710#   8897    8898    8899    8905    8946    8947    9007    9018    9029    9063    9074    9085    9120    9131
         9142    9248    9349    9539    9548    9554    9576    9617    9629    9658    9695    9801    9803    9812    9813
         9859    9882    9883    9885    9887    9890    9930    9942    9945    9953    9957    9988    9999   10040   10054
        10092   10105   10106   10108   10112   10122   10124   10125   10129   10211   10213   10222   10224   10232   10234
        10242   10244   10252   10254   10262   10264   10272   10274   10282   10284   10359   10362   10378   10380   10381
        10383   10384   10386   10388   10391   10395   10422   10436   10455   10459   10462   10468   10472   10475   10477
        10481   10482   10487   10520   10524   10526   10529   10535   10537   10539   10542   10547   10551   10553   10556
        10562   10564   10566   10569   10574   10578   10580   10583   10589   10591   10593   10596   10601   10605   10607
        10610   10616   10618   10620   10623   10731   10771   10772   10931
$OPCOM   7461#
$OPDEF   5909#   8892    8894    8896    8897    8898    8899    8900    8905    8906    8908    8909    8928    8933    8935
         8937    8941    8946    8947    8948    8992    8997    8999    9006    9007    9009    9016    9018    9020    9027
         9029    9032    9053    9055    9062    9063    9065    9072    9074    9076    9083    9085    9088    9110    9112
         9119    9120    9122    9129    9131    9133    9140    9142    9145    9163    9165    9192    9196    9199    9200
         9207    9211    9214    9227    9231    9234    9237    9241    9242    9247    9248    9249    9253    9255    9256
         9258    9259    9281    9285    9288    9293    9296    9303    9307    9325    9329    9333    9337    9339    9349
         9350    9368    9372    9376    9380    9382    9393    9395    9413    9414    9416    9419    9425    9430    9432
         9435    9441    9443    9446    9449    9459    9460    9471    9474    9483    9489    9490    9493    9498    9507
         9534    9537    9539    9543    9545    9548    9554    9556    9559    9565    9568    9574    9576    9578    9581
         9583    9606    9612    9614    9615    9617    9623    9627    9629    9632    9635    9640    9643    9655    9658
         9660    9662    9665    9667    9689    9695    9700    9701    9708    9710    9711    9718    9720    9727    9728
         9729    9734    9739    9745    9758    9780    9782    9783    9786    9787    9790    9791    9792    9793    9794
         9795    9801    9803    9808    9812    9813    9814    9815    9816    9819    9822    9825    9829    9854    9856
         9859    9877    9879    9881    9882    9883    9884    9885    9886    9887    9888    9890    9918    9920    9921
         9926    9928    9930    9932    9934    9935    9939    9940    9942    9944    9945    9947    9949    9951    9953
         9955    9956    9957    9968    9971    9974    9979    9986    9988    9990    9993    9999   10000   10004   10006
        10008   10009   10015   10034   10035   10037   10038   10040   10052   10053   10054   10078   10080   10081   10087
        10088   10090   10091   10092   10093   10099   10100   10101   10103   10105   10106   10108   10110   10112   10114
        10116   10122   10124   10125   10126   10127   10128   10129   10132   10144   10152   10154   10155   10157   10181
        10208   10210   10211   10213   10219   10221   10222   10224   10229   10231   10232   10234   10239   10241   10242
        10244   10249   10251   10252   10254   10259   10261   10262   10264   10269   10271   10272   10274   10279   10281
        10282   10284   10344   10348   10350   10351   10354   10355   10357   10358   10359   10360   10361   10362   10363
        10369   10371   10372   10374   10376   10378   10380   10381   10383   10384   10385   10386   10388   10391   10392
        10393   10394   10395   10398   10401   10402   10403   10404   10409   10411   10413   10415   10416   10417   10418
        10419   10420   10422   10423   10426   10431   10434   10436   10437   10444   10445   10446   10447   10449   10450
        10452   10453   10455   10457   10459   10461   10462   10464   10467   10468   10471   10472   10475   10477   10479
