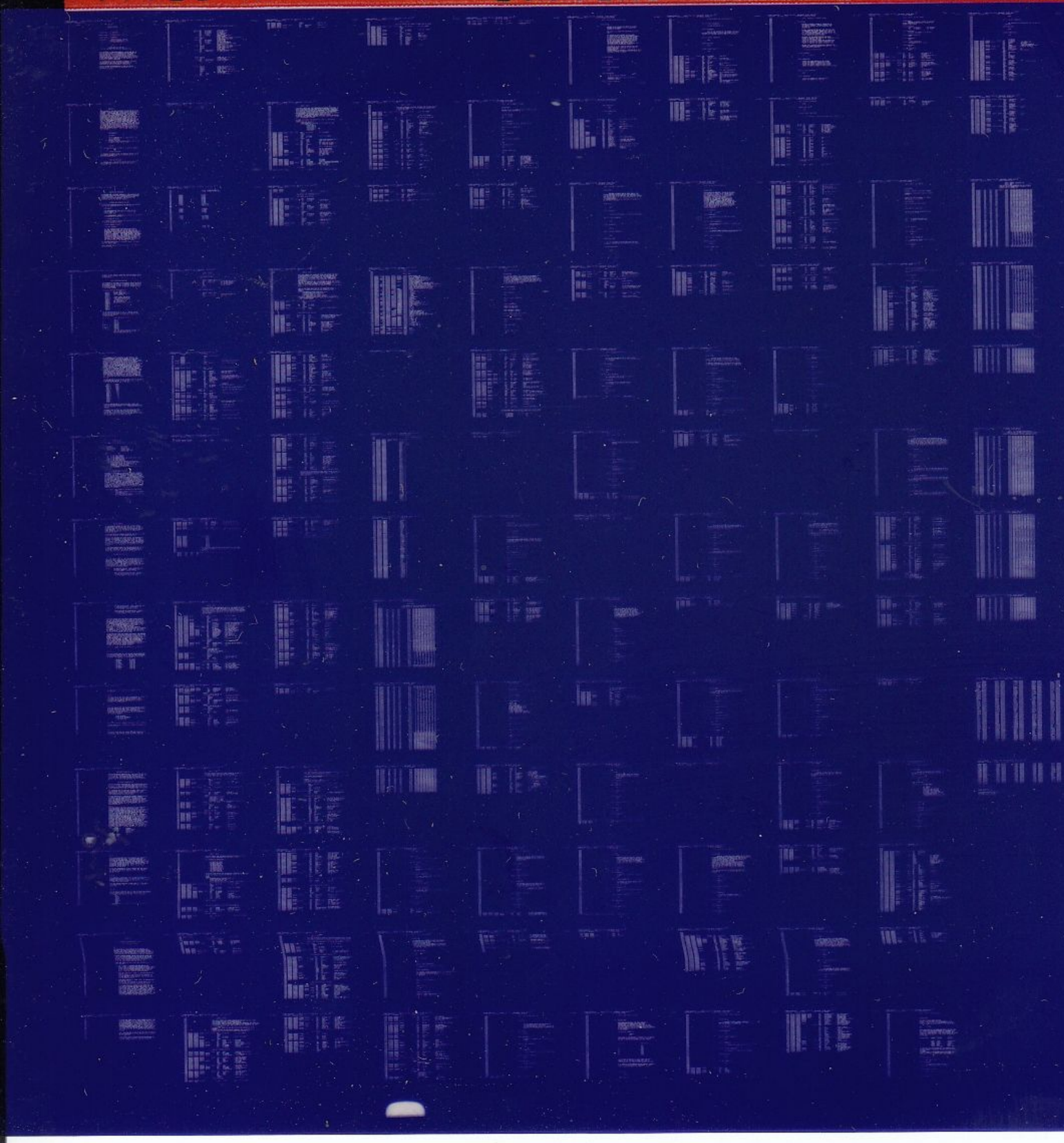
VTV30-K

VTV30-K LOGIC TEST
CVVTCAO

AH-S052A-MC
FICHE 1 OF 1

MAY 1981
COPYRIGHT© 1981
MADE IN USA



```
000000                                  .REPT 0

                                        IDENTIFICATION
                                        --------------

                        PRODUCT CODE:   AC-S051A-MC

                        PRODUCT NAME:   CVVTCA0 VTV30-K LOGIC TEST

                        PRODUCT DATE:   20-MAR-80

                        MAINTAINER:     COMPUTER SPECIAL SYSTEMS,
                                        DIGITAL EQUIPMENT CO. LTD.,
                                        READING, BERKSHIRE. U.K.
```

COPYRIGHT (C) 1981 BY
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

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

1. ABSTRACT

THIS DIAGNOSTIC HAS BEEN DESIGNED TO EXERCISE UP TO 20 (DECIMAL) VTV30-K, LSI BUS, VIDEO DISPLAY CONTROLLERS IN SEQUENCE. THE PROGRAM HAS SIX DEFAULT TESTS AND TWO OPTIONAL TESTS. THE PROGRAM ALSO HAS THE CAPABILITY TO RUN WITHOUT OPERATOR INTERVENTION, ONCE THE INITIAL PARAMETERS HAVE BEEN SET UP. IF MORE THAN ONE DEVICE IS BEING TESTED, THE OPERATOR MUST ENSURE THAT ALL THE DEVICES ARE SET UP IN ASCENDING ORDER AND THAT THEY ALL HAVE THE SAME NUMBER OF DISPLAY LINES AND CHARACTER SET TYPE. FOR EXAMPLE, THE DEVICE ADDRESSES RANGE THRU 174000, 174002, 174010, 174012, ETC., AND ALL ARE SET UP FOR 625-LINE OPERATION WITH A RAM BASED CHARACTER STORE. THE OPERATOR HAS THE FACILITY FOR HURRYING UP THE TESTS BY TYPING SPACE ON THE CONSOLE TERMINAL. THIS WILL ABORT AND PROGRAM CONTROLLED WAITS, AND PROCEED TO THE NEXT SECTION OF THE TEST. THIS FEATURE SHOULD NOT BE USED IN TEST 1.

2. REQUIREMENTS

2.1 EQUIPMENT

A. PDP-11 COMPUTER WITH A Q-BUS ADAPTOR, OR LSI-11 COMPUTER
B. CONSOLE TELETYPE
C. UP TO 20 VTV30-K COLOUR DISPLAY CONTROLLERS
D. DIAGNOSTIC TAPE AND LISTINGS

2.2 STORAGE

THIS PROGRAM REQUIRES A MINIMUM OF 8K WORDS OF MEMORY.

3. LOADING PROCEDURE

THE PROGRAM IS LOADED USING THE ABSOLUTE BINARY LOADER, OR USING THE NORMAL XXDP LOAD PROCEDURE AND IS IN ABSOLUTE BINARY FORMAT.

4. STARTING PROCEDURE

THE PROGRAM HAS A LOAD AND GO FEATURE WHICH AUTOMATICALLY STARTS THE PROGRAM AT ADDRESS 1000 UPON A SUCCESSFUL LOAD.

5. RESTARTING PROCEDURE

THE PROGRAM HAS A RESTART ADDRESS AT 1200 WHICH ALLOWS THE PROGRAM TO BE RESTARTED WITHOUT HAVING TO RE-ENTER THE BUS ADDRESSES. IF IT IS NECESSARY TO RESTART THE PROGRAM WITH NEW BUS ADDRESSES, THE ADDRESS 1000 SHOULD BE USED AS THE RESTART ADDRESS.

6. PROGRAM AND OPERATOR ACTION

THE FOLLOWING OPERATOR REQUESTS ARE MADE BY THE PROGRAM PRIOR TO THE COMMENCEMENT OF THE ACTUAL TESTS:-

M736?/M737?..

(THE OPERATOR MUST TYPE THE LAST DIGIT OF THE MODULE NUMBER-- 8,9,0,1 OR 2 -- TO SELECT THE DEVICE UNDER TEST

(IF A PROGRAMMABLE CHARACTER SET VERSION IS SELECTED, THE OPERATOR IS THEN ASKED....)

T.V.-ENCODED?(Y OR N)..

(..WHICH IS USED TO SELECT THE CHARACTER SETS LOADED INTO THE CHARACTER STORE.)

DECIMAL NUMBER OF UNITS (1)..

FIRST BUS ADDRESS =

(IF THE PROGRAM FOUND THAT ONE OF THE DEVICE REGISTERS SELECTED WAS NOT ON THE BUS, I.E. ITS ADDRESS WAS NON EXISTANT, THEN AN ERROR MESSAGE WILL BE PRINTED. FOR A CSR ADDRESS BEING NON EXISTANT, ERROR 77 IS REPORTED. FOR A DBUF ADDRESS, THE ERROR REPORTED IS ERROR 76. THE PARAMETER DATA CONTAINS THE ADDRESS FOUND TO BE NON EXISTANT. BY SELECTING TRAP+10, THE OPERATOR CAN LOOP ON THIS ADDRESS. IF THE FAULT IS NOT CORRECTED, THEN WHEN THE TESTS ARE RUN ON THAT DEVICE, THE PROGRAM WILL CRASH.)

THE OPERATOR SHOULD REPLY TO REQUESTS ABOVE, BY INPUT-TING THE CORRECT DATA.

SELECT DESIRED SWITCH REGISTER SETTINGS, TYPE CNTRL-C TO CONTINUE OR SWR = 0.

```
IN REPLY TO THE REQUEST ABOVE THE OPERATOR SHOULD SELECT
DESIRED SWITCH REGISTER OPTIONS AS SET OUT UNDER SWITCH
OPTIONS BELOW.


7.          SWITCH REGISTER OPTIONS

THIS PROGRAM IS DESIGNED TO RUN EQUALLY EASILY ON PDP-11
PROCESSORS WITH, OR WITHOUT, A HARDWARE SWITCH REGISTER.
ON STARTING, A TEST IS DONE TO SEE IF A HARDWARE SWITCH
REGISTER IS PRESENT.  IF IT IS PRESENT, IT MAY BE USED
IN THE NORMAL MANNER.

THE SWITCH REGISTER SETTINGS ARE:-

    SWR15=1     INHIBIT ERROR HALT
    SWR14=1     INHIBIT ERROR PRINT-OUT
    SWR13=1     FAST ITERATION
    SWR12=1     :
    SWR11=1     :SCOPE LOOPS, SEE BELOW
    SWR10=1     :FOR A DESCRIPTION, AND
    SWR09=1     :APPENDIX A FOR EXAMPLES
    SWR08=1     :OF THEIR USE
    SWR07=1     NON INTERVENTION
    SWR06=1     PRE-SELECTED TEST INDICATOR
    SWR05=1     :
    SWR04=1     :
    SWR03=1     :TEST NOS.
    SWR02=1     :
    SWR01=1     :
    SWR00=1     :

THE SETTING OF BITS 7, 8, 9, 10, 11, AND 12 IN THE
SWITCH REGISTER ARE USED TO SELECT THE TRAP OPTIONS
PRESENT IN THE PROGRAM. THE SELECTION IS MADE IN THE
FOLLOWING MANNER:

BIT(S) SET      TRAP FUNCTION SELECTED

7               TRAP+2
8               TRAP+4
9               TRAP+10
10              TRAP+20
9 AND 10        TRAP+30
11              TRAP+40
9 AND 11        TRAP+50
10 AND 11       TRAP+60
9, 10 AND 11    TRAP+70
12              USES THE SWITCH REGISTER SETTING THAT
                WAS IN FORCE WHEN THE LAST TRAP
                INSTRUCTION WAS EXECUTED.
```

IF A HARDWARE SWITCH REGISTER IS NOT PRESENT, THE PROGRAM ASSIGNS A LOCATION IN MEMORY AS A SOFTWARE SWITCH REGISTER, THE OPTIONS REMAINING AS ABOVE. THIS MEANS THAT ALL MODIFICATIONS TO THE SWITCH REGISTER MAY BE MADE USING THE CONSOLE TELETYPE VIA A MONITOR ROUTINE. THIS MONITOR IS CALLED BY TYPING CTRL-G AT THE CONSOLE TELETYPE AND RESPONDS BY PRINTING THE CURRENT CONTENTS OF THE SOFTWARE SWITCH REGISTER, FOLLOWED BY A PROMPT CHARACTER (>). THE OPERATOR SHOULD THEN TYPE IN THE NEW SWITCH REGISTER SETTINGS AS AN OCTAL NUMBER, FOLLOWED BY A CARRIAGE RETURN. TYPING CARRIAGE RETURN ALONE WILL CAUSE THE SETTING TO REMAIN UNCHANGED. THE SWITCH REGISTER IS THEN LOADED WITH THE NEW VALUE AND PROGRAM EXECUTION CONTINUES. IF A SETTING IS ENTERED WHICH INCLUDES THE PRE-SELECTED TEST BIT (SWR06), THE TEST INDICATED BY SWR 00-05 WILL BE SELECTED IMMEDIATELY. THIS DOES NOT APPLY WHEN DEFAULTING ON AN EXISTING SETTING.

THE SWR MONITOR IS ALSO CALLED AUTOMATICALLY IF AN ERROR IS DETECTED AND SWR BIT 15 IS NOT SET. OCTAL EQUIVALENTS FOR THE SWITCHES ARE AS FOLLOWS:

```
SWR15 =    100000
SWR14 =     40000
SWR13 =     20000
SWR12 =     10000
SWR11 =      4000
SWR10 =      2000
SWR09 =      1000
SWR08 =       400
SWR07 =       200
SWR06 =       100
SWR05 =        40
SWR04 =        20
SWR03 =        10
SWR02 =         4
SWR01 =         2
SWR00 =         1
```

TO SET A COMBINATION OF THESE SWITCHES, SIMPLY ADD TOGETHER THE CORRESPONDING OCTAL NUMBERS AND ENTER THE TOTAL IN RESPONSE TO ''SWR= X>''. (LEADING ZEROS MAY BE IGNORED).

FOR WORST CASE TESTING, ALL SWITCHES SHOULD BE ZERO. IT IS POSSIBLE, WITH THESE SWITCH REGISTER OPTIONS, TO EXECUTE ONLY A PRE-SELECTED TEST WITH THE FACILITY TO LOOP ON THAT TEST OR TO START THE PROGRAM PASS OR FINISH THE PROGRAM PASS AT ANY PARTICULAR TEST.

```
                    8.          ERROR REPORTS

                    THE FORMAT OF THE ERROR REPORTS IS AS FOLLOWS:-


                        E# AABB                 AT PC:CCCC
                        GOOD: DDDD              BAD: EEEE
                        STATUS: FFFF            ADDRES: GGGG
                        DATA: KKKK              CALLED FROM: HHHH
                                                ERROR COUNT = JJJJ

                    WHERE:

                        AA   IS THE TEST NUMBER
                        BB   IS THE ERROR NUMBER
                        CCCC IS THE ADDRESS WHERE THE ERROR REPORT OCCURS.
                        DDDD IS THE DATA EXPECTED
                        EEEE IS THE DATA RECEIVED
                        FFFF GGGG AND KKKK ARE CONTENTS OF REGISTERS.
                        HHHH IS THE ADDRESS IN THE MAINLINE CODE WHERE THE
                             SUBROUTINE, WHERE THE ERROR REPORT IS
                             GENERATED, IS CALLED FROM.
                        JJJJ IS THE NUMBER OF ERRORS REPORTED TO DATE IN
                             THIS SECTION.


                    9.          ONLINE MODIFICATIONS

                    AVAILABLE TO THE USER IS A ROUTINE TO MODIFY PROGRAM
                    LOCATIONS.  IT IS ENTERED BY TYPING CNTRL-O DURING THE
                    RUNNING OF THE TESTS.  ON ENTRY, A PROMPT '$' IS MADE
                    FOR THE ADDRESS TO BE MODIFIED.  IF NO ADDRESS IS GIVEN,
                    IT IS ASSUMED THAT NO MODIFICATIONS ARE REQUIRED AND THE
                    ROUTINE WILL COMPLETE.  IF AN ADDRESS IS SPECIFIED, IT
                    WILL BE CHECKED TO SEE IF IT IS EVEN AND IN EXISTANCE.
                    ONCE IT HAS BEEN CHECKED, ITS CONTENTS ARE DISPLAYED AND
                    A PROMPT '/' IS MADE FOR THE NEW CONTENTS.  IF NO NEW
                    VALUE IS GIVEN, THE EXISTING VALUE WILL BE LOADED.
                    HAVING DEALT WITH THAT ADDRESS, THE ROUTINE WILL THEN
                    EXAMINE THE TERMINATING CHARACTER TO DETERMINE THE NEXT
                    OPERATION TO PERFORM.

                    TYPING  <ESC>  WILL COMPLETE THE MODIFICATIONS BEING
                                   DONE.
                            <CR>   WILL CAUSE A PROMPT FOR THE NEXT ADDRESS
                                   TO BE MODIFIED.
                            <LF>   WILL TAKE THE NEXT ADDRESS TO BE MODIFIED
                                   AS THE CURRENT ADDRESS+2.
```

10. MISCELLANEOUS

IF THE OPERATOR WISHES TO SOAK TEST WITHOUT MANUAL INTERVENTION, THEN HE MUST SET SWR BITS 15, 14 AND 7. THIS WILL RUN THE TESTS WITHOUT DISPLAYING AND TEST OR ERROR MESSAGES. IF, AT THE END OF THE RUN, A SUMMARY ERROR REPORT IS REQUIRED, THEN TEST 10 SHOULD BE SELECTED AND RUN.

TO RUN THE TESTS WITH ERROR REPORTING, BUT WITH NO ERROR HALT, ONLY SWR BITS 15 AND 7 NEED TO BE SET.

ONCE ALL THE DEVICES HAVE BEEN TESTED, THE PROGRAM WILL ATTEMPT TO DETERMINE IF IT WAS CALLED FROM THE XXDP MONITOR. IF LOCATIONS 42/43 ARE NOT ZERO, THE PROGRAM WILL ISSUE A CALL TO THAT ADDRESS. IF THE OPERATOR DOES NOT WISH THIS RETURN TO OCCUR, THEN HE MUST EITHER RUN WITH SWR BIT 7 SET OR ENSURE THAT LOCATIONS 42/43 ARE CLEAR.

IF THE FIELD VIDEO SIGNAL HAS TOO MUCH NOISE ON IT FOR A STATISIZED READING, ONE OF TWO ERROR REPORTS MAY BE GENERATED. ERROR 70: AN INITIAL SAMPLY COULD NOT BE TAKEN. ERROR 67: THE SIGNAL DOES NOT CHANGE STATE.

11. PROGRAM DESCRIPTION

TEST 0

THIS TEST CHECKS THAT ALL THE READ/WRITE BITS IN THE CSR CAN BE SET AND CLEARED, AND THAT THE DATA BUFFER IS UNAFFECTED. THE PROGRAM LOADS A SLIDING ONES PATTERN INTO THE CSR, THEN READS IT BACK TO CHECK THAT THE PATTERN WAS LOADED. THE DBUFF IS ALSO CHECKED FOR ZERO. HAVING CHECKED A SLIDING ONES PATTERN, THE PROCESS IS THEN REPEATED FOR SLIDING ZEROS.

ERROR 1 BIT NOT SET IN CSR. GOOD = EXPECTED, BAD = RECEIVED, ADDRESS = CSR ADDRESS, DATA = PATTERN NUMBER. TRAP+10 TO LOOP.

ERROR 2 DBUFF AFFECTED BY SETTING CSR. STATUS = RECEIVED DATA, ADDRESS = DBUFF ADDRESS, DATA = PATTERN NUMBER. TRAP+20 TO LOOP.

```
                        ERROR 3     BIT NOT CLEAR IN CSR.  GOOD = EXPECTED, BAD
                                    = RECEIVED, ADDRESS = CSR ADDRESS, DATA =
                                    PATTERN NUMBER.  TRAP+30 TO LOOP.

                        ERROR 4     DBUFF AFFECTED BY CLEARING CSR.  STATUS =
                                    RECEIVED DATA, ADDRESS = DBUFF ADDRESS, DATA
                                    = PATTERN NUMBER.  TRAP+40 TO LOOP.


                        TEST 1

                        THIS TEST CHECKS THAT THE FIELD VIDEO SIGNAL TOGGLES,
                        AND THAT THE PERIOD IS ROUGHLY 1/50TH OR 1/60TH OF A
                        SECOND, DEPENDING UPON WHAT THE OPERATOR HAS SELECTED.
                        THE PROGRAM FIRST CHECKS THAT THE FIELD VIDEO SIGNAL CAN
                        GO THROUGH A HI-LOW-HI-LOW, AND A LOW-HI-LOW-HI TRANSI-
                        TION.  AS MENTIONED PREVIOUSLY, IF THE SIGNAL IS TOO
                        NOISY TO GET A READING, ERROR 67 MAY BE
                        GENERATED. IF THE SIGNAL DOES NOT WORK, ERROR 70
                        IS FLAGGED.   IF THE TEST DOES NOT COMPLETE IN A REASONA-
                        BLE AMOUNT OF TIME, IT CAN BE ASSUMED THAT THE SIGNAL
                        HAS NOT TOGGLED.  TO GET OUT OF THIS, THE OPERATOR MUST
                        TYPE SPACE TO TERMINATE THIS SECTION.

                        HAVING VERIFIED THAT THE SIGNAL CAN TOGGLE, THE PROGRAM
                        THEN USES THIS INFORMATION TO CREATE A WAIT LOOP OF ONE
                        SECOND.  THIS WAIT LOOP IS THEN USED TO TIME FIVE BY ONE
                        SECOND INTERVALS.  AT THE END OF EACH INTERVAL, THE
                        CONSOLE TELETYPE BELL IS RUNG.  THIS INFORMATION CAN BE
                        USED TO JUDGE WHETHER THE FRAME TIME IS REASONABLE.
                        SINCE THERE ARE MANY INSTRUCTIONS EXECUTED BETWEEN
                        SAMPLING POINTS, THIS TIMING INFORMATION IS NOT ACCURATE
                        AND MUST THEREFORE BE JUDGED ACCORDINGLY.  SETTING
                        TRAP+20 WILL LOOP ON RINGING THE BELL.

                        ERROR 1     NOISY SIGNAL OR ABORTED TEST.  TRAP+10 TO
                                    LOOP.


                        TEST 2

                        THIS TEST CHECKS THAT ALL THE COLOURS CAN BE DISPLAYED.
                        BY SELECTING CHARACTER CODE 0, AND THE SAME FOREGROUND
                        AND BACKGROUND COLOURS, A FULL SCREEN OF THAT COLOUR IS
                        DISPLAYED.  THE SEQUENCE OF OPERATIONS IS AS FOLLOWS:-

                                    F/B WHITE           NO VIDEO
                                    F/B BLACK           VIDEO ON
                                    F/B RED             VIDEO ON
                                    F/B GREEN           VIDEO ON
                                    F/B YELLOW          VIDEO ON
                                    F/B BLUE            VIDEO ON
                                    F/B MAGENTA         VIDEO ON
                                    F/B CYAN            VIDEO ON
                                    F/B WHITE           VIDEO ON
```

SETTING TRAP+10 WILL LOOP ON THE CURRENT COLOUR
SELECTION.

TEST 3

THIS TEST CHECKS OUT THE X AND Y ADDRESSING OF THE DEVICE.
THE SCREEN IS SET TO BLUE AND TWO DIAGONALS ARE
DISPLAYED. ONE IS MADE OF MAGENTA BLOCKS, STARTING AT
THE TOP RIGHT CORNER, AND THE OTHER IS IN WHITE,
STARTING AT THE BOTTOM LEFT.

TEST 4

THIS TEST CHECKS OUT THE FOREGROUND/BACKGROUND COLOUR
SELECTION WITH BLINK AND FLASH. CHARACTER CODE ZERO IS
SELECTED, SINCE IT IS HALF FOREGROUND AND HALF
BACKGROUND, DIVIDED VERTICALLY. THE WHOLE DISPLAY IS
LOADED WITH THE COLOUR, BLINK AND FLASH INFORMATION IN
THE FOLLOWING ORDER:-

BLACK ON WHITE
BLINK WHITE ON BLACK
FLASH WHITE ON BLACK
FLASH AND BLINK

NOTE: THE FLASH WILL NORMALLY APPEAR TO 'HOP' SIDEWAYS.

SETTING TRAP+10 WILL LOOP ON THE CURRENT COLOUR/BLINK/-
FLASH SELECTION.

TEST 5

THIS TEST CHECKS OUT THE CHARACTER STORE ADDRESSING OF
THE DEVICE. THE DISPLAY IS SET TO BLACK, THEN THE
ENTIRE CHARACTER SET IS DISPLAYED WITH RED ON BLUE.

ROM BASED CHARACTER SET

FOR THE ROM BASED CHARACTER SET, THE CHARACTER SET WILL DISPLAY AS RED ON BLUE IN AN AREA FROM Y = 18 THRU 21 TO X = 7 THRU 40. THE CHARACTER SET IS DISPLAYED AS CHARACTER STORE ADDRESSES INCREMENTING HORIZONTALLY. SETTING TRAP+10 WILL LOOP ON DISPLAYING THE CHARACTER SET.

RAM BASED CHARACTER SET

FOR THE RAM BASED CHARACTER SET, THE DISPLAY AREA STARTS AT X = 7, Y = 18. THE CHARACTER CODES ARE DISPLAYED VERTICALLY IN COLUMNS FIVE CHARACTERS LONG, I.E. COLUMN 1 RUNS FROM CODE 0 TO CODE 4, COLUMN 2 FROM 5 TO 9, ETC.

THE CHARACTER SET IS FIRST SET TO ZERO, THEN A VERTICAL LINE IS MOVED THRU THE ENTIRE CHARACTER SET, FIVE CHARACTERS AT A TIME. THIS WILL APPEAR AS A VERTICAL LINE MOVING ACROSS THE DISPLAY AREA.

SETTING TRAP+20 WILL LOOP ON THE CURRENT CHARACTER CODE. IF TRAP+70 IS SET, THE OPERATOR CAN SELECT WHICH CHARACTER CODE HE WANTS TO TEST. HAVING SELECTED THE CODE, THE PROGRAM THEN ALLOWS THE OPERATOR TO CHANGE SWITCH REGISTER SETTINGS, SUCH AS SETTING TRAP+20. ONCE TRAP+70 HAS BEEN SET, IT IS THEN CLEARED, THE PROGRAM WILL CONTINUE TESTING FROM THE CHARACTER CODE TO OPERATOR SELECTED.

TEST 6
THIS TEST CHECKS THE DATA INTEGRITY OF THE CHARACTER STORE ADDRESSES, HELD IN THE PICTURE STORE. THE CHARACTER STORE ADDRESS IS HELD IN TWO BY 4-BIT RAMS PER PICTURE LOCATION, WITH THE HIGH BIT OF THE TOP RAM UNUSED. A FIVE CHARACTER STORE ADDRESS PATTERN IS USED, SUCH THAT THE SAME BIT IN EACH RAM IS SET. THE ADDRESSES USED ARE 21, 42, 104, 10 AND 40. ALTHOUGH CHARACTER ADDRESS 40 DOES NOT SET BITS IN EACH RAM, IT IS USED TO MAKE THE SEQUENCE ODD SO THAT IF THERE IS A POWER OF TWO FAULT IN THE MEMORY, IT WILL GET PICKED UP. EACH NEW ROW STARTS WITH THE NEXT CHARACTER IN THE SEQUENCE, SO THAT THE CODES APPEAR TO BE DRAWN IN DIAGONAL LINES RUNNING TOP RIGHT TO BOTTOM LEFT. EACH TIME THE COMPLETE SCREEN IS UPDATED, THE CHARACTER CODES ARE MOVED ONE PLACE LEFT, GIVING THE EFFECT TO THE DIAGONAL LINES MOVING RIGHT TO LEFT. THE COLOURS USED ARE WHITE ON BLACK. SETTING TRAP+4 WILL LOOP ON THE TEST.

THE PATTERN IS.

DIAMOND
DOUBLE QUOTES
CAPITAL 'D'
HEAVY PLUS (+) SIGN
(SPACE)

```
TEST 7

THIS IS AN EXERCISER TEST THAT DISPLAYS ALL CHARACTERS
IN EVERY PICTURE LOCATION.  CHARACTER CODES ARE
DISPLAYED IN DIAGONAL ROWS, TOP RIGHT TO BOTTOM LEFT.
THE DIAGONAL LINE WILL BE MOVED FROM BOTTOM RIGHT TO TOP
LEFT.  THE COLOURS ARE DISPLAYED IN VERTICAL BANDS, THE
FOREGROUND COLOURS BEING RED, GREEN, THEN BLUE, AND THE
BACKGROUND COLOURS BEING RED, GREEN, YELLOW, BLUE,
MAGENTA, CYAN AND WHITE.

AS THE TEST PROGRESSES, IT WILL APPEAR THAT THE CHARAC-
TER CODES ARE MOVING UP VERTICAL LINES, BUT OFFSET BY
ONE PICTURE ROW, THUS GIVING THE EFFECT OF A MOVING
DIAGONAL LINE.


TEST 10

THIS OPTIONALLY SELECTABLE TEST IS USED TO PRINT OUT THE
ACCUMULATED ERRORS TO DATE.  IT CAN BE USED AFTER
RUNNING SOAK TESTING TO GIVE AN INDICATION OF THE NUMBER
OF ERRORS THAT HAVE OCCURED.

THIS TEST PRINTS A MESSAGE 'ERROR COUNT = '', FOLLOWED BY
THE CURRENT ERROR COUNT.  THE PROGRAM THEN CLEARS THE
ERROR COUNT.


TEST 11

THIS OPTIONALLY SELECTED TEST IS USED TO CHECK THE RESPONSE
OF THE VIDEO ADAPTERS. THERE ARE 7 BLOCKS OF 4 ROWS EACH.
EACH BLOCK IS A DIFFERENT COLOUR, IN THE SEQUENCE..
        RED
        GREEN
        YELLOW
        BLUE
        MAGENTA
        CYAN
        WHITE
....EACH ROW HAS .5,1,2 AND 4 MHZ VERTICAL BARS TO TEST
THE RESPONSE.
```

APPENDIX A

PDP-11 DIAGNOSTIC LOOPING FACILITIES VIA SWITCH REGISTER OPTIONS

N.B.

THE INTENTION OF THIS APPENDIX IS TO EXPLAIN, IN GENERAL, THE SCOPE FACILITIES WITHIN PROGRAM CODING. THE USER MUST FIRST EXAMINE THE CODING ABOUT THE AREA HE WISHES TO USE SCOPING FACILITIES, TO ASCERTAIN THAT THE PARTICULAR FACILITY HE REQUIRES IS, IN FACT, AVAILABLE.

