

VTV30-K

VTV30-K LOGIC TEST  
CVVTCA0

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 INPUTTING 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<br>WAS IN FORCE WHEN THE LAST TRAP<br>INSTRUCTION WAS EXECUTED. |



198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249

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.



251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300

## 8. ERROR REPORTS

THE FORMAT OF THE ERROR REPORTS IS AS FOLLOWS:-

|              |                    |
|--------------|--------------------|
| E# ABBB      | AT PC:CCCC         |
| GOOD: DDDD   | BAD: EEEE          |
| STATUS: FFFF | ADDRESS: 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.



302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347

## 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.



349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403

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 |

405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444

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)

446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501

503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546

#### 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 CHARACTER 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 (SWR 10) - 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.

548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599

601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619

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          ;RETURN 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.)
        .
        .
        .
;
```

VTV30K DIAGNOSTIC  
CVTCA.SRC 08-JAN-81 10:36 MACY11 30A(1052) 08-JAN-81 10:58 PAGE 15<sup>C</sup> 2

SEQ 0015

673

.ENDR



|     |        |        |                     |
|-----|--------|--------|---------------------|
| 675 |        | .TITLE | VTV30K DIAGNOSTIC   |
| 676 |        | .SBTTL | GENERAL DEFINITIONS |
| 677 |        |        |                     |
| 678 |        |        |                     |
| 679 | 000000 | R0=    | %00                 |
| 680 | 000001 | R1=    | %01                 |
| 681 | 000002 | R2=    | %02                 |
| 682 | 000003 | R3=    | %03                 |
| 683 | 000004 | R4=    | %04                 |
| 684 | 000005 | R5=    | %05                 |
| 685 | 000006 | SP=    | %06                 |
| 686 | 000007 | PC=    | %07                 |
| 687 | 177776 | PSW=   | 177776              |
| 688 | 177570 | HSWR=  | 177570              |
| 689 |        |        |                     |
| 690 |        |        |                     |
| 691 | 100000 | G=     | 100000              |
| 692 | 040000 | D=     | 40000               |
| 693 | 020000 | A=     | 20000               |
| 694 | 010000 | S=     | 10000               |
| 695 |        |        |                     |
| 696 |        |        |                     |
| 697 | 177564 | TPS=   | 177564              |
| 698 | 177566 | TPB=   | 177566              |
| 699 | 177560 | TKS=   | 177560              |
| 700 | 177562 | TKB=   | 177562              |
| 701 |        |        |                     |
| 702 |        |        |                     |
| 703 |        |        |                     |
| 704 |        |        |                     |
| 705 |        |        |                     |
| 706 |        | .ENABL | ABS                 |
| 707 |        | .ENABL | AMA                 |
| 708 |        |        |                     |
| 709 |        |        |                     |
| 710 |        |        |                     |
| 711 |        |        |                     |
| 712 |        |        |                     |

SEQ 0017

714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736

**.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
```



```

738                                     .SBTTL  INITIALISATION
739                                     .ASECT
740                                     .=0
741                                     .=42
742 000042 000000 0                                     : XXDP HOOK INITIALIZATION.
743 000200 000200 0001000 001000 0001000 .=200
744 000200 000137 0001000 001000 JMP @#START : JMP TO START AT 200
745 000200 0001000 001000 .=1000
746
747
748 001000 012706 001000 START: MOV #,SP :INITIALISE STACK POINTER
749 001004 PSWSET #340
750
751 001020 012737 012704 000004 MOV #SWRSET,4 :TEST FOR HARDWARE SWR
752 001026 012737 000340 000006 MOV #340,6 :TRAPS TO 4 IF IT
753 001034 012737 177570 007502 MOV #HSWR,SWR :DOES NOT EXIST
754 001042 005777 006434 TST @SWR
755 001046 005037 007504 CLR SSWR :INITIALISE SOFTWARE SWR
756 001052 004737 012640 JSR PC,SILLSI : CHECK WHICH PROCESSOR WE ARE ON
757 001056 004737 012714 JSR PC,VECTOR :FILL 0-774 WITH HALT TRAPS
758 001062 013737 007306 007436 MOV CSRO,DEVCSR : ASSUME 1 DEVICE
759 001070 013737 007310 007440 MOV DBUF0,DEVBUF
760 001076 005037 007302 CLR DEVFLG
761 001102 004737 014734 JSR PC,TYPOUT
762 001106 006005 GOMSG :DIAGNOSTIC
763 001110 004737 012162 JSR PC,SETDEV : SET UP THE DEVICE
764 001114 004737 015776 JSR PC,SELBUS
765 001120 000137 001200 JMP RSTART :THEN GOTO THE RESTART ADDRESS
766 001174 001174 .=1174
767 001174 000000 REACT: 0
768 001176 000425 XXDPG0: BR XXDP2 : EXTERNAL TEST-EQUIPMENT HOOK
769
770 001200 012706 001000 RSTART: MOV #START,SP :INITIALISE STACK POINTER
771 001204 PSWSET #340
772 001220 005037 007302 CLR DEVFLG
773 001224 013737 007306 007436 MOV CSRO,DEVCSR
774 001232 013737 007310 007440 MOV DBUF0,DEVBUF
775
776 001240 004737 014734 JSR PC,TYPOUT
777 001244 006317 WMSG :SELECT DESIRED CONSOLE SWITCHES
778 001246 004737 013546 JSR PC,MONIT :GO TO SWR MONITOR
779
780 001252 012706 001000 :XXDP2: MOV #START,SP : RESTORE THE STACK
781 001256 PSWSET #340
782 001272 005037 007444 CLR ERRFLG : CLEAR ERROR FLAG
783 001276 005037 007574 CLR ERRDIS
784 001302 005037 007442 CLR ERRDI1
785 001306 013737 007306 007436 START1: MOV CSRO,DEVCSR : START WITH THE FIRST CSR
786 001314 013737 007310 007440 MOV DBUF0,DEVBUF :
787 001322 005037 007302 CLR DEVFLG
788
789 001326 012706 001000 START2: MOV #START,SP :INITIALISE STACK POINTER
790 001332 PSWSET #340
791
792 001346 032777 000100 006126 BIT #100,@SWR :CHECK FOR PRE-SELECTED TEST
793 001354 001002 BNE START3

```

VTV30K DIAGNOSTIC MACY11 30A(1052) 08-JAN-81 10:58 PAGE 18-1<sup>G 2</sup>  
CVVTC.A.SRC 08-JAN-81 10:36 INITIALISATION

SEQ 0019

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



798 001366 017700 006110  
 799 001372 042700 177700  
 800 001376 022700 000011  
 801 001402 002676  
 802  
 803 001404 006300  
 804 001406 000170 001412  
 805  
 806  
 807  
 808  
 809

TABLE: MOV @SWR,R0 ;GET SELECTED TEST NO  
 BIC #177700,R0  
 CMP #11,R0 ;CHECK FOR VALID TEST NO  
 BLT RSTART  
 ASL R0  
 JMP @TABLE1(R0) ;JUMP TO PRE-SELECTED TEST

810 001412 001450  
 811 001414 002040  
 812 001416 002174  
 813 001420 002354  
 814 001422 002616  
 815 001424 003110  
 816 001426 004234  
 817 001430 004660  
 818 001432 005264  
 819 001434 005362

TABLE1: TEST0  
 TEST1  
 TEST2  
 TEST3  
 TEST4  
 TEST5  
 TEST6  
 TEST7  
 TEST10  
 TEST11

; THE FOLLOWING TABLE HAS ONE ENTRY FOR EACH OF THE ABOVE  
 ; IF ZERO, IT DISABLES THE STALL FACILITY IN 'WAITS' FOR  
 ; THAT TEST.