        10481   10482   10483   10484   10485   10487   10490   10492   10496   10519   10520   10521   10522   10524   10526
        10527   10529   10534   10535   10536   10537   10539   10542   10546   10547   10548   10549   10551   10553   10554
```

```
        10556  10561  10562  10563  10564  10566  10569  10573  10574  10575  10576  10578  10580  10581  10583
        10588  10589  10590  10591  10593  10596  10600  10601  10602  10603  10605  10607  10608  10610  10615
        10616  10617  10618  10620  10623  10705  10706  10707  10714  10715  10716  10717  10721  10723  10726
        10730  10731  10732  10738  10739  10741  10748  10770  10771  10772  10773  10774  10776  10931  11055
$OPEQU   7642#
$OPNAN   7627#
$OPNEG   7473#
$OPNOR   7621#
$OPNOT   7616#  9018   9074   9131   9548   9576   9617   9658   9812   9813   9859   9882   9887   9999  10213
        10224  10234  10244  10254  10264  10274  10284  10542  10569  10596  10623
$OPOR    7603#  9007   9029   9063   9085   9120   9142   9349   9554   9629   9801   9803   9890   9930   9942
         9945   9953   9957   9988  10105  10108  10380  10383  10468  10526  10553  10580  10607
$OPROT   7493#
$OPRO    7752#  8892   8894   8896   8900   8909   8928   8933   8935   8937   8997   9006   9016   9027   9053
         9062   9072   9083   9110   9119   9129   9140   9163   9196   9199   9200   9207   9227   9234   9253
         9256   9285   9288   9303   9329   9333   9372   9376   9393   9425   9430   9441   9446   9459   9471
         9483   9489   9534   9537   9543   9574   9578   9581   9612   9614   9615   9623   9655   9660   9662
         9695   9708   9718   9780   9782   9783   9786   9787   9790   9791   9793   9795   9856   9877   9881
         9926   9928   9932   9939   9986  10004  10006  10053  10078  10080  10087  10088  10090  10091  10099
        10100  10101  10110  10144  10152  10154  10155  10157  10208  10210  10219  10221  10229  10231  10239
        10241  10249  10251  10259  10261  10269  10271  10279  10281  10344  10348  10350  10354  10355  10357
        10358  10360  10361  10369  10371  10372  10374  10401  10402  10403  10404  10409  10411  10413  10416
        10417  10419  10420  10426  10434  10444  10445  10446  10447  10449  10450  10453  10457  10461  10464
        10519  10521  10534  10536  10546  10548  10561  10563  10573  10575  10588  10590  10600  10602  10615
        10617  10705  10706  10707  10715  10717  10723  10770  10771  10772
$OPR1    7799#  9695  10771  10772
$OPR2    7868#  8897   8898   8899   8905   8946   8947   9007   9018   9029   9063   9074   9085   9120   9131
         9142   9248   9349   9539   9548   9554   9576   9617   9629   9658   9801   9803   9812   9813   9859
         9882   9883   9885   9887   9890   9930   9942   9945   9953   9957   9988   9999  10040  10054  10092
        10105  10106  10108  10112  10122  10124  10125  10129  10211  10213  10222  10224  10232  10234  10242
        10244  10252  10254  10262  10264  10272  10274  10282  10284  10359  10362  10378  10380  10381  10383
        10384  10386  10388  10391  10395  10422  10436  10455  10459  10462  10468  10472  10475  10477  10481
        10482  10487  10520  10524  10526  10529  10535  10537  10539  10542  10547  10551  10553  10556  10562
        10564  10566  10569  10574  10578  10580  10583  10589  10591  10593  10596  10601  10605  10607  10610
        10616  10618  10620  10623  10731  10931
$OPSHF   7510#  10124  10388  10455  10459  10482
$OPSUB   7586#  10112  10436  10529  10539  10556  10566  10583  10593  10610  10620  10731
$OPSWB   7485#
$OPXOR   7633#
$OR      6065#  9281   9325   9368   9606   9689   9758   9879   9918
$PUT     7022#  9211   9241   9293   9337   9380   9432   9443   9490   9545   9556   9565   9583   9627   9632
         9640   9665   9700   9710   9934   9990  10000  10008  10114  10730
$STRUC   5765#
$SUBON   5811#  8908   8910   8944   8948   8994   9002   9012   9023   9035   9058   9068   9079   9091   9115
         9125   9136   9148   9172   9194   9218   9237   9245   9246   9254   9255   9257   9259   9283   9299
         9310   9327   9343   9354   9370   9386   9399   9416   9418   9419   9421   9438   9455   9468   9480