PROGRAM LOOPING CONTROL CAN BE SELECTED BY USING SWR 12 - 07. THE PROGRAM HANDLES THIS BY USE OF THE TRAPSV ROUTINE, WHICH IS ENTERED USING THE TRAP INSTRUCTION. BASICALLY, THE ROUTINE CHECKS EQUALITY BETWEEN BITS 05 - 00 OF THE TRAP INSTRUCTION AND SWR 12 - 07.

THERE ARE THREE DISTINCT FUNCTIONS CONTROLLED BY THE TRAP ROUTINE:-

A) RUN - (TRAP + 2 INSTRUCTION AND SWR 07 SET)
USUALLY USED TO INHIBIT TEST NUMBER PRINTOUT; USEFUL IN THE CASE OF NON-INTERVENTION TESTS. WHEN SWR 07 IS SET, ALL TEST NUMBER MESSAGES ARE SUPPRESSED. NOTE: IT DOES NOT SUPPRESS ERROR PRINTOUTS.

B) LOOP ON A SUB-TEST - (TRAP + 4 INSTRUCTION AND SWR 08 SET)
THIS IS GENERALLY USED TO ALLOW THE OPERATOR, BY SETTING SWR 08, TO CONTINUOUSLY LOOP ON ONE LOGICAL TEST OR GROUP OF TESTS.

C) SCOPE ON A SUB-TEST - (TRAP + 10 - 70 INSTRUCTIONS AND SWR 09 - 11)
THIS ALLOWS THE OPERATOR TO SELECT SEVEN LEVELS OF LOOPING FACILITY WITHIN A SELECTION OR TEST.

IT IS USED TYPICALLY WITHIN A TEST WHERE ONE SUB-TEST SETS A FLAG AND THE NEXT ONE CLEARS IT. BY USING SCOPE LEVEL 1 (SWR 09) - TRAP + 10), HE COULD LOOP ON THE FIRST SUB-TEST, SCOPE LEVEL 2 (SWR10) - TRAP + 20), HE COULD LOOP ON THE SECOND SUB-TEST OR SCOPE LEVEL 3 (SWR 10 & 09 - TRAP + 30) TO LOOP CONSTANTLY THROUGH BOTH SUB-TESTS SEQUENTIALLY. THE SCOPE RETURN ADDRESS ALWAYS APPEARS AS THE ARGUMENT OR NEXT WORD AFTER THE TRAP INSTRUCTION.

TO ALLOW THE SCOPE LEVEL TO BE CHANGED WITHOUT STOP-
PING THE PROGRAM, E.G. TO CHANGE FROM LEVEL 1 TO 2,
WHICH WOULD ALMOST CERTAINLY CAUSE LEVEL 3 OR 0 TO
BE SEEN MOMENTARILY), A 'PRESERVE SCOPE' FACILITY IS
PROVIDED WITH SWR12. WHEN THIS IS SELECTED, THE
PROGRAM NO LONGER INSPECTS SWR 11 - 09 BUT USES THE
SETTING MEMORISED FROM BEFORE SWR 12 WAS SELECTED.
THE SCOPE LEVEL MAY NOW BE CHANGED WITH NO EFFECT
UNTIL SWR 12 IS SET TO 0, WHEN THE NEW SCOPE SETTING
APPLIES.

N.B. SETTING SWR 12 SHOULD ONLY BE USED TO PRESERVE
AS EXISTING SCOPE LEVEL, AS PREVIOUSLY SET ON THE
SWITCH REGISTER.

WHICH SCOPE LEVEL TO SELECT MAY BE DETERMINED THE
LISTING; LEVELS 1 THROUGH 7 ARE CALLED BY TRAP + 10
(SWR 09) THROUGH TRAP + 70 (SWR 09, 10 AND 11).