820  
 821  
 822  
 823  
 824 001436 000 000 001  
 001441 001 001 000  
 001444 001 000 000  
 001447 000

TABLE2: .BYTE 0,0,1,1,1,0,1,0,0,0

825 .EVEN

```
827                                     .SBTTL TEST0
828
829                                     : THIS TEST CHECKS THE READ/WRITE BITS OF THE CONTROL REGISTER
830                                     : BY WRITING A SLIDING ONES PATTERN TO THE REGISTER, TO CHECK
831                                     : THAT BITS CAN SET. THEN, BY WRITING A SLIDING ZEROES PATTERN
832                                     : CHECK THAT THE BITS CAN CLEAR. AT EACH STAGE OF THE OPERATION
833                                     : THE DATA BUFFER IS CHECKED TO SEE THAT THE SETTING/CLEARING
834                                     : HAD NO EFFECT.
835
836 001450 005037 007454 TEST0: CLR TESTNO ; SET TEST NUMBER
837 001454 104402 TRAP+2
838 001456 001472 T0001
839 001460 004737 011624 JSR PC,TESTR ; PRINT TEST NUMBER
840 001464 004737 014734 JSR PC,TYPOUT
841 001470 006360 TSOMSG
842 001472 012704 000200 T0001: MOV #200,R4 ; SUGGEST AN ITERATION COUNT
843 001476 004737 013306 JSR PC,FASTSW ; IS FAST ITERATION SET
844 001502 010437 007506 MOV R4,REPCNT ; SET ACTUAL ITERATION COUNT
845 001506 012737 000020 007566 T0002: MOV #16,DATA ; GET LOOP COUNT
846 001514 012703 000004 T0003: MOV #4,R3 ; SELECT SLIDING ONES
847 001520 004737 016370 JSR PC,GENER ; GET THE PATTERN
848 001524 042737 034000 007562 BIC #34000,GOOD ; MASK READ ONLY BITS
849 001532 013737 007436 007572 T0004: MOV DEVCSR,ADDRES ; GET CURRENT ADDRESS
850 001540 013777 007562 006024 MOV GOOD,@ADDRES ; SET DATA IN CSR
851 001546 017737 006020 007564 MOV @ADDRES,BAD ; READ REPLY
852 001554 042737 034000 007564 BIC #34000,BAD ; MASK UNWANTED
853 001562 023737 007562 007564 CMP GOOD,BAD ; ARE THEY THE SAME
854 001570 001403 BEQ T0005 ; YES
855 001572 004737 016666 JSR PC,ERROR ; NO REPORT AN ERROR
856 001576 160001 1+D+G+A
857 001600 104410 T0005: TRAP+10 ; LOOP ON THIS DATA
858 001602 001532 T0004
859 001604 013737 007440 007572 MOV DEVBUF,ADDRES ; SET DATA BUFFER ADDRESS
860 001612 017737 005754 007570 MOV @ADDRES,STATUS ; THE DATA BUFFER SHOULD BE CLEAR
861 001620 005737 007570 TST STATUS ; IS IT
862 001624 001403 BEQ T0007 ; YES
863
864 : ERROR-
865 : CHECK 'LEVA LOAD CAR H'
866 : 'LEVA READ CAR L'
867 : +3VA ON E46-1 IF BAD=0
868 : RELEVANT BIT ON LEVB B8 (E25,E46)
869 001626 004737 016666 JSR PC,ERROR
870 001632 070002 2+A+S+D
871 001634 104420 T0007: TRAP+20 ; LOOP ON COMPLETE
872 001636 001532 T0004 ; ALL PATTERNS DONE
873 001640 005337 007566 DEC DATA ; NO
874 001644 001323 BNE T0003
875
876 : NOW DO SLIDING ZEROES
877
878 001646 012737 000020 007566 T0009: MOV #16,DATA ; GET LOOP COUNT
879 001654 012703 000005 T0009: MOV #5,R3 ; SELECT SLIDING ZEROES
880 001660 004737 016370 JSR PC,GENER ; GET THE PATTERN
881 001664 042737 034000 007562 BIC #34000,GOOD ; MASK READ ONLY BITS
882 001672 013737 007436 007572 T0010: MOV DEVCSR,ADDRES ; GET CURRENT ADDRESS
883 001700 013777 007562 005664 MOV GOOD,@ADDRES ; SET DATA IN CSR
```



```

883 001706 017737 005660 007564      MOV      @ADDRESS,BAD      ; READ REPLY
884 001714 042737 034000 007564      BIC      #34000,BAD      ; MASK UNWANTED
885 001722 023737 007562 007564      CMP      GOOD,BAD      ; ARE THEY THE SAME
886 001730 001403                BEQ      T0011      ; YES
887                                ; ERROR-
888                                ; REPLACE LEVB B8 E25 OR E46?
889 001732 004737 016666                JSR      PC,ERROR      ; NO REPORT AN ERROR
890 001736 160003                3+D+G+A
891 001740 104430                T0011: TRAP+30      ; LOOP ON THIS DATA
892 001742 001672                T0010
893 001744 013737 007440 007572      MOV      DEVBUF,ADDRESS ; SET DBUF ADDRESS
894 001752 017737 005614 007570      MOV      @ADDRESS,STATUS ; THE DATA BUFFER SHOULD BE CLEAR
895 001760 005737 007570                TST      STATUS      ; IS IT
896 001764 001403                BEQ      T0013      ; YES
897                                ; ERROR-
898                                ; REPLACE LEVB B8 E25 OR E46?
899 001766 004737 016666                JSR      PC,ERROR
900 001772 070004                4+A+S+D
901 001774 104440                T0013: TRAP+40
902 001776 001672                T0010      ; LOOP ON COMPLETE
903 002000 005337 007566                DEC      DATA      ; ALL PATTERNS DONE
904 002004 001323                BNE      T0009      ; NO
905                                ;
906                                ; END OF TEST ?
907                                ;
908                                TRAP+4
909 002006 104404                T0003      ; LOOP ON TEST ?
910 002010 001514                DEC      REPCNT      ; ITERATE TILL DONE
911 002012 005337 007506                BEQ      T0002
912 002016 001633                CLR      @DEVCSR      ; TIDY UP AFTER TEST.
913 002020 005077 005412                BIT      #100,@SWR      ; PRESELECT TEST ?
914 002024 032777 000100 005450      BEQ      TEST1      ; NO
915 002032 001402                JMP      START2      ; YES GO DO IT
915 002034 000137 001326