         9496   9505   9511   9562   9571   9608   9638   9646   9670   9692   9695   9705   9715   9723   9733
         9734   9738   9739   9744   9745   9760   9794   9796   9808   9819   9822   9825   9828   9829   9832
         9833   9857   9884   9888   9891   9892   9920   9924   9925   9938   9943   9946   9954   9955   9958
         9959   9960   9970   9973   9978   9984   9996  10012  10018  10037  10041  10042  10050  10055  10081
        10093  10119  10120  10130  10131  10132  10188  10351  10363  10385  10387  10396  10397  10398  10418
        10421  10423  10431  10437  10469  10473  10479  10480  10488  10489  10490  10495  10499  10527  10530
        10554  10557  10581  10584  10608  10611  10716  10718  10724  10732  10740  10748  10771  10772  10773
        10774  10776  11055
$THEN    6027#  8906   8941   8992   8999   9009   9020   9032   9055   9065   9076   9088   9112   9122   9133
         9145   9165   9192   9231   9247   9249   9281   9307   9325   9350   9368   9395   9413   9414   9449
```

	9460	9474	9498	9507	9568	9606	9643	9689	9720	9727	9728	9729	9739	9758	9792
	9814	9815	9816	9819	9822	9854	9879	9886	9918	9921	9940	9944	9947	9949	9951
	9956	9968	9971	9974	9979	10015	10034	10035	10038	10052	10103	10126	10128	10181	10376
	10392	10394	10415	10452	10467	10471	10483	10485	10492	10496	10522	10549	10576	10603	10714
	10721	10738													
$TILA	6526#														
$TILO	6563#														
$UNTL2	6621#	9808	10479	10490											
$UNTL3	6600#														
$WHILE	6460#														
$$CMRE	8857#														
$$CMTM	8857#														
$$DEFA	7354#														
$$ENDS	7305#														
$$ERRO	6012#														
$$ESCA	1711#	8724#													
$$GEN	5888#	8908	8910	8934	8944	8994	9002	9012	9023	9035	9058	9068	9079	9091	9115
	9125	9136	9148	9172	9194	9197	9218	9237	9245	9246	9254	9255	9257	9259	9281
	9283	9299	9310	9325	9327	9343	9354	9368	9370	9386	9399	9416	9418	9419	9421
	9438	9455	9468	9480	9496	9505	9511	9562	9571	9606	9608	9638	9646	9670	9689
	9692	9695	9705	9715	9723	9733	9734	9738	9739	9744	9745	9758	9760	9794	9796
	9807	9808	9819	9822	9825	9828	9829	9832	9833	9857	9879	9884	9888	9891	9892
	9918	9920	9924	9925	9938	9943	9946	9954	9955	9958	9959	9960	9970	9973	9978
	9984	9996	10012	10018	10037	10041	10042	10050	10055	10079	10089	10102	10119	10120	10130
	10131	10188	10349	10356	10375	10385	10387	10396	10397	10414	10418	10421	10430	10435	10451
	10466	10469	10473	10479	10480	10488	10489	10490	10495	10499	10527	10530	10554	10557	10581
	10584	10608	10611	10704	10712	10716	10718	10724	10732	10740	10748	10768	10771	10772	10773
	10774	10776	11042	11055											
$$GETS	5851#	8908	8910	8944	8948	8994	9002	9012	9023	9035	9058	9068	9079	9091	9115
	9125	9136	9148	9172	9194	9218	9237	9245	9246	9254	9255	9257	9258	9259	9283
	9299	9310	9327	9343	9354	9370	9386	9399	9416	9418	9419	9421	9438	9455	9468
	9480	9496	9505	9511	9562	9571	9608	9638	9646	9670	9692	9695	9705	9715	9723
	9733	9734	9738	9739	9744	9745	9760	9794	9796	9808	9819	9822	9825	9828	9829
	9832	9833	9857	9884	9888	9891	9892	9920	9924	9925	9938	9943	9946	9954	9955
	9958	9959	9960	9970	9973	9978	9984	9996	10012	10018	10037	10041	10042	10050	10055
	10081	10093	10119	10120	10130	10131	10132	10188	10351	10363	10385	10387	10396	10397	10398
	10418	10421	10423	10431	10437	10469	10473	10479	10480	10488	10489	10490	10495	10499	10527
	10530	10554	10557	10581	10584	10608	10611	10716	10718	10724	10726	10732	10740	10748	10771
	10772	10773	10774	10776	11055										
$$GETT	5861#	8908	9237	9255	9258	9416	9419	9734	9739	9794	9819	9822	9825	9829	9884
	9888	9920	9955	10037	10385	10418	10527	10554	10581	10608	10716	10726			
$$LPCN	6887#	9695	10771	10772											
$$NEWT	1662#	8724#	8923	8986	9044	9104	9157	9186	9275	9319	9362	9407	9528	9601	9684
	9753	9910	10029	10070	10338										
$$POP	5875#	8908	8910	8944	8948	8994	9002	9012	9023	9035	9058	9068	9079	9091	9115
	9125	9136	9148	9172	9194	9218	9237	9245	9246	9254	9255	9257	9259	9283	9299
	9310	9327	9343	9354	9370	9386	9399	9416	9418	9419	9421	9438	9455	9468	9480
	9496	9505	9511	9562	9571	9608	9638	9646	9670	9692	9695	9705	9715	9723	9733
	9734	9738	9739	9744	9745	9760	9794	9796	9808	9819	9822	9825	9828	9829	9832
	9833	9857	9884	9888	9891	9892	9920	9924	9925	9938	9943	9946	9954	9955	9958