EXAMPLE OF SCOPE LOOP FACILITIES WITHIN DIAGNOSTIC

```
;
TEST10: MOV     #40,COUNT       ;ITERATION COUNT.
T101:   BIS     #1,STATUS       ;SET GO BIT.
        BIT     #1,STATUS       ;IS GO BIT SET?
        BNE     T10SCP          ;YES, BRANCH.
        JSR     PC,ERROR        ;ERROR!!! GO NOT SET.
        10                      ;ERROR NUMBER.
T10SCP: TRAP+10                 ;SCOPE TEST 10 IF LEVEL 1 SELECTED.
        T101                    ;RETURN LABEL FOR SCOPE.
TEST11: BIC     #1,STATUS       ;CLEAR GO BIT
        BIT     #1,STATUS       ;GO BIT CLEAR  ?
        BEQ     T11SCP          ;GO BIT CLEAR BRANCH.
        JSR     PC,ERROR        ;NO. GO BIT FAILED TO CLEAR.
        11                      ;ERROR!!! NO. 11.
T11SCP: TRAP+30                 ;SCOPE TEST 11 IF LEVEL 3 SELECTED.
        TEST11                  ;RETURN LABEL FOR SCOPE.
        TRAP+20                 ;SCOPE TESTS 10 & 11.
        T101                    ;RTURN LABEL.
TEST12: BIS     #100,STATUS     ;NO SCOPE SELECTED. CARRY ON.
        BIT     #100,STATUS     ;IS DONE BIT SET ?
        (ETC.)
        .
        .
        .
T12SCR: TRAP+10                 ;SCOPE INT. EN. SET, LEVEL 1.
        TEST 12                 ;RETURN LABEL.
TEST13: BIC     #100,STATUS     ;CLEAR INT ENABLE BIT.
        BIT     #100,STATUS     ;IS BIT CLEAR?
        (ETC.)
        .
        .
        .
T13SCP: TRAP+30                 ;SCOPE INT CLEAR LEVEL 3.
        TEST13                  ;RETURN LABEL.
        TRAP+20                 ;SCOPE INT. SET AND CLEAR.
        TEST12                  ;LEVEL 2.
        TRAP+4                  ;SCOPE LOOP ON THIS SET OF TESTS.
        T101                    ;RETURN LABEL.
        DEC     COUNT           ;DO 40 TIMES ANYWAY
        BGT     T101
TEST14: (ETC.)
        .
        .
        .
;
```

```
673                                     .ENDR
```

```
                                        .TITLE  VTV30K DIAGNOSTIC
                                        .SBTTL  GENERAL DEFINITIONS


        000000                          R0=%00
        000001                          R1=%01
        000002                          R2=%02
        000003                          R3=%03
        000004                          R4=%04
        000005                          R5=%05
        000006                          SP=%06
        000007                          PC=%07
        177776                          PSW=177776
        177570                          HSWR=177570


        100000                          G=100000
        040000                          D=40000
        020000                          A=20000
        010000                          S=10000


        177564                          TPS=177564
        177566                          TPB=177566
        177560                          TKS=177560
        177562                          TKB=177562


                                        .ENABL  ABS
                                        .ENABL  AMA
```

```
                        .SBTTL  MACROS
                ;
                ;
                ; SET PROCESSOR PRIORITY
                ;
                ;
                        .MACRO  PSWSET  $ARG,?LAB
                        MOV     $ARG,-(SP)      ; SET UP NEW PSW AS $ARG
                        MOV     #LAB,-(SP)      ; SET RETURN ADDRESS
                        RTI                     ; RTI TO SET PRIORITY
                LAB:    NOP                     ; RETURN ADDRESS
                        .ENDM
                ;
                ;
                ; READ PROCESSOR PRIORITY
                ;
                ;
                        .MACRO  PSWREA  $ARG
                        EMT                     ; ISSUE EMT TO READ PSW
                        MOV     FSAVPW,$ARG     ; READ PSW IN $ARG
                        .ENDM
                ;
                ;
```

```
                                                .SBTTL  INITIALISATION
        000000                                  .ASECT
        000000                                  .=0
        000042                                  .=42
000042  000000                                  0                       ; XXDP HOOK INITIALIZATION.
        000200                                  .=200
000200  000137  001000                          JMP     @#START         ; JMP TO START AT 200
        001000                                  .=1000


001000  012706  001000          START:          MOV     #.,SP           ;INITIALISE STACK POINTER
001004                                          PSWSET  #340

001020  012737  012704  000004                  MOV     #SWRSET,4       ;TEST FOR HARDWARE SWR
001026  012737  000340  000006                  MOV     #340,6          ;TRAPS TO 4 IF IT
001034  012737  177570  007502                  MOV     #HSWR,SWR       ;DOES NOT EXIST
001042  005777  006434                          TST     @SWR
001046  005037  007504                          CLR     SSWR            ;INITIALISE SOFTWARE SWR
001052  004737  012640                          JSR     PC,SILLSI       ; CHECK WHICH PROCESSOR WE ARE ON
001056  004737  012714                          JSR     PC,VECTOR       ;FILL 0-774 WITH HALT TRAPS
001062  013737  007306  007436                  MOV     CSR0,DEVCSR     ; ASSUME 1 DEVICE
001070  013737  007310  007440                  MOV     DBUF0,DEVBUF
001076  005037  007302                          CLR     DEVFLG
001102  004737  014734                          JSR     PC,TYPOUT
001106  006005                                  GOMSG                   ;DIAGNOSTIC
001110  004737  012162                          JSR     PC,SETDEV       ; SET UP THE DEVICE
001114  004737  015776                          JSR     PC,SELBUS
001120  000137  001200                          JMP     RSTART          ;THEN GOTO THE RESTART ADDRESS
        001174                                  .=1174
001174  000000                  REACT:          0                       ; EXTERNAL TEST-EQUIPMENT HOOK
001176  000425                  XXDPGO:         BR      XXDP2

001200  012706  001000          RSTART:         MOV     #START,SP       ;INITIALISE STACK POINTER
001204                                          PSWSET  #340
001220  005037  007302                          CLR     DEVFLG
001224  013737  007306  007436                  MOV     CSR0,DEVCSR
001232  013737  007310  007440                  MOV     DBUF0,DEVBUF

001240  004737  014734                          JSR     PC,TYPOUT
001244  006317                                  WMSG                    ;SELECT DESIRED CONSOLE SWITCHES
001246  004737  013546                          JSR     PC,MONIT        ;GO TO SWR MONITOR
                                        ;
001252  012706  001000          XXDP2:          MOV     #START,SP       ; RESTORE THE STACK
001256                                          PSWSET  #340
001272  005037  007444                          CLR     ERRFLG          ; CLEAR ERROR FLAG
001276  005037  007574                          CLR     ERRDIS
001302  005037  007442                          CLR     ERRDI1
001306  013737  007306  007436  START1:         MOV     CSR0,DEVCSR     ; START WITH THE FIRST CSR
001314  013737  007310  007440                  MOV     DBUF0,DEVBUF    ;
001322  005037  007302                          CLR     DEVFLG

001326  012706  001000          START2:         MOV     #START,SP       ;INITIALISE STACK POINTER
001332                                          PSWSET  #340

001346  032777  000100  006126                  BIT     #100,@SWR       ;CHECK FOR PRE-SELECTED TEST
001354  001002                                  BNE     START3
```

```
794
795  001356  000137  001450                JMP     TEST0           ;TO TEST 0
796  001362  000137  001366        START3: JMP     TABLE           ;TO LOOK-UP TABLE
```

```
001366  017700  006110          TABLE:  MOV     @SWR,R0         ;GET SELECTED TEST NO
001372  042700  177700                  BIC     #177700,R0
001376  022700  000011                  CMP     #11,R0          ;CHECK FOR VALID TEST NO
001402  002676                          BLT     RSTART

001404  006300                          ASL     R0
001406  000170  001412                  JMP     @TABLE1(R0)     ;JUMP TO PRE-SELECTED TEST



001412  001450                  TABLE1: TEST0
001414  002040                          TEST1
001416  002174                          TEST2
001420  002354                          TEST3
001422  002616                          TEST4
001424  003110                          TEST5
001426  004234                          TEST6
001430  004660                          TEST7
001432  005264                          TEST10
001434  005362                          TEST11
                                ; THE FOLLOWING TABLE HAS ONE ENTRY FOR EACH OF THE ABOVE
                                ; IF ZERO, IT DISABLES THE STALL FACILITY IN 'WAITS' FOR
                                ; THAT TEST.
                                ;
001436     000     000     001  TABLE2: .BYTE   0,0,1,1,1,0,1,0,0,0
001441     001     001     000
001444     001     000     000
001447     000
                                        .EVEN
```

```
                                        .SBTTL  TEST0
                                ;
                                ; THIS TEST CHECKS THE READ/WRITE BITS OF THE CONTROL REGISTER
                                ; BY WRITING A SLIDING ONES PATTERN TO THE REGISTER, TO CHECK
                                ; THAT BITS CAN SET. THEN, BY WRITING A SLIDING ZEROES PATTERN
                                ; CHECK THAT THE BITS CAN CLEAR. AT EACH STAGE OF THE OPERATION
                                ; THE DATA BUFFER IS CHECKED TO SEE THAT THE SETTING/CLEARING
                                ; HAD NO EFFECT.
                                ;
001450  005037  007454          TEST0:  CLR     TESTNO          ; SET TEST NUMBER
001454  104402                          TRAP+2
001456  001472                          T0001
001460  004737  011624                  JSR     PC,TESTR        ; PRINT TEST NUMBER
001464  004737  014734                  JSR     PC,TYPOUT
001470  006360                          TSOMSG
001472  012704  000200          T0001:  MOV     #200,R4         ; SUGGEST AN ITERATION COUNT
001476  004737  013306                  JSR     PC,FASTSW       ; IS FAST ITERATION SET
001502  010437  007506                  MOV     R4,REPCNT       ; SET ACTUAL ITERATION COUNT
001506  012737  000020  007566  T0002:  MOV     #16.,DATA       ; GET LOOP COUNT
001514  012703  000004          T0003:  MOV     #4,R3           ; SELECT SLIDING ONES
001520  004737  016370                  JSR     PC,GENER        ; GET THE PATTERN
001524  042737  034000  007562          BIC     #34000,GOOD     ; MASK READ ONLY BITS
001532  013737  007436  007572  T0004:  MOV     DEVCSR,ADDRES   ; GET CURRENT ADDRESS
001540  013777  007562  006024          MOV     GOOD,@ADDRES    ; SET DATA IN CSR
001546  017737  006020  007564          MOV     @ADDRES,BAD     ; READ REPLY
001554  042737  034000  007564          BIC     #34000,BAD      ; MASK UNWANTED
001562  023737  007562  007564          CMP     GOOD,BAD        ; ARE THEY THE SAME
001570  001403                          BEQ     T0005           ; YES
001572  004737  016666                  JSR     PC,ERROR        ; NO REPORT AN ERROR
001576  160001                          1+D+G+A
001600  104410                  T0005:  TRAP+10                 ; LOOP ON THIS DATA
001602  001532                          T0004
001604  013737  007440  007572          MOV     DEVBUF,ADDRES   ; SET DATA BUFFER ADDRESS
001612  017737  005754  007570          MOV     @ADDRES,STATUS  ; THE DATA BUFFER SHOULD BE CLEAR
001620  005737  007570                  TST     STATUS          ; IS IT
001624  001403                          BEQ     T0007           ; YES
                                ; ERROR-
                                ; CHECK 'LEVA LOAD CAR H'
                                ;       'LEVA READ CAR L'
                                ;       +3VA ON E46-1 IF BAD=0
                                ;       RELEVANT BIT ON LEVB B8 (E25,E46)
001626  004737  016666                  JSR     PC,ERROR
001632  070002                          2+A+S+D
001634  104420                  T0007:  TRAP+20
001636  001532                          T0004                   ; LOOP ON COMPLETE
001640  005337  007566                  DEC     DATA            ; ALL PATTERNS DONE
001644  001323                          BNE     T0003           ; NO
                                ;
                                ; NOW DO SLIDING ZEROES
                                ;
001646  012737  000020  007566          MOV     #16.,DATA       ; GET LOOP COUNT
001654  012703  000005          T0009:  MOV     #5,R3           ; SELECT SLIDING ZEROES
001660  004737  016370                  JSR     PC,GENER        ; GET THE PATTERN
001664  042737  034000  007562          BIC     #34000,GOOD     ; MASK READ ONLY BITS
001672  013737  007436  007572  T0010:  MOV     DEVCSR,ADDRES   ; GET CURRENT ADDRESS
001700  013777  007562  005664          MOV     GOOD,@ADDRES    ; SET DATA IN CSR
```

```
001706  017737  005660  007564          MOV     @ADDRES,BAD       ; READ REPLY
001714  042737  034000  007564          BIC     #34000,BAD        ; MASK UNWANTED
001722  023737  007562  007564          CMP     GOOD,BAD          ; ARE THEY THE SAME
001730  001403                          BEQ     T0011             ; YES
                                ; ERROR-
                                ; REPLACE LEVB B8 E25 OR E46?
001732  004737  016666                  JSR     PC,ERROR          ; NO REPORT AN ERROR
001736  160003                          3+D+G+A
001740  104430                  T0011:  TRAP+30                   ; LOOP ON THIS DATA
001742  001672                          T0010
001744  013737  007440  007572          MOV     DEVBUF,ADDRES     ; SET DBUF ADDRESS
001752  017737  005614  007570          MOV     @ADDRES,STATUS    ; THE DATA BUFFER SHOULD BE CLEAR
001760  005737  007570                  TST     STATUS            ; IS IT
001764  001403                          BEQ     T0013             ; YES
                                ; ERROR-
                                ; REPLACE LEVB B8 E25 OR E46?
001766  004737  016666                  JSR     PC,ERROR
001772  070004                          4+A+S+D
001774  104440                  T0013:  TRAP+40
001776  001672                          T0010                     ; LOOP ON COMPLETE
002000  005337  007566                  DEC     DATA              ; ALL PATTERNS DONE
002004  001323                          BNE     T0009             ; NO
                                ;
                                ; END OF TEST ?
                                ;
002006  104404                          TRAP+4                    ; LOOP ON TEST ?
002010  001514                          T0003
002012  005337  007506                  DEC     REPCNT            ; ITERATE TILL DONE
002016  001633                          BEQ     T0002
002020  005077  005412                  CLR     @DEVCSR           ; TIDY UP AFTER TEST.
002024  032777  000100  005450          BIT     #100,@SWR         ; PRESELECT TEST ?
002032  001402                          BEQ     TEST1             ; NO
002034  000137  001326                  JMP     START2            ; YES GO DO IT
```

```
                                                .SBTTL  TEST1
                                        ;
                                        ; THIS TEST CHECKS THE FIELD VIDEO TIMING BY ATTEMPTING
                                        ; TO RING THE BELL EVERY SECOND FOR 5 SECONDS
                                        ; IF THE BELL DOES NOT RING, TYPE CNTRL-X TO ABORT THE TEST
                                        ;
                                        ;
002040  012737  000001  007454  TEST1:  MOV     #1,TESTNO       ; SET TEST NUMBER
002046  104402                          TRAP+2
002050  002064                          T0101
002052  004737  011624                  JSR     PC,TESTR        ; PRINT TEST NUMBER
002056  004737  014734                  JSR     PC,TYPOUT
002062  006365                          TS1MSG
002064                          T0101:
002064  012737  000100  007506          MOV     #100,REPCNT     ; SET ITERATION COUNT
                                        ;
                                        ; TRY TO SYNC UP
                                        ;
002072  004737  011644          T0102:  JSR     PC,SYNCUP       ; SYNC UP
002076  000401                          BR      T0103           ; NO SYNC OR ABORT
002100  000403                          BR      T0104           ; GO A SYNC SIGNAL
                                        ; ERROR-
                                        ; CHECK 'LEVB CLOCK H' (8 MHZ)
                                        ;       'LEVB CLOCK L'
                                        ;       'LEVB VA02 H' (1 MHZ)
                                        ;       'LEVA VA02 L' (1 MHZ)
                                        ;       'LEVB VA17 H' (16.7 MILLISECS FOR 525, 20 FOR 625)
002102  004737  016666          T0103:  JSR     PC,ERROR        ; ERROR
002106  000001                          1
002110  104410                  T0104:  TRAP+10                 ; LOOP
002112  002072                          T0102                   ; TO HERE
002114  005337  007506                  DEC     REPCNT          ; REDUCE ITERATION
002120  001364                          BNE     T0102           ; KEEP GOING
                                        ;
                                        ; NOW RING THAT BELL
                                        ;
002122  012737  000005  007566  T0105:  MOV     #5,DATA         ; FINE BELLS
002130  004737  012040          T0106:  JSR     PC,WAITS        ; WAIT ONE SECOND
002134  000001                          1
002136  004737  013166                  JSR     PC,BELL         ; DING DONG !!
002142  005337  007566                  DEC     DATA
002146  001370                          BNE     T0106           ; KEEP GOING
002150  104420                          TRAP+20
002152  002122                          T0105                   ; LOOP ?
                                        ;
                                        ;
002154  104404                          TRAP+4                  ; LOOP ON TEST
002156  002064                          T0101
002160  032777  000100  005314          BIT     #100,@SWR       ; PRESELECT TEST
002166  001402                          BEQ     TEST2
002170  000137  001326                  JMP     START2          ; YES- GO DO IT
```

```
                                        .SBTTL  TEST2
                                ;
                                ; THE TEST CHECKS THE PICTURE BY WRITING TO EVERY LOCATION,
                                ; USING THE SAME FOREGROUND AND BACKGROUND COLOUR.
                                ;
                                ; THE SEQUENCE OF OPERATIONS IS:
                                ;
                                ;       F/B WHITE NO VIDEO
                                ;       F/B BLACK VIDEO ON
                                ;       F/B RED VIDEO ON
                                ;       F/B GREEN VIDEO ON
                                ;       F/B YELLOW VIDEO ON
                                ;       F/B BLUE VIDEO ON
                                ;       F/B MAGENTA VIDEO ON
                                ;       F/B CYAN VIDEO ON
                                ;       F/B WHITE VIDEO ON
                                ;
                                ; CHARACTER STORE ERRORS ARE DELIBERATELY BYPASSED IN THIS TEST.
                                ; IF THE TEST WORKS, COLOUR INFORMATION CAN BE USED IN THE NEXT
                                ; TEST.
                                ; ERRORS-
                                ; CHECK LOW-ORDER BITS O PICTURE STORE (LEVC)
                                ;       OUTPUT LATCH ON LEVD, E23.
                                ;       OUTPUT MUX ON LEVD, E18.
                                ;       STATICISERS E6 & E12, WITH STROBES 'LEVB STROBE A H', '-B H'
                                ;
002174  012737  000002  007454  TEST2:  MOV     #2,TESTNO       ; SET TEST NUMBER
002202  104402                          TRAP+2
002204  002220                          T0201
002206  004737  011624                  JSR     PC,TESTR        ; PRINT TEST NUMBER
002212  004737  014734                  JSR     PC,TYPOUT
002216  006411                          TS2MSG
002220  012737  000007  007464  T0201:  MOV     #7,FORGND       ; SET F/B TO WHITE
002226  012737  000007  007466          MOV     #7,BAKGND
002234  005037  007470                  CLR     BLINK           ; NO BLINK
002240  005037  007472                  CLR     FLASH           ; FLASH,
002244  005037  007450                  CLR     VIDEON          ; OR VIDEO.
002250  005037  007550                  CLR     CHAR            ; USE CHARACTER ZERO
002254  004737  012410                  JSR     PC,LOADFS       ; LOAD THE DISPLAY
                                ;
                                ; NOW WAIT AROUND
                                ;
                                ;
                                ;
                                ; NOW LOAD THE REST OF THE PICTURES WITH VIDEO ON
                                ;
002260  005037  007464                  CLR     FORGND          ; START WITH BLACK
002264  005037  007466                  CLR     BAKGND          ; ON BLACK
002270  012737  040000  007450          MOV     #40000,VIDEON   ; BUT WITH VIDEO ON
002276  004737  012410          T0202:  JSR     PC,LOADFS       ; LOAD THE SCREEN
                                ;
                                ; NOW WAIT AROUND
                                ;
002302  004737  012040                  JSR     PC,WAITS                ; WAIT FOR 2 SECONDS
002306  000002                          2
002310  104410                          TRAP+10
```

```
002312  002276                          T0202                   ; LOOP TO HERE
                                ;
                                ; NOW UPDATE THE COLOURS
                                ;
002314  005237  007464                  INC     FORGND          ; NEW FORGROUND
002320  005237  007466                  INC     BAKGND          ; AND BACKGROUND
002324  023727  007464  000007          CMP     FORGND,#7       ; ALL DONE
002332  003761                          BLE     T0202           ; NO
                                ;
                                ; COMPLETE
                                ;
002334  104404                          TRAP+4                  ; LOOP ON TEST ?
002336  002220                          T0201
002340  032777  000100  005134          BIT     #100,@SWR       ; PRESELECT TEST
002346  001402                          BEQ     TEST3           ; NO
002350  000137  001326                  JMP     START2          ; YES GO TO IT
```

```
                                        .SBTTL  TEST3
                                ;
                                ; THIS TEST CHECKS THE CURSOR ADDRESSING FEATURES OF THE VTV30K.
                                ; THE PICTURE IS FIRST SET TO BLUE ON BLUE.
                                ; TWO DIAGONAL ROWS OF SQUARE CHARACTERS ARE PRODUCED. THE FIRST IN
                                ; MAGENTA RUNS FROM THE TOP RIGHT CORNER TO THE BOTTOM OF THE SCREEN.
                                ; THE SECOND IN WHITE RUNS FROM THE BOTTOM LEFT CORNER TO THE TOP.
                                ;
                                ; IF THIS TEST WORKS THE CAR AND THE PICTURE ADDRESSING IS OK.
                                ; ERRORS-
                                ; BROKEN, MISSING OR DUPLICATED DIAGONALS ARE CAUSED BY FAULTS
                                ; ASSOCIATED WITH THE PICTURE ELEMENT SELECTOR ON LEVC.
                                ;
002354  012737  000003  007454  TEST3:  MOV     #3,TESTNO       ; SET TEST NUMBER
002362  104402                          TRAP+2
002364  002400                          T0301
002366  004737  011624                  JSR     PC,TESTR        ; PRINT TEST NUMBER
002372  004737  014734                  JSR     PC,TYPOUT
002376  006416                          TS3MSG
002400  005037  007470          T0301:  CLR     BLINK           ; NO BLINK
002404  005037  007472                  CLR     FLASH           ; NO FLASH
002410  012737  040000  007450          MOV     #40000,VIDEON   ; ENABLE VIDEO
002416  005037  007550                  CLR     CHAR            ; SELECT CHARACTER ZERO
002422  012737  000004  007464          MOV     #4,FORGND       ; SELECT BLUE
002430  012737  000004  007466          MOV     #4,BAKGND       ; ON BLUE
002436  004737  012410                  JSR     PC,LOADFS       ; LOAD THE FULL SCREEN
                                ;
                                ; NOW DO X ADDRESS TESTING
                                ;
002442  005037  007476          T0303:  CLR     CARY            ; START AT TOP OF SCREEN..
002446  012737  000057  007474          MOV     #47.,CARX       ; AT RIGHT.
002454  012737  000005  007464          MOV     #5,FORGND       ; SET CHARACTERS TO CYAN
002462  013737  007464  007466          MOV     FORGND,BAKGND   ; ...ALL OVER!
                                ;
002470  004737  012522          T0302:  JSR     PC,LOELEM       ; LOAD ONE ELEMENT
002474  005237  007476                  INC     CARY            ; NEXT LINE DOWN...
002500  005337  007474                  DEC     CARX            ; SHIFT LEFT ONCE.
002504  023727  007476  000040          CMP     CARY,#32.       ; OVER TOP?
002512  001366                          BNE     T0302           ; NO, NEXT BLOCK
002514  104410                          TRAP+10                 ; LOOP ON THIS BIT
002516  002442                          T0303
002520  005037  007474          T0304:  CLR     CARX            ; NOW START AT RIGHT
002524  013737  007500  007476          MOV     YMAX,CARY       ; AT BOTTOM
002532  012737  000007  007464          MOV     #7,FORGND       ; DISPLAY IN WHITE..
002540  013737  007464  007466          MOV     FORGND,BAKGND   ; ALL OVER.
                                ;
002546  004737  012522          T0305:  JSR     PC,LOELEM       ; LOAD ONE ELEMENT
002552  005237  007474                  INC     CARX            ; SHIFT RIGHT
002556  005337  007476                  DEC     CARY            ; SHIFT UP
002562  100371                          BPL     T0305           ; PAST TOP OF SCREEN?
002564  104420                          TRAP+20
002566  002520                          T0304                   ; LOOP ON THIS
                                ; NOW WAIT AROUND
                                ;
002570  004737  012040                  JSR     PC,WAITS        ; WAIT FOR 2 SECONDS
002574  000002                          2
```

```
002576  104404                          TRAP+4                  ; LOOP ON TEST
002600  002400                          T0301
002602  032777  000100  004672          BIT     #100,@SWR       ; PRESELECT TEST
002610  001402                          BEQ     TEST4
002612  000137  001326                  JMP     START2          ; YES- GO DO IT
                                ;
```

```
                                        .SBTTL  TEST4
                                ;
                                ; THIS TEST CHECKS OUT THE FOREGROUND BACKGROUND SELECTION, BY
                                ; DISPLAYING CHARACTER CODE ZERO IN ALL LOCATIONS. CHARACTER
                                ; ZERO IS DEFINED AS BEING HALF FOREGROUND AND HALF BACKGROUND,
                                ; IN A VERTICAL STRIPE (CHARACTER DATA 360). THE FOREGROUND
                                ; AND BACKGROUND COLOURS ARE THEN ALTERED AND THE WHOLE DISPLAY
                                ; IS REFRESHED. THE SEQUENCE BEING: BLACK ON WHITE,
                                ; BLINK SET, THEN BLINK CLEAR BUT FLASH SET. FINALLY BLINK AND
                                ; FLASH ARE SET TOGETHER, CAUSING A PICTURE THAT 'HOPS' SIDEWAYS
                                ; WITH A BACKGROUND FLASHING.
                                ;
                                ; IF THIS TEST WORKS THE OUTPUT MUX (LEVB) IS FUNCTIONING.
                                ; ERRORS-
                                ; CHECK 'LEVB F/B SELECT H'
                                ;       'LEVB FLASH B H' IF BLINK OR FLASH DOES NOT WORK.
                                ;       LEVA SIGNALS ON E4...
                                ;               'LEVB FLASH A'
                                ;               'LEVB FLASH B'
                                ;               'LEVD BLINK A'
                                ;               'LEVD BLINK B'
                                ;               'LEVD DATA'
                                ;
002616  012737  000004  007454  TEST4:  MOV     #4,TESTNO       ; SET TESTNUMBER
002624  104402                          TRAP+2
002626  002642                          T0401
002630  004737  011624                  JSR     PC,TESTR        ; PRINT TEST NUMBER
002634  004737  014734                  JSR     PC,TYPOUT
002640  006426                          TS4MSG
002642  005037  007550          T0401:  CLR     CHAR            ; START WITH CHARACTER CODE 0
002646  005737  007462                  TST     PROM            ; PROM CHARACTER SET ?
002652  001022                          BNE     T0403           ; YES
002654  005001                          CLR     R1              ; BUT THERE ARE 8 LINES TO IT
002656  013700  007550          T0402:  MOV     CHAR,R0         ; GET CHARACTER NUMBER
002662  006300                          ASL     R0
002664  006300                          ASL     R0              ; SET IT IN THE RIGHT PLACE
002666  006300                          ASL     R0
002670  050100                          BIS     R1,R0           ; ADD IN LINE NUMBER
002672  052700  100000                  BIS     #100000,R0      ; SAY ITS SELECT CHSR
002676  010077  004534                  MOV     R0,@DEVCSR      ; LOAD THE CSR
002702  012777  000360  004530          MOV     #360,@DEVBUF    ; LOAD THE DATA PATTERN
002710  005201                          INC     R1              ; GET NEXT LINE
002712  020127  000007                  CMP     R1,#7           ; ALL LINES DONE
002716  003757                          BLE     T0402           ; NO YET
                                ;
002720                          T0403:
002720  005037  007472                  CLR     FLASH           ; NO FLASH
002724  005037  007470                  CLR     BLINK           ; NO BLINK
002730  012737  000007  007464          MOV     #7,FORGND       ; START WITH BLACK FOREGROUND
002736  012737  000000  007466  T0406:  MOV     #0,BAKGND       ; AND IT COMPLEMENT BACKGROUND
002744  012737  040000  007450          MOV     #40000,VIDEON   ; SET VIDEO ON
002752  004737  012410                  JSR     PC,LOADFS       ; FILL THE SCREEN
                                ;
                                ; WAIT AROUND
                                ;
002756  004737  012040                  JSR     PC,WAITS                ; WAIT FOR 2 SECONDS
```

```
002762  000002                          2
002764  104410                          TRAP+10
002766  002736                          T0406                   ; LOOP ON THIS
                                ;
                                ; NOW UPDATE THE FOREGROUND COLOUR
                                ;
                                ; NOW CHANGE BLINK STATUS
                                ;
002770  012737  000100  007470          MOV     #100,BLINK
002776  004737  012410          T0407:  JSR     PC,LOADFS       ; LOAD DISPLAY
003002  004737  012040                  JSR     PC,WAITS        ; WAIT 2 SECS
003006  000002                          2
003010  104420                          TRAP+20
003012  002776                          T0407                   ; LOOP ON PICTURE
003014  005037  007470                  CLR     BLINK           ; RESET BLINK
003020  012737  000200  007472          MOV     #200,FLASH      ;  SET FLASH
003026  004737  012410          T0409:  JSR     PC,LOADFS       ; LOAD PICTURE
003032  004737  012040                  JSR     PC,WAITS        ; WAIT 2 SECS
003036  000002                          2
003040  104430                          TRAP+30
003042  003026                          T0409                   ; LOOP ON PICTURE
003044  012737  000100  007470          MOV     #100,BLINK      ; NOW SET BLINK AND FLASH
003052  004737  012410          T0410:  JSR     PC,LOADFS       ; FILL SCREEN
003056  004737  012040                  JSR     PC,WAITS        ; STALL FOR A WHILE!
003062  000002                          2                       ; 2 SECS
003064  104440                          TRAP+40
003066  003052                          T0410                   ; LOOP ON PICTURE
003070  104404                          TRAP+4                  ; LOOP ON TEST
003072  002642                          T0401                   ;
003074  032777  000100  004400          BIT     #100,@SWR       ; PRESELECT TEST
003102  001402                          BEQ     TEST5
003104  000137  001326                  JMP     START2          ; YES GO TO IT
```

```
                                        .SBTTL  TEST5
                                ;
                                ; THIS TEST CHECKS THE CHARACTER STORE FOR EITHER THE PROM
                                ; OR RAM VERSIONS OF THE VTV30K. FOR THE PROM VERSION THE
                                ; CHARACTER SET WILL BE DISPLAYED AS CYAN ON BLUE IN AN
                                ; AREA FROM Y=10 ,X=3 THRU Y=12, X=45. THE CHARACTER SET
                                ; IS DISPLAYED AS CHARACTER STORE ADDRESSES INCREMENTING
                                ; HORIZONTALLY.
                                ; FOR THE RAM CHARACTER SET, THE PICTURE IS LAYED OUT IN THE
                                ; SAME WAY. A VERTICAL LINE IS THEN MOVED THROUGH THE CHARACTER
                                ; STORE TO CHECK THE ACCESSABILITY OF THE MEMORY.
                                ; NEXT A LINE IS ROTATED DOWN THE SCREEN TO CHECK THE LOW-ORDER
                                ; BITS.
                                ;
                                ; THIS IS THE FIRST REAL TEST OF THE CHARACTER STORE. IF IT
                                ; WORKS ALL CHARACTER STORE DATA AND ADDRESSING WORKS.
                                ; ERRORS-
                                ; CHECK HIGH-ORDER PICTURE STORE BITS (LEVB)
                                ;       PICTURE STORE STATICISER (LEVD)
                                ;       CHARACTER STORE ICS (LEVD)
                                ;       C.S. WRITE DATA,C.S. WRITE ADDRESS
                                ; VERTICALLY CORRUPTED PROM CHARS, OR FAULTS IN HORIZONTAL
                                ;  LINES IN RAM VERSIONS, IMPLY FAULTS IN THE CHARACTER STORE
                                ;  LINE SELECTOR (LEVD)
                                ;
003110  012737  000005  007454  TEST5:  MOV     #5,TESTNO       ; SET TEST NUMBER
003116  104402                          TRAP+2
003120  003134                          T0501
003122  004737  011624                  JSR     PC,TESTR
003126  004737  014734                  JSR     PC,TYPOUT
003132  006433                          TS5MSG
003134  005737  007462          T0501:  TST     PROM            ; IS THIS A PROM CHARACTER
003140  001032                          BNE     T0502           ; SET
                                ;
                                ;
                                ; RAM CHARACTER SET PRESENT
                                ;
003142  005037  007550          TEST5A: CLR     CHAR            ; START WITH CHARACTER CODE ZERO
003146  005001                  5$:     CLR     R1              ; THERE ARE 8 LINES PER CHAR
003150  013700  007550          10$:    MOV     CHAR,R0         ; GET CHARACTER NUMBER
003154  006300                          ASL     R0              ; SET IN RIGHT PLACE
003156  006300                          ASL     R0
003160  006300                          ASL     R0
003162  050100                          BIS     R1,R0           ; ADD IN LINE NUMBER
003164  052700  100000                  BIS     #100000,R0      ; SELECT CHSR
003170  010077  004242                  MOV     R0,@DEVCSR      ; SELECT CHARACTER TO ADDRESS
003174  012777  000000  004236          MOV     #0,@DEVBUF      ; LOAD ZEROES IN
003202  005201                          INC     R1              ; UPDATE LINE NUMBER
003204  020127  000007                  CMP     R1,#7           ; ALL LINES DONE
003210  003757                          BLE     10$             ; NO
003212  005237  007550                  INC     CHAR            ; SELECT NEXT CHAR
003216  023727  007550  000177          CMP     CHAR,#177       ; ALL DONE
003224  003750                          BLE     5$              ; NO
                                ;
                                ; NOW FILL THE SCREEN
                                ;
```

```
003226  005037  007472          T0502:  CLR     FLASH           ; NO FLASH
003232  005037  007470                  CLR     BLINK           ; NO BLINK
003236  012737  000000  007464          MOV     #0,FORGND       ; BLACK ON BLACK
003244  012737  000000  007466          MOV     #0,BAKGND
003252  005037  007550                  CLR     CHAR            ; CHARACTER CODE 0
003256  012737  040000  007450          MOV     #40000,VIDEON   ; WITH VIDEO ON
003264  004737  012410                  JSR     PC,LOADFS
003270                          T0520:
003270  012737  000003  007474          MOV     #3,CARX         ; START AT X=3
003276  012737  000012  007476          MOV     #10.,CARY       ; Y=10.
003304  005037  007550                  CLR     CHAR            ; CHARACTER CODE 0
003310  012737  000006  007464          MOV     #6,FORGND       ; CYAN ON
003316  012737  000004  007466          MOV     #4,BAKGND       ; BLUE
003324  005037  007470                  CLR     BLINK
003330  005037  007472                  CLR     FLASH           ; NO FLASH OR BLINK
003334  012737  040000  007450          MOV     #40000,VIDEON   ; VIDEO ON
003342  004737  012522          T0518:  JSR     PC,LOELEM       ; LOAD DATA OUT
003346  005237  007474                  INC     CARX            ; GET NEW COORD
003352  023727  007474  000056          CMP     CARX,#46.       ; END OF DISPLAY
003360  001005                          BNE     T0519           ; NO
003362  012737  000003  007474          MOV     #3.,CARX        ; YES START NEW LINE
003370  005237  007476                  INC     CARY
003374  005237  007550          T0519:  INC     CHAR            ;SELECT NEXT CHARACTER
003400  023727  007550  000177          CMP     CHAR,#177       ; ALL DONE ?
003406  003755                          BLE     T0518           ; NO
                                ;
                                ; NOW INSERT LAST CHARACTER AS FILER...
                                ;
003410  005737  007462                  TST     PROM            ; DIFFERENT FOR RAM...
003414  001414                          BEQ     2$              ; CONTINUE FOR RAM..