```

```

917          .SBTTL TEST1
918
919          : THIS TEST CHECKS THE FIELD VIDEO TIMING BY ATTEMPTING
920          : TO RING THE BELL EVERY SECOND FOR 5 SECONDS
921          : IF THE BELL DOES NOT RING, TYPE CNTRL-X TO ABORT THE TEST
922
923
924 002040 012737 000001 007454 TEST1: MOV #1,TESTNO ; SET TEST NUMBER
925 002046 104402 TRAP+2
926 002050 002064 T0101
927 002052 004737 011624 JSR PC,TESTR ; PRINT TEST NUMBER
928 002056 004737 014734 JSR PC,TYPOUT
929 002062 006365 TS1MSG
930 002064
931 002064 012737 000100 007506 T0101: MOV #100,REPCNT ; SET ITERATION COUNT
932
933          : TRY TO SYNC UP
934
935 002072 004737 011644 T0102: JSR PC,SYNCUP ; SYNC UP
936 002076 000401 BR T0103 ; NO SYNC OR ABORT
937 002100 000403 BR T0104 ; GO A SYNC SIGNAL
938
939          : ERROR-
940          : CHECK 'LEVB CLOCK H' (8 MHZ)
941          : 'LEVB CLOCK L'
942          : 'LEVB VA02 H' (1 MHZ)
943          : 'LEVA VA02 L' (1 MHZ)
944          : 'LEVB VA17 H' (16.7 MILLISECS FOR 525, 20 FOR 625)
945 002102 004737 016666 T0103: JSR PC,ERROR ; ERROR
946 002106 000001 1
947 002110 104410 T0104: TRAP+10 ; LOOP
948 002112 002072 T0102 ; TO HERE
949 002114 005337 007506 DEC REPCNT ; REDUCE ITERATION
950 002120 001364 BNE T0102 ; KEEP GOING
951
952          : NOW RING THAT BELL
953
954 002122 012737 000005 007566 T0105: MOV #5,DATA ; FINE BELLS
955 002130 004737 012040 T0106: JSR PC,WAIT5 ; WAIT ONE SECOND
956 002134 000001 1
957 002136 004737 013166 JSR PC,BELL ; DING DONG !!
958 002142 005337 007566 DEC DATA
959 002146 001370 BNE T0106 ; KEEP GOING
960 002150 104420 TRAP+20
961 002152 002122 T0105 ; LOOP ?
962
963          :
964          :
965 002154 104404 TRAP+4 ; LOOP ON TEST
966 002156 002064 T0101
967 002160 032777 000100 005314 BIT #100,ASWR ; PRESELECT TEST
968 002166 001402 BEQ TEST2
969 002170 000137 001326 JMP START2 ; YES- GO DO IT

```



```

969          .SBTTL TEST2
970
971          : THE TEST CHECKS THE PICTURE BY WRITING TO EVERY LOCATION,
972          : USING THE SAME FOREGROUND AND BACKGROUND COLOUR.
973
974          : THE SEQUENCE OF OPERATIONS IS:
975
976          : F/B WHITE NO VIDEO
977          : F/B BLACK VIDEO ON
978          : F/B RED VIDEO ON
979          : F/B GREEN VIDEO ON
980          : F/B YELLOW VIDEO ON
981          : F/B BLUE VIDEO ON
982          : F/B MAGENTA VIDEO ON
983          : F/B CYAN VIDEO ON
984          : F/B WHITE VIDEO ON
985
986          : CHARACTER STORE ERRORS ARE DELIBERATELY BYPASSED IN THIS TEST.
987          : IF THE TEST WORKS, COLOUR INFORMATION CAN BE USED IN THE NEXT
988          : TEST.
989          : ERRORS-
990          : CHECK LOW-ORDER BITS OF PICTURE STORE (LEVC)
991          : OUTPUT LATCH ON LEVD, E23.
992          : OUTPUT MUX ON LEVD, E18.
993          : STATISERS E6 & E12, WITH STROBES 'LEVB STROBE A H', '-B H'
994
995          002174 012737 000002 007454 TEST2: MOV #2,TESTNO ; SET TEST NUMBER
996          002202 104402 TRAP+2
997          002204 002220 T0201
998          002206 004737 011624 JSR PC,TESTR ; PRINT TEST NUMBER
999          002212 004737 014734 JSR PC,TYPOUT
1000         002216 006411 TS2MSG
1001         002220 012737 000007 007464 T0201: MOV #7,FORGND ; SET F/B TO WHITE
1002         002226 012737 000007 007466 MOV #7,BAKGND
1003         002234 005037 007470 CLR BLINK ; NO BLINK
1004         002240 005037 007472 CLR FLASH ; FLASH,
1005         002244 005037 007450 CLR VIDEON ; OR VIDEO.
1006         002250 005037 007550 CLR CHAR ; USE CHARACTER ZERO
1007         002254 004737 012410 JSR PC,LOADFS ; LOAD THE DISPLAY
1008
1009          : NOW WAIT AROUND
1010
1011
1012
1013          : NOW LOAD THE REST OF THE PICTURES WITH VIDEO ON
1014
1015          002260 005037 007464 CLR FORGND ; START WITH BLACK
1016          002264 005037 007466 CLR BAKGND ; ON BLACK
1017          002270 012737 040000 007450 MOV #40000,VIDEON ; BUT WITH VIDEO ON
1018          002276 004737 012410 T0202: JSR PC,LOADFS ; LOAD THE SCREEN
1019
1020          : NOW WAIT AROUND
1021
1022          002302 004737 012040 JSR PC,WAIT5 ; WAIT FOR 2 SECONDS
1023          002306 000002 2
1024          002310 104410 TRAP+10
  