	9959	9960	9970	9973	9978	9984	9996	10012	10018	10037	10041	10042	10050	10055	10081
	10093	10119	10120	10130	10131	10132	10188	10351	10363	10385	10387	10396	10397	10398	10418
	10421	10423	10431	10437	10469	10473	10479	10480	10488	10489	10490	10495	10499	10527	10530
	10554	10557	10581	10584	10608	10611	10716	10718	10724	10732	10740	10748	10771	10772	10773
	10774	10776	11055												
$$PUSH	5839#	8906	8908	8934	8941	8992	8999	9009	9020	9032	9055	9065	9076	9088	9112

```
          9122    9133    9145    9165    9192    9197    9214    9231    9237    9242    9247    9249    9255    9259    9281
          9296    9307    9325    9339    9350    9368    9382    9395    9413    9414    9416    9419    9435    9449    9460
          9474    9493    9498    9507    9559    9568    9606    9635    9643    9667    9689    9695    9701    9711    9720
          9727    9728    9729    9734    9739    9758    9792    9794    9807    9814    9815    9816    9819    9822    9825
          9829    9854    9879    9884    9886    9888    9918    9920    9921    9935    9940    9944    9947    9949    9951
          9955    9956    9968    9971    9974    9979    9993   10009   10015   10034   10035   10037   10038   10052   10079
         10089   10102   10103   10116   10126   10128   10181   10349   10356   10375   10376   10385   10392   10394   10414
         10415   10418   10430   10435   10451   10452   10466   10467   10471   10483   10485   10492   10496   10522   10527
         10549   10554   10576   10581   10603   10608   10704   10712   10714   10716   10721   10732   10738   10768   10771
         10772   11042
$$SELE    7244#
$$SET    11087#
$$SETM    8871#
$$SETS    5820#   8906    8908    8934    8941    8992    8999    9009    9020    9032    9055    9065    9076    9088    9112
          9122    9133    9145    9165    9192    9197    9214    9231    9237    9242    9247    9249    9255    9259    9281
          9296    9307    9325    9339    9350    9368    9382    9395    9413    9414    9416    9419    9435    9449    9460
          9474    9493    9498    9507    9559    9568    9606    9635    9643    9667    9689    9695    9701    9711    9720
          9727    9728    9729    9734    9739    9758    9792    9794    9807    9814    9815    9816    9819    9822    9825
          9829    9854    9879    9884    9886    9888    9918    9920    9921    9935    9940    9944    9947    9949    9951
          9955    9956    9968    9971    9974    9979    9993   10009   10015   10034   10035   10037   10038   10052   10079
         10089   10102   10103   10116   10126   10128   10181   10349   10356   10375   10376   10385   10392   10394   10414
         10415   10418   10430   10435   10451   10452   10466   10467   10471   10483   10485   10492   10496   10522   10527
         10549   10554   10576   10581   10603   10608   10704   10712   10714   10716   10721   10732   10738   10768   10771
         10772   11042
$$SETT    5830#   9739    9819    9822
$$SKIP    1746#   8724#   8993    9193    9260    9282    9326    9369    9417    9420    9454    9467    9479    9504    9513
          9691    9704    9714    9732    9737    9743    9834   10020   10190
.EQUAT     189#   8609#   8724
.HEADE      61#   8608#   8721
.INIT     5658#   8724#  11093
.SETUP    1213#   8608#   8868
.SWRHI     104#   8609#   8723
.SWRLO    8723#
.$ACT1    5064#   8610#   8854
.$APTB    5109#   8610#   8857#
.$APTH    5370#   8610#   8856
.$APTY    5547#   8610#  11072
.$ASTA    5417#
.$CATC     932#   8608#   8834
.$CMTA    1047#   8608#   8857
.$DB2D    4686#
.$DB2O    4812#
.$DIV     4587#
.$EOP     2214#   8608#  11063
.$ERRO    2700#   8609#  11075
.$ERRT    2896#
.$MULT    4523#
.$POWE    4229#
.$RAND    4307#
.$RDDE    3891#
.$RDOC    3797#
.$READ    3395#   8608#  11069
.$R2AZ    4958#
.$SAVE    3969#
.$SB2D    4771#
.$SB2O    4874#
```

```
.$SCOP   2454#   8609#  11078
.$SIZE   4361#
.$SUPR   4913#
.$TRAP   4073#   8609#  11087
.$TYPB   3287#
.$TYPD   3209#   8609#  11081
.$TYPE   2985#   8608#  11066
.$TYPO   3112#   8610#  11084
.$40CA    972#


. ABS.  017274     000

 ERRORS DETECTED:  0

 CNDLAA,CNDLAA/CRF/NL:TOC=CNMAC2.SML,CNMAC.MAC,CNDLAA.P11
 RUN-TIME: 80 82 5 SECONDS
 RUN-TIME RATIO: 374/168=2.2
 CORE USED:  42K  (84 PAGES)
```