003416  012737  000040  007550          MOV     #' ,CHAR        ; LOAD NEW CHARACTER
003424  004737  012522                  JSR     PC,LOELEM       ; IN LAST SPACE AS SPACE.
                                ; NOW WAIT AROUND
                                ;
003430  004737  012040                  JSR     PC,WAITS        ; WAIT A BIT
003434  000005                          5                       ; 5 SECONDS
003436  104410                          TRAP+10
003440  003226                          T0502
003442  000137  004214                  JMP     T05XIT
                                ;
003446  012737  000125  007550  2$:     MOV     #85.,CHAR       ; FILL IN FINAL CHARACTER
003454  004737  012522                  JSR     PC,LOELEM
003460  004737  012040                  JSR     PC,WAITS        ; SHOW PRIME PICTURE AS CUE.
003464  000001                          1
                                ;
003466  005005                  5$:     CLR     R5              ; POINTS TO LINE POSITION
                                ;
003470  105737  177560          10$:    TSTB    TKS             ; ANY ABORT?
003474  100012                          BPL     12$             ; NO..
003476  004737  014206                  JSR     PC,READ
003502  020027  000040                  CMP     R0,#'           ; SPCAE-HURRYUP?
003506  001525                          BEQ     30$             ; YES...
003510  020027  000007                  CMP     R0,#'G&37       ; BELL?
003514  001002                          BNE     12$             ; NO...
003516  004737  013546                  JSR     PC,MONIT
                                ;
```

```
003522  004737  011644          12$:    JSR     PC,SYNCUP          ; DELAY 2 PERIODS
003526  000240                          NOP
003530  004737  011644                  JSR     PC,SYNCUP
003534  000240                          NOP
003536  010500                          MOV     R5,R0              ; GET LINE POSITION..
003540  042700  177770                  BIC     #177770,R0         ; GET POS IN CHAR..
003544  052700  000010                  BIS     #10,R0
003550  012704  100000                  MOV     #100000,R4         ; PUT LINE IN INITIAL POS.
003554  000241                          CLC
003556  005737  007622                  TST     PAL                ; IF TV-ENCODED...
003562  001401                          BEQ     20$
003564  000261                          SEC                        ; DOUBLE-UP LINE
                                ;
003566  006004                  20$:    ROR     R4                 ; SHIFT LINE TO POSN.
003570  005300                          DEC     R0
003572  001375                          BNE     20$
003574  000304                          SWAB    R4                 ; SWITCH FOR CLEAN-UP TRAILER
003576  010500                          MOV     R5,R0              ; GET CURRENT CHAR..
003600  042700  000007                  BIC     #7,R0              ; WITHOUT LINE..
003604  162700  000010                  SUB     #10,R0             ; GET LAST CHAR...
003610  100003                          BPL     22$                ; VALID CHAR?
003612  005000                          CLR     R0                 ; NO, START AT FIRST..
003614  105004                          CLRB    R4                 ; CLEAR LINE
003616  000304                          SWAB    R4                 ; SWAP LINES OVER..
                                ;
003620  020027  000520          22$:    CMP     R0,#42.*10         ; OVER TOP?
003624  103402                          BCS     23$                ; NO..
003626  042704  177400                  BIC     #177400,R4         ; ELSE CLEAR HIGH ORDER...
                                ;
                                ; NOW WE HAVE 2 CHARACTERS TO LOAD. EACH CHANGE IN THE LINE
                                ; INVOLVES LOADING 2 ADJACENT CHARACTERS.... THE ONE THE
                                ; LINE IS/COULD BE SCROLLING OUT OF, AND THE ONE IT IS
                                ; SCROLLING INTO.
                                ;
003632  012702  000003          23$:    MOV     #3,R2              ; 3 ROWS OF CHARACTERS.
                                ;
003636  012703  000010          24$:    MOV     #10,R3             ; 8 BYTES/CHARACTER
003642  052700  140000                  BIS     #140000,R0         ; CHAR STORE SEL,VIDEO-ON
003646  010077  003564                  MOV     R0,@DEVCSR
003652  032777  002000  003556          BIT     #2000,@DEVCSR      ; INHIBIT WRAP-ROUND
003660  001034                          BNE     29$
                                ;
003662  010477  003552          26$:    MOV     R4,@DEVBUF         ; LOAD DATUM
003666  005277  003544                  INC     @DEVCSR
003672  005303                          DEC     R3
003674  001372                          BNE     26$                ; FIRST CHARACTER DONE IN PAIR?
003676  032777  002000  003532          BIT     #2000,@DEVCSR      ; WRAP-ROUND?
003704  001022                          BNE     29$                ; YES...
003706  000304                          SWAB    R4                 ; YES, GET NEXT DATUM
003710  012703  000010                  MOV     #10,R3
                                ;
003714  010477  003520          27$:    MOV     R4,@DEVBUF         ; LOAD DATUM
003720  005277  003512                  INC     @DEVCSR
003724  005303                          DEC     R3
003726  001372                          BNE     27$
003730  000304                          SWAB    R4                 ; RESTORE ORIGINAL DATUM..
```

```
003732  062700  000530                  ADD     #43.*10,R0          ; UPDATE TO NEXT LINE..
003736  032777  002000  003472          BIT     #2000,@DEVCSR       ; OVER TOP?
003744  001002                          BNE     29$                 ; YES...DONT WRAP AROUND!
003746  005302                          DEC     R2                  ; ANY MORE...
003750  001332                          BNE     24$                 ; YES..
                                ;
003752  005205                  29$:    INC     R5                  ;
003754  020527  000530                  CMP     R5,#43.*10          ; ALL DONE?
003760  001243                          BNE     10$                 ; YES...
                                ;
003762  104470                  30$:    TRAP+70                     ; LOOP?
003764  003466                          5$
003766  004737  012040                  JSR     PC,WAITS            ; WAIT A SECOND...
003772  000001                          1
                                ;...THEN PROCEED TO VERTICAL-LINE...
```

```
003774  005005                  35$:    CLR     R5              ; POSITION MARKER FOR LINE.
003776  012704  004000                  MOV     #4000,R4        ; ZERO CHSR
                                ;
004002  010500                  40$:    MOV     R5,R0
004004  006300                          ASL     R0              ; GET CHARACTER NUMBER
004006  006300                          ASL     R0
004010  105000                          CLRB    R0              ; ZERO COLOUR+BLINK
004012  052700  000047                  BIS     #47,R0          ; WHITE ON BLUE
004016  010577  003414                  MOV     R5,@DEVCSR      ; GET PICTURE ADDRESS
004022  010077  003412                  MOV     R0,@DEVBUF      ; LOAD ELEMENT
004026  005205                          INC     R5              ; NEXT PICTURE CELL!
004030  005304                          DEC     R4              ; ANY MORE PICTURE?
004032  001363                          BNE     40$             ; ..YES..
004034  012705  002000                  MOV     #2000,R5        ; NOW CLEAR CHARACTER STRE
004040  012777  100000  003370          MOV     #100000,@DEVCSR
                                ;
004046  005077  003366          45$:    CLR     @DEVBUF         ; ZERO BYTE IN CHSR
004052  005277  003360                  INC     @DEVCSR
004056  005305                          DEC     R5              ; ANY MORE?
004060  001372                          BNE     45$             ; YES..
                                ;
004062  105737  177560          50$:    TSTB    TKS             ; ANY ABORT?
004066  100012                          BPL     52$             ; NO
004070  004737  014206                  JSR     PC,READ
004074  020027  000040                  CMP     R0,#'           ; ABORT?
004100  001440                          BEQ     70$             ; ..YES...
004102  020027  000007                  CMP     R0,#'G&37       ; BELL?
004106  001002                          BNE     52$             ; NO..
004110  004737  013546                  JSR     PC,MONIT        ; CALL MONITOR.
                                ;
004114  004737  011644          52$:    JSR     PC,SYNCUP       ; WAIT..
004120  000240                          NOP
004122  004737  011644                  JSR     PC,SYNCUP
004126  000240                          NOP
004130  010500                          MOV     R5,R0           ; GETCHARACTER..
004132  005300                          DEC     R0              ; GET LAST FOR CLEARING
004134  100406                          BMI     53$
004136  052700  140000                  BIS     #140000,R0      ; VIDEO ON, CHSR SELECT.
004142  010077  003270                  MOV     R0,@DEVCSR      ; GET ADDRESS..
004146  005077  003266                  CLR     @DEVBUF         ; ZERO CHARACTER BYTE.
                                ;
004152  010500                  53$:    MOV     R5,R0           ; GET CURRENT CHAR.
004154  052700  140000                  BIS     #140000,R0
004160  010077  003252                  MOV     R0,@DEVCSR      ; LOAD ADDRESS
004164  012777  000377  003246          MOV     #377,@DEVBUF    ; LIGHT LINE.
004172  005205                          INC     R5              ; STEP TO NEXT LINE.
004174  020527  000400                  CMP     R5,#400         ; NEXT LINE?
004200  001330                          BNE     50$             ; YES, LOOP
                                ;
004202  104420                  70$:    TRAP+20
004204  003774                          35$                     ; LOOP ON SUBTEST
004206  004737  012040                  JSR     PC,WAITS
004212  000001                          1                       ; WAIT FOR 1 SEC
                                ;
                                ;
                                ; END OF TEST..
```

```
                                        ;
004214  104404                          T05XIT: TRAP+4                  ; LOOP ON TEST?
004216  003134                                  T0501
004220  032777  000100  003254                  BIT     #100,@SWR       ; PRESELECTED TEST?
004226  001402                                  BEQ     TEST6
004230  000137  001326                          JMP     START2
```

```
                                        .SBTTL  TEST6
                                ;
                                ; THIS TEST DISPLAYS A ROTATING CHARACTER PATTERN IN ALL PICTURE
                                ; LOCATIONS, WITH COLOUR WHITE ON BLACK.
                                ;
                                ; THIS CHECKS THE HIGH-ORDER PICTURE STORE BITS (THESE HOLD THE
                                ; SELECTED CHARACTER NUMBER)
                                ;
                                ; ERRORS-
                                ; HIGH ORDER PICTURE STORE BITS (LEVC)
                                ; PICTURE STORE STATICISER (LEVD)
                                ;
004234  012737  000006  007454  TEST6:  MOV     #6,TESTNO       ; SET TEST NUMBER
004242  104402                          TRAP+2
004244  004260                          T0601
004246  004737  011624                  JSR     PC,TESTR        ; PRINT TESTNUMBER
004252  004737  014734                  JSR     PC,TYPOUT
004256  006441                          TS6MSG
004260  012737  000005  007506  T0601:  MOV     #5,REPCNT
004266  005737  007462                  TST     PROM            ; IS THERE A PROM ON BOARD
004272  001037                          BNE     T0604           ; YES DON'T LOAD CHSR
                                ;
                                ; RAM CHARACTER SET NEEDS LOADING
                                ;
004274  012700  007624                  MOV     #P02CHR,R0      ; GET START OF DATA TO LOAD
004300  005037  007550                  CLR     CHAR            ; START AT CHARACTER ZERO
004304  005001                  T0602:  CLR     R1              ; 8 LINES PER CHARACTER
004306  013702  007550          T0603:  MOV     CHAR,R2         ; GET CHAR CODE
004312  006302                          ASL     R2
004314  006302                          ASL     R2              ; SET IN RIGHT PLACE
004316  006302                          ASL     R2
004320  050102                          BIS     R1,R2           ; ADD IN LINE NUMBER
004322  052702  100000                  BIS     #100000,R2      ; SELECT CHSR
004326  010277  003104                  MOV     R2,@DEVCSR      ; LOAD CSR
004332  112003                          MOVB    (R0)+,R3        ; GET DATA
004334  000303                          SWAB    R3
004336  105003                          CLRB    R3              ; ENSURE 8 BITS LOADED
004340  000303                          SWAB    R3
004342  010377  003072                  MOV     R3,@DEVBUF      ; LOAD DATA OUT
004346  005201                          INC     R1              ; UPDATE LINE NUMBER
004350  020127  000007                  CMP     R1,#7           ; ALL LINES DONE
004354  003754                          BLE     T0603           ; NO
004356  005237  007550                  INC     CHAR            ; UPDATE CHAR COUNT
004362  023727  007550  000177          CMP     CHAR,#177       ; ALL DONE
004370  003745                          BLE     T0602           ; NOT YET
                                ;
                                ; NOW START PUTTING UP PICTURE
                                ;
004372                          T0604:
004372  012737  040000  007450          MOV     #40000,VIDEON   ; VIDEO ON
004400  012737  000000  007466          MOV     #0,BAKGND       ; BLACK BACKGROUND
004406  012737  000007  007464          MOV     #7,FORGND       ; WHITE FOREGROUND
004414  005037  007470                  CLR     BLINK
004420  005037  007472                  CLR     FLASH           ; NO BLINK OR FLASH
004424  012700  007266                  MOV     #CHXLST,R0      ; POINT TO START OF LIST
004430  010001                  T0605:  MOV     R0,R1           ; GET PICTURE STARTER
```

```
004432  005037  007476                  CLR     CARY            ; START AT TOP LEFT
004436  010102                  T0606:  MOV     R1,R2           ; GET LINE STARTER
004440  005037  007474                  CLR     CARX            ; START OF LINE
004444  011237  007550          T0607:  MOV     (R2),CHAR       ; GET CHAR CODE
004450  004737  012522                  JSR     PC,LOELEM       ; DISPLAY THE CHARACTER
004454  062702  000002                  ADD     #2,R2           ; GET NEXT POINTER
004460  021227  177777                  CMP     (R2),#-1        ; END OF TABLE ?
004464  001002                          BNE     1$
004466  012702  007266                  MOV     #CHXLST,R2      ; YES RESET IT
004472  005237  007474          1$:     INC     CARX            ; GET NEXT COORD
004476  023737  007474  007446          CMP     CARX,XMAX       ; END OF ROW ?
004504  003757                          BLE     T0607           ; NOT YET
004506  062701  000002                  ADD     #2,R1           ; GET NEW LINE STARTER
004512  021127  177777                  CMP     (R1),#-1        ; END OF TABLE
004516  001002                          BNE     2$              ; NO
004520  012701  007266                  MOV     #CHXLST,R1      ; YES RESET IT
004524  005237  007476          2$:     INC     CARY            ; UPDATE ROW
004530  023737  007476  007500          CMP     CARY,YMAX       ; END OF DISPLAY ?
004536  003737                          BLE     T0606           ; NOT YET
                                ;
                                ; WAIT AROUND
                                ;
004540  004737  012040                  JSR     PC,WAITS                ; WAIT 2 SECONDS
004544  000002                          2
004546  005337  007506                  DEC     REPCNT          ; ALL DONE
004552  001406                          BEQ     T0615           ; YES
                                ;
                                ;
                                ;       UPDATE NEW PICTURE STARTER
                                ;
004554  062700  000002                  ADD     #2,R0           ; NEW STARTER
004560  021027  177777                  CMP     (R0),#-1        ; ALL DONE ?
004564  001321                          BNE     T0605           ; NO
004566  000701                          BR      T0604           ; YES RESET TABLE MIXTURE
004570  104404                  T0615:  TRAP+4
004572  004372                          T0604
004574  032777  000100  002700          BIT     #100,@SWR       ; PRESELECT TEST
004602  001402                          BEQ     1$
004604  000137  001326                  JMP     START2          ; YES GO DO IT
004610                          1$:
004610  005237  007302                  INC     DEVFLG          ; UPDATE UNIT COUNT
004614  023737  007302  007304          CMP     DEVFLG,DEVCNT   ; ALL UNITS DONE
004622  002402                          BLT     2$              ; NOT YET
004624  000137  013062                  JMP     ENDIT           ; YES, SIGNAL END OF PASS
004630  013700  007302          2$:     MOV     DEVFLG,R0       ; GET CURRENT UNIT
004634  005200                          INC     R0
004636  006300                          ASL     R0              ; AS A WORD OFFSET
004640  016037  007306  007436          MOV     CSR0(R0),DEVCSR ; GET NEW CSR ADDRESS
004646  016037  007310  007440          MOV     DBUF0(R0),DEVBUF; AND DATA BUFFER
004654  000137  001326                  JMP     START2
```

```
                                        .SBTTL  TEST7
                                ;
                                ; THIS TEST DISPLAYS A ROTATING CHARACTER PATTERN IN ALL PICTURE
                                ; LOCATIONS. IN ADDITION THE COLOUR, IS CHANGED FOR EACH COLUMN.
                                ;
                                ; THIS IS AN EXERCISER ONLY, NOT PART OF THE DIAGNOSTIC ROUTINES
                                ;
004660  012737  000007  007454  TEST7:  MOV     #7,TESTNO       ; SET TEST NUMBER
004666  104402                          TRAP+2
004670  004704                          T0701
004672  004737  011624                  JSR     PC,TESTR        ; PRINT TESTNUMBER
004676  004737  014734                  JSR     PC,TYPOUT
004702  006454                          TS7MSG
004704  012737  000200  007506  T0701:  MOV     #200,REPCNT
004712  005737  007462                  TST     PROM            ; IS THERE A PROM ON BOARD
004716  001037                          BNE     T0704           ; YES DON'T LOAD CHSR
                                ;
                                ; RAM CHARACTER SET NEEDS LOADING
                                ;
004720  012700  007624                  MOV     #P02CHR,R0      ; GET START OF DATA TO LOAD
004724  005037  007550                  CLR     CHAR            ; START AT CHARACTER ZERO
004730  005001                  T0702:  CLR     R1              ; 8 LINES PER CHARACTER
004732  013702  007550          T0703:  MOV     CHAR,R2         ; GET CHAR CODE
004736  006302                          ASL     R2
004740  006302                          ASL     R2              ; SET IN RIGHT PLACE
004742  006302                          ASL     R2
004744  050102                          BIS     R1,R2           ; ADD IN LINE NUMBER
004746  052702  100000                  BIS     #100000,R2      ; SELECT CHSR
004752  010277  002460                  MOV     R2,@DEVCSR      ; LOAD CSR
004756  112003                          MOVB    (R0)+,R3        ; GET DATA
004760  000303                          SWAB    R3
004762  105003                          CLRB    R3              ; ENSURE 8 BITS LOADED
004764  000303                          SWAB    R3
004766  010377  002446                  MOV     R3,@DEVBUF      ; LOAD DATA OUT
004772  005201                          INC     R1              ; UPDATE LINE NUMBER
004774  020127  000007                  CMP     R1,#7           ; ALL LINES DONE
005000  003754                          BLE     T0703           ; NO
005002  005237  007550                  INC     CHAR            ; UPDATE CHAR COUNT
005006  023727  007550  000177          CMP     CHAR,#177       ; ALL DONE
005014  003745                          BLE     T0702           ; NOT YET
                                ;
                                ; NOW START PUTTING UP PICTURE
                                ;
005016                          T0704:
005016  012737  040000  007450          MOV     #40000,VIDEON   ; ENABLE VIDEO
005024  012737  000000  007460          MOV     #0,BASE1        ; START AT CHARAACTER 0
005032  005037  007476          T0705:  CLR     CARY            ; START AT FIRST ROW
005036  013737  007460  007456          MOV     BASE1,BASE      ; SET THIS PICTURE START
005044  013737  007456  007550  T0706:  MOV     BASE,CHAR       ; SET START FOR THIS ROW
005052  005037  007474                  CLR     CARX            ; START OF COLUMN
005056  005037  007472          T0707:  CLR     FLASH           ; NO FLASH
005062  005037  007470                  CLR     BLINK           ; NO BLINK
005066  012737  000001  007466          MOV     #1,BAKGND       ; RED ON
005074  012737  000001  007464  T0710:  MOV     #1,FORGND       ; RED
005102  004737  012522          T0711:  JSR     PC,LOELEM       ; DISPLAY THE CHARACTER
005106  005237  007550                  INC     CHAR            ; SELECT NEXT ONE
```

```
005112  042737  177600  007550          BIC     #177600,CHAR    ; IN RANGE 0-177
005120  005237  007474                  INC     CARX            ; GET NEXT COORD
005124  023737  007474  007446          CMP     CARX,XMAX       ; END OF ROW ?
005132  003432                          BLE     T0712           ; NOT YET
005134  005237  007456                  INC     BASE            ; ELSE SET NEW ROW STARTER
005140  042737  177600  007456          BIC     #177600,BASE    ; IN RANGE 0-177
005146  005237  007476                  INC     CARY            ; UPDATE ROW
005152  023737  007476  007500          CMP     CARY,YMAX       ; END OF DISPLAY ?
005160  003731                          BLE     T0706           ; NOT YET
005162  005237  007460                  INC     BASE1           ; SET NEW PICTURE STARTER
005166  042737  177600  007460          BIC     #177600,BASE1   ; IN RANGE 0-177
                                ;
                                ; WAIT AROUND
                                ;
005174  004737  012040                  JSR     PC,WAITS                ; WAIT 1 SECONDS
005200  000001                          1
005202  005337  007506                  DEC     REPCNT
005206  001422                          BEQ     T0715
005210  000137  005032                  JMP     T0705                   ; YES
005214  000137  005254                  JMP     T0715           ; NO EXIT TEST
                                ;
005220  006337  007464          T0712:  ASL     FORGND          ; GET NEW FOREGROUND
005224  023727  007464  000004          CMP     FORGND,#4       ; ALL DONE
005232  003723                          BLE     T0711           ; NO
005234  005237  007466                  INC     BAKGND          ; GET NEXT BACK GROUND
005240  023727  007466  000007          CMP     BAKGND,#7       ; ALL DONE
005246  003712                          BLE     T0710           ; NO
005250  000137  005056                  JMP     T0707           ; YES
005254  104404                  T0715:  TRAP+4
005256  004704                          T0701
005260  000137  001326                  JMP     START2          ; YES GO DO IT
```

```
                                                .SBTTL  TEST10
                                        ;
                                        ; THIS TEST PRINTS THE CURRENT ACCUMULATION OF ERRORS
                                        ;
005264  012737  000010  007454  TEST10: MOV     #10,TESTNO      ; SET TEST NUMBER
005272  104402                          TRAP+2
005274  005310                          T1001
005276  004737  011624                  JSR     PC,TESTR        ; PRINT TEST NUMBER
005302  004737  014734                  JSR     PC,TYPOUT
005306  006462                          TS10MS
005310  004737  014700          T1001:  JSR     PC,CRLF         ; NEW LINE
005314  004737  014734                  JSR     PC,TYPOUT       ; SAY
005320  007205                          EMSG9                   ; ERROR COUNT =
005322  013700  007442                  MOV     ERRDI1,R0       ; PRINT HI WORD
005326  004737  015024                  JSR     PC,PROCT
005332  013700  007574                  MOV     ERRDIS,R0       ; AND LO WORD
005336  004737  015024                  JSR     PC,PROCT
005342  005037  007444                  CLR     ERRFLG
005346  005037  007574                  CLR     ERRDIS
005352  005037  007442                  CLR     ERRDI1
005356  000137  001200                  JMP     RSTART          ; RESTART
```

```
                                        .SBTTL  TEST11
                                ;
                                ; THIS TEST DISPLAYS A PICTURE SUITABLE FOR TESTING ADAPTERS.
                                ; 7 BLOCKS OF 4 ROWS EACH OF THE FOLLOWING COLOURS,WITH 1,
                                ; 2, AND 4 MHZ BARS
                                ;
005362  012737  000011  007454  TEST11: MOV     #11,TESTNO      ; SET TEST NUMBER
005370  104402                          TRAP+2
005372  005400                          5$
005374  004737  011624                  JSR     PC,TESTR        ; PRINT TEST NUMBER
                                ;
005400  005005                  5$:     CLR     R5              ; FIRST PICTURE ELEMENT
                                ;
005402  010500                  10$:    MOV     R5,R0           ; GET COLOUR
005404  000300                          SWAB    R0
005406  005200                          INC     R0              ; FIRST COLOUR RED,NOT BLACK
005410  042700  177770                  BIC     #177770,R0      ; FOREGROUND
005414  010501                          MOV     R5,R1           ; GET CHARACTER
005416  042701  177700                  BIC     #177700,R1
005422  010577  002010                  MOV     R5,@DEVCSR      ; SET POSITION
005426  020127  000014                  CMP     R1,#12.         ; 1/2 MHZ?
005432  103013                          BCC     20$             ; NO..
005434  042701  000076                  BIC     #76,R1          ; EVEN?
005440  001406                          BEQ     12$             ; YES...
005442  010002                          MOV     R0,R2           ; SET F/B SAME..
005444  006302                          ASL     R2              ; COPY TO BACKGROUND
005446  006302                          ASL     R2
005450  006302                          ASL     R2
005452  050200                          BIS     R2,R0           ; COPY BACK..
005454  000422                          BR      77$
                                ;
005456  005000                  12$:    CLR     R0
005460  000420                          BR      77$             ; LOAD IT..
                                ;
005462  020127  000030          20$:    CMP     R1,#24.         ; 1 MHZ?
005466  103001                          BCC     40$             ; NO..
005470  000414                          BR      77$
                                ;
005472  020127  000044          40$:    CMP     R1,#36.         ; 2 MHZ?
005476  103003                          BCC     60$             ; NO..
005500  052700  011400                  BIS     #19.*400,R0     ; LOAD 2 MHZ..
005504  000406                          BR      77$
                                ;
005506  020127  000060          60$:    CMP     R1,#48.         ; 4 MHZ?
005512  103005                          BCC     100$            ; NO, DISCARD..
005514  052700  011000                  BIS     #18.*400,R0
005520  000400                          BR      77$
                                ;
005522  010077  001712          77$:    MOV     R0,@DEVBUF      ; LOAD CHARACTER
                                ;
005526  005205                  100$:   INC     R5
005530  020527  004000                  CMP     R5,#4000        ; DONE?
005534  001322                          BNE     10$             ; NO..

005536  005737  007462                  TST     PROM            ; LOAD CHSR?
005542  001015                          BNE     120$            ; NO..
```

```
                                        ;
005544  012700  007624                          MOV     #P02CHR,R0          ; YAS...
005550  012701  002000                          MOV     #2000,R1
005554  012777  140000  001654                  MOV     #140000,@DEVCSR
                                        ;
005562  112077  001652          102$:           MOVB    (R0)+,@DEVBUF       ; LOAD BYTE INTO CHSR
005566  005277  001644                          INC     @DEVCSR
005572  005301                                  DEC     R1
005574  001372                                  BNE     102$
                                        ;
005576  012777  040000  001632  120$:           MOV     #40000,@DEVCSR      ; TURN ON VIDEO...
005604  000137  001200                          JMP     RSTART
```

```
                                                .SBTTL  ASCII STRINGS

                                                .NLIST  BEX

005610  046533  031467  037466  SET56:  .ASCII  $[M736?/M737?..a$
005627     133  027124  027126  SETENC: .ASCII  /[T.V. ENCODED?(Y OR N)..a/
005660  042133  041505  046511  NUMDEV: .ASCII  /[DECIMAL NUMBER OF UNITS(1)..a/
005716  042533  052116  051105  CHRCOD: .ASCII  /[ENTER IN OCTAL THE CHARACTER/
005753     133  047503  042504          .ASCII  /[CODE TO BE DISPLAYED....a/
006005     133  052126  031526  GOMSG:  .ASCII  /[VTV30K DIAGNOSTIC/
006027     133  052123  051101          .ASCII  /[START ADDRESS 1000, RESTART 1200/
006070  050133  042514  051501          .ASCII  /[PLEASE READ THE LISTING BEFORE PROCEEDING!a/
006144  052133  042510  042040  GOMS1:  .ASCII  /[THE DEVICES TO BE TESTED MUST/
006202  044133  053101  020105          .ASCII  /[HAVE THEIR ADDRESSES IN ASCENDING ORDER, AND/
006257     133  040510  042526          .ASCII  /[HAVE THE SAME CHARACTERISTICS[a/
006317     133  042523  042514  WMSG:   .ASCII  /[SELECT SWITCH REGISTER OPTIONS[a/
006360  041440  051123     100  TS0MSG: .ASCII  / CSRa/
006365     040  046106  020104  TS1MSG: .ASCII  / FLD VIDEO BELL TSTa/
006411     040  047503  040114  TS2MSG: .ASCII  / COLa/
006416  041440  051125  047523  TS3MSG: .ASCII  / CURSORa/
006426  043040  041055     100  TS4MSG: .ASCII  / F-Ba/
006433     040  044103  051123  TS5MSG: .ASCII  / CHSRa/
006441     040  044520  020103  TS6MSG: .ASCII  / PIC STOREa/
006454  042440  042530  040122  TS7MSG: .ASCII  / EXERa/
006462  042533  051122  051440  TS10MS: .ASCII  /[ERR SUMa/
006473     133  051524  040124  TESMSG: .ASCII  /[TSTa/
006500  042533  042116  047440  PASMSG: .ASCII  /[END OF PASS[a/
006516  052133  040522  020120  ILVMSG: .ASCII  /[TRAP TO ILLEGAL VECTOR  a/
006550  043133  047522  020115  FRMMSG: .ASCII  /[FROM ADDRESS  a/
006570  052133  050131  020105  REDMES: .ASCII  /[TYPE CNTRL-C TO CONTINUE a/
006623     133  053523  020122  SWRMSG: .ASCII  /[SWR =   a/
006635     133  020044  040040  MODADM: .ASCII  /[$  a/
006642  027440  040040          MODSPA: .ASCII  ? / a?
006646  025133  020040     100  MODPRM: .ASCII  /[*  a/
006653     133  044506  051522  BAMSG:  .ASCII  /[FIRST BUS ADDRESS IS .....a/
006707     133  047111  040526  ODAMSG: .ASCII  /[INVALID ADDRESS  a/
006732  047133  047117  042440  NXMSG:  .ASCII  /[NON EXISTANT ADDRESS  a/
006762  042133  043105  052501  NODEFM: .ASCII  /[DEFAULT SETTINGS ARE NOT ALLOWED/
007023     133  046120  040505          .ASCII  /[PLEASE RE-ENTER THE VALUE .....a/
007064  042533  040043          EMSG1:  .ASCII  /[E#a/
007070  020040  052101  050040  EMSG2:  .ASCII  /  AT PC  a/
007102  043533  047517  020104  EMSG3:  .ASCII  /[GOOD : a/
007113     040  020040  040502  EMSG4:  .ASCII  /   BAD :a/
007124  042133  052101  020101  EMSG5:  .ASCII  /[DATA : a/
007135     040  020040  042101  EMSG6:  .ASCII  /   ADDRESS  :a/
007153     133  052123  052101  EMSG7:  .ASCII  /[STATUS  :a/
007166  041533  046101  042514  EMSG8:  .ASCII  /[CALLED FROM :a/
007205     040  020040  051105  EMSG9:  .ASCII  /   ERROR COUNT = a/
        007230                          .EVEN
007230  000000  000000  000000  OCTNUM: .WORD   0,0,0
007236     100     000                  .BYTE   100,0
007240  040502  042523  030061  BASE11: .ASCII  /BASE10a/
        007250                          .EVEN
007250  000000  000000  000000  DECMSG: .WORD   0,0,0
007256  000000  000000  000000  OCTMSG: .WORD   0,0,0,0
                                        .EVEN
```

```
1783                                    .LIST   BEX
```

```
                                                .SBTTL  PROGRAM VARIABLES

007266  000021  000042  000104  CHXLST: .WORD   21,42,104,10,40,-1
007274  000010  000040  177777
007302  000000                  DEVFLG: 0
007304  000001                  DEVCNT: 1
007306  174000                  CSR0:   174000
007310  174002                  DBUF0:  174002
007312  000000                  CSR1:   0
007314  000000                  DBUF1:  0
007316  000000                  CSR2:   0
007320  000000                  DBUF2:  0
007322  000000                  CSR3:   0
007324  000000                  DBUF3:  0
007326  000000                  CSR4:   0
007330  000000                  DBUF4:  0
007332  000000                  CSR5:   0
007334  000000                  DBUF5:  0
007336  000000                  CSR6:   0
007340  000000                  DBUF6:  0
007342  000000                  CSR7:   0
007344  000000                  DBUF7:  0
007346  000000                  CSR8:   0
007350  000000                  DBUF8:  0
007352  000000                  CSR9:   0
007354  000000                  DBUF9:  0
007356  000000                  CSR10:  0
007360  000000                  DBUF10: 0
007362  000000                  CSR11:  0
007364  000000                  DBUF11: 0
007366  000000                  CSR12:  0
007370  000000                  DBUF12: 0
007372  000000                  CSR13:  0
007374  000000                  DBUF13: 0
007376  000000                  CSR14:  0
007400  000000                  DBUF14: 0
007402  000000                  CSR15:  0
007404  000000                  DBUF15: 0
007406  000000                  CSR16:  0
007410  000000                  DBUF16: 0
007412  000000                  CSR17:  0
007414  000000                  DBUF17: 0
007416  000000                  CSR18:  0
007420  000000                  DBUF18: 0
007422  000000                  CSR19:  0
007424  000000                  DBUF19: 0
007426  000000                  CSR20:  0
007430  000000                  DBUF20: 0
007432  000000  000000                  .WORD   0,0
007436  000000                  DEVCSR: 0
007440  000000                  DEVBUF: 0
007442  000000                  ERRDI1: 0
007444  000000                  ERRFLG: 0
007446  000000                  XMAX:   0
007450  000000                  VIDEON: 0
007452  000000                  WAIT1:  0
```

```
007454  000000                  TESTNO: 0
007456  000000                  BASE:   0
007460  000000                  BASE1:  0
007462  000000                  PROM:   0
007464  000000                  FORGND: 0
007466  000000                  BAKGND: 0
007470  000000                  BLINK:  0
007472  000000                  FLASH:  0
007474  000000                  CARX:   0
007476  000000                  CARY:   0
007500  000000                  YMAX:   0
007502  177570                  SWR:    HSWR
007504  000000                  SSWR:   0
007506  000000                  REPCNT: 0
007510  000137  001000          JM600:  JMP     @#START
007514  000001                  FSTCNT: 1
007516  000000                  TRPARG: 0
007520  000000                  TRPSEL: 0
007522  000000                  TRPMEM: 0
007524  000000                  SAVPC:  0
007526  000000                  SAVPC1: 0
007530  000000                  TYPOTA: 0
007532  000000                  RAND:   0
007534  000000                  TYPD1:  0
007536  000000                  MODADR: 0
007540  000000                  MODSAV: 0
007542  000000                  BASADD: 0
007544  000000                  TRPERR: 0
007546  000000                  PARITY: 0
007550  000000                  CHAR:   0
007552  052525                  RANDN:  52525
007554  000001                  RANSEL: 1
007556  000006                  RANMTA: 6
007560  052525                  RANST:  52525
007562  000000                  GOOD:   0
007564  000000                  BAD:    0
007566  000000                  DATA:   0
007570  000000                  STATUS: 0
007572  000000                  ADDRES: 0
007574  000000                  ERRDIS: 0
007576  000000                  ERRARG: 0
007600  000000                  CALLPC: 0
007602  000000                  CNVFLG: 0
007604  000000                  STRADD: 0
007606  000000                  STRLEN: 0
007610  000000                  LOWCHR: 0
007612  000000                  UPPCHR: 0
007614  000000                  RUBFLG: 0
007616  000000                  RANDC:  0
007620  000000                  FSAVPW: 0
007622  000000                  PAL:    0
```

```
                                        .NLIST BEX
                                ;
                                ; THIS CHARACTER SET IS OVERLAYED BY THE RELEVANT CHARACTER SET
                                ; FOR ENCODED OR UNENCODED DEVICES.
007624                          P02CHR:
007624   360    360    360              .BYTE   360,360,360,360,360,360,360,360
007634   010    010    010              .BYTE   010,010,010,010,010,010,010,010
007644   030    030    030              .BYTE   030,030,030,030,030,030,030,030
007654   074    074    074              .BYTE   074,074,074,074,074,074,074,074
007664   000    000    000              .BYTE   000,000,000,000,377,000,000,000
007674   000    000    000              .BYTE   000,000,000,000,377,377,000,000
007704   000    000    377              .BYTE   000,000,377,377,377,377,000,000
007714   010    010    010              .BYTE   010,010,010,010,377,010,010,010
007724   030    030    030              .BYTE   030,030,030,377,377,030,030,030
007734   074    074    377              .BYTE   074,074,377,377,377,377,074,074 ;10
007744   001    002    004              .BYTE   001,002,004,010,020,040,100,200
007754   200    100    040              .BYTE   200,100,040,020,010,004,002,001
007764   000    014    022              .BYTE   000,014,022,040,170,040,100,176
007774   201    102    044              .BYTE   201,102,044,030,030,044,102,201
010004   074    176    377              .BYTE   074,176,377,377,377,377,176,074
010014   030    074    176              .BYTE   030,074,176,377,377,176,074,034
010024   010    010    024              .BYTE   010,010,024,042,301,042,024,010