```

|      |        |        |        |        |                          |                  |
|------|--------|--------|--------|--------|--------------------------|------------------|
| 1025 | 002312 | 002276 |        |        | T0202                    | : LOOP TO HERE   |
| 1026 |        |        |        |        | :                        |                  |
| 1027 |        |        |        |        | : NOW UPDATE THE COLOURS |                  |
| 1028 |        |        |        |        | :                        |                  |
| 1029 | 002314 | 005237 | 007464 |        | INC FORGND               | : NEW FORGROUND  |
| 1030 | 002320 | 005237 | 007466 |        | INC BAKGND               | : AND BACKGROUND |
| 1031 | 002324 | 023727 | 007464 | 000007 | CMP FORGND,#7            | : ALL DONE       |
| 1032 | 002332 | 003761 |        |        | BLE T0202                | : NO             |
| 1033 |        |        |        |        | :                        |                  |
| 1034 |        |        |        |        | : COMPLETE               |                  |
| 1035 |        |        |        |        | :                        |                  |
| 1036 | 002334 | 104404 |        |        | TRAP+4                   | : LOOP ON TEST ? |
| 1037 | 002336 | 002220 |        |        | T0201                    |                  |
| 1038 | 002340 | 032777 | 000100 | 005134 | BIT #100,@SWR            | : PRESELECT TEST |
| 1039 | 002346 | 001402 |        |        | BEQ TEST3                | : NO             |
| 1040 | 002350 | 000137 | 001326 |        | JMP START2               | : YES GO TO IT   |



1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097

## .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.

TEST3: MOV #3,TESTNO ; SET TEST NUMBER  
TRAP+2  
T0301  
JSR PC,TESTR ; PRINT TEST NUMBER  
JSR PC,TYPOUT  
TS3MSG  
T0301: CLR BLINK ; NO BLINK  
CLR FLASH ; NO FLASH  
MOV #40000,VIDEON ; ENABLE VIDEO  
CLR CHAR ; SELECT CHARACTER ZERO  
MOV #4,FORGND ; SELECT BLUE  
MOV #4,BAKGND ; ON BLUE  
JSR PC,LOADFS ; LOAD THE FULL SCREEN  
  
; NOW DO X ADDRESS TESTING  
T0303: CLR CARY ; START AT TOP OF SCREEN..  
MOV #47,CARX ; AT RIGHT.  
MOV #5,FORGND ; SET CHARACTERS TO CYAN  
MOV FORGND,BAKGND ; ...ALL OVER!  
  
T0302: JSR PC,LOELEM ; LOAD ONE ELEMENT  
INC CARY ; NEXT LINE DOWN...  
DEC CARX ; SHIFT LEFT ONCE.  
CMP CARY,#32. ; OVER TOP?  
BNE T0302 ; NO, NEXT BLOCK  
TRAP+10 ; LOOP ON THIS BIT  
T0303  
T0304: CLR CARX ; NOW START AT RIGHT  
MOV YMAX,CARY ; AT BOTTOM  
MOV #7,FORGND ; DISPLAY IN WHITE..  
MOV FORGND,BAKGND ; ALL OVER.  
  
T0305: JSR PC,LOELEM ; LOAD ONE ELEMENT  
INC CARY ; SHIFT RIGHT  
DEC CARY ; SHIFT UP  
BPL T0305 ; PAST TOP OF SCREEN?  
TRAP+20 ; LOOP ON THIS  
T0304  
; NOW WAIT AROUND  
JSR PC,WAIT5 ; WAIT FOR 2 SECONDS  
2

|      |        |        |        |        |        |           |                  |
|------|--------|--------|--------|--------|--------|-----------|------------------|
| 1098 | 002576 | 104404 |        |        | TRAP+4 |           | ; LOOP ON TEST   |
| 1099 | 002600 | 002400 |        |        | T0301  |           |                  |
| 1100 | 002602 | 032777 | 000100 | 004672 | BIT    | #100,ASWR | ; PRESELECT TEST |
| 1101 | 002610 | 001402 |        |        | BEQ    | TEST4     |                  |
| 1102 | 002612 | 000137 | 001326 |        | JMP    | START2    | ; YES- GO DO IT  |
| 1103 |        |        |        |        |        |           |                  |



```

1105 .SBTTL TEST4
1106
1107 : THIS TEST CHECKS OUT THE FOREGROUND BACKGROUND SELECTION, BY
1108 : DISPLAYING CHARACTER CODE ZERO IN ALL LOCATIONS. CHARACTER
1109 : ZERO IS DEFINED AS BEING HALF FOREGROUND AND HALF BACKGROUND,
1110 : IN A VERTICAL STRIPE (CHARACTER DATA 360). THE FOREGROUND
1111 : AND BACKGROUND COLOURS ARE THEN ALTERED AND THE WHOLE DISPLAY
1112 : IS REFRESHED. THE SEQUENCE BEING: BLACK ON WHITE,
1113 : BLINK SET, THEN BLINK CLEAR BUT FLASH SET. FINALLY BLINK AND
1114 : FLASH ARE SET TOGETHER, CAUSING A PICTURE THAT 'HOPS' SIDEWAYS
1115 : WITH A BACKGROUND FLASHING.
1116
1117 : IF THIS TEST WORKS THE OUTPUT MUX (LEVB) IS FUNCTIONING.
1118 : ERRORS-
1119 : CHECK 'LEVB F/B SELECT H'
1120 : 'LEVB FLASH B H' IF BLINK OR FLASH DOES NOT WORK.
1121 : LEVA SIGNALS ON E4...
1122 : 'LEVB FLASH A'
1123 : 'LEVB FLASH B'
1124 : 'LEVD BLINK A'
1125 : 'LEVD BLINK B'
1126 : 'LEVD DATA'
1127
1128 002616 012737 000004 007454 TEST4: MOV #4,TESTNO ; SET TESTNUMBER
1129 002624 104402 TRAP+2
1130 002626 002642 T0401
1131 002630 004737 011624 JSR PC,TESTR ; PRINT TEST NUMBER
1132 002634 004737 014734 JSR PC,TYP0UT
1133 002640 006426 TS4MSG
1134 002642 005037 007550 T0401: CLR CHAR ; START WITH CHARACTER CODE 0
1135 002646 005737 007462 TST PROM ; PROM CHARACTER SET ?
1136 002652 001022 BNE T0403 ; YES
1137 002654 005001 CLR R1 ; BUT THERE ARE 8 LINES TO IT
1138 002656 013700 007550 T0402: MOV CHAR,R0 ; GET CHARACTER NUMBER
1139 002662 006300 ASL R0
1140 002664 006300 ASL R0 ; SET IT IN THE RIGHT PLACE
1141 002666 006300 ASL R0
1142 002670 050100 BIS R1,R0 ; ADD IN LINE NUMBER
1143 002672 052700 100000 BIS #100000,R0 ; SAY ITS SELECT CHSR
1144 002676 010077 004534 MOV R0,@DEVCSR ; LOAD THE CSR
1145 002702 012777 000360 004530 MOV #360,@DEVBUF ; LOAD THE DATA PATTERN
1146 002710 005201 INC R1 ; GET NEXT LINE
1147 002712 020127 000007 CMP R1,#7 ; ALL LINES DONE
1148 002716 003757 BLE T0402 ; NO YET
1149
1150 002720 T0403:
1151 002720 005037 007472 CLR FLASH ; NO FLASH
1152 002724 005037 007470 CLR BLINK ; NO BLINK
1153 002730 012737 000007 007464 MOV #7,F0RGND ; START WITH BLACK FOREGROUND
1154 002736 012737 000000 007466 T0406: MOV #0,BAKGD ; AND IT COMPLEMENT BACKGROUND
1155 002744 012737 040000 007450 MOV #40000,VIDEON ; SET VIDEO ON
1156 002752 004737 012410 JSR PC,LOADFS ; FILL THE SCREEN
1157
1158 : WAIT AROUND
1159
1160 002756 004737 012040 JSR PC,WAIT5 ; WAIT FOR 2 SECONDS