010034   010    010    034              .BYTE   010,010,034,076,377,076,034,010
010044   252    252    252              .BYTE   252,252,252,252,252,252,252,252
010054   314    314    314              .BYTE   314,314,314,314,314,314,314,314 ;20
010064   377    000    377              .BYTE   377,000,377,000,377,000,377,000
010074   377    377    000              .BYTE   377,377,000,000,377,377,000,000
010104   377    203    205              .BYTE   377,203,205,211,221,241,301,377
010114   377    201    201              .BYTE   377,201,201,201,201,201,201,377
010124   377    376    374              .BYTE   377,376,374,370,360,340,300,200
010134   377    177    077              .BYTE   377,177,077,037,017,007,003,001
010144   200    300    340              .BYTE   200,300,340,360,370,374,376,377
010154   001    003    007              .BYTE   001,003,007,017,037,077,177,377
010164   003    007    016              .BYTE   003,007,016,034,070,160,340,300
010174   300    340    260              .BYTE   300,340,260,070,034,016,007,003 ;30
010204   303    347    176              .BYTE   303,347,176,074,074,176,347,303
010214   000    160    120              .BYTE   000,160,120,160,000,000,000,000
010224   000    000    000              .BYTE   000,000,000,000,000,000,000,000
010234   010    010    010              .BYTE   010,010,010,010,010,000,010,000
010244   024    024    024              .BYTE   024,024,024,000,000,000,000,000
010254   024    024    066              .BYTE   024,024,066,000,066,024,024,000
010264   010    036    040              .BYTE   010,036,040,034,002,074,010,000
010274   000    062    062              .BYTE     0, 62, 62,  4, 10, 20, 46, 46
010304   000    020    050              .BYTE     0, 20, 50, 50, 20, 52, 44, 32
010314   000    030    030              .BYTE     0, 30, 30, 30,  0,  0,  0,  0 ;40(')
010324   000    010    020              .BYTE     0, 10, 20, 40, 40, 40, 20, 10
010334   000    010    004              .BYTE     0, 10,  4,  2,  2,  2,  4, 10
010344   000    052    034              .BYTE     0, 52, 34, 76, 34, 52,  0,  0
010354   000    000    010              .BYTE     0,  0, 10, 10, 76, 10, 10,  0
010364   000    000    000              .BYTE     0,  0,  0,  0, 30, 30, 10, 20
010374   000    000    000              .BYTE     0,  0,  0,  0, 76,  0,  0,  0
010404   000    000    000              .BYTE     0,  0,  0,  0,  0,  0, 30, 30
010414   000    002    002              .BYTE     0,  2,  2,  4, 10, 20, 40, 40
010424   000    034    042              .BYTE   000,034,042,046,052,062,042,034
010434   000    010    030              .BYTE   000,010,030,010,010,010,010,034 ;50(1)
010444   000    034    042              .BYTE   000,034,042,002,034,040,040,076
```

```
010454   000   034   042        .BYTE   000,034,042,002,014,002,042,034
010464   000   004   014        .BYTE   000,004,014,024,044,074,004,004
010474   000   076   040        .BYTE     0, 76, 40, 74,  2,  2, 42, 34
010504   000   014   020        .BYTE     0, 14, 20, 40, 74, 42, 42, 34
010514   000   076   002        .BYTE     0, 76,  2,  4, 10, 20, 20, 20
010524   000   034   042        .BYTE     0, 34, 42, 42, 34, 42, 42, 34
010534   000   034   042        .BYTE     0, 34, 42, 42, 36,  2,  4, 30
010544   000   000   030        .BYTE     0,  0, 30, 30,  0, 30, 30,  0
010554   000   030   030        .BYTE     0, 30, 30,  0, 30, 30, 10, 20 ;60(;)
010564   000   004   010        .BYTE     0,  4, 10, 20, 40, 20, 10,  4
010574   000   000   000        .BYTE     0,  0,  0, 76,  0, 76,  0,  0
010604   000   020   010        .BYTE     0, 20, 10,  4,  2,  4, 10, 20
010614   000   030   044        .BYTE     0, 30, 44,  4, 10, 10,  0, 10
010624   000   074   102        .BYTE     0, 74,102,132,132,104, 40,  0
010634   000   010   024        .BYTE     0, 10, 24, 42, 42, 76, 42, 42
010644   000   074   022        .BYTE     0, 74, 22, 22, 34, 22, 22, 74
010654   000   034   042        .BYTE     0, 34, 42, 40, 40, 40, 42, 34
010664   000   074   022        .BYTE     0, 74, 22, 22, 22, 22, 22, 74
010674   000   076   040        .BYTE     0, 76, 40, 40, 70, 40, 40, 76 ;70(E)
010704   000   076   040        .BYTE     0, 76, 40, 40, 70, 40, 40, 40
010714   000   036   040        .BYTE     0, 36, 40, 40, 46, 42, 42, 36
010724   000   042   042        .BYTE     0, 42, 42, 42, 76, 42, 42, 42
010734   000   034   010        .BYTE     0, 34, 10, 10, 10, 10, 10, 34
010744   000   002   002        .BYTE     0,  2,  2,  2,  2,  2, 42, 34
010754   000   042   044        .BYTE     0, 42, 44, 50, 60, 50, 44, 42
010764   000   040   040        .BYTE     0, 40, 40, 40, 40, 40, 40, 76
010774   000   042   066        .BYTE     0, 42, 66, 52, 42, 42, 42, 42
011004   000   042   062        .BYTE     0, 42, 62, 52, 46, 42, 42, 42
011014   000   034   042        .BYTE     0, 34, 42, 42, 42, 42, 42, 34 ;80(O)
011024   000   074   042        .BYTE     0, 74, 42, 42, 74, 40, 40, 40
011034   000   034   042        .BYTE     0, 34, 42, 42, 42, 52, 44, 32
011044   000   074   042        .BYTE     0, 74, 42, 42, 74, 50, 44, 42
011054   000   036   040        .BYTE     0, 36, 40, 40, 34,  2,  2, 74
011064   000   076   010        .BYTE     0, 76, 10, 10, 10, 10, 10, 10
011074   000   042   042        .BYTE     0, 42, 42, 42, 42, 42, 42, 34
011104   000   042   042        .BYTE     0, 42, 42, 42, 24, 24, 10, 10
011114   000   042   042        .BYTE     0, 42, 42, 42, 42, 52, 66, 42
011124   000   042   042        .BYTE     0, 42, 42, 24, 10, 24, 42, 42
011134   000   042   042        .BYTE     0, 42, 42, 24, 10, 10, 10, 10 ;90(Y)
011144   000   076   002        .BYTE     0, 76,  2,  4, 10, 20, 40, 76
011154   000   070   040        .BYTE   000,070,040,040,040,040,040,070
011164   000   040   040        .BYTE   000,040,040,020,010,004,002,002
011174   000   016   002        .BYTE   000,016,002,002,002,002,002,016
011204   030   074   176        .BYTE   030,074,176,377,030,030,030,030
011214   010   014   016        .BYTE   010,014,016,377,377,016,014,010
011224   030   030   030        .BYTE   030,030,030,030,377,176,074,030
011234   020   060   160        .BYTE   020,060,160,377,377,160,060,020
011244   000   377   377        .BYTE   000,377,377,377,377,377,377,377 ;100
011254   000   000   377        .BYTE   000,000,377,377,377,377,377,377
011264   000   000   000        .BYTE   000,000,000,377,377,377,377,377
011274   000   000   000        .BYTE   000,000,000,000,377,377,377,377
011304   000   000   000        .BYTE   000,000,000,000,000,377,377,377
011314   000   000   000        .BYTE   000,000,000,000,000,000,377,377
011324   000   000   000        .BYTE   000,000,000,000,000,000,000,377
011334   376   376   376        .BYTE   376,376,376,376,376,376,376,376
011344   374   374   374        .BYTE   374,374,374,374,374,374,374,374
```

```
2004  011354  370  370  370        .BYTE   370,370,370,370,370,370,370,370
2005  011364  360  360  360        .BYTE   360,360,360,360,360,360,360,360 ;110
2006  011374  340  340  340        .BYTE   340,340,340,340,340,340,340,340
2007  011404  300  300  300        .BYTE   300,300,300,300,300,300,300,300
2008  011414  200  200  200        .BYTE   200,200,200,200,200,200,200,200
2009  011424  200  200  200        .BYTE   200,200,200,200,200,200,200,377
2010  011434  377  001  001        .BYTE   377,001,001,001,001,001,001,001
2011  011444  001  001  001        .BYTE   001,001,001,001,001,001,001,377
2012  011454  200  200  200        .BYTE   200,200,200,200,200,200,200,377
2013  011464  000  000  000        .BYTE   000,000,000,000,000,000,000,200
2014  011474  200  000  000        .BYTE   200,000,000,000,000,000,000,000
2015  011504  001  000  000        .BYTE   001,000,000,000,000,000,000,000 ;120
2016  011514  000  000  000        .BYTE   000,000,000,000,000,000,000,001
2017  011524  000  010  024        .BYTE   000,010,024,042,343,000,000,000
2018  011534  010  010  030        .BYTE   010,010,030,040,100,040,030,010
2019  011544  000  074  146        .BYTE   000,074,146,303,201,000,000,000
2020  011554  030  060  140        .BYTE   030,060,140,100,100,140,060,030
2021  011564  000  000  000        .BYTE   000,000,000,000,017,010,010,010
2022  011574  010  010  010        .BYTE   010,010,010,010,370,000,000,000
2023  011604  010  010  010        .BYTE   010,010,010,010,013,000,000,000
2024  011614  000  000  000        .BYTE   000,000,000,000,370,010,010,010
2025
```

```
                                        .SBTTL  PRINT TEST NUMBER
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO PRINT THE TEST NUMBER IN OCTAL
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,TESTR
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               TESTNO CONTAINS THE OCTAL TEST NUMBER TO BE
                                ;               PRINTED
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               THE LABEL TESMSG IS THE START ADDRESS OF AN
                                ;               ASCII STRING 'TEST NO'
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               R0 WILL BE CORRUPTED
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               THE MESSAGE 'TEST NO N' WILL BE PRINTED ON THE
                                ;               CONSOLE TERMINAL
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
011624  004737  014734          TESTR:  JSR     PC,TYPOUT
011630  006473                          TESMSG                  ;TEST NO
011632  013700  007454                  MOV     TESTNO,R0
011636  004737  015024                  JSR     PC,PROCT        ;PRINT OCTAL TEST NO
011642  000207                          RTS     PC              ;EXIT
```

```
                                        .SBTTL  SYNC UP THE PROGRAM
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;       THIS ROUTINE WILL ATTEMPT TO SYNC THE PROGRAM UP
                                ;       TO EITHER A HI-LO, OR LO-HI TRANSITION START
                                ;       OF THE FIELD VIDEO PERIOD.
                                ;       A HURRY UP FEATURE IS AVAILABLE, BY TYPING THE
                                ;       SPACE KEY ON THE CONSOLE TERMINAL. THIS CAUSES
                                ;       THE ROUTINE TO EXIT PREMATURELY, INDICATING THAT
                                ;       IT COULD NOT SYNC UP.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,SYNCUP
                                ;       A:
                                ;       B:
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;       DEVCSR CONTAINS THE CURRENT CSR ADDRESS
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;       TWO POSSIBLE RETURN ADDRESSES CAN BE USED DEPENDING UPON
                                ;       THE SUCEES OF THE SYNC ATTEMPT.
                                ;       A: IMPOSSIBLE TO SYNC UP OR SPACE TYPED ON CONSOLE
                                ;       B: SYNCED UP
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;       NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;       ERROR 70- CANNOT FIND AN INITIAL LEVEL
                                ;       ERROR 67- CANNOT CHANGE STATE
                                ;
```

```
011644  004737  013740          SYNCUP: JSR     PC,SAVREG       ; SAVE GOODIES
011650  005005                          CLR     R5              ; SETS TO 1 WHEN HIGH LEVEL DETECTED
011652  005004                          CLR     R4              ; SET SYNC INDETERMINATE
011654  012703  037777                  MOV     #037777,R3      ; SETUP TIMEOUT
                                ;
011660  012702  000040                  MOV     #40,R2          ; SET FILTER TIME CONSTANT
                                ;
011664  017701  175546          10$:    MOV     @DEVCSR,R1      ; GET CONTENTS
011670  042701  157777                  BIC     #157777,R1      ; ISOLATE VIDEO-ON
011674  162701  010000                  SUB     #010000,R1      ; NORMALISE AROUND ZERO.
011700  100405                          BMI     20$             ; UP OR DOWN?
011702  020427  040000                  CMP     R4,#040000      ; AT END STOP?
011706  001406                          BEQ     30$             ; YES..
011710  060104                          ADD     R1,R4           ; ELSE, UPDATE.
011712  000404                          BR      30$
                                ;
011714  020427  140000          20$:    CMP     R4,#140000      ; AT END STOP?
011720  001401                          BEQ     30$             ; YES..
011722  060104                          ADD     R1,R4
                                ;
011724  005704                  30$:    TST     R4              ; SIGN?
011726  100413                          BMI     35$             ; ZERO...
011730  020427  030000                  CMP     R4,#030000      ; VALID ONE?
011734  103417                          BCS     40$             ; NO...
011736  012702  000040                  MOV     #40,R2          ; RESET TIME CONSTANT
011742  020327  037764                  CMP     R3,#037764      ; READY TO TEST?
011746  103012                          BCC     40$             ; NO...
011750  012705  000001                  MOV     #1,R5           ; SET FLAG..
011754  000407                          BR      40$
                                ;
011756  020427  160000          35$:    CMP     R4,#160000      ; VALID ZERO?
011762  103004                          BCC     40$             ; NO..
011764  005705                          TST     R5              ; HIGH LEVEL YET?
011766  001017                          BNE     50$             ; YES, A-OK!
011770  012702  000040                  MOV     #40,R2          ; RESET TIME CONSTANT
                                ;
011774  005302                  40$:    DEC     R2              ; TIME CONSTANT TALLY..
011776  001005                          BNE     45$             ; OK SO FAR IF BRANCH..
                                ; ERROR-
                                ; CHECK 'LEVB CLOCK H'  STABILITY
                                ;       'LEVB VA17 H'
012000  004737  016666                  JSR     PC,ERROR        ; ELSE NOISY SYNC IGNAL!
012004  000070                          70
012006  000137  012032                  JMP     NOSYNC          ; ERROR EXIT
                                ;
012012  005303                  45$:    DEC     R3              ; TIMOUT FOR SYNC?
012014  001323                          BNE     10$             ; NO....
                                ; ERROR-
                                ; CHECK 'LEVB CLOCK H' (8 MHZ)
                                ;       'LEVB CLOCK L'
                                ;       'LEVA VA02 L'
                                ;       'LEVB VA17 H'
012016  004737  016666                  JSR     PC,ERROR        ; SYNC STOPPED!
012022  000067                          67
012024  000402                          BR      NOSYNC          ; ERROR RETURN
                                ;
```

```
012026  062716  000002          50$:     ADD     #2,(SP)          ; UPDATE TO GOOD RETURN
                                 ;
012032  004737  013776          NOSYNC:  JSR     PC,RSTREG        ; RECOVER REGISTERS.
012036  000207                           RTS     PC               ; ..AND EXIT
```

```
                                        .SBTTL  WAIT A NUMBER OF SECONDS
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;       ROUTINE TO WAIT A NUMBER OF UNITS (SECONDS), BASED
                                ;       UPON THE FIELD VIDEO PERIOD.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,WAITS
                                ;       XX
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;       XX IS THE NUMBER OF UNITS (SECONDS) TO WAIT
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;       NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;       NONE
                                ;
                                ;
012040  004737  013740          WAITS:  JSR     PC,SAVREG       ; SAVE REGISTERS
012044  017601  000000                  MOV     @(SP),R1        ; GET SECOND COUNT
012050  062716  000002                  ADD     #2,(SP)         ; PROTECT RETURN
012054  013702  007454                  MOV     TESTNO,R2       ; GET TEST NUMBER
012060  105762  001436                  TSTB    TABLE2(R2)      ; IS IT STALL-ABLE?
012064  001405                          BEQ     1$              ; NO IF BRANCH
012066  005737  001174                  TST     REACT           ; HAS USER OVERRIDDEN TIMER?
012072  001402                          BEQ     1$              ; NO..
012074  013701  001174                  MOV     REACT,R1        ; ELSE RESET TIME.
                                ;
012100  013702  007452          1$:     MOV     WAIT1,R2        ; GET TIME FOR ONE SECOND.
```

```
                                        ;
012104  105737  177560          2$:     TSTB    TKS             ; ANY CONSOLE INPUT?
012110  100012                          BPL     3$              ; NO...
012112  004737  014206                  JSR     PC,READ         ; ELSE GET CHARACTER..
012116  020027  000040                  CMP     R0,#'           ; EXIT ON SPACE
012122  001414                          BEQ     7$
012124  020027  000007                  CMP     R0,#'G&37       ; 'BELL'?
012130  001002                          BNE     3$              ; NO..
012132  004737  013546                  JSR     PC,MONIT        ; ..ELSE CALL UP MONITOR.
                                        ;
012136  004737  011644          3$:     JSR     PC,SYNCUP       ; SYNC UP
012142  000240                          NOP                     ; IGNORE ERRORS
012144  005302                          DEC     R2              ; ONE SECOND?
012146  001356                          BNE     2$              ; NOT YET..
012150  005301                          DEC     R1              ; MORE SECONDS?
012152  001352                          BNE     1$              ; YES, KEEP LOOPING
                                        ;
012154  004737  013776          7$:     JSR     PC,RSTREG       ; EXIT AFTER RECOVERING REGS.
012160  000207                          RTS     PC
```

```
                                        .SBTTL  SELECT 525/625
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;       THIS ROUTINE PROMPTS THE OPERATOR TO DETERMINE WHAT
                                ;       DEVICE(S) ARE BEING TESTED. THE ROUTINE PROMPTS
                                ;       'VTV30-K?'', AND LOOKS FOR A RESPONSE OF A,B,C,D OR G,
                                ;       TO SELECT THE APPROPRIATE DEVICE. HAVING FOUND WHICH
                                ;       DEVICE HAS BEEN SELECT, THE ROUTINE THEN SETS UP FLAGS
                                ;       AS TO THE SIZE OF THE DISPLAY AREA AND WHETHER IT HAS
                                ;       A ROM OR RAM CHARACTER STORE.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,SETDEV
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;       PROM=0 FOR RAM CHSR
                                ;       PROM=1 FOR PROM CHSR
                                ;
                                ;       XMAX= MAXIMUM X ADDRESS
                                ;       YMAX= MAXIMUM Y ADDRESS
                                ;
                                ;       WAIT1= TIMER VALUE FOR 1 SECOND (?)
                                ;
                                ;       RELEVANT RAM CHARACTER SET
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;       NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;       NONE
```

```
                                        ;
012162  004737  013740          SETDEV: JSR     PC,SAVREG       ; SAVE OLD ENVIRONMENT
                                        ;
012166  004737  014734          1$:     JSR     PC,TYPOUT       ; ASK MODULE TYPE?
012172  005610                          SET56
012174  004737  014206                  JSR     PC,READ         ; READ REPLY...A/B/C/D/G
012200  012701  012374                  MOV     #DEVLST,R1      ; GET CHECK TABLE ADDRESS
                                        ;
012204  121100                  3$:     CMPB    (R1),R0         ; THIS MODULE?
012206  001404                          BEQ     10$             ; YES...
012210  005721                          TST     (R1)+           ; ELSE SKIP ON TO NEXT
012212  005711                          TST     (R1)            ; END OF TABLE?
012214  001373                          BNE     3$              ; NO, TRY NEXT
012216  000763                          BR      1$              ; ELSE RESTART.
                                        ;
012220  005037  007622          10$:    CLR     PAL             ; PRESET TO NON-TV-ENCODED.
012224  032711  001000                  BIT     #1000,(R1)      ; IS IT 525?
012230  001407                          BEQ     12$             ; NO IF BRANCH
012232  012737  000033  007500          MOV     #27.,YMAX       ; MAXIMUM ROW NO FOR 525
012240  012737  000074  007452          MOV     #60.,WAIT1      ; 60 FRAMES/SEC
012246  000406                          BR      13$
                                        ;
012250  012737  000037  007500  12$:    MOV     #31.,YMAX       ; 32 ROWS IN 625
012256  012737  000062  007452          MOV     #50.,WAIT1      ; 50 FRAMES/SEC
                                        ;
012264  012737  000077  007446  13$:    MOV     #63.,XMAX       ; 48 VISIBLE,16 INVISIBLE CELLS.
012272  012737  000001  007462          MOV     #1,PROM         ; PRESET TO PROM CHAR SET.
012300  032711  000400                  BIT     #400,(R1)       ; PROM?
012304  001030                          BNE     20$             ; YES...
012306  005037  007462                  CLR     PROM            ; ELSE RESET TO RAM CHAR STORE.
012312  004737  014734                  JSR     PC,TYPOUT       ; ASK IF ENCODED FOR TV!
012316  005627                          SETENC
012320  004737  014206                  JSR     PC,READ         ; GET REPLY..
012324  012701  017242                  MOV     #NORCH,R1       ; DEFAULT TO NORMAL CHARACTER SET
012330  120027  000131                  CMPB    R0,#'Y          ; IF YES, THEN WE ARE ENCODED.
012334  001005                          BNE     15$
012336  012737  000001  007622          MOV     #1,PAL          ; SET ENCODED (ONLY RELEVANT FOR RAM)
012344  012701  021242                  MOV     #PALCH,R1       ; GET THIS CHAR SET
                                        ;
012350  012700  007624          15$:    MOV     #P02CHR,R0      ; GET DESTINATION.
012354  012702  001000                  MOV     #1000,R2        ; 512 WORDS=128 CHARS.
                                        ;
012360  012120                  16$:    MOV     (R1)+,(R0)+     ; TRANSFER 2 BYTES.
012362  005302                          DEC     R2
012364  001375                          BNE     16$             ; ANY MORE?
                                        ;
012366  004737  013776          20$:    JSR     PC,RSTREG       ; RECOVER REGISTERS.
012372  000207                          RTS     PC
                                        ;
                                        ; THE FOLLOWING TABLE HAS AN ASCII CHACK BYTE IN EACH WORD,
                                        ; TOGETHER WITH 400 IF IT IS A PROM UNIT, AND 1000 FOR 525.
                                        ;
012374  000470                  DEVLST: '8+400+0000     ; INTERFACE, -KA
012376  001471                          '9+400+1000     ;            -KB
012400  000060                          '0+000+0000     ;            -KC
012402  001061                          '1+000+1000     ;            -KD
```

```
 2387  012404  000462                    '2+400+0000     ;          -KG
 2388  012406  000000                    0
```

```
                                        .SBTTL  LOAD A FULL PICTURE
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;       ROUTINE TO LOAD A FULL PICTURE WITH A GIVEN
                                ;       CHARACTER CODE, COLOUR, BLINK AND FLASH SETTING
                                ;
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,LOADFS
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;       FLASH,BLINK,CHAR,FORGND,BAKGND, VIDEON ARE TO AS
                                ;       FLASH CODE, BLINK CODE, CHARACTER CODE, FOREGROUND
                                ;       COLOUR, BACKGROUND COLOUR, AND THE VIDEO ON ENABLE
                                ;       RESPECTIVELY.
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;       CARX, AND CARY ARE DESTROYED
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;       NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;       NONE
                                ;
                                ;
012410  004737  013740          LOADFS: JSR     PC,SAVREG
012414  005037  007476                  CLR     CARY            ; CLEAR OUR Y COORD
012420  005037  007474                  CLR     CARX            ; AND OUR X COORD
012424  013702  007550                  MOV     CHAR,R2         ; PRESET THE PICT TO THIS CHAR
012430  000302                          SWAB    R2              ; HIGH ORDER BYTE
012432  053702  007470                  BIS     BLINK,R2        ; LOOAD BLINK AND FLASH BITS
012436  053702  007472                  BIS     FLASH,R2
```

```
012442  053702  007464                  BIS     FORGND,R2       ; LOAD FOREGROUND COLOUR
012446  013701  007466                  MOV     BAKGND,R1       ; GET BACKGROUND COLOUR
012452  006301                          ASL     R1              ; TO CORRECT BITS
012454  006301                          ASL     R1
012456  006301                          ASL     R1
012460  050102                          BIS     R1,R2           ; AND LOAD INTO DATUM WORD
012462  013700  007436                  MOV     DEVCSR,R0       ; GET THE CSR ADDRESS
012466  013701  007440                  MOV     DEVBUF,R1       ; ALSO THE DATA BUFFER
012472  013703  007450                  MOV     VIDEON,R3       ; GET THE VIDEO ON BIT..
012476  012704  004000                  MOV     #4000,R4        ; PICTURE STORE COUNT
                                ;
012502  010310                  2$:     MOV     R3,(R0)         ; LOAD CSR WITH ADDRESS
012504  010211                          MOV     R2,(R1)         ; LOAD PICTURE ELEMENT
012506  005203                          INC     R3              ; NEXT PICTURE ELEMENT
012510  005304                          DEC     R4              ; ANY MORE?
012512  001373                          BNE     2$              ; YES...
012514  004737  013776                  JSR     PC,RSTREG       ; RESTORE THE REGISTERS
012520  000207                          RTS     PC              ; AND EXIT.
```

```
                                        .SBTTL  LOAD A PICTURE ELEMENT
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;       ROUTINE TO LOAD A SINGLE PICTURE ELEMENT
                                ;;
                                ;       CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,LOELEM
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;       CARY= Y POS
                                ;       CARX= X POS
                                ;       CHAR= CHAR CODE
                                ;       FORGND= FOREGROUND COLOUR
                                ;       BAKGND= BACKGROUND COLOUR
                                ;       BLINK= BLINK STATUS
                                ;       FLASH= FLASH STATUS
                                ;       VIDEON= VIDEO ENABLE STATUS
                                ;       DEVCSR= ADDRESS OF CURRENT CSR
                                ;       DEVBUF= ADDRESS OF CURRENT DBUFF
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;       NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;       NONE
                                ;
012522  004737  013740          LOELEM: JSR     PC,SAVREG
012526  013702  007474                  MOV     CARX,R2         ; SET UP OUR SAR REGISTER
012532  013701  007476                  MOV     CARY,R1
```

```
012536  000301                  SWAB    R1              ; GET Y ADDRESS IN TOP BYTE
012540  105001                  CLRB    R1
012542  006201                  ASR     R1
012544  006201                  ASR     R1              ; SHIFT IT DOWN
012546  042701  174000          BIC     #174000,R1      ; CLEAR RUBBISH
012552  053701  007450          BIS     VIDEON,R1       ; SET VIDEO ON
012556  050102                  BIS     R1,R2           ; THEN SET IT UP
012560  010277  174652          MOV     R2,@DEVCSR      ; SET OUR X AND Y ADDRESSES
012564  013702  007464          MOV     FORGND,R2               ; START WITH FOREGROUND
012570  013701  007466          MOV     BAKGND,R1       ; GET BACKGROUND COLOUR
012574  006301                  ASL     R1
012576  006301                  ASL     R1              ; SET IT IN CORRECT POSITION
012600  006301                  ASL     R1
012602  050102                  BIS     R1,R2           ; ADD TO COMPOSITE
012604  053702  007470          BIS     BLINK,R2                ; SET BLINK IF REQUIRED
012610  053702  007472          BIS     FLASH,R2                ; AND FLASH
012614  013701  007550          MOV     CHAR,R1         ; GET CHARACTER
012620  000301                  SWAB    R1              ; THE SET IN THE CORRECT POSITION
012622  105001                  CLRB    R1
012624  050102                  BIS     R1,R2           ; ADD IT IN
012626  010277  174606          MOV     R2,@DEVBUF      ; OUT
012632  004737  013776          JSR     PC,RSTREG
012636  000207                  RTS     PC
```

```
                                        .SBTTL  'SILLSI' SUBROUTINE
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO ESTABLISH WHETHER OR NOT THE
                                ;               DIAGNOSTIC IS RUNNING  ON  A  PROCESSOR
                                ;               WHICH POSSESSES ONLY ONE INTERRUPT  BUS
                                ;               PRIORITY LEVEL.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,SILLSI
                                ;
                                ;       INPUT PARAMTERS:
                                ;
                                ;               NONE
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               THE VARIABLE 'LSIFLG' WILL BE SET UP TO
                                ;               REFLECT WHETHER OR NOT THE PROCESSOR HAS
                                ;               A SINGLE INTERRUPT PRIORITY LEVEL.
                                ;
                                ;               LSIFLG  =  0  => MULTIPLE INT. PRIORITIES
                                ;                       <> 0  => SINGLE INT. PRIORITY
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
012640  005037  012702          SILLSI: CLR     LSIFLG          ;SET UP FLAG FOR NON-LSI
012644  012737  012670  000004          MOV     #1$,4           ;INSTALL TRAP THRU 4 VECTOR
012652  012737  000340  000006          MOV     #340,6          ;AND CORRESPONDING PRIORITY
```

```
                                        ;
012660  005737  177776                  TST     PSW             ;TRY ADDRESSING THE PSW --
012664  000240                          NOP                     ;IF NOT THERE, TRAP THRU 4
012666  000404                          BR      2$              ;WILL OCCUR, ELSE BRANCH.
                                        ;
012670  022626                          1$:     CMP     (SP)+,(SP)+     ;PERFORM A DUMMY RTI
012672  012737  177777  012702          MOV     #-1,LSIFLG      ;PROCESSOR HAS ONE INT.LEVEL
                                        ;
012700  000207                          2$:     RTS     PC              ;RETURN TO MAINLINE CALL.
                                        ;
                                        ;
                                        ;
012702  000000                          LSIFLG: 0                       ;0 => MULTIPLE INT.PRIORITIES
                                                                        ;-1=> SINGLE INT. PRIORITY
                                        ;
                                        ;
                                        ;
```

```
                                        .SBTTL  NON EXISTANT SWR TRAP
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               THE TRAP WHEN TESTING FOR THE HARDWARE
                                ;               SWITCH REGISTER WILL OCCUR HERE
                                ;               THE LOACTION SWR WILL BE SET TO CONTAIN THE
                                ;               ADDRESS OF THE SOFTWARE SWITCH REGISTER SSWR
                                ;
                                ;       ENTRY POINT
                                ;
                                ;               SWRSET
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               OCCURS IF A HARDWARE SWITCH REGISTER IN NOT
                                ;               PRESENT
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               THE LOCATION SWR WILL BE SET TO CONTAIN SSWR
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               AN RTI IS PERFORMED
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;

012704  012737  007504  007502  SWRSET: MOV     #SSWR,SWR
012712  000002                          RTI
```

```
                        .SBTTL  SET UP ILLEGAL VECTOR TRAPS
;
;
;
;       DESCRIPTION:
;
;               ROUTINE TO SET CATCHES FOR TRAPS TO ILLEGAL
;               VECTORS IN THE RANGE 0 TO 772, DURING THE
;               RUNNING OF THE TESTS.
;
;               THE CATCH IS TO FORCE THE EXECUTION OF AN IOT
;               TRAP.
;
;               THE VECTOR 14 (ODT VECTOR) IS LEFT FREE,
;               34 (TRAP VECTOR) IS SET TO THE ADDRESS TRAPSV
;               TO SERVICE THE TRAP INSTRUCTION. THE VECTOR
;               20 (IOT) IS SET TO ILLVEC TO SERVICE ILLEGAL
;               VECTOR TRAPS, AND THE ADDRESSES 200 AND 202
;               ARE SET WITH A JUMP TO START, THUS ALLOWING
;               THE BE BE RESTARTED FROM ADDRESS 200.
;               LOCATIONS 30 AND 32 AND SET TO CATCH EMT CALLS
;               AND HENCE READ THE PROCESSOR PRIORITY.
;
;               LOCATION 40 IS LEFT FREE TO CONTAIN THE LOAD
;               MEDIUM INDICATORS, LOCATION IS LEFT FREE TO
;               CONTAIN THE XXDP
;               RETURN ADDRESS (IF PRESENT). LOCATION 46 IS SET TO
;               CONTAIN A POINTER TO THE XXDP RETURN ADDRESS AND
;               LOCATION 52 IS SET TO ZERO.
;
;
;
;       CALLING SEQUENCE:
;
;               JSR     PC,VECTOR
;
;       INPUT PARAMETERS:
;
;               NONE
;
;
;       IMPLICIT INPUT PARAMETERS:
;
;               NONE
;
;
;       OUTPUT PARAMETERS:
;
;               R0 AND R1 WILL BE CORRUPTED
;
;               ADDRESSES 0 THRU TO 774 WILL BE SET WITH
;               APPROPRIATE VALUES
;
;       IMPLICIT OUTPUT PARAMETERS:
;
;               NONE
```

```
                                        ;
                                        ;
                                        ;       COMPLETION CODES:
                                        ;
                                        ;               NONE
                                        ;
                                        ;
                                        ;       POSSIBLE ERROR CODES:
                                        ;
                                        ;               NONE
                                        ;
                                        ;
                                        ;
                                        ;
 012714  005000                         VECTOR: CLR     R0              ;FILL 0-572 WITH IOT TRAPS
 012716  012701  000002                         MOV     #2,R1
 012722  020027  000014                 FILL:   CMP     R0,#14          ;ODT TRAP?
 012726  001002                                 BNE     1$
 012730  022020                                 CMP     (R0)+,(R0)+     ;YES BUMP R0
 012732  000410                                 BR      FILL1
 012734  020027  000040                 1$:     CMP     R0,#40          ;XXDP RETURN ADDRESS
 012740  001002                                 BNE     2$              ;NO
 012742  022020                                 CMP     (R0)+,(R0)+     ;YES BUMP R0
 012744  000403                                 BR      FILL1
 012746  010120                         2$:     MOV     R1,(R0)+        ;'.+2'
 012750  012720  000004                         MOV     #4,(R0)+        ;'IOT'
 012754  062701  000004                 FILL1:  ADD     #4,R1
 012760  020027  000774                         CMP     R0,#774
 012764  002756                                 BLT     FILL
 012766  012737  013326  000034                 MOV     #TRAPSV,34      ;TRAP (LOOP CONTROL)
 012774  012737  000340  000036                 MOV     #340,36
 013002  012737  013156  000030                 MOV     #FADR,30        ; PLUG EMT FOR READING
 013010  012737  000340  000032                 MOV     #340,32         ; THE PROCESSOR STATUS
 013016  012737  013226  000020                 MOV     #ILLVEC,20      ;PLUG 20 FOR IOTS
 013024  012737  000340  000022                 MOV     #340,22
 013032  013737  007510  000200                 MOV     JM600,200       ;SET UP JMP START IN LOC 200
 013040  013737  007512  000202                 MOV     JM600+2,202
 013046  012737  013142  000046                 MOV     #$ENDAD,46      ;POINT TO RETURN TO XXDP
 013054  005037  000052                         CLR     52              ;CLEAR 52
 013060  000207                                 RTS     PC              ;THEN EXIT
```

```
                        .SBTTL  XXDP END OF PASS HOOKS
;
;
;       DESCRIPTION:
;
;
;               ROUTINE TO SIGNIFY END OF PASS, AND IF THE
;       PROGRAM HAS BEEN LOADED USING AN XXDP MONITOR A CALL
;       WILL BE MADE BACK TO THE MONITOR. THE LOCATIONS USED BY
;       XXDP ARE 40, AND 41 FOR THE LOAD MEDIUM AND 42, 43 FOR
;       THE RETURN ADDRESS.
;       IF A PRESELECTED TEST IS IN OPERATION THE PROGRAM WILL
;       GO AND SELECT THAT TEST.
;
;
;       ENTRY POINT:
;
;
;               ENDIT
;
;
;       INPUT PARAMETERS:
;
;
;               LOCATION 42/43 CONTAINS THE XXDP RETURN ADDRESS
;
;
;       IMPLICIT INPUT PARAMETERS:
;
;
;               NONE
;
;
;       OUTPUT PARAMETERS:
;
;
;               NONE
;
;
;       IMPLICIT OUTPUT PARAMETERS:
;
;
;               NONE
;
;
;       COMPLETION CODES:
;
;
;               IF LOCATIONS 42/43 ARE NON ZERO THEY ARE
;       ASSUMMED TO CONTAIN THE XXDP MONITOR RETURN ADDRESS
;
;
;       POSSIBLE ERROR CODES:
;
;
;               NONE
```

```
                                        ;
                                        ;
013062  012706  001000          ENDIT:  MOV     #START,SP       ; RESET THE STACK
013066                                  PSWSET  #340            ; AND PROCESSOR PRIORITY
013102  032777  000100  174372          BIT     #100,@SWR       ; IS THERE A PRESELECT ON
013110  001402                          BEQ     1$              ; NO
013112  000137  001326                  JMP     START2          ; YES GO SELECT THE TEST
                                        ;
                                        ; NO PRESELECT ON SO SIGNAL END OF PASS
                                        ;
013116  104402                  1$:     TRAP+2                  ; BYPASS MESSAGE
013120  001306                          START1