```

|      |        |        |        |        |                                    |                   |                           |
|------|--------|--------|--------|--------|------------------------------------|-------------------|---------------------------|
| 1161 | 002762 | 000002 |        |        | 2                                  |                   |                           |
| 1162 | 002764 | 104410 |        |        | TRAP+10                            |                   |                           |
| 1163 | 002766 | 002736 |        |        | T0406                              | ; LOOP ON THIS    |                           |
| 1164 |        |        |        |        | :                                  |                   |                           |
| 1165 |        |        |        |        | : NOW UPDATE THE FOREGROUND COLOUR |                   |                           |
| 1166 |        |        |        |        | :                                  |                   |                           |
| 1167 |        |        |        |        | : NOW CHANGE BLINK STATUS          |                   |                           |
| 1168 |        |        |        |        | :                                  |                   |                           |
| 1169 | 002770 | 012737 | 000100 | 007470 | MOV                                | #100,BLINK        |                           |
| 1170 | 002776 | 004737 | 012410 |        | T0407: JSR                         | PC,LOADFS         | ; LOAD DISPLAY            |
| 1171 | 003002 | 004737 | 012040 |        | JSR                                | PC,WAIT5          | ; WAIT 2 SECS             |
| 1172 | 003006 | 000002 |        |        | 2                                  |                   |                           |
| 1173 | 003010 | 104420 |        |        | TRAP+20                            |                   |                           |
| 1174 | 003012 | 002776 |        |        | T0407                              | ; LOOP ON PICTURE |                           |
| 1175 | 003014 | 005037 | 007470 |        | CLR                                | BLINK             | ; RESET BLINK             |
| 1176 | 003020 | 012737 | 000200 | 007472 | MOV                                | #200,FLASH        | ; SET FLASH               |
| 1177 | 003026 | 004737 | 012410 |        | T0409: JSR                         | PC,LOADFS         | ; LOAD PICTURE            |
| 1178 | 003032 | 004737 | 012040 |        | JSR                                | PC,WAIT5          | ; WAIT 2 SECS             |
| 1179 | 003036 | 000002 |        |        | 2                                  |                   |                           |
| 1180 | 003040 | 104430 |        |        | TRAP+30                            |                   |                           |
| 1181 | 003042 | 003026 |        |        | T0409                              | ; LOOP ON PICTURE |                           |
| 1182 | 003044 | 012737 | 000100 | 007470 | MOV                                | #100,BLINK        | ; NOW SET BLINK AND FLASH |
| 1183 | 003052 | 004737 | 012410 |        | T0410: JSR                         | PC,LOADFS         | ; FILL SCREEN             |
| 1184 | 003056 | 004737 | 012040 |        | JSR                                | PC,WAIT5          | ; STALL FOR A WHILE!      |
| 1185 | 003062 | 000002 |        |        | 2                                  | ; 2 SECS          |                           |
| 1186 | 003064 | 104440 |        |        | TRAP+40                            |                   |                           |
| 1187 | 003066 | 003052 |        |        | T0410                              | ; LOOP ON PICTURE |                           |
| 1188 | 003070 | 104404 |        |        | TRAP+4                             | ; LOOP ON TEST    |                           |
| 1189 | 003072 | 002642 |        |        | T0401                              | :                 |                           |
| 1190 | 003074 | 032777 | 000100 | 004400 | BIT                                | #100,@SWR         | ; PRESELECT TEST          |
| 1191 | 003102 | 001402 |        |        | BEG                                | TEST5             |                           |
| 1192 | 003104 | 000137 | 001326 |        | JMP                                | START2            | ; YES GO TO IT            |



1194  
 1195  
 1196  
 1197  
 1198  
 1199  
 1200  
 1201  
 1202  
 1203  
 1204  
 1205  
 1206  
 1207  
 1208  
 1209  
 1210  
 1211  
 1212  
 1213  
 1214  
 1215  
 1216  
 1217  
 1218  
 1219  
 1220  
 1221  
 1222  
 1223  
 1224  
 1225  
 1226  
 1227  
 1228  
 1229  
 1230  
 1231  
 1232  
 1233  
 1234  
 1235  
 1236  
 1237  
 1238  
 1239  
 1240  
 1241  
 1242  
 1243  
 1244  
 1245  
 1246  
 1247  
 1248  
 1249

.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)

TEST5: MOV #5,TESTNO ; SET TEST NUMBER  
 TRAP+2  
 T0501  
 JSR PC,TESTR  
 JSR PC,TYP0UT  
 TSSMSG  
 T0501: TST PROM ; IS THIS A PROM CHARACTER  
 BNE T0502 ; SET

RAM CHARACTER SET PRESENT

TEST5A: CLR CHAR ; START WITH CHARACTER CODE ZERO  
 5\$: CLR R1 ; THERE ARE 8 LINES PER CHAR  
 10\$: MOV CHAR,R0 ; GET CHARACTER NUMBER  
 ASL R0 ; SET IN RIGHT PLACE  
 ASL R0  
 ASL R0  
 BIS R1,R0 ; ADD IN LINE NUMBER  
 BIS #100000,R0 ; SELECT CHSR  
 MOV R0,@DEVCSR ; SELECT CHARACTER TO ADDRESS  
 MOV #0,@DEVBUF ; LOAD ZEROES IN  
 INC R1 ; UPDATE LINE NUMBER  
 CMP R1,#7 ; ALL LINES DONE  
 BLE 10\$ ; NO  
 INC CHAR ; SELECT NEXT CHAR  
 CMP CHAR,#177 ; ALL DONE  
 BLE 5\$ ; NO

NOW FILL THE SCREEN

| Address | Op Code | Op 1 | Op 2 | Op 3 | Op 4 | Op 5 | Op 6 | Op 7 | Op 8 | Op 9 | Op 10 | Op 11 | Op 12 | Op 13 | Op 14 | Op 15 | Op 16 | Op 17 | Op 18 | Op 19 | Op 20 | Op 21 | Op 22 | Op 23 | Op 24 | Op 25 | Op 26 | Op 27 | Op 28 | Op 29 | Op 30 | Op 31 | Op 32 | Op 33 | Op 34 | Op 35 | Op 36 | Op 37 | Op 38 | Op 39 | Op 40 | Op 41 | Op 42 | Op 43 | Op 44 | Op 45 | Op 46 | Op 47 | Op 48 | Op 49 | Op 50 | Op 51 | Op 52 | Op 53 | Op 54 | Op 55 | Op 56 | Op 57 | Op 58 | Op 59 | Op 60 | Op 61 | Op 62 | Op 63 | Op 64 | Op 65 | Op 66 | Op 67 | Op 68 | Op 69 | Op 70 | Op 71 | Op 72 | Op 73 | Op 74 | Op 75 | Op 76 | Op 77 | Op 78 | Op 79 | Op 80 | Op 81 | Op 82 | Op 83 | Op 84 | Op 85 | Op 86 | Op 87 | Op 88 | Op 89 | Op 90 | Op 91 | Op 92 | Op 93 | Op 94 | Op 95 | Op 96 | Op 97 | Op 98 | Op 99 | Op 100 | Op 101 | Op 102 | Op 103 | Op 104 | Op 105 | Op 106 | Op 107 | Op 108 | Op 109 | Op 110 | Op 111 | Op 112 | Op 113 | Op 114 | Op 115 | Op 116 | Op 117 | Op 118 | Op 119 | Op 120 | Op 121 | Op 122 | Op 123 | Op 124 | Op 125 | Op 126 | Op 127 | Op 128 | Op 129 | Op 130 | Op 131 | Op 132 | Op 133 | Op 134 | Op 135 | Op 136 | Op 137 | Op 138 | Op 139 | Op 140 | Op 141 | Op 142 | Op 143 | Op 144 | Op 145 | Op 146 | Op 147 | Op 148 | Op 149 | Op 150 | Op 151 | Op 152 | Op 153 | Op 154 | Op 155 | Op 156 | Op 157 | Op 158 | Op 159 | Op 160 | Op 161 | Op 162 | Op 163 | Op 164 | Op 165 | Op 166 | Op 167 | Op 168 | Op 169 | Op 170 | Op 171 | Op 172 | Op 173 | Op 174 | Op 175 | Op 176 | Op 177 | Op 178 | Op 179 | Op 180 | Op 181 | Op 182 | Op 183 | Op 184 | Op 185 | Op 186 | Op 187 | Op 188 | Op 189 | Op 190 | Op 191 | Op 192 | Op 193 | Op 194 | Op 195 | Op 196 | Op 197 | Op 198 | Op 199 | Op 200 | Op 201 | Op 202 | Op 203 | Op 204 | Op 205 | Op 206 | Op 207 | Op 208 | Op 209 | Op 210 | Op 211 | Op 212 | Op 213 | Op 214 | Op 215 | Op 216 | Op 217 | Op 218 | Op 219 | Op 220 | Op 221 | Op 222 | Op 223 | Op 224 | Op 225 | Op 226 | Op 227 | Op 228 | Op 229 | Op 230 | Op 231 | Op 232 | Op 233 | Op 234 | Op 235 | Op 236 | Op 237 | Op 238 | Op 239 | Op 240 | Op 241 | Op 242 | Op 243 | Op 244 | Op 245 | Op 246 | Op 247 | Op 248 | Op 249 | Op 250 | Op 251 | Op 252 | Op 253 | Op 254 | Op 255 | Op 256 | Op 257 | Op 258 | Op 259 | Op 260 | Op 261 | Op 262 | Op 263 | Op 264 | Op 265 | Op 266 | Op 267 | Op 268 | Op 269 | Op 270 | Op 271 | Op 272 | Op 273 | Op 274 | Op 275 | Op 276 | Op 277 | Op 278 | Op 279 | Op 280 | Op 281 | Op 282 | Op 283 | Op 284 | Op 285 | Op 286 | Op 287 | Op 288 | Op 289 | Op 290 | Op 291 | Op 292 | Op 293 | Op 294 | Op 295 | Op 296 | Op 297 | Op 298 | Op 299 | Op 300 | Op 301 | Op 302 | Op 303 | Op 304 | Op 305 | Op 306 | Op 307 | Op 308 | Op 309 | Op 310 | Op 311 | Op 312 | Op 313 | Op 314 | Op 315 | Op 316 | Op 317 | Op 318 | Op 319 | Op 320 | Op 321 | Op 322 | Op 323 | Op 324 | Op 325 | Op 326 | Op 327 | Op 328 | Op 329 | Op 330 | Op 331 | Op 332 | Op 333 | Op 334 | Op 335 | Op 336 | Op 337 | Op 338 | Op 339 | Op 340 | Op 341 | Op 342 | Op 343 | Op 344 | Op 345 | Op 346 | Op 347 | Op 348 | Op 349 | Op 350 | Op 351 | Op 352 | Op 353 | Op 354 | Op 355 | Op 356 | Op 357 | Op 358 | Op 359 | Op 360 | Op 361 | Op 362 | Op 363 | Op 364 | Op 365 | Op 366 | Op 367 | Op 368 | Op 369 | Op 370 | Op 371 | Op 372 | Op 373 | Op 374 | Op 375 | Op 376 | Op 377 | Op 378 | Op 379 | Op 380 | Op 381 | Op 382 | Op 383 | Op 384 | Op 385 | Op 386 | Op 387 | Op 388 | Op 389 | Op 390 | Op 391 | Op 392 | Op 393 | Op 394 | Op 395 | Op 396 | Op 397 | Op 398 | Op 399 | Op 400 | Op 401 | Op 402 | Op 403 | Op 404 | Op 405 | Op 406 | Op 407 | Op 408 | Op 409 | Op 410 | Op 411 | Op 412 | Op 413 | Op 414 | Op 415 | Op 416 | Op 417 | Op 418 |
|---------|---------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
|---------|---------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|



```

1306 003522 004737 011644      12$: JSR    PC,SYNCUP      ; DELAY 2 PERIODS
1307 003526 000240              NOP
1308 003530 004737 011644      JSR    PC,SYNCUP
1309 003534 000240              NOP
1310 003536 010500              MOV    R5,R0      ; GET LINE POSITION..
1311 003540 042700 177770      BIC    #177770,R0 ; GET POS IN CHAR..
1312 003544 052700 000010      BIS    #10,R0
1313 003550 012704 100000      MOV    #100000,R4 ; PUT LINE IN INITIAL POS.
1314 003554 000241              CLC
1315 003556 005737 007622      TST    PAL      ; IF TV-ENCODED...
1316 003562 001401              BEQ    20$
1317 003564 000261              SEC      ; DOUBLE-UP LINE
1318
1319 003566 006004      20$:  ROR    R4      ; SHIFT LINE TO POSN.
1320 003570 005300      DEC    R0
1321 003572 001375      BNE    20$
1322 003574 000304      SWAB   R4      ; SWITCH FOR CLEAN-UP TRAILER
1323 003576 010500      MOV    R5,R0      ; GET CURRENT CHAR..
1324 003600 042700 000007      BIC    #7,R0      ; WITHOUT LINE..
1325 003604 162700 000010      SUB    #10,R0      ; GET LAST CHAR...
1326 003610 100003      BPL    22$      ; VALID CHAR?
1327 003612 005000      CLR    R0      ; NO, START AT FIRST..
1328 003614 105004      CLRB   R4      ; CLEAR LINE
1329 003616 000304      SWAB   R4      ; SWAP LINES OVER..
1330
1331 003620 020027 000520      22$:  CMP    R0,#42.*10 ; OVER TOP?
1332 003624 103402          BCS    23$      ; NO..
1333 003626 042704 177400      BIC    #177400,R4 ; ELSE CLEAR HIGH ORDER...
1334
1335      ; NOW WE HAVE 2 CHARACTERS TO LOAD. EACH CHANGE IN THE LINE
1336      ; INVOLVES LOADING 2 ADJACENT CHARACTERS.... THE ONE THE
1337      ; LINE IS/COULD BE SCROLLING OUT OF, AND THE ONE IT IS
1338      ; SCROLLING INTO.
1339
1340 003632 012702 000003      23$:  MOV    #3,R2      ; 3 ROWS OF CHARACTERS.
1341
1342 003636 012703 000010      24$:  MOV    #10,R3      ; 8 BYTES/CHARACTER
1343 003642 052700 140000      BIS    #140000,R0 ; CHAR STORE SEL,VIDEO-ON
1344 003646 010077 003564      MOV    R0,@DEVCSR
1345 003652 032777 002000 003556  BIT    #2000,@DEVCSR ; INHIBIT WRAP-ROUND
1346 003660 001034          BNE    29$
1347
1348 003662 010477 003552      26$:  MOV    R4,@DEVBUF ; LOAD DATUM
1349 003666 005277 003544      INC    @DEVCSR
1350 003672 005303      DEC    R3
1351 003674 001372          BNE    26$      ; FIRST CHARACTER DONE IN PAIR?
1352 003676 032777 002000 003532  BIT    #2000,@DEVCSR ; WRAP-ROUND?
1353 003704 001022          BNE    29$      ; YES...
1354 003706 000304      SWAB   R4      ; YES, GET NEXT DATUM
1355 003710 012703 000010      MOV    #10,R3
1356
1357 003714 010477 003520      27$:  MOV    R4,@DEVBUF ; LOAD DATUM
1358 003720 005277 003512      INC    @DEVCSR
1359 003724 005303      DEC    R3
1360 003726 001372          BNE    27$
1361 003730 000304      SWAB   R4      ; RESTORE ORIGINAL DATUM..