013122  004737  014734          2$:     JSR     PC,TYPOUT       ; END OF PASS
013126  006500                          PASMSG
013130  013700  000042                  MOV     @#42,R0         ; GET RETURN ADDRESS TO XXDP
013134  001002                          BNE     $ENDAD          ; IF IT IS ZERO THERE IS NO
013136  000137  001306                  JMP     START1          ; MONITOR SO RESTART DIAG
                                        ;
                                        ;
013142  004710                  $ENDAD: JSR     PC,(R0)         ; CALLED VIA XXDP SO RETURN
013144  000240                          NOP                     ; THERE
013146  000240                          NOP
013150  000240                          NOP                     ; ALLOW A SLIGHT PAUSE
013152  000137  001306                  JMP     START1          ; THEN RESTART THE DIAGNOSTIC
                                        ;
                                        ;
                                        ;
```

```
                                        .SBTTL  READ PROCESSOR PRIORITY
                                ;
                                ; DESCRIPTION:
                                ;
                                ;       THIS IS THE EMT HANDLER AND IS USED TO READ
                                ;       THE PROCESSOR PRIORITY OFF THE STACK AND RETURNING
                                ;       IT IN FSAVPW
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       CALLED BY ISSUING AN EMT
                                ;
                                ; INPUT PARAMETERS:
                                ;
                                ;       NONE
                                ;
                                ; IMPLICIT INPUT PARAMETERS:
                                ;
                                ;       THE LOCATIONS 30 AND 32 MUST HAVE BEEN SET
                                ;       UP TO POINT TO THIS ROUTINE
                                ;
                                ; OUTPUT PARAMETERS:
                                ;
                                ;       THE CONTENTS OF THE PROCESSOR PRIORITY
                                ;       ARE RETURNED IN THE LOCATION FSAVPW
                                ;
                                ; COMPLETION CODES:
                                ;
                                ;       NONE
                                ;
                                ; POSSIBLE ERROR CODES:
                                ;
                                ;       NONE
                                ;
                                ;
013156  016637  000002  007620  FADR:   MOV     2(SP),FSAVPW    ; READ PRIORITY
013164  000002                          RTI                     ; RETURN TO CALLER
                                ;
```

```
                                        .SBTTL  RING TTY BELL
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO RING THE BELL ON THE CONSOLE
                                ;               TERMINAL, IF BIT 5 IS SET IN THE SWITCH
                                ;               REGISTER.
                                ;
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,BELL
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               THE SWITCH REGISTER MUST HAVE BEEN
                                ;               SET UP
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               THE TELETYPE BELL WILL BE RUNG IF
                                ;               APPROPRIATE
                                ;
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;
```

```
2954  013216  112737  000000  177566          MOVB    #0,TPB          ;PRINT NULL
2955  013224  000207                          RTS     PC              ;GO OUT
```

```
                                        .SBTTL  ILLEGAL VECTOR TRAP CATCH
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               TRAPS TO ILLEGAL VECTORS WILL BE
                                ;               REPORTED HERE. THE VECTOR TO WHICH
                                ;               THE TRAP OCCURRED WILL BE PRINTED
                                ;               AS WELL AS THE ADDRESS IN THE MAIN
                                ;               LINE CODE FROM WHICH THE TRAP OCCURRED.
                                ;               A PROGRAM RESTART IS THEN PERFORMED,
                                ;               UNLESS A NEW TEST HAS BEEN SELECTED
                                ;               WHILE RUNNING UNDER A SOFTWARE
                                ;               SWITCH REGISTER.
                                ;
                                ;
                                ;       ENTRY POINT
                                ;
                                ;               ILLVEC
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               ENTRY IS CAUSED BY AN ILLEGAL
                                ;               VECTOR TRAP
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               THE VECTOR AND MAINLINE ADDRESS
                                ;               WILL BE PRINTED
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               A PROGRAM RESTART OR A NEW
                                ;               TEST SELECTION .
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;               NONE
                                ;
```

```
                                        ;
                                                ;
                                                ;
                                                ;A TRAP TO AN UNRECOGNISED VECTOR WILL
                                                ;BE REPORTED FROM HERE.
                                                ;
013226  004737  014734          ILLVEC: JSR     PC,TYPOUT
013232  006516                          ILVMSG
013234  012600                          MOV     (SP)+,R0
013236  162700  000004                  SUB     #4,R0
013242  004737  015024                  JSR     PC,PROCT        ;PRINT VECTOR
013246  004737  014734                  JSR     PC,TYPOUT
013252  006550                          FRMMSG                  ;PRINT MAINLINE ADDRES
013254  005726                          TST     (SP)+
013256  012600                          MOV     (SP)+,R0
013260  004737  015024                  JSR     PC,PROCT
013264  005726                          TST     (SP)+
013266  004737  013546                  JSR     PC,MONIT
013272  000137  001200                  JMP     RSTART
```

```
                                        .SBTTL  WAIT FOR PRINTER READY
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO WAIT UNTIL THE PRINTER
                                ;               ON THE CONSOLE TERMINAL IS READY,
                                ;               IE: IT IS READY TO PRINT THE NEXT
                                ;               CHARACTER.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,TPREDY
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;
013276  105737  177564          TPREDY: TSTB    TPS     ;ROUTINE TO WAIT FOR PRINTER READY
013302  100375                          BPL     TPREDY
013304  000207                          RTS PC
```

```
                                        .SBTTL  SET ITERATION COUNT
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO SET UP THE TEST ITERATION
                                ;               COUNT. A PROPOSED COUNT IS SET IN R4
                                ;               THEN UNLESS BIT 13 IN THE SWITCH REGISTER
                                ;               IS SET, THE SAME VALUE IS RETURNED.
                                ;               IF BIT 13 IS SET THEN FAST ITERATION IS
                                ;               ASSUMED AND A VALUE OF 1 IS RETURNED IN
                                ;               R4.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,FASTSW
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               R4 CONTAINS THE PROPOSED ITERATION
                                ;               COUNT
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               SETTING BIT 13 IN THE SWR WILL INDICATE
                                ;               FAST ITERATION, AND A SINGLE PASS
                                ;               WILL BE REQUESTED
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               R4 WILL CONTAIN THE ACTUAL ITERATION
                                ;               COUNT
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;;              NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;

013306  032777  020000  174166  FASTSW: BIT     #20000,@SWR
```

```
3140  013314  001001                        BNE     1$
3141  013316  000207                        RTS     PC
3142  013320  013704  007514        1$:     MOV     FSTCNT,R4
3143  013324  000207                        RTS     PC
```

```
                        .SBTTL  TRAP SERVICE ROUTINE
;
;
;       DESCRIPTION:
;
;       TRAP HANDLING ROUTINE. THE TRAP HANDLER IS
;       ENTERED UPON THE EXECUTION OF ANY TRAP
;       INSTRUCTION. IT COMPARES THE LOWER BYTE OF THE
;       TRAP INSTRUCTION WITH THE CONTENTS ON THE SWITCH
;       REGISTER, AND IF A MATCH IS FOUND TAKES THE CONTENTS
;       OF THE ADDRESS FOLLOWING THE TRAP INSTRUCTION AS THE
;       RETURN ADDRESS.
;
;       THE EXPECTED FORMAT IS:
;
;                       TRAP+N
;                       ADDR
;
;       WHERE ADDR IS THE ADDRESS TO PROCEED TO IF N MATCHES
;       THE SWITCH REGISTER SETTINGS. THE TRAP ARGUEMENT IS
;       RELATED TO THE SWITCH REGISTER SETTINGS THUS:
;
;
;       TRAP ARG                        SWR SETTING
;
;       2                               200
;       4                               400
;       10                              1000
;       20                              2000
;       30                              3000
;       40                              4000
;       50                              5000
;       60                              6000
;       70                              7000
;
;
;
;       THE SETTING OF SWR BIT 12, WILL FORCE THE TRAP
;       HANDLER TO USE THE SWR SETTINGS THAT WERE IN
;       FORCE WHEN THE LAST TRAP INSTRUCTION WAS EXECUTED.
;
;       IF A CNTRL-G IS OUTSTANDING ON THE CONSOLE TERMINAL
;       WHEN THE TRAP WAS EXECUTED, THEN MONIT IS CALLED. IF
;       CNTRL-O WAS OUTSTANDING THEN MODIFY IS CALLED
;
;
;
;
;       ENTRY POINT
;
;               TRAPSV
;
;
;       INPUT PARAMETERS:
;
```

```
                                        ;               ON EXECUTION OF ANY TRAP INSTRUCTION
                                        ;
                                        ;
                                        ;       IMPLICIT INPUT PARAMETERS:
                                        ;
                                        ;               THE SWR HAS BEEN SET UP
                                        ;
                                        ;
                                        ;       OUTPUT PARAMETERS:
                                        ;
                                        ;               EXIT TO THE CONTENTS OF THE ADDRESS FOLLOWING
                                        ;       THE TRAP INSTRUCTION IF A MATCH WAS FOUND, ELSE THE
                                        ;       PROGRAM WILL CONTINUE FROM THE ADDRESS AFTER THAT.
                                        ;
                                        ;
                                        ;       IMPLICIT OUTPUT PARAMETERS:
                                        ;
                                        ;               NONE
                                        ;
                                        ;
                                        ;       COMPLETION CODES:
                                        ;
                                        ;               NONE
                                        ;
                                        ;
                                        ;       POSSIBLE ERROR CODES:
                                        ;
                                        ;               NONE
                                        ;
                                        ;
                                        ;
013326  004737  013740                  TRAPSV: JSR     PC,SAVREG       ;SAVE REGS
013332  011600                                  MOV     (SP),R0
013334  016037  177776  007516                  MOV     -2(R0),TRPARG   ;GET TRAP CALL
013342  105737  177560                          TSTB    TKS             ;LOOK FOR MONITOR CALL
013346  100015                                  BPL     3$
013350  004737  014206                          JSR     PC,READ
013354  120027  000017                          CMPB    R0,#17          ;IS IT CNTRL-O ?
013360  001003                                  BNE     1$              ;NO
013362  004737  015450                          JSR     PC,MODIFY       ;YES CALL MODIFIER ROUTINE
013366  000405                                  BR      3$
013370  120027  000007                  1$:     CMPB    R0,#7           ;IS IT CTRL-G?
013374  001002                                  BNE     3$
013376  004737  013546                          JSR     PC,MONIT        ;YES, GO TO SWR MONITOR
013402  017737  174074  007520          3$:     MOV     @SWR,TRPSEL
013410  132737  000020  007521                  BITB    #20,TRPSEL+1    ;IS IT PRESERVE SCOPE
013416  001012                                  BNE     TRPSCP          ;YES
013420  013737  007520  007522                  MOV     TRPSEL,TRPMEM   ;NO SO SAVE SWITCH SETTING
013426  042737  170777  007522                  BIC     #170777,TRPMEM  ;GET IT SO WE CAN COMPARE
013434  132737  000016  007521                  BITB    #16,TRPSEL+1    ;ANY SCOPE LEVELS SET
013442  001412                                  BEQ     TRPLP           ;NO
013444  013700  007516                  TRPSCP: MOV     TRPARG,R0       ;YES
013450  006000                                  ROR     R0              ;GET TO POSITION FOR COMPARE
013452  006000                                  ROR     R0              ;FOR 10 & ABOVE
013454  000300                                  SWAB    R0              ;ONLY FOR SCOPE BITS(9-11)
```

```
013456  042700  170777                  BIC     #170777,R0          ;ARGUMENT SCOPE BITS
013462  020037  007522                  CMP     R0,TRPMEM           ;AS SELECTED ?
013466  001422                          BEQ     TRPBAK              ;YES, GO BACK
013470  013700  007520          TRPLP:  MOV     TRPSEL,R0           ;NO, TEST LOOP
013474  006100                          ROL     R0                  ;FOR BITS 7&8
013476  006100                          ROL     R0
013500  000300                          SWAB    R0
013502  142700  000371                  BICB    #371,R0             ;CUT OUT SW06
013506  142737  000370  007516          BICB    #370,TRPARG         ;ONLY SWITCHES 7 OR 8 NOW
013514  130037  007516                  BITB    R0,TRPARG           ;ANY SELECTED ?
013520  001005                          BNE     TRPBAK              ;YES
013522  004737  013776                  JSR     PC,RSTREG
013526  062716  000002                  ADD     #2.(SP)             ;NO SCOPE OR LOOP,SO RUN
013532  000002                          RTI                         ;PAST ARGUMENT AND RETURN
013534  004737  013776          TRPBAK: JSR     PC,RSTREG           ;RESTORE REGS
013540  017616  000000                  MOV     @(SP),(SP)          ;RETURN ADDRESS TO STACK
013544  000002                          RTI                         ;EXIT, LOOPING
```

```
                        .SBTTL  SWITCH REGISTER MONITOR
;
;
;       DESCRIPTION:
;
;       SWITCH REGISTER MONITOR.  CALLED BY AN ERROR
;       WITH SWR BIT 15 CLEAR, OR BY TYPING CTRL-G
;       ON THE CONSOLE TELETYPE.  IF USING HARDWARE
;       SWR, SIMPLY ASKS FOR CTRL-C TO CONTINUE.
;       OTHERWISE, IT PRINTS THE CURRENT
;       CONTENTS OF THE SOFTWARE SWITCH REGISTER,
;       FOLLOWED BY A PROMPT (>). THE NEW SWITCH
;       REGISTER SETTINGS SHOULD THEN BE ENTERED AS
;       AN OCTAL NUMBER, TERMINATED BY CARRIAGE RETURN.
;       TYPING CARRIAGE RETURN ALONE WILL CAUSE THE
;       SWITCH REGISTER CONTENTS TO REMAIN UNCHANGED.
;       IF THE SWITCH REGISTER IS UPDATED TO SELECT
;       A TEST (BIT 6 SET) THE NEW TEST WILL BE
;       ENTERED IMMEDIATELY.
;
;       CALLING SEQUENCE:
;
;               JSR     PC,MONIT
;
;       INPUT PARAMETERS:
;
;               BY TYPING CNTRL-G DURING THE RUNNING
;       OF THE TESTS.
;
;
;       IMPLICIT INPUT PARAMETERS:
;
;               THE SOFTWARE SWITCH REGISTER, IF BEING USED
;       MUST HAVE BEEN SET UP.
;
;
;       OUTPUT PARAMETERS:
;
;               IF RUNNING UNDER SOFTWARE SWITCH REGISTER MODE
;       A NEW SETTING OF THE SWR COULD HAVE BEEN SET UP.
;
;
;       IMPLICIT OUTPUT PARAMETERS:
;
;               NONE
;
;
;       COMPLETION CODES:
;
;               NONE
;
;
;       POSSIBLE ERROR CODES:
;
;               NONE
```

```
                                        ;
                                        ;
                                        ;
013546  010046                  MONIT:  MOV     R0,-(SP)        ;SAVE R0
013550  023727  007502  177570          CMP     SWR,#HSWR       ;HARDWARE SWR?
013556  001430                          BEQ     MONITA          ;IF YES, GO TO END
013560  004737  014734                  JSR     PC,TYPOUT       ;SWR=
013564  006623                          SWRMSG
013566  013700  007504                  MOV     SSWR,R0
013572  004737  015024                  JSR     PC,PROCT
013576  012700  000076                  MOV     #76,R0
013602  004737  015012                  JSR     PC,PCHR         ;PRINT '>''
013606  004737  014034                  JSR     PC,OCTIN        ;GET NEW SETTING
013612  005737  007532                  TST     RAND            ;ANY INPUT?
013616  001413                          BEQ     MONITX
013620  010037  007504                  MOV     R0,SSWR         ;YES UPDATE SSWR
013624  032737  000100  007504          BIT     #100,SSWR       ;TEST SELECTED?
013632  001405                          BEQ     MONITX
013634  000137  001326                  JMP     START2          ;YES GO DO IT
013640  004737  013656          MONITA: JSR     PC,TYPCTC       ;CTRL-C TO CONTINUE
013644  000775                          BR      MONITA
013646  004737  014700          MONITX: JSR     PC,CRLF
013652  012600                          MOV     (SP)+,R0        ;RESTORE R0
013654  000207                          RTS     PC
```

```
                                        .SBTTL  WAIT FOR CNTRL-C
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO WAIT FOR THE USER TO TYPE
                                ;       CNTRL-C ON THE CONSOLE. IF CNTRL-O IS HIT MODIFY
                                ;       IS CALLED, AND IF CNTRL-G IS HIT MONIT IS CALLED.
                                ;
                                ;       IF NONE OF THESE ARE HIT, THE ROUTINE WILL RETURN
                                ;       TO THE NEXT LOCATION AFTER THE CALL. IF CNTRL-C WAS
                                ;       HIT THE ROUTINE WILL RETURN TO THE NEXT LOCATION+2.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,TYPCTC
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               INPUT IS REQUESTED FROM THE CONSOLE
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               R0 IS CORRUPTED
                                ;
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               A RETURN TO THE FIRST OR SECOND LOCATION AFTER
                                ;       THE CALL IS PERFORMED.
                                ;
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;
013656  004737  014734          TYPCTC: JSR     PC,TYPOUT       ;PRINTS TYPE CTRL/C WHEN READY
013662  006570                          REDMES
013664  004737  014206                  JSR     PC,READ         ;CTRL/C ENTERED
013670  020027  000003                  CMP     R0,#3
013674  001005                          BNE     QEXIT2          ;NO
```

```
013676  004737  014700                  JSR     PC,CRLF
013702  062716  000002                  ADD     #2,(SP)         ;YES SO JUMP OVER NO FIND RETURN
013706  000207                  QEXIT1: RTS     PC              ;GO BACK
013710  020027  000017          QEXIT2: CMP     R0,#17          ;CNTRL-O ?
013714  001003                          BNE     QEXIT3          ;NO
013716  004737  015450                  JSR     PC,MODIFY       ;YES CALL MODIFY PROGRAM
013722  000755                          BR      TYPCTC          ;THEN GO BACK TO START
013724  020027  000007          QEXIT3: CMP     R0,#7           ;CTRL-G?
013730  001366                          BNE     QEXIT1
013732  004737  013546                  JSR     PC,MONIT        ;GO TO SWR MONITOR
013736  000747                          BR      TYPCTC          ;LOOK FOR CTRL-C AGAIN
```

```
                                        .SBTTL  SAVE REGISTERS
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO SAVE ALL THE GENERAL PURPOSE
                                ;       REGISTERS ON THE STACK, AND LEAVE THE ADDRESS OF THE
                                ;       CALLING ROUTINE ON THE STACK. THE ROUTINE WILL RUN AT
                                ;       PRIORITY 7 TO AVOID ANY INTERRUPTS
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,SAVREG
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               REGISTERS 0 THRU 5 ARE SAVED ON THE STACK
                                ;       AND THE RETURN ADDRESS OF THE CALLING ROUTINE IS
                                ;       SET AS THE LAST ENTRY ON THE STACK
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
013740  012637  007524          SAVREG: MOV     (SP)+,SAVPC     ;SAVE PC FOR RETURN FROM THIS ROUTINE
013744  012637  007526                  MOV     (SP)+,SAVPC1
013750  010546                          MOV     R5,-(SP)
013752  010446                          MOV     R4,-(SP)
013754  010346                          MOV     R3,-(SP)
```

```
013756  010246                          MOV     R2,-(SP)
013760  010146                          MOV     R1,-(SP)
013762  010046                          MOV     R0,-(SP)
013764  013746  007526                  MOV     SAVPC1,-(SP)
013770  013746  007524                  MOV     SAVPC,-(SP)     ;PUT PC READY FOR
013774  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  RESTORE REGISTERS
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               RESTORE TO RESTORE THE GENERAL PURPOSE
                                ;       REGISTERS. THE STACK IS LEFT IN THE SAME STATE AS IT
                                ;       WAS WHEN SAVREG WAS CALLED.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,RSTREG
                                ;
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               R0 THRU R5 RESTORED
                                ;
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;
013776  012637  007524          RSTREG: MOV     (SP)+,SAVPC
014002  012637  007526                  MOV     (SP)+,SAVPC1
014006  012600                          MOV     (SP)+,R0
014010  012601                          MOV     (SP)+,R1
014012  012602                          MOV     (SP)+,R2
014014  012603                          MOV     (SP)+,R3
014016  012604                          MOV     (SP)+,R4
014020  012605                          MOV     (SP)+,R5
014022  013746  007526                  MOV     SAVPC1,-(SP)
014026  013746  007524                  MOV     SAVPC,-(SP)     ;PUT PC READY FOR
```

```
  3548  014032  000207                          RTS     PC              ;RETURN
```

```
                        .SBTTL  ENTER AN OCTAL NUMBER
;
;
;       DESCRIPTION:
;
;               ROUTINE TO ENTER AN OCTAL NUMBER ON THE CONSOLE.
;       THE NUMBER ENTERED IS RETURNED IN R0, AND THE VALUE RAND
;       IS SET TO NON ZERO IF ANY CHARACTERS WERE ENTERED.
;       TYPING CARRIAGE RETURN, LINE FEED OR ESCAPE WILL
;       TERMINATE THE LINE BEING ENTERED.
;       RUBOUT WILL DELETE THE LAST CHARACTER ENTERED,
;       CNTRL-U WILL RUBOUT THE WHOLE LINE, AND CNTRL-R
;       WILL TYPE OUT THE CHARACTERS SO FAR ENTERED.
;       RAND WILL BE SET TO NON ZERO IF ANY CHARACTERS
;       WERE HIT AND RANDC WILL CONTAIN THE TERMINATING
;       CHARACTER.
;
;       A '!' WILL BE PRINTED IF AN OVERFLOW CONDITION IS
;       DETECTED AND A '?' IF AN INVALID CHARACTER WAS
;       ENTERED.
;
;
;       CALLING SEQUENCE:
;
;               JSR     PC,OCTIN
;
;       INPUT PARAMETERS:
;
;               NONE
;
;
;       IMPLICIT INPUT PARAMETERS:
;
;               NONE
;
;
;       OUTPUT PARAMETERS:
;
;               R0 CONTAINS THE NUMBER ENTERED.
;               RAND=0 IF NO CHARACTERS WERE ENTERED.
;
;
;       IMPLICIT OUTPUT PARAMETERS:
;
;               NONE
;
;
;       COMPLETION CODES:
;
;               NONE
;
;
;       POSSIBLE ERROR CODES:
;
;               NONE
```

```
                                        ;
                                        ;
                                        ;
014034  004737  013740          OCTIN:  JSR     PC,SAVREG       ;SAVE THE REGISTERS
014040  005000                  TYPOTB: CLR     R0              ;CLEAR R0
014042  005037  007532                  CLR     RAND            ;CLEAR FLAG WORD
014046  012737  007256  007604          MOV     #OCTMSG,STRADD  ; SET START ADDRESS OF STRING
014054  012737  000006  007606          MOV     #6,STRLEN       ; AND ITS SIZE
014062  012737  000060  007610          MOV     #60,LOWCHR      ; MINIMUM CHAR
014070  012737  000067  007612          MOV     #67,UPPCHR      ; MAXIMUM CHAR
014076  005037  007602                  CLR     CNVFLG          ; DON'T CONVERT TO UPPER CASE
014102  004737  014232                  JSR     PC,GETSTR       ; GET A STRING
014106  013701  007606                  MOV     STRLEN,R1       ; ZERO LENGTH ?
014112  001420                          BEQ     TYPOTD          ; YES
014114  012702  007256                  MOV     #OCTMSG,R2      ; NO GET START ADDRES OF STRING
014120  112203                  TYPOTC: MOVB    (R2)+,R3        ; GET A CHARACTER
014122  162703  000060                  SUB     #60,R3          ; PUT IN RANGE
014126  000241                          CLC
014130  006300                          ASL     R0              ; SHIFT OUT RESULT
014132  103417                          BCS     TYPOTE          ; C BIT SET ERROR
014134  006300                          ASL     R0              ; TIMES 4
014136  103415                          BCS     TYPOTE          ; C BIT ERROR
014140  006300                          ASL     R0              ; TIMES 8
014142  050300                          BIS     R3,R0           ; SET NEW BITS IN
014144  005237  007532                  INC     RAND            ; SET GOOD
014150  005301                          DEC     R1
014152  001362                          BNE     TYPOTC          ; LOOP IF MORE TO COME
014154  010037  007530          TYPOTD: MOV     R0,TYPOTA       ; SAVE FINAL NUMBER
014160  004737  013776                  JSR     PC,RSTREG       ; RESTORE REGISTERS
014164  013700  007530                  MOV     TYPOTA,R0       ; GET RESULT
014170  000207                          RTS     PC
014172  012700  000041          TYPOTE: MOV     #41,R0          ; OVERFLOW
014176  004737  015012                  JSR     PC,PCHR         ; PRINT A !
014202  000137  014040                  JMP     TYPOTB          ; THEN GO AGAIN
                                        ;
```

```
                                        .SBTTL  READ A SINGLE CHARACTER
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO READ A SINGLE CHARACTER
                                ;       FROM THE CONSOLE. THE CHARACTER IS RETURN IN R0
                                ;       AND HAS THE 200 BIT STRIPPED OFF.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,READ
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               R0 CONTAINS THE CHARACTER READ IN.
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               THE CHARACTER IS ECHOED ON THE TERMINAL
                                ;
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;
014206  105737  177560          READ:   TSTB    TKS
014212  100375                          BPL     READ
014214  013700  177562                  MOV     TKB,R0
014220  042700  177600                  BIC     #177600,R0
014224  004737  015012                  JSR     PC,PCHR
014230  000207                          RTS     PC
```

```
                        .SBTTL  ENTERING A CHARACTER STRING
;
;
; DESCRIPTION:
;
;       ROUTINE TO ENTER A STRING OF CHARACTERS ON
;       THE CONSOLE TERMINAL. VARIOUS CONTROL CODES
;       ARE USED TO CONTROL HOW THE CHARACTERS ARE
;       INTERPTETED:
;
;       ESCAPE, CARRIAGE RETURN AND LINE FEED ARE USED
;       TO TERMINATE THE STRING. RUBOUT WILL DELETE THE
;       LAST CHARACTER ENTERED, CNTRL-U WILL DELETE THE
;       ALL THE CHARACTERS ENTERED, AND CNTRL-R WILL ECHO
;       THOSE CHARACTERS ALREADY ENTERED.
;
;       ON ENTRY THE FOLLOWING POINTERS ARE USED:
;       STRADD TO INDICATE THE START OF THE CHARACTER STRING
;       STRLEN TO INDICATE ITS SIZE
;       CVNFLG SET TO CONVERT LOWER CASE TO UPPER CASE
;       UPPCHR TO INDICATE THE HIGHEST CHARACTER CODE
;       LOWCHR TO INDICATE THE LOWEST CHARACTER CODE
;
;       ON EXIT THE  LOCATION STRLEN WILL INDICATE THE
;       NUMBER OF CHARACTERS ENTERED AND RANDC WILL CONTAIN
;       THE TERMINATING CHARACTER
;
;       IF AN INVALID CHARACTER WAS HIT A '?' IS PRINTED
;
;
; CALLING SEQUENCE:
;
;       JSR     PC,GETSTR
;
;
; INPUT PARAMETERS:
;
;       STRADD THE START ADDRESS OF THE STRING
;       STRLEN THE NUMBER OF CHARACTERS TO READ
;       UPPCHR THE HIGHEST CHARACTER CODE ALLOWED
;       LOWCHR THE LOWEST CHARACTER CODE ALLOWED
;       CNVFLG TO INDICATE LOWER TO UPPER CASE CONVERSION
;
;
;
; IMPLICIT INPUT PARAMETERS:
;
;       NONE
;
;
; OUTPUT PARAMETERS:
;
;       STRLEN THE NUMBERS OF CHARACTERS READ
;       RANDC THE TERMINATING CHARACTER
;
;
```

```
                                        ; IMPLICIT OUTPUT PARAMETERS:
                                        ;
                                        ;       NONE
                                        ;
                                        ;
                                        ; COMPELETION CODES:
                                        ;
                                        ;       NONE
                                        ;
                                        ;
                                        ; POSSIBLE ERROR CONDITIONS:
                                        ;
                                        ;
                                        ;       A '?' IS PRINTED IF AN INVALID CHARACTER IS ENTERED
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
014232  004737  013740                  GETSTR: JSR     PC,SAVREG       ;SAVE REGISTERS
014236  005037  007614                  GETCH1: CLR     RUBFLG          ; CLEAR RUBOUT FLAG
014242  013702  007604                          MOV     STRADD,R2       ; GET STARTING POINTER
014246  005001                                  CLR     R1              ; SET INTITAL LENGTH
014250  105737  177560                  GETCH2: TSTB    TKS             ; ANY TTY INPUT
014254  100375                                  BPL     GETCH2          ; NO WAIT FOR IT
014256  013700  177562                          MOV     TKB,R0          ; GET CHARACTER
014262  042700  177600                          BIC     #177600,R0      ; CLEAR RUBBISH
                                        ;
                                        ; CHECK FOR CONTROL CODES
                                        ;
014266  020027  000015                          CMP     R0,#15          ; WAS IT <CR> ?
014272  001002                                  BNE     1$              ; NO
014274  000137  014656                          JMP     CHRFIN
014300  020027  000012                  1$:     CMP     R0,#12          ; WAS IT <LF>
014304  001002                                  BNE     3$              ; NO
014306  000137  014656                          JMP     CHRFIN
014312  020027  000033                  3$:     CMP     R0,#33          ; WAS IT <ESC>
014316  001002                                  BNE     5$              ; NO
014320  000137  014656                          JMP     CHRFIN
014324  020027  000177                  5$:     CMP     R0,#177         ; RUBOUT ?
014330  001002                                  BNE     7$              ; NO
014332  000137  014504                          JMP     RUBCHR          ; YES
014336  020027  000022                  7$:     CMP     R0,#22          ; CNTRL-R
014342  001002                                  BNE     9$              ; NO
014344  000137  014602                          JMP     LINECH          ; YES EXCO LINE
014350  020027  000025                  9$:     CMP     R0,#25          ; CNTRL-U
014354  001002                                  BNE     GETCH3          ; NO
014356  000137  014552                          JMP     LINDEL          ; YES DELETE LINE
                                        ;
                                        ; IT WAS NOT A CONTROL CODE CHECK
                                        ; FOR LEGALITY ?
                                        ;
                                        ;
014362  020027  000140                  GETCH3: CMP     R0,#140         ; IS IT LOWER CASE
014366  002405                                  BLT     3$              ; NO
```

```
014370  005737  007602                  TST     CNVFLG          ; YES DO WE CONVERT TO UPPER
014374  001402                          BEQ     3$              ; NO
014376  042700  000040                  BIC     #40,R0          ; YES STRIP 40 OFF
014402  020037  007610          3$:     CMP     R0,LOWCHR       ; IS THE CHAR TOO SMALL
014406  002002                          BGE     5$              ; NO
014410  000137  014636                  JMP     ILLCHR          ; YES
014414  020037  007612          5$:     CMP     R0,UPPCHR       ; IS THE CHAR TOO BIG ?
014420  003402                          BLE     GETCH4          ; NO
014422  000137  014636                  JMP     ILLCHR          ; YES ITS ILLEGAL
                                ;
                                ; WE HAVE A LEGAL CHARACTER
                                ;
014426  010003                  GETCH4: MOV     R0,R3           ; SAVE CHAR
014430  005737  007614                  TST     RUBFLG          ; IS RUBOUT SET
014434  001406                          BEQ     3$              ; NO
014436  012700  000057                  MOV     #57,R0          ; YES PRINT A '/'
014442  004737  015012                  JSR     PC,PCHR
014446  005037  007614                  CLR     RUBFLG          ; CLEAR RUBOUT FLAG
014452  005201                  3$:     INC     R1              ; UPDATE CHAR COUNT
014454  020137  007606                  CMP     R1,STRLEN       ; END OF STRING SEEN
014460  003403                          BLE     5$              ; NO
014462  005000                          CLR     R0              ; YES, FORCE A NULL TERMINATOR
014464  000137  014656                  JMP     CHRFIN          ; AND COMPLETE
014470  010300                  5$:     MOV     R3,R0           ; RESTORE CHAR
014472  004737  015012                  JSR     PC,PCHR         ; ECHO IT
014476  110022                          MOVB    R0,(R2)+        ; SAVE IT IN BUFFER
014500  000137  014250                  JMP     GETCH2          ; NO GET NEXT ONE
                                ;
                                ; RUBOUT WAS HIT
                                ;
014504  005701                  RUBCHR: TST     R1              ; ANY CHARACTERS TO RUBOUT
014506  001002                          BNE     3$              ; YES
014510  000137  014572                  JMP     LINDL1          ; NO
014514  005737  007614          3$:     TST     RUBFLG          ; IS RUBOUT SET ALREADY ?
014520  001006                          BNE     5$              ; YES
014522  012700  000057                  MOV     #57,R0          ; NO PRINT A '/'
014526  004737  015012                  JSR     PC,PCHR
014532  005237  007614                  INC     RUBFLG          ; SET RUBOUT
014536  114200                  5$:     MOVB    -(R2),R0        ; GET LAST CHAR ENTERED
014540  004737  015012                  JSR     PC,PCHR         ; PRINT IT
014544  005301                          DEC     R1              ; REDUCE COUNT
014546  000137  014250                  JMP     GETCH2          ; GET ANOTHER CHAR
                                ;
                                ; RUBOUT LINE WAS HIT
                                ;
014552  012700  000136          LINDEL: MOV     #136,R0         ; PRINT A ^
014556  004737  015012                  JSR     PC,PCHR
014562  012700  000125                  MOV     #125,R0         ; THEN U
014566  004737  015012                  JSR     PC,PCHR
014572  004737  014700          LINDL1: JSR     PC,CRLF         ; START ON A NEWLINE
014576  000137  014236                  JMP     GETCH1          ; A GO BACK TO BEGINNING
                                ;
                                ; CNTRL-R WAS HIT
                                ;
014602  112712  000100          LINECH: MOVB    #'@,(R2)        ; PUT IN A TERMINATOR
014606  004737  014700                  JSR     PC,CRLF         ; START ON A NEW LINE
```

```
014612  013737  007604  014630          MOV     STRADD,3$       ; SET START ADDRESS
014620  005037  007614                  CLR     RUBFLG          ; CLEAR RUBOUTS
014624  004737  014734                  JSR     PC,TYPOUT       ; PRINT LINE
014630  000000                  3$:     0
014632  000137  014250                  JMP     GETCH2          ; AND GET ANOTHER CHAR
                                ;
                                ; ILLEGAL CHARACTER ENTERED
                                ;
014636  004737  015012          ILLCHR: JSR     PC,PCHR         ; ECHO IT
014642  012700  000077                  MOV     #'?,R0          ; PRINT A ?
014646  004737  015012                  JSR     PC,PCHR
014652  000137  014602                  JMP     LINECH          ; THE ECHO LINE
                                ;
                                ; A TERMINATOR WAS FOUND
                                ;
014656  010037  007616          CHRFIN: MOV     R0,RANDC        ; SAVE TERMINATOR
014662  163702  007604                  SUB     STRADD,R2       ; CALCULATE BYTE COUNT
014666  010237  007606                  MOV     R2,STRLEN       ; SET IT FOR RETURN
014672  004737  013776                  JSR     PC,RSTREG       ; RESTORE REGISTERS
014676  000207                          RTS     PC              ; AND EXIT
```

```
                                        .SBTTL  CARRIAGE RETURN LINE FEED
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO DO A <CR> <LF> ON THE
                                ;       CONSOLE TERMINAL
                                ;
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,CRLF
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               THE CARRIAGE WILL BE PLACED ON A NEW LINE
                                ;
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;
```

3940

```
                        .SBTTL  PRINT AN ASCII MESSAGE
;
;
;
;       DESCRIPTION:
;
;               ROUTINE TO PRINT A STRING OF ASCII
;       CHARACTERS ON THE CONSOLE TERMINAL. CERTAIN
;       CHARACTERS WITHIN THE STRING ARE INTERPRETED
;       AS CONTROL CODES, THESE ARE:
;
;       133 ([)         WILL GENERATE A <CR>,<LF>
;       100 (@)         WILL SIGNIFY END OF MESSAGE
;
;       THE ADDRESS OF THE MESSAGE STRING TO BE PRINTED
;       WILL BE HELD IN THE LOCATION FOLLOWING THE CALL
;       TO THE ROUTINE, IE:
;
;               JSR     PC,TYPOUT
;               ADDR
;
;
;
;       CALLING SEQUENCE:
;
;               JSR     PC,TYPOUT
;
;       INPUT PARAMETERS:
;
;               THE ADDRESS OF THE MESSAGE STRING FOLLOWS
;       THE SUBROUTINE CALL.
;
;
;       IMPLICIT INPUT PARAMETERS:
;
;               NONE
;
;
;       OUTPUT PARAMETERS:
;
;               NONE
;
;
;       IMPLICIT OUTPUT PARAMETERS:
;
;               THE SPECIFIED MESSAGE WILL BE PRINTED
;
;
;       COMPLETION CODES:
;
;               NONE
;
;
;       POSSIBLE ERROR CODES:
;
;               NONE
```

```
                                        ;
                                        ;
                                        ;
014734  004737  013740          TYPOUT: JSR     PC,SAVREG       ;SAVE REGS
014740  017601  000000                  MOV     @(SP),R1        ;R1 POINTS AT STRING
014744  062716  000002                  ADD     #2,(SP)         ;JUMPS OVER ARGUMENT
014750  111100                  PMSG1:  MOVB    @R1,R0          ;PRINT THE MESSAGE POINTED
014752  022700  000100                  CMP     #100,R0         ;TO BY R1 UNTIL @, WHICH IS END.
014756  001412                          BEQ     PMSG4           ;[ MEANS CR-LF
014760  022700  000133                  CMP     #133,R0
014764  001003                          BNE     PMSG2
014766  004737  014700                  JSR     PC,CRLF
014772  000402                          BR      PMSG3
014774  004737  015012          PMSG2:  JSR     PC,PCHR
015000  005201                  PMSG3:  INC     R1
015002  000762                          BR      PMSG1
015004  004737  013776          PMSG4:  JSR     PC,RSTREG       ;RESTORE REGS
015010  000207                          RTS     PC
```

```
                                        .SBTTL  PRINT A CHARACTER
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO PRINT A CHARACTER ON THE
                                ;       CONSOLE. R0 CONTAINS THE CHARACTER TO BE PRINTED
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,PCHR
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               R0 CONTAINS THE CHARACTER TO BE PRINTED
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               THE CHARACTER SELECTED WILL BE PRINTED
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
015012  004737  013276          PCHR:   JSR     PC,TPREDY       ;PRINTER READY
015016  010037  177566                  MOV     R0,TPB
015022  000207                          RTS     PC
```

```
                                        .SBTTL  PRINT AN OCTAL NUMBER
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO PRINT AN OCTAL NUMBER ON
                                ;       THE CONSOLE TERMINAL. R0 CONTAINS THE BINARY
                                ;       REPRESENTATION ON THE NUMBER THAT IS TO BE
                                ;       PRINTED.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,PROCT
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               R0 CONTAINS THE NUMBER THAT IS TO BE
                                ;       PRINTED
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;               NONE
                                ;
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               THE NUMBER SPECIFIED WILL BE PRINTED
                                ;
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;
015024  004737  013740          PROCT:  JSR     PC,SAVREG       ;SAVE REGS
015030  012701  007230                  MOV     #OCTNUM,R1      ;POINTER TO MESSAGE
015034  112721  000040          PROCT1: MOVB    #40,(R1)+       ;FILL WITH SPACES
015040  020127  007236                  CMP     R1,#OCTNUM+6    ;ALL DONE
015044  001373                          BNE     PROCT1          ;NO
015046  000241                  PRCT1A: CLC                     ;CLEAR AT START
015050  010002                  PROCT2: MOV     R0,R2   ;SAVE CHARA
015052  042702  177770                  BIC     #177770,R2      ;CLEAR ALL BUT BOTTOM 3 BITS
015056  062702  000060                  ADD     #60,R2          ;NOW ASCII
```

```
015062  110241                          MOVB    R2,-(R1)          ;STORE IT IN MESSAGE
015064  042700  000007                  BIC     #7,R0             ;NOW CLEAR BOTTOM 3 BITS
015070  001404                          BEQ     PROCT3            ;ALL DONE
015072  006000                          ROR     R0                ;ROTATE NEXT 3 BITS
015074  006000                          ROR     R0
015076  006000                          ROR     R0
015100  000763                          BR      PROCT2            ;GO DO NEXT
015102  004737  014734          PROCT3: JSR     PC,TYPOUT         ;TYPE MESSAGE
015106  007230                          OCTNUM                    ;MESSAGE ADD.
015110  004737  013776                  JSR     PC,RSTREG                 ;RESTORE REGS
015114  000207                          RTS     PC
                                        ;
                                        ;
                                        ;ROUTINE TO PRINT LEAST SIG.9 BITS OF R0

015116  004737  013740          PRNT3:  JSR     PC,SAVREG
015122  042700  177000                  BIC     #177000,R0        ;CLEAR UNWANTER
015126  012737  002040  007230          MOV     #2040,OCTNUM      ;SET UP MESSAGE
015134  012737  040040  007232          MOV     #40040,OCTNUM+2   ;WITH END CODE
015142  012701  007233                  MOV     #OCTNUM+3,R1      ;& POINTER
015146  000737                          BR      PRCT1A            ;JUMP INTO MAIN ROUTINE
                                                                  ;WITH POINTERS SET UP
```

```
                                        .SBTTL  ENTER A DECIMAL NUMBER
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO ENTER A DECIMAL NUMBER ON
                                ;       THE CONSOLE TERMINAL. R0 WILL CONTAIN THE NUMBER
                                ;       ENTERED AS A BINARY VALUE. RAND WILL BE NON ZERO
                                ;       IF ANY CHARACTERS WERE READ IN. THE INCOMING STRING
                                ;       CAN BE TERMINATED BY <CR>, <LF> OR <ESC>.
                                ;       TYPING CNTRL-U WILL DELETE ALL CHARACTERS ENTERED,
                                ;       CNTRL-R WILL ECHO THE CHARACTERS ENTERED AND RUBOUT
                                ;       WILL DELETE THE LAST CHARACTER ENTERED
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,TYPDEC
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               R0 CONTAINS THE BINARY REPRESENTATION
                                ;       OF THE DECIMAL NUMBER ENTERED, AND RAND WILL
                                ;       INDICATE WHETHER ANY CHARACTERS HAVE BEEN ENTERED.
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
015150  004737  013740          TYPDEC: JSR     PC,SAVREG       ;SAVE REGISTERS
015154  005000                  TPDEC1: CLR     R0
015156  005037  007532                  CLR     RAND            ;CLEAR INDICATOR
```

```
015162  012737  007250  007604          MOV     #DECMSG,STRADD  ; SET START ADDRESS
015170  012737  000005  007606          MOV     #5,STRLEN       ; ONLY 5 CHARS
015176  012737  000060  007610          MOV     #60,LOWCHR      ; MINIMUM CHAR
015204  012737  000071  007612          MOV     #71,UPPCHR      ; MAXIMUM CHAR
015212  005037  007602                  CLR     CNVFLG          ; DON'T CONVERT TO UPPER
015216  004737  014232                  JSR     PC,GETSTR       ; GET A STRING
015222  013704  007606                  MOV     STRLEN,R4       ; GET LENGHT READ IN
015226  001433                          BEQ     TPDEC5          ; ZERO MEANS COMLPETE
015230  012702  007250                  MOV     #DECMSG,R2      ; POINT TO STRING
015234  112201                  TPDEC2: MOVB    (R2)+,R1        ; GET A CHARACTER
015236  000241                          CLC
015240  006300                          ASL     R0              ; RESULT TIMES 2
015242  103413                          BCS     TPDEC3          ; C BIT SET MEANS ERROR
015244  010003                          MOV     R0,R3           ; ASVE IT
015246  006300                          ASL     R0              ; TIMES 4
015250  103410                          BCS     TPDEC3          ; C BIT MEANS ERROR
015252  006300                          ASL     R0              ; TIMES 8
015254  103406                          BCS     TPDEC3          ; C BIT MEANS OVERFLOW
015256  060300                          ADD     R3,R0           ; ADD IN TO BE TIMES 12
015260  103404                          BCS     TPDEC3          ; C BIT MEANS ERROR
015262  042701  177760                  BIC     #177760,R1      ; CLEAR UNWANTED BITS
015266  060100                          ADD     R1,R0           ; ADD IN NEW VALUE
015270  103006                          BCC     TPDEC4          ; C BIT
015272  012700  000041          TPDEC3: MOV     #41,R0          ; OVERFLOW PRINT A !
015276  004737  015012                  JSR     PC,PCHR
015302  000137  015154                  JMP     TPDEC1          ; AND GO BAK TO START
015306  005237  007532          TPDEC4: INC     RAND            ; SET CHAR SEEN
015312  005304                          DEC     R4              ; REDUCE COUNT
015314  001347                          BNE     TPDEC2          ; NON ZERO MEANS MORE
015316  010037  007534          TPDEC5: MOV     R0,TYPD1        ; ALL DONE SAVE RESULT
015322  004737  013776                  JSR     PC,RSTREG       ; RESTORE REGISTERS
015326  013700  007534                  MOV     TYPD1,R0        ; RESTORE RESULT
015332  000207                          RTS     PC              ; AND EXIT
                                ;
```

```
                                        .SBTTL  PRINT A DECIMAL NUMBER
                                ;
                                ;       DESCRIPTION:
                                ;               ROUTINE TO PRINT A SIGNED OR UNSIGNED
                                ;       DECIMAL NUMBER ON THE CONSOLE. R0 HOLDS THE
                                ;       NUMBER TO BE PRINTED.
                                ;
                                ;       CALLING SEQUENCE:
                                ;               JSR     PC,BASE10       ; FOR UNSIGNED
                                ;               JSR     PC,BASM10       ; FOR SIGNED
                                ;
                                ;       INPUT PARAMETERS:
                                ;               R0 HOLDS THE NUMBER TO BE PRINTED
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;               NONE.
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;               NONE
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;               THE NUMBER IN R0 IS PRINTED.
                                ;
                                ;       COMPLETION CODES:
                                ;               NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;               NONE
                                ;
                                ;
015334  005700                  BASM10: TST     R0              ; GET SIGN OF R0
015336  100005                          BPL     BASE10          ; POSITIVE..
015340  005400                          NEG     R0              ; IF MINUS,MAKE POSITIVE..
015342  112737  000055  007240          MOVB    #55,BASE11      ; AND PRINT '-'
015350  000403                          BR      BAS10A          ; THEN CONTINUE..
                                ;
015352  112737  000040  007240  BASE10: MOVB    #' ,BASE11      ; PRINT LEADING SPACE
                                ;
015360  004737  013740          BAS10A: JSR     PC,SAVREG       ; NOW SAVE OLD REGS..
015364  012703  007241                  MOV     #BASE11+1,R3    ; SETUP OUTPUT POINTER..
015370  112723  000040          BASE1A: MOVB    #' ,(R3)+       ; PRELOAD STRING WITH SPACES.
015374  022703  007246                  CMP     #BASE11+6,R3    ; ALL DONE?
015400  001373                          BNE     BASE1A          ; NO..
                                ;
015402  005001                  BASE1D: CLR     R1              ; R1 IS RECEIVER...
                                ;
015404  020027  000012          BASE1B: CMP     R0,#12          ; MORE THAN 10?
015410  103404                          BLO     BASE1C          ; YES IF BRANCH
015412  162700  000012                  SUB     #12,R0          ; DIV BY REPEAT SUB
015416  005201                          INC     R1              ; TALLY COUNT
015420  000771                          BR      BASE1B          ; DO NEXT
                                ;
015422  062700  000060          BASE1C: ADD     #'0,R0          ; MAKE INTO NUMBER
015426  110043                          MOVB    R0,-(R3)        ; SAVE IN STRING
015430  010100                          MOV     R1,R0
```

```
015432  001363                  BNE     BASE1D
015434  004737  013776          JSR     PC,RSTREG       ; RESTORE REGISTERS..
015440  004737  014734          JSR     PC,TYPOUT       ; TYPE MESSAGE
015444  007240                  BASE11                  ; THIS ONE...
015446  000207                  RTS     PC              ; AND EXIT
```

```
                                        .SBTTL  MODIFY THE PROGRAM
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;       ROUTINE TO MODIFY A LOCATION IN MEMORY
                                ;
                                ;       ENTERED BY TYPING CNTRL-O ON THE CONSOLE
                                ;       TERMINAL.
                                ;
                                ;
                                ;       PROMPTS ($) FOR AN ADDRESS TO EXAMINE
                                ;       AND PRINTS IT IN THE FORM
                                ;
                                ;       ADDR CONTENTS /
                                ;
                                ;       THEN A NEW VALUE CAN BE ENTERED
                                ;
                                ;       VIS:
                                ;
                                ;       ADDR CONTENTS / NEW VALUE
                                ;
                                ;
                                ;       THE NEW VALUE CAN BE TERMINATED USING
                                ;       <CR>, <LF> , OR <ESC>
                                ;
                                ;
                                ;
                                ;       <LF>    TO EXAMINE THE NEXT LOC
                                ;       <CR>    TO SELECT ANOTHER ADDRESS
                                ;       <ESC>   TO EXIT
                                ;
                                ;
                                ;
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,MODIFY
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               ENTERED BY TYPING CNTRL-O DURING
                                ;       THE RUNNING OF THE TESTS
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
```

```
                                        ;
                                        ;       IMPLICIT OUTPUT PARAMETERS:
                                        ;
                                        ;               THE LOCATIONS SPECIFIED WILL HAVE BEEN
                                        ;       MODIFIED
                                        ;
                                        ;
                                        ;       COMPLETION CODES:
                                        ;
                                        ;               NONE
                                        ;
                                        ;
                                        ;       POSSIBLE ERROR CODES:
                                        ;
                                        ;               NONE
                                        ;
                                        ;
015450  004737  013740          MODIFY: JSR     PC,SAVREG       ;SAVE REGISTERS
015454                                  PSWREA  MODSAV
015464                                  PSWSET  #340
015500  004737  014734          MODI1:  JSR     PC,TYPOUT       ;PROMPT $ FOR AN ADDRESS
015504  006635                          MODADM                  ;TO HAVE A LOOK AT
015506  004737  014034                  JSR     PC,OCTIN        ;READ REPLY
015512  005737  007532                  TST     RAND            ;ANYTHONG READ ?
015516  001506                          BEQ     MODXIT          ;NO, SO EXIT
015520  010037  007536                  MOV     R0,MODADR       ;ELSE SAVE OUR ADDRESS
015524  032737  000001  007536  MODI2:  BIT     #1,MODADR       ;IS IT EVEN ?
015532  001404                          BEQ     MODI3           ;YES WE CAN USE IT
015534  004737  014734                  JSR     PC,TYPOUT       ;ELSE SAY IT IS AN ODD
015540  006707                          ODAMSG                  ;ADDRESS, AND REPROMPT
015542  000756                          BR      MODI1
015544  012737  016354  000004  MODI3:  MOV     #NXMTRP,4       ;PLUG TRAP THRU 4
015552  012737  000340  000006          MOV     #340,6          ;FOR NXM TESTS
015560  005037  007544                  CLR     TRPERR          ;CLEAR NXM FLAG
015564  005777  171746                  TST     @MODADR         ;TEST OUR ADDRESS
015570  005737  007544                  TST     TRPERR          ;DOES IT EXIST
015574  001341                          BNE     MODI1           ;NO TRY AGAIN
015576  004737  014700                  JSR     PC,CRLF         ;START ON A NEW LINE
015602  013700  007536                  MOV     MODADR,R0       ;PRINT OUR ADDRESS
015606  004737  015024                  JSR     PC,PROCT
015612  012700  000040                  MOV     #40,R0          ;THEN A SPACE
015616  004737  015012                  JSR     PC,PCHR         ;AS A SEPARATOR
015622  017700  171710                  MOV     @MODADR,R0      ;THEN PRINT THE CONTENTS
015626  004737  015024                  JSR     PC,PROCT
015632  004737  014734                  JSR     PC,TYPOUT       ;PROMPT FOR THE NEW
015636  006642                          MODSPA                  ;CONTENTS '/'
015640  004737  014034                  JSR     PC,OCTIN        ;READ REPLY
015644  005737  007532                  TST     RAND            ;ANY THING GIVEN
015650  001402                          BEQ     MODI4           ;NO, DON'T UPDATE
015652  010077  171660                  MOV     R0,@MODADR      ;ELSE SET NEW VALUE
015656  013700  007616          MODI4:  MOV     RANDC,R0        ; GET TERMINATOR
015662  001706                          BEQ     MODI1           ; NULL MEANS <CR>
015664  020027  000015                  CMP     R0,#15          ;WAS IT <CR> ?
015670  001703                          BEQ     MODI1           ;YES GET NEXT ADDRESS
015672  020027  000012                  CMP     R0,#12          ;WAS IT <LF> ?
015676  001006                          BNE     MODI5           ;NO
```

```
015700  004737  014700                  JSR     PC,CRLF         ; NEWLINE
015704  062737  000002  007536          ADD     #2,MODADR       ;UPDATE THE ADDRESS BY 2
015712  000704                          BR      MODI2           ; THEN TRY THIS ADDRESS
015714  020027  000033          MODI5:  CMP     R0,#33          ;WAS IT EXIT ?
015720  001405                          BEQ     MODXIT          ;YES DISSAPPEAR
015722  012700  000077                  MOV     #77,R0          ;ELSE GIVE A ?
015726  004737  015012                  JSR     PC,PCHR         ;AS AN ERROR
015732  000751                          BR      MODI4           ;THEN TRY AGAIN
015734  012737  000006  000004  MODXIT: MOV     #6,4            ;PLUG BACK TRAP THRU 4
015742  012737  000004  000006          MOV     #4,6            ;WITH WHAT IS WAS
015750  004737  014700                  JSR     PC,CRLF         ;PUT OURSELVES ON A NEW LINE
015754                                  PSWSET  MODSAV
015770  004737  013776                  JSR     PC,RSTREG       ;RESTORE REGISTERS
015774  000207                          RTS     PC              ;AND GO AWAY
```

```
                                        .SBTTL  SELECT BUS ADDRESSES
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               THIS ROUTINE PROMPTS THE OPERATOR FOR THE
                                ;       BUS ADDRESS OF THE FIRST VTV30K ON THE SYSTEM, AND ALSO
                                ;       THE NUMBER OF DEVICES THAT ARE ON THE SYSTEM. THE PROGRAM
                                ;       WILL ALLOW UP TO TWENTY (DECIMAL) DEVICES TO BE SELECTED,
                                ;       HOWEVER THE DEVICES MUST BE CONFIGURED IN ASCENDING ORDER.
                                ;       FOR EXAMPLE: THE FIRST DEVICE WOULD BE AT 1XXX00, 1XXX02,
                                ;       THEN THE SECOND ONE WOULD BE AT 1XXX10, AND 1XXX12, ETC.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR PC,SELBUS
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               THE ADDRESSES OF THE CSR AND DATA BUFFERS SELECTED
                                ;       WILL BE LOADED INTO LOCATIONS FOLLOWING THE ADDRESS ''CSR0''.
                                ;       THE LOCATION ''DEVCNT'' IS SET TO SAY HOW MANY DEVICES WERE
                                ;       SELECTED.
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               EACH OF THE ADDRESSES SELECTED ARE TESTED FOR NXM,
                                ;       AND AN ERROR REPORT IS GENERATED IF THE SELECTED ADDRESS
                                ;       REPORTS AS NXM. ERROR 77 FOR A NON EXISTANT CSR AND 76 FOR
                                ;       THE DATA BUFFER.
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               ERROR 77 WILL OCCUR IF A SELECTED CSR IS NOT ON
                                ;       THE BUS. ERROR 76 WILL OCCUR FOR A NON EXISTANT DATA BUFFER
                                ;       ADDRESS. SETTING TRAP+10 WILL LOOP ON THIS ADDRESS.
```

```
                                        ;
015776  005037  007454          SELBUS: CLR     TESTNO
016002  004737  013740                  JSR     PC,SAVREG               ; SAVE REGISTERS
016006  012737  016354  000004          MOV     #NXMTRP,4               ; PLUG TRAP THRU 4
016014  012737  000340  000006          MOV     #340,6
016022  004737  014734          1$:     JSR     PC,TYPOUT               ; ASK FOR NUMBER OF DEVICES
016026  005660                          NUMDEV
016030  004737  015150                  JSR     PC,TYPDEC               ; READ DECIMAL REPLY
016034  005737  007532                  TST     RAND                    ; NULL REPLY MEANS 1
016040  001002                          BNE     2$
016042  012700  000001                  MOV     #1,R0
016046  020027  000024          2$:     CMP     R0,#20.                 ; IS THE NUMBER TOO BIG ?
016052  003363                          BGT     1$                      ; YES
016054  005700                          TST     R0                      ; OR TOO SMALL
016056  003761                          BLE     1$                      ; OOPS, REPROMPT
016060  010037  007304                  MOV     R0,DEVCNT               ; ELSE SAVE COUNT AWAY
016064  023727  007304  000001          CMP     DEVCNT,#1               ; ONE DEVICE SELECTED
016072  001406                          BEQ     BUSSE1                  ; YES
016074  004737  014734                  JSR     PC,TYPOUT
016100  006144                          GOMS1                           ; NO- ALL DEVICES MUST BE
016102  004737  013656          3$:     JSR     PC,TYPCTC               ; SEQUENTIAL
016106  000775                          BR      3$
                                        ;
                                        ; NOW GET OUR FIRST ADDRESS
                                        ;
016110  004737  014734          BUSSE1: JSR     PC,TYPOUT               ; FIRST BUS ADDRESS =
016114  006653                          BAMSG
016116  004737  014034          1$:     JSR     PC,OCTIN                ; READ OCTAL REPLY
016122  005737  007532                  TST     RAND                    ; DON'T ALLOW DEFAULTS
016126  001004                          BNE     2$
016130  004737  014734                  JSR     PC,TYPOUT
016134  006762                          NODEFM
016136  000767                          BR      1$
016140  032700  000005          2$:     BIT     #5,R0                   ; THESE BITS CLEAR
016144  001406                          BEQ     3$                      ; YES
016146  004737  014734          4$:     JSR     PC,TYPOUT
016152  006707                          ODAMSG                          ; NO ITS AN INVALID ADDRESS
016154  004737  015024                  JSR     PC,PROCT
016160  000753                          BR      BUSSE1
016162  032700  100000          3$:     BIT     #100000,R0              ; IT MUST BE ON THE IO PAGE
016166  001767                          BEQ     4$
016170  010037  007542          BUSSE2: MOV     R0,BASADD               ; SAVE START ADDRESS
016174  013701  007304                  MOV     DEVCNT,R1               ; GET DEVICE COUNT
016200  012702  007306                  MOV     #CSR0,R2                ; AND WHERE DATA GOES
016204  013737  007542  007566          MOV     BASADD,DATA
016212  005037  007544          1$:     CLR     TRPERR                  ; CLEAR NXM FLAG
016216  005777  171344                  TST     @DATA                   ; CHECKIT
016222  005737  007544                  TST     TRPERR                  ; OK ?
016226  001411                          BEQ     2$                      ; YES
                                        ; ERROR-
                                        ; CHECK 'LEVB CLOCK H'
                                        ;       'LEVB VA02 H'
                                        ; BUS CYCLE TIMING ON LEVA.
                                        ; 8136 ADDRESS COMPARATORS ON LEVA
016230  013737  007566  007572          MOV     DATA,ADDRES
016236  004737  016666                  JSR     PC,ERROR                ; NO- THIS IS NAUGHTY
```

```
016242  020077                          77+A
016244  104410                          TRAP+10
016246  016212                          1$
016250  000717                          BR      BUSSET
016252  013722  007566          2$:     MOV     DATA,(R2)+          ; SAVE DATA AWAY
016256  062737  000002  007566          ADD     #2,DATA             ; POINT TO DBUFF ADDRESS
016264  005037  007544          3$:     CLR     TRPERR              ; CLEAR NXM FLAG
016270  005777  171272                  TST     @DATA               ; CHECK FOR NXM
016274  005737  007544                  TST     TRPERR              ; IS IT ?
016300  001403                          BEQ     4$                  ; NO- ALL IS WELL
                                ; ERROR-
                                ; CHECK E4 ON LEVA
                                ; BUS CYCLE TIMING ON LEVA
016302  004737  016666                  JSR     PC,ERROR            ; OOPS- NAUGHTY
016306  040076                          76+D
016310  104410                  4$:     TRAP+10                     ; LOOP ON
016312  016264                          3$
016314  013722  007566                  MOV     DATA,(R2)+          ; SAVE DBUFF ADDRESS
016320  062737  000006  007566          ADD     #6,DATA             ; POINT TO NEXT CSR
016326  005301                          DEC     R1                  ; ALL CSRS DONE
016330  001330                          BNE     1$                  ; NO
016332  012737  000006  000004          MOV     #6,4                ; RESET TRAP POINTERS
016340  012737  000004  000006          MOV     #4,6                ; WITH THE IOT TRAP
016346  004737  013776                  JSR     PC,RSTREG
016352  000207                          RTS     PC                  ; YES EXIT
                                ;
```

```
4566  016354  004737  014734          NXMTRP: JSR     PC,TYPOUT
4567  016360  006732                          NXMSG
4568  016362  005237  007544                  INC     TRPERR
4569  016366  000002                          RTI
```

```
                                        .SBTTL  NUMBER GENERATOR
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               ROUTINE TO GENERATE DATA PATTERNS,
                                ;       THE TYPE OF PATTERN IS SELECTED BY R3, AND THE
                                ;       PATTERN GENERATED IS RETURNED IN R0 AND LOCATION
                                ;       GOOD.
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,GENER
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;       R3 CONTAINS THE PATTERN NUMBER
                                ;
                                ;       R3=0            ALL ZEROES
                                ;       1               ALL ONES
                                ;       2               010101 ETC BIT PATTERN
                                ;       3               101010 ETC BIT PATTERN
                                ;       4               ROTATING 1 IN A ZERO WORD
                                ;       5               ROTATING 0 IN AN ALL ONE WORD
                                ;       6               PSEUDO RANDOM NUMBER
                                ;       7               INCREMENTING DATA PATTERN, GOOD
                                ;                       CONTAINS THE VALUE TO BE UPDATED
                                ;
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               THE NUMBER GENERATED IS HELD IN
                                ;       R0 AND GOOD.
                                ;
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
```

```
                                        ;
                                        ;
                                        ;
016370  042703  177770          GENER:  BIC     #177770,R3
016374  004737  013740                  JSR     PC,SAVREG
016400  006303                          ASL     R3
016402  000173  016406                  JMP     @GENSEL(R3)
016406  016426                  GENSEL: GEN0                    ;ALL ZERO WORD
016410  016432                          GEN1                    ;ALL ONE WORD
016412  016440                          GEN52                   ;52 PATTERN
016414  016446                          GEN25                   ;25 PATTERN
016416  016454                          GENR1                   ;ROTATE '1' EACH CALL
016420  016464                          GENR0                   ;ROTATE '0' EACH CALL
016422  016522                          GENRAN                  ;RANDOM NUMBER
016424  016642                          GENINC                  ;INCREMENTING COUNT
016426  005000                  GEN0:   CLR     R0              ;0>R0
016430  000507                          BR      GENEX
016432  005000                  GEN1:   CLR     R0              ;NOT0>R0
016434  005100                          COM     R0
016436  000504                          BR      GENEX
016440  012700  052525          GEN52:  MOV     #52525,R0       ;5252>R0
016444  000501                          BR      GENEX
016446  012700  125252          GEN25:  MOV     #125252,R0      ;125252>R0
016452  000476                          BR      GENEX
016454  000241                  GENR1:  CLC
016456  004737  016476                  JSR     PC,GENROT       ;SHIFT 1 > R0
016462  000472                          BR      GENEX
016464  000241                  GENR0:  CLC
016466  004737  016476                  JSR     PC,GENROT       ;
016472  005100                          COM     R0              ;SHIFT 0 > R0
016474  000465                          BR      GENEX
016476  006037  016520          GENROT: ROR     GENISH          ;ROTATE  1 PATTERN
016502  001003                          BNE     GENER1          ;= 0?
016504  012737  100000  016520          MOV     #100000,GENISH  ;YES, SET MSB
016512  013700  016520          GENER1: MOV     GENISH,R0       ;PUT 1 IN R0
016516  000207                          RTS     PC              ;AND EXIT
016520  000001                  GENISH: 1
016522  012737  000005  007554  GENRAN: MOV     #5,RANSEL       ;SET SELECT VALUE TO 5
016530  004737  016542                  JSR     PC,RANGEN       ;GENERATE RANDOM NUMBER IN R0
016534  013700  007552                  MOV     RANDN,R0        ;
016540  000443                          BR      GENEX           ;
016542  013702  007552          RANGEN: MOV     RANDN,R2        ;
016546  001002                          BNE     RAN1            ;IS RANDOM = 0
016550  013702  007560                  MOV     RANST,R2        ;YES, PUT RANDOM START VALUE IN
016554  032737  000777  007554  RAN1:   BIT     #777,RANSEL     ;NO;IS RANSEL SELECT VALUE = 0
016562  001003                          BNE     RAN2            ;NO
016564  012737  000001  007554          MOV     #1,RANSEL       ;YES: SET RANSEL = 1
016572  013703  007554          RAN2:   MOV     RANSEL,R3       ;
016576  013702  007552                  MOV     RANDN,R2
016602  033702  007556                  BIT     RANMTA,R2       ;GET R2 <0 AND 1>
016606  001405                          BEQ     RANCLC          ;
016610  005102                          COM     R2
016612  033702  007556                  BIT     RANMTA,R2
016616  001401                          BEQ     RANCLC
016620  000402                          BR      RANSEC
016622  000241                  RANCLC: CLC
```

```
016624  000401                          BR      RAN4
016626  000261                  ..ANSEC: SEC
016630  006037  007552          RAN4:   ROR     RANDN           ;ROTATE C TO B15
016634  005303                          DEC     R3              ;IS THIS NUMBER REQUIRED?
016636  001357                          BNE     RAN2+4          ;NO, GET ANOTHER
016640  000207                          RTS     PC              ;YES, EXIT
016642  013700  007562          GENINC: MOV     GOOD,R0         ;INCREMENTS LOC. 'GOOD'
016646  005200                          INC     R0
016650  010037  007562          GENEX:  MOV     R0,GOOD
016654  010066  000002                  MOV     R0,2(SP)        ;LOAD R0 MAINROUTINE
016660  004737  013776                  JSR     PC,RSTREG
016664  000207                          RTS     PC
```

```
                                        .SBTTL  PRINT  ERROR MESSAGES
                                ;
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;       ROUTINE TO PRINT ERROR MESSAGES
                                ;       IF BIT 14 IN THE SWR IS SET NO MESSAGES WILL BE
                                ;       PRINTED, IF BIT 15 IS SET THE PROGRAM WILL NO WAIT
                                ;       AFTER AN ERROR HAS BEEN PRINTED. IT IS CALLED THUS:
                                ;
                                ;       JSR     PC,ERROR
                                ;       ARG
                                ;
                                ;       WHERE ARG CONTAINS THE ERROR CODE AND IS OF THE FORM
                                ;       X+N, WHERE N IS THE ERROR NUMBER IN THE RANGE 0-177
                                ;       AND X IS A COMBINATION OF FLAGS THAT INDICATE WHAT
                                ;       VALUES ARE TO BE PRINTED. THESE VALUES SHOULD BE LOADED
                                ;       BEFORE THE ERROR ROUTINE IS CALLED AND ARE DEFINED
                                ;       AS FOLLOWS:
                                ;
                                ;       FLAG SETTING    NUMBER  LOCATION        MESSAGE
                                ;       ------------    ------  --------        -------
                                ;
                                ;       C               4000    CALLPC          CALLED FROM
                                ;       S               10000   STATUS          STATUS
                                ;       A               20000   ADDRES          ADDRESS
                                ;       D               40000   DATA            DATA
                                ;       G               100000  GOOD, BAD       GOOD= BAD =
                                ;
                                ;       IN ADDITION THE ERROR NUMBER WILL BE COMBINED WITH
                                ;       THE TEST NUMBER TO INDICATE IN WHICH TEST THE ERROR
                                ;       OCCURRED.
                                ;       AN ERROR COUNT IS MAINTAINED AN ON EACH ERROR THE
                                ;       COUNT IS UPDATED. IF RUNNING UNDER A SOFTWARE SWITCH
                                ;       REGISTER, IT IS POSSIBLE TO SELECT NEW OPTIONS
                                ;
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,ERROR
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               THE LOACTION FOLLOWING THE CALL
                                ;       CONTAINS THE ERROR CODE AND FLAG SETTINGS
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
```

```
                                        ;               NONE
                                        ;
                                        ;       IMPLICIT OUTPUT PARAMETERS:
                                        ;
                                        ;               THE APPROPRAITE ERROR MESSAGE WILL BE PRINTED
                                        ;       IF PERMITTED.
                                        ;
                                        ;       COMPLETION CODES:
                                        ;
                                        ;               NONE
                                        ;
                                        ;       POSSIBLE ERROR CODES:
                                        ;
                                        ;               NONE
                                        ;
                                        ;
                                        ;               20-FEB-80
016666  004737  013740                  ERROR:  JSR     PC,SAVREG
016672  012737  000001  007444                  MOV     #1,ERRFLG       ; SET ERROR FLAG
016700  005237  007574                          INC     ERRDIS          ;INC COUNT
016704  001002                                  BNE     1$              ; CHECK FOR OVERFLOW
016706  005237  007442                          INC     ERRDI1          ; THEN INC OVERFLOW COUNT
016712  032777  040000  170562          1$:     BIT     #40000,@SWR     ;SUPPRESS PRINTOUT
016720  001133                                  BNE     ERHALT                          ;YES
016722  017637  000000  007576                  MOV     @(SP),ERRARG    ;NO
016730  004737  014734                          JSR     PC,TYPOUT       ;TYPE MESSAGE
016734  007064                                  EMSG1
016736  013700  007454                          MOV     TESTNO,R0
016742  000300                                  SWAB    R0
016744  006300                                  ASL     R0
016746  153700  007576                          BISB    ERRARG,R0
016752  004737  015024                          JSR     PC,PROCT
016756  004737  014734                          JSR     PC,TYPOUT
016762  007070                                  EMSG2
016764  011600                                  MOV     (SP),R0
016766  004737  015024                          JSR     PC,PROCT                        ;PRINT 6 DIGITS
016772  132737  000200  007577                  BITB    #200,ERRARG+1
017000  001416                                  BEQ     ERRSV1
017002  004737  014734                          JSR     PC,TYPOUT
017006  007102                                  EMSG3
017010  013700  007562                          MOV     GOOD,R0
017014  004737  015024                          JSR     PC,PROCT
017020  004737  014734                          JSR     PC,TYPOUT
017024  007113                                  EMSG4
017026  013700  007564                          MOV     BAD,R0
017032  004737  015024                          JSR     PC,PROCT
017036  132737  000100  007577  ERRSV1:         BITB    #100,ERRARG+1   ;D SET ?
017044  001407                                  BEQ     ERRSV2          ;NO
017046  004737  014734                          JSR     PC,TYPOUT
017052  007124                                  EMSG5
017054  013700  007566                          MOV     DATA,R0
017060  004737  015024                          JSR     PC,PROCT
```

```
017064  132737  000040  007577  ERRSV2: BITB    #40,ERRARG+1     ;A SET ?
017072  001407                          BEQ     ERRSV3           ;NO
017074  004737  014734                  JSR     PC,TYPOUT
017100  007135                          EMSG6
017102  013700  007572                  MOV     ADDRES,R0
017106  004737  015024                  JSR     PC,PROCT
017112  132737  000020  007577  ERRSV3: BITB    #20,ERRARG+1     ;S SET ?
017120  001407                          BEQ     ERRSV4           ;NO
017122  004737  014734                  JSR     PC,TYPOUT
017126  007153                          EMSG7
017130  013700  007570                  MOV     STATUS,R0
017134  004737  015024                  JSR     PC,PROCT
017140  132737  000010  007577  ERRSV4: BITB #10,ERRARG+1        ;C SET
017146  001407                          BEQ     ERRSV5           ;NO
017150  004737  014734                  JSR     PC,TYPOUT
017154  007166                          EMSG8
017156  013700  007600                  MOV     CALLPC,R0
017162  004737  015024                  JSR     PC,PROCT
017166  004737  014700          ERRSV5: JSR     PC,CRLF
017172  004737  014734                  JSR     PC,TYPOUT
017176  007205                          EMSG9
017200  013700  007574                  MOV     ERRDIS,R0
017204  004737  015352                  JSR     PC,BASE10
017210  032777  100000  170264  ERHALT: BIT     #100000,@SWR
017216  001004                          BNE     NOHALT
017220  013700  007574                  MOV     ERRDIS,R0
017224  004737  013546                  JSR     PC,MONIT         ; GO TO SWR
017230  004737  013776          NOHALT: JSR     PC,RSTREG
017234  062716  000002                  ADD     #2,(SP)
017240  000207                          RTS     PC
```

```
                                        .SBTTL  NORMAL CHARACTER SET.
                                        .NLIST  BEX
                                ;THIS IS THE VTV30K CHARACTER SET IN ITS MAIN
                                ;VERSION, WITH A POUND SIGN AT ADDRESS 13,& A
                                ;DEGREE SIGN AT ADDRESS 32.

017242    360     360     360   NORCH:  .BYTE   360,360,360,360,360,360,360,360
017252    010     010     010           .BYTE   010,010,010,010,010,010,010,010
017262    030     030     030           .BYTE   030,030,030,030,030,030,030,030
017272    074     074     074           .BYTE   074,074,074,074,074,074,074,074
017302    000     000     000           .BYTE   000,000,000,000,377,000,000,000
017312    000     000     000           .BYTE   000,000,000,000,377,377,000,000
017322    000     000     377           .BYTE   000,000,377,377,377,377,000,000
017332    010     010     010           .BYTE   010,010,010,010,377,010,010,010
017342    030     030     030           .BYTE   030,030,030,377,377,030,030,030
017352    074     074     377           .BYTE   074,074,377,377,377,377,074,074 ;10
017362    001     002     004           .BYTE   001,002,004,010,020,040,100,200
017372    200     100     040           .BYTE   200,100,040,020,010,004,002,001
017402    000     014     022           .BYTE   000,014,022,040,170,040,100,176
017412    201     102     044           .BYTE   201,102,044,030,030,044,102,201
017422    074     176     377           .BYTE   074,176,377,377,377,377,176,074
017432    030     074     176           .BYTE   030,074,176,377,377,176,074,030
017442    010     010     024           .BYTE   010,010,024,042,301,042,024,010
017452    010     010     034           .BYTE   010,010,034,076,377,076,034,010
017462    252     252     252           .BYTE   252,252,252,252,252,252,252,252
017472    314     314     314           .BYTE   314,314,314,314,314,314,314,314 ;20
017502    377     000     377           .BYTE   377,000,377,000,377,000,377,000
017512    377     377     000           .BYTE   377,377,000,000,377,377,000,000
017522    377     203     205           .BYTE   377,203,205,211,221,241,301,377
017532    377     201     201           .BYTE   377,201,201,201,201,201,201,377
017542    377     376     374           .BYTE   377,376,374,370,360,340,300,200
017552    377     177     077           .BYTE   377,177,077,037,017,007,003,001
017562    200     300     340           .BYTE   200,300,340,360,370,374,376,377
017572    001     003     007           .BYTE   001,003,007,017,037,077,177,377
017602    003     007     016           .BYTE   003,007,016,034,070,160,340,300
017612    300     340     160           .BYTE   300,340,160,070,034,016,007,003 ;30
017622    303     347     176           .BYTE   303,347,176,074,074,176,347,303
017632    000     160     120           .BYTE   000,160,120,160,000,000,000,000
017642    000     000     000           .BYTE   000,000,000,000,000,000,000,000
017652    010     010     010           .BYTE   010,010,010,010,010,000,010,000
017662    024     024     024           .BYTE   024,024,024,000,000,000,000,000
017672    024     024     066           .BYTE   024,024,066,000,066,024,024,000
017702    010     036     040           .BYTE   010,036,040,034,002,074,010,000
017712    000     062     062           .BYTE     0, 62, 62,  4, 10, 20, 46, 46
017722    000     020     050           .BYTE     0, 20, 50, 50, 20, 52, 44, 32
017732    000     030     030           .BYTE     0, 30, 30, 30,  0,  0,  0,  0 ;40(')
017742    000     010     020           .BYTE     0, 10, 20, 40, 40, 40, 20, 10
017752    000     010     004           .BYTE     0, 10,  4,  2,  2,  2,  4, 10
017762    000     052     034           .BYTE     0, 52, 34, 76, 34, 52,  0,  0
017772    000     000     010           .BYTE     0,  0, 10, 10, 76, 10, 10,  0
020002    000     000     000           .BYTE     0,  0,  0,  0, 30, 30, 10, 20
020012    000     000     000           .BYTE     0,  0,  0,  0, 76,  0,  0,  0
020022    000     000     000           .BYTE     0,  0,  0,  0,  0,  0, 30, 30
020032    000     002     002           .BYTE     0,  2,  2,  4, 10, 20, 40, 40
020042    000     034     042           .BYTE   000,034,042,046,052,062,042,034
020052    000     010     030           .BYTE   000,010,030,010,010,010,010,034 ;50(1)
```

```
020062   000   034   042        .BYTE   000,034,042,002,034,040,040,076
020072   000   034   042        .BYTE   000,034,042,002,014,002,042,034
020102   000   004   014        .BYTE   000,004,014,024,044,076,004,004
020112   000   076   040        .BYTE     0, 76, 40, 74,  2,  2, 42, 34
020122   000   014   020        .BYTE     0, 14, 20, 40, 74, 42, 42, 34
020132   000   076   002        .BYTE     0, 76,  2,  4, 10, 20, 20, 20
020142   000   034   042        .BYTE     0, 34, 42, 42, 34, 42, 42, 34
020152   000   034   042        .BYTE     0, 34, 42, 42, 36,  2,  4, 30
020162   000   000   030        .BYTE     0,  0, 30, 30,  0, 30, 30,  0
020172   000   030   030        .BYTE     0, 30, 30,  0, 30, 30, 10, 20 ;60(;)
020202   000   004   010        .BYTE     0,  4, 10, 20, 40, 20, 10,  4
020212   000   000   000        .BYTE     0,  0,  0, 76,  0, 76,  0,  0
020222   000   020   010        .BYTE     0, 20, 10,  4,  2,  4, 10, 20
020232   000   030   044        .BYTE     0, 30, 44,  4, 10, 10,  0, 10
020242   000   074   102        .BYTE     0, 74,102,132,132,104, 40,  0
020252   000   010   024        .BYTE     0, 10, 24, 42, 42, 76, 42, 42
020262   000   074   022        .BYTE     0, 74, 22, 22, 34, 22, 22, 74
020272   000   034   042        .BYTE     0, 34, 42, 40, 40, 40, 42, 34
020302   000   074   022        .BYTE     0, 74, 22, 22, 22, 22, 22, 74
020312   000   076   040        .BYTE     0, 76, 40, 40, 70, 40, 40, 76 ;70(E)
020322   000   076   040        .BYTE     0, 76, 40, 40, 70, 40, 40, 40
020332   000   036   040        .BYTE     0, 36, 40, 40, 46, 42, 42, 36
020342   000   042   042        .BYTE     0, 42, 42, 42, 76, 42, 42, 42
020352   000   034   010        .BYTE     0, 34, 10, 10, 10, 10, 10, 34
020362   000   002   002        .BYTE     0,  2,  2,  2,  2,  2, 42, 34
020372   000   042   044        .BYTE     0, 42, 44, 50, 60, 50, 44, 42
020402   000   040   040        .BYTE     0, 40, 40, 40, 40, 40, 40, 76
020412   000   042   066        .BYTE     0, 42, 66, 52, 42, 42, 42, 42
020422   000   042   062        .BYTE     0, 42, 62, 52, 46, 42, 42, 42
020432   000   034   042        .BYTE     0, 34, 42, 42, 42, 42, 42, 34 ;80(O)
020442   000   074   042        .BYTE     0, 74, 42, 42, 74, 40, 40, 40
020452   000   034   042        .BYTE     0, 34, 42, 42, 42, 52, 44, 32
020462   000   074   042        .BYTE     0, 74, 42, 42, 74, 50, 44, 42
020472   000   036   040        .BYTE     0, 36, 40, 40, 34,  2,  2, 74
020502   000   076   010        .BYTE     0, 76, 10, 10, 10, 10, 10, 10
020512   000   042   042        .BYTE     0, 42, 42, 42, 42, 42, 42, 34
020522   000   042   042        .BYTE     0, 42, 42, 42, 24, 24, 10, 10
020532   000   042   042        .BYTE     0, 42, 42, 42, 42, 52, 66, 42
020542   000   042   042        .BYTE     0, 42, 42, 24, 10, 24, 42, 42
020552   000   042   042        .BYTE     0, 42, 42, 24, 10, 10, 10, 10 ;90(Y)
020562   000   076   002        .BYTE     0, 76,  2,  4, 10, 20, 40, 76
020572   000   070   040        .BYTE   000,070,040,040,040,040,040,070
020602   000   040   040        .BYTE   000,040,040,020,010,004,002,002
020612   000   016   002        .BYTE   000,016,002,002,002,002,002,016
020622   030   074   176        .BYTE   030,074,176,377,030,030,030,030
020632   010   014   016        .BYTE   010,014,016,377,377,016,014,010
020642   030   030   030        .BYTE   030,030,030,030,377,176,074,030
020652   020   060   160        .BYTE   020,060,160,377,377,160,060,020
020662   000   377   377        .BYTE   000,377,377,377,377,377,377,377
020672   000   000   377        .BYTE   000,000,377,377,377,377,377,377 ;100
020702   000   000   000        .BYTE   000,000,000,377,377,377,377,377
020712   000   000   000        .BYTE   000,000,000,000,377,377,377,377
020722   000   000   000        .BYTE   000,000,000,000,000,377,377,377
020732   000   000   000        .BYTE   000,000,000,000,000,000,377,377
020742   000   000   000        .BYTE   000,000,000,000,000,000,000,377
020752   376   376   376        .BYTE   376,376,376,376,376,376,376,376
```

```
020762   374   374   374        .BYTE   374,374,374,374,374,374,374,374
020772   370   370   370        .BYTE   370,370,370,370,370,370,370,370
021002   360   360   360        .BYTE   360,360,360,360,360,360,360,360
021012   340   340   340        .BYTE   340,340,340,340,340,340,340,340  ;110
021022   300   300   300        .BYTE   300,300,300,300,300,300,300,300
021032   200   200   200        .BYTE   200,200,200,200,200,200,200,200
021042   200   200   200        .BYTE   200,200,200,200,200,200,200,377
021052   377   001   001        .BYTE   377,001,001,001,001,001,001,001
021062   001   001   001        .BYTE   001,001,001,001,001,001,001,377
021072   377   200   200        .BYTE   377,200,200,200,200,200,200,200
021102   000   000   000        .BYTE   000,000,000,000,000,000,000,200
021112   200   000   000        .BYTE   200,000,000,000,000,000,000,000
021122   001   000   000        .BYTE   001,000,000,000,000,000,000,000
021132   000   000   000        .BYTE   000,000,000,000,000,000,000,001  ;120
021142   000   010   024        .BYTE   000,010,024,042,343,000,000,000
021152   010   010   030        .BYTE   010,010,030,040,100,040,030,010
021162   000   074   146        .BYTE   000,074,146,303,201,000,000,000
021172   030   060   140        .BYTE   030,060,140,100,100,140,060,030
021202   000   000   000        .BYTE   000,000,000,000,017,010,010,010
021212   010   010   010        .BYTE   010,010,010,010,370,000,000,000
021222   010   010   010        .BYTE   010,010,010,010,017,000,000,000
021232   000   000   000        .BYTE   000,000,000,000,370,010,010,010
```

```
                                        .SBTTL  T.V.-ENCODED CHARACTER SET.
                                ; THIS CHARACTER SET IS SPECIALLY SELECTED TO REDUCE THE
                                ; BANDWIDTH REQUIREMENTS FOR PAL ENCODING PURPOSES.
                                ;
021242  360  360  360  PALCH:   .BYTE 360,360,360,360,360,360,360,360
021252  030  030  030          .BYTE 030,030,030,030,030,030,030,030
021262  030  030  030          .BYTE 030,030,030,030,030,030,030,030
021272  074  074  074          .BYTE 074,074,074,074,074,074,074,074
021302  000  000  000          .BYTE 000,000,000,000,377,000,000,000
021312  000  000  000          .BYTE 000,000,000,000,377,377,000,000
021322  000  000  377          .BYTE 000,000,377,377,377,377,000,000
021332  030  030  030          .BYTE 030,030,030,030,377,030,030,030
021342  030  030  030          .BYTE 030,030,030,377,377,030,030,030
021352  074  074  377          .BYTE 074,074,377,377,377,377,074,074   ; 9
021362  003  006  014          .BYTE 003,006,014,030,060,140,300,200
021372  200  300  140          .BYTE 200,300,140,060,030,014,006,003
021402  000  034  062          .BYTE 000,034,062,060,170,060,140,176
021412  203  306  154          .BYTE 203,306,154,070,070,154,306,203
021422  074  176  377          .BYTE 074,176,377,377,377,377,176,074
021432  030  074  176          .BYTE 030,074,176,377,377,176,074,030
021442  030  030  074          .BYTE 030,030,074,146,303,146,074,030
021452  030  030  074          .BYTE 030,030,074,176,377,176,074,030
021462  146  146  146          .BYTE 146,146,146,146,146,146,146,146
021472  314  314  314          .BYTE 314,314,314,314,314,314,314,314
021502  377  000  377          .BYTE 377,000,377,000,377,000,377,000   ; 20
021512  377  377  000          .BYTE 377,377,000,000,377,377,000,000
021522  377  303  307          .BYTE 377,303,307,317,333,363,343,377
021532  377  303  303          .BYTE 377,303,303,303,303,303,303,377
021542  377  376  374          .BYTE 377,376,374,370,360,340,300,200
021552  377  377  177          .BYTE 377,377,177,077,037,017,007,003
021562  200  300  340          .BYTE 200,300,340,360,370,374,376,377
021572  003  007  017          .BYTE 003,007,017,037,077,177,377,377
021602  003  007  016          .BYTE 003,007,016,034,070,160,340,300
021612  300  340  160          .BYTE 300,340,160,070,034,016,007,003
021622  303  347  176          .BYTE 303,347,176,074,074,176,347,303
021632  000  160  120          .BYTE 000,160,120,160,000,000,000,000
021642  000  000  000          .BYTE 000,000,000,000,000,000,000,000
021652  000  030  030          .BYTE 000,030,030,030,030,030,000,030
021662  000  066  066          .BYTE 000,066,066,000,000,000,000,000
021672  066  066  167          .BYTE 066,066,167,000,167,066,066,000
021702  030  076  140          .BYTE 030,076,140,074,006,076,030,000
021712  000  146  146          .BYTE 000,146,146,014,030,060,146,146
021722  000  060  154          .BYTE 000,060,154,154,060,156,154,076
021732  000  030  030          .BYTE 000,030,030,030,000,000,000,000
021742  000  030  060          .BYTE 000,030,060,140,140,140,060,030   ; 40
021752  000  030  014          .BYTE 000,030,014,006,006,006,014,030
021762  000  066  034          .BYTE 000,066,034,076,034,066,000,000
021772  000  000  030          .BYTE 000,000,030,030,076,030,030,000
022002  000  000  000          .BYTE 000,000,000,000,000,030,030,060
022012  000  000  000          .BYTE 000,000,000,000,076,000,000,000
022022  000  000  000          .BYTE 000,000,000,000,000,000,030,030
022032  000  006  006          .BYTE 000,006,006,014,030,060,140,140
022042  000  074  146          .BYTE 000,074,146,146,146,146,146,074
022052  000  030  070          .BYTE 000,030,070,030,030,030,030,074
022062  000  074  146          .BYTE 000,074,146,006,036,060,140,176   ; 50
022072  000  074  146          .BYTE 000,074,146,006,030,006,146,074
```

```
022102   000   014   034        .BYTE 000,014,034,074,154,176,014,014
022112   000   176   140        .BYTE 000,176,140,174,006,006,146,074
022122   000   034   060        .BYTE 000,034,060,140,174,146,146,074
022132   000   176   006        .BYTE 000,176,006,014,030,060,060,060
022142   000   074   146        .BYTE 000,074,146,146,030,146,146,074
022152   000   074   146        .BYTE 000,074,146,146,036,006,014,070
022162   000   000   030        .BYTE 000,000,030,030,000,030,030,000
022172   000   000   030        .BYTE 000,000,030,030,000,030,030,060
022202   000   014   030        .BYTE 000,014,030,060,140,060,030,014   ; 60
022212   000   000   000        .BYTE 000,000,000,076,000,076,000,000
022222   000   060   030        .BYTE 000,060,030,014,006,014,030,060
022232   000   074   146        .BYTE 000,074,146,006,034,030,000,030
022242   000   030   146        .BYTE 000,030,146,156,156,146,060,000
022252   000   030   066        .BYTE 000,030,066,146,146,176,146,146
022262   000   174   146        .BYTE 000,174,146,146,170,146,146,174
022272   000   074   146        .BYTE 000,074,146,140,140,140,146,074
022302   000   174   146        .BYTE 000,174,146,146,146,146,146,174
022312   000   176   140        .BYTE 000,176,140,140,170,140,140,176
022322   000   176   140        .BYTE 000,176,140,140,170,140,140,140   ;70
022332   000   076   140        .BYTE 000,076,140,140,156,146,146,076
022342   000   146   146        .BYTE 000,146,146,146,176,146,146,146
022352   000   074   030        .BYTE 000,074,030,030,030,030,030,074
022362   000   006   006        .BYTE 000,006,006,006,006,006,146,074
022372   000   146   154        .BYTE 000,146,154,170,160,170,154,146
022402   000   140   140        .BYTE 000,140,140,140,140,140,140,176
022412   000   146   176        .BYTE 000,146,176,176,146,146,146,146
022422   000   146   146        .BYTE 000,146,146,166,156,146,146,146
022432   000   074   146        .BYTE 000,074,146,146,146,146,146,074
022442   000   174   146        .BYTE 000,174,146,146,174,140,140,140
022452   000   074   146        .BYTE 000,074,146,146,146,156,156,166
022462   000   174   146        .BYTE 000,174,146,146,174,170,154,146
022472   000   076   140        .BYTE 000,076,140,140,074,006,006,174
022502   000   176   030        .BYTE 000,176,030,030,030,030,030,030
022512   000   146   146        .BYTE 000,146,146,146,146,146,146,074
022522   000   146   146        .BYTE 000,146,146,146,146,074,074,030
022532   000   146   146        .BYTE 000,146,146,146,146,176,176,146
022542   000   146   146        .BYTE 000,146,146,074,030,074,146,146
022552   000   146   146        .BYTE 000,146,146,074,030,030,030,030
022562   000   176   006        .BYTE 000,176,006,014,030,060,140,176
022572   000   170   140        .BYTE 000,170,140,140,140,140,140,170
022602   000   140   140        .BYTE 000,140,140,060,030,014,006,006
022612   000   036   006        .BYTE 000,036,006,006,006,006,006,036
022622   030   074   176        .BYTE 030,074,176,377,030,030,030,030
022632   030   034   036        .BYTE 030,034,036,377,377,036,034,030
022642   030   030   030        .BYTE 030,030,030,030,377,176,074,030
022652   030   070   170        .BYTE 030,070,170,377,377,170,070,030
022662   000   377   377        .BYTE 000,377,377,377,377,377,377,377
022672   000   000   377        .BYTE 000,000,377,377,377,377,377,377
022702   000   000   000        .BYTE 000,000,000,377,377,377,377,377
022712   000   000   000        .BYTE 000,000,000,000,377,377,377,377
022722   000   000   000        .BYTE 000,000,000,000,000,377,377,377
022732   000   000   000        .BYTE 000,000,000,000,000,000,377,377
022742   000   000   000        .BYTE 000,000,000,000,000,000,000,377
022752   374   374   374        .BYTE 374,374,374,374,374,374,374,374
022762   374   374   374        .BYTE 374,374,374,374,374,374,374,374
022772   370   370   370        .BYTE 370,370,370,370,370,370,370,370
```

```
023002     360     360     360          .BYTE 360,360,360,360,360,360,360,360
023012     340     340     340          .BYTE 340,340,340,340,340,340,340,340
023022     300     300     300          .BYTE 300,300,300,300,300,300,300,300    ; 110
023032     300     300     300          .BYTE 300,300,300,300,300,300,300,300
023042     300     300     300          .BYTE 300,300,300,300,300,300,300,377
023052     377     003     003          .BYTE 377,003,003,003,003,003,003,003
023062     003     003     003          .BYTE 003,003,003,003,003,003,003,377
023072     377     300     300          .BYTE 377,300,300,300,300,300,300,300
023102     000     000     000          .BYTE 000,000,000,000,000,000,000,300
023112     300     000     000          .BYTE 300,000,000,000,000,000,000,000
023122     003     000     000          .BYTE 003,000,000,000,000,000,000,000
023132     000     000     000          .BYTE 000,000,000,000,000,000,000,003
023142     000     030     074          .BYTE 000,030,074,146,347,000,000,000
023152     030     030     070          .BYTE 030,030,070,060,140,060,070,030
023162     000     074     146          .BYTE 000,074,146,303,303,000,000,000
023172     030     070     160          .BYTE 030,070,160,140,140,160,070,030
023202     000     000     000          .BYTE 000,000,000,000,037,030,030,030
023212     030     030     030          .BYTE 030,030,030,030,370,000,000,000
023222     030     030     030          .BYTE 030,030,030,030,037,000,000,000
023232     000     000     000          .BYTE 000,000,000,000,370,030,030,030
          001000                       .END    START
```