```

VTV30K DIAGNOSTIC MACY11 30A(1052) 08-JAN-81 10:58 PAGE 25-3<sup>H 3</sup>  
 CVVTCA.SRC 08-JAN-81 10:36 TEST5

SEQ 0033

|      |        |        |        |        |               |               |   |                                      |
|------|--------|--------|--------|--------|---------------|---------------|---|--------------------------------------|
| 1362 | 003732 | 062700 | 000530 |        | ADD           | #43.*10,R0    | : | UPDATE TO NEXT LINE..                |
| 1363 | 003736 | 032777 | 002000 | 003472 | BIT           | #2000,@DEVCSR | : | OVER TOP?                            |
| 1364 | 003744 | 001002 |        |        | BNE           | 29\$          | : | YES...DONT WRAP AROUND!              |
| 1365 | 003746 | 005302 |        |        | DEC           | R2            | : | ANY MORE...                          |
| 1366 | 003750 | 001332 |        |        | BNE           | 24\$          | : | YES..                                |
| 1367 |        |        |        |        |               |               |   |                                      |
| 1368 | 003752 | 005205 |        |        | 29\$: INC     | R5            | : |                                      |
| 1369 | 003754 | 020527 | 000530 |        | CMP           | R5,#43.*10    | : | ALL DONE?                            |
| 1370 | 003760 | 001243 |        |        | BNE           | 10\$          | : | YES...                               |
| 1371 |        |        |        |        |               |               |   |                                      |
| 1372 | 003762 | 104470 |        |        | 30\$: TRAP+70 |               | : | LOOP?                                |
| 1373 | 003764 | 003466 |        |        | 5\$           |               |   |                                      |
| 1374 | 003766 | 004737 | 012040 |        | JSR           | PC.WAITS      | : | WAIT A SECOND...                     |
| 1375 | 003772 | 000001 |        |        | 1             |               |   |                                      |
| 1376 |        |        |        |        |               |               |   | ....THEN PROCEED TO VERTICAL-LINE... |



```

1378 003774 005005 35$: CLR R5 ; POSITION MARKER FOR LINE.
1379 003776 012704 004000 MOV #4000,R4 ; ZERO CHSR
1380
1381 004002 010500 40$: MOV R5,R0 ; GET CHARACTER NUMBER
1382 004004 006300 ASL R0
1383 004006 006300 ASL R0
1384 004010 105000 CLR R0 ; ZERO COLOUR+BLINK
1385 004012 052700 000047 BIS #47,R0 ; WHITE ON BLUE
1386 004016 010577 003414 MOV R5,@DEVCSR ; GET PICTURE ADDRESS
1387 004022 010077 003412 MOV R0,@DEVBUF ; LOAD ELEMENT
1388 004026 005205 INC R5 ; NEXT PICTURE CELL!
1389 004030 005304 DEC R4 ; ANY MORE PICTURE?
1390 004032 001363 BNE 40$ ; ..YES..
1391 004034 012705 002000 MOV #2000,R5 ; NOW CLEAR CHARACTER STRE
1392 004040 012777 100000 003370 MOV #100000,@DEVCSR
1393
1394 004046 005077 003366 45$: CLR @DEVBUF ; ZERO BYTE IN CHSR
1395 004052 005277 003360 INC @DEVCSR
1396 004056 005305 DEC R5 ; ANY MORE?
1397 004060 001372 BNE 45$ ; YES..
1398
1399 004062 105737 177560 50$: TSTB TKS ; ANY ABORT?
1400 004066 100012 BPL 52$ ; NO
1401 004070 004737 014206 JSR PC,READ
1402 004074 020027 000040 CMP R0,#' ; ABORT?
1403 004100 001440 BEQ 70$ ; ..YES...
1404 004102 020027 000007 CMP R0,#'G&37 ; BELL?
1405 004106 001002 BNE 52$ ; NO..
1406 004110 004737 013546 JSR PC,MONIT ; CALL MONITOR.
1407
1408 004114 004737 011644 52$: JSR PC,SYNCUP ; WAIT..
1409 004120 000240 NOP
1410 004122 004737 011644 JSR PC,SYNCUP
1411 004126 000240 NOP
1412 004130 010500 MOV R5,R0 ; GET CHARACTER..
1413 004132 005300 DEC R0 ; GET LAST FOR CLEARING
1414 004134 100406 BMI 53$
1415 004136 052700 140000 BIS #140000,R0 ; VIDEO ON, CHSR SELECT.
1416 004142 010077 003270 MOV R0,@DEVCSR ; GET ADDRESS..
1417 004146 005077 003266 CLR @DEVBUF ; ZERO CHARACTER BYTE.
1418
1419 004152 010500 53$: MOV R5,R0 ; GET CURRENT CHAR.
1420 004154 052700 140000 BIS #140000,R0
1421 004160 010077 003252 MOV R0,@DEVCSR ; LOAD ADDRESS
1422 004164 012777 000377 003246 MOV #377,@DEVBUF ; LIGHT LINE.
1423 004172 005205 INC R5 ; STEP TO NEXT LINE.
1424 004174 020527 000400 CMP R5,#400 ; NEXT LINE?
1425 004200 001330 BNE 50$ ; YES, LOOP
1426
1427 004202 104420 70$: TRAP+20
1428 004204 003774 35$ ; LOOP ON SUBTEST
1429 004206 004737 012040 JSR PC,WAITS
1430 004212 000001 1 ; WAIT FOR 1 SEC
1431
1432
1433 ; END OF TEST..