A	= 020000	DBUF0	007310	FORGND	007464	MONITX	013646	S	= 010000
ADDRES	007572	DBUF1	007314	FRMMSG	006550	NODEFM	006762	SAVPC	007524
BAD	007564	DBUF10	007360	FSAVPW	007620	NOHALT	017230	SAVPC1	007526
BAKGND	007466	DBUF11	007364	FSTCNT	007514	NORCH	017242	SAVREG	013740
BAMSG	006653	DBUF12	007370	G	= 100000	NOSYNC	012032	SELBUS	015776
BASADD	007542	DBUF13	007374	GENER	016370	NUMDEV	005660	SETDEV	012162
BASE	007456	DBUF14	007400	GENER1	016512	NXMSG	006732	SETENC	005627
BASE1	007460	DBUF15	007404	GENEX	016650	NXMTRP	016354	SET56	005610
BASE1A	015370	DBUF16	007410	GENINC	016642	OCTIN	014034	SILLSI	012640
BASE1B	015404	DBUF17	007414	GENISH	016520	OCTMSG	007256	SSWR	007504
BASE1C	015422	DBUF18	007420	GENRAN	016522	OCTNUM	007230	START	001000
BASE1D	015402	DBUF19	007424	GENROT	016476	ODAMSG	006707	START1	001306
BASE10	015352	DBUF2	007320	GENR0	016464	PAL	007622	START2	001326
BASE11	007240	DBUF20	007430	GENR1	016454	PALCH	021242	START3	001362
BASM10	015334	DBUF3	007324	GENSEL	016406	PARITY	007546	STATUS	007570
BAS10A	015360	DBUF4	007330	GEN0	016426	PASMSG	006500	STRADD	007604
BELL	013166	DBUF5	007334	GEN1	016432	PCHR	015012	STRLEN	007606
BELL1	013200	DBUF6	007340	GEN25	016446	PMSG1	014750	SWR	007502
BLINK	007470	DBUF7	007344	GEN52	016440	PMSG2	014774	SWRMSG	006623
BUSSE1	016110	DBUF8	007350	GETCH1	014236	PMSG3	015000	SWRSET	012704
BUSSE2	016170	DBUF9	007354	GETCH2	014250	PMSG4	015004	SYNCUP	011644
CALLPC	007600	DECMSG	007250	GETCH3	014362	PRCT1A	015046	TABLE	001366
CARX	007474	DEVBUF	007440	GETCH4	014426	PRNT3	015116	TABLE1	001412
CARY	007476	DEVCNT	007304	GETSTR	014232	PROCT	015024	TABLE2	001436
CHAR	007550	DEVCSR	007436	GOMSG	006005	PROCT1	015034	TESMSG	006473
CHRCOD	005716	DEVFLG	007302	GOMS1	006144	PROCT2	015050	TESTNO	007454
CHRFIN	014656	DEVLST	012374	GOOD	007562	PROCT3	015102	TESTR	011624
CHXLST	007266	EMSG1	007064	HSWR	= 177570	PROM	007462	TEST0	001450
CNVFLG	007602	EMSG2	007070	ILLCHR	014636	PSW	= 177776	TEST1	002040
CRLF	014700	EMSG3	007102	ILLVEC	013226	P02CHR	007624	TEST10	005264
CSR0	007306	EMSG4	007113	ILVMSG	006516	QEXIT1	013706	TEST11	005362
CSR1	007312	EMSG5	007124	JM600	007510	QEXIT2	013710	TEST2	002174
CSR10	007356	EMSG6	007135	LINDEL	014552	QEXIT3	013724	TEST3	002354
CSR11	007362	EMSG7	007153	LINDL1	014572	RANCLC	016622	TEST4	002616
CSR12	007366	EMSG8	007166	LINECH	014602	RAND	007532	TEST5	003110
CSR13	007372	EMSG9	007205	LOADFS	012410	RANDC	007616	TEST5A	003142
CSR14	007376	ENDIT	013062	LOELEM	012522	RANDN	007552	TEST6	004234
CSR15	007402	ERHALT	017210	LOWCHR	007610	RANGEN	016542	TEST7	004660
CSR16	007406	ERRARG	007576	LSIFLG	012702	RANMTA	007556	TKB	= 177562
CSR17	007412	ERRDIS	007574	MODADM	006635	RANSEC	016626	TKS	= 177560
CSR18	007416	ERRDI1	007442	MODADR	007536	RANSEL	007554	TPB	= 177566
CSR19	007422	ERRFLG	007444	MODIFY	015450	RANST	007560	TPDEC1	015154
CSR2	007316	ERROR	016666	MODI1	015500	RAN1	016554	TPDEC2	015234
CSR20	007426	ERRSV1	017036	MODI2	015524	RAN2	016572	TPDEC3	015272
CSR3	007322	ERRSV2	017064	MODI3	015544	RAN4	016630	TPDEC4	015306
CSR4	007326	ERRSV3	017112	MODI4	015656	REACT	001174	TPDEC5	015316
CSR5	007332	ERRSV4	017140	MODI5	015714	READ	014206	TPREDY	013276
CSR6	007336	ERRSV5	017166	MODPRM	006646	REDMES	006570	TPS	= 177564
CSR7	007342	FADR	013156	MODSAV	007540	REPCNT	007506	TRAPSV	013326
CSR8	007346	FASTSW	013306	MODSPA	006642	RSTART	001200	TRPARG	007516
CSR9	007352	FILL	012722	MODXIT	015734	RSTREG	013776	TRPBAK	013534
D	= 040000	FILL1	012754	MONIT	013546	RUBCHR	014504	TRPERR	007544
DATA	007566	FLASH	007472	MONITA	013640	RUBFLG	007614	TRPUP	013470

```
TRPMEM  007522      TYPOTD  014154      T0106   002130      T0518   003342      T0710   005074
TRPSCP  013444      TYPOTE  014172      T0201   002220      T0519   003374      T0711   005102
TRPSEL  007520      TYPOUT  014734      T0202   002276      T0520   003270      T0712   005220
TS0MSG  006360      T0001   001472      T0301   002400      T0601   004260      T0715   005254
TS1MSG  006365      T0002   001506      T0302   002470      T0602   004304      T1001   005310
TS10MS  006462      T0003   001514      T0303   002442      T0603   004306      UPPCHR  007612
TS2MSG  006411      T0004   001532      T0304   002520      T0604   004372      VECTOR  012714
TS3MSG  006416      T0005   001600      T0305   002546      T0605   004430      VIDEON  007450
TS4MSG  006426      T0007   001634      T0401   002642      T0606   004436      WAITS   012040
TS5MSG  006433      T0009   001654      T0402   002656      T0607   004444      WAIT1   007452
TS6MSG  006441      T0010   001672      T0403   002720      T0615   004570      WMSG    006317
TS7MSG  006454      T0011   001740      T0406   002736      T0701   004704      XMAX    007446
TYPCTC  013656      T0013   001774      T0407   002776      T0702   004730      XXDPGO  001176
TYPDEC  015150      T0101   002064      T0409   003026      T0703   004732      XXDP2   001252
TYPD1   007534      T0102   002072      T0410   003052      T0704   005016      YMAX    007500
TYPOTA  007530      T0103   002102      T05XIT  004214      T0705   005032      $ENDAD  013142
TYPOTB  014040      T0104   002110      T0501   003134      T0706   005044      .     = 023242
TYPOTC  014120      T0105   002122      T0502   003226      T0707   005056

. ABS.  023242      000


ERRORS DETECTED:  0

CVVTCA.BIN,CVVTCA.LPT/NL:TOC=CVVTCA.SRC
RUN-TIME: 14 28 1 SECONDS
RUN-TIME RATIO: 85/44=1.9
CORE USED:  13K  (25 PAGES)
```