```

|      |        |        |        |        |  |         |            |  |                     |
|------|--------|--------|--------|--------|--|---------|------------|--|---------------------|
| 1434 |        |        |        |        |  |         |            |  |                     |
| 1435 | 004214 | 104404 |        |        |  | T05XIT: | TRAP+4     |  | : LOOP ON TEST?     |
| 1436 | 004216 | 003134 |        |        |  |         | T0501      |  |                     |
| 1437 | 004220 | 032777 | 000100 | 003254 |  | BIT     | #100, @SWR |  | : PRESELECTED TEST? |
| 1438 | 004226 | 001402 |        |        |  | BEQ     | TEST6      |  |                     |
| 1439 | 004230 | 000137 | 001326 |        |  | JMP     | START2     |  |                     |



```

1441          .SBTTL TEST6
1442          :
1443          : THIS TEST DISPLAYS A ROTATING CHARACTER PATTERN IN ALL PICTURE
1444          : LOCATIONS, WITH COLOUR WHITE ON BLACK.
1445          :
1446          : THIS CHECKS THE HIGH-ORDER PICTURE STORE BITS (THESE HOLD THE
1447          : SELECTED CHARACTER NUMBER)
1448          :
1449          : ERRORS-
1450          : HIGH ORDER PICTURE STORE BITS (LEVC)
1451          : PICTURE STORE STATISER (LEVD)
1452          :
1453          004234 012737 000006 007454 TEST6: MOV      #6,TESTNO      ; SET TEST NUMBER
1454          004242 104402          TRAP+2
1455          004244 004260          T0601
1456          004246 004737 011624      JSR      PC,TESTR      ; PRINT TESTNUMBER
1457          004252 004737 014734      JSR      PC,TYPOUT
1458          004256 006441          TS6MSG
1459          004260 012737 000005 007506 T0601: MOV      #5,REPCNT
1460          004266 005737 007462      TST      PROM          ; IS THERE A PROM ON BOARD
1461          004272 001037          BNE      T0604          ; YES DON'T LOAD CHSR
1462          :
1463          : RAM CHARACTER SET NEEDS LOADING
1464          :
1465          004274 012700 007624      MOV      #P02CHR,R0      ; GET START OF DATA TO LOAD
1466          004300 005037 007550      CLR      CHAR          ; START AT CHARACTER ZERO
1467          004304 005001          T0602: CLR      R1          ; 8 LINES PER CHARACTER
1468          004306 013702 007550      T0603: MOV      CHAR,R2      ; GET CHAR CODE
1469          004312 006302          ASL      R2
1470          004314 006302          ASL      R2          ; SET IN RIGHT PLACE
1471          004316 006302          ASL      R2
1472          004320 050102          BIS      R1,R2          ; ADD IN LINE NUMBER
1473          004322 052702 100000      BIS      #100000,R2      ; SELECT CHSR
1474          004326 010277 003104      MOV      R2,@DEVCSR      ; LOAD CSR
1475          004332 112003          MOVB     (R0)+,R3          ; GET DATA
1476          004334 000303          SWAB     R3
1477          004336 105003          CLRB     R3          ; ENSURE 8 BITS LOADED
1478          004340 000303          SWAB     R3
1479          004342 010377 003072      MOV      R3,@DEVBUF      ; LOAD DATA OUT
1480          004346 005201          INC      R1          ; UPDATE LINE NUMBER
1481          004350 020127 000007      CMP      R1,#7          ; ALL LINES DONE
1482          004354 003754          BLE      T0603          ; NO
1483          004356 005237 007550      INC      CHAR          ; UPDATE CHAR COUNT
1484          004362 023727 007550 000177 CMP      CHAR,#177      ; ALL DONE
1485          004370 003745          BLE      T0602          ; NOT YET
1486          :
1487          : NOW START PUTTING UP PICTURE
1488          :
1489          004372          T0604: MOV      #40000,VIDEON      ; VIDEO ON
1490          004372 012737 040000 007450 MOV      #0,BAKGND      ; BLACK BACKGROUND
1491          004400 012737 000000 007466 MOV      #7,FORGND      ; WHITE FOREGROUND
1492          004406 012737 000007 007464 MOV      BLINK
1493          004414 005037 007470      CLR      FLASH
1494          004420 005037 007472      CLR      FLASH      ; NO BLINK OR FLASH
1495          004424 012700 007266      MOV      #CHXLST,R0      ; POINT TO START OF LIST
1496          004430 010001          T0605: MOV      R0,R1          ; GET PICTURE STARTER
  
```

```

1497 004432 005037 007476          CLR      CARY          ; START AT TOP LEFT
1498 004436 010102          T0606: MOV     R1,R2          ; GET LINE STARTER
1499 004440 005037 007474          CLR      CARX          ; START OF LINE
1500 004444 011237 007550          T0607: MOV     (R2),CHAR        ; GET CHAR CODE
1501 004450 004737 012522          JSR      PC,LOELEM        ; DISPLAY THE CHARACTER
1502 004454 062702 000002          ADD     #2,R2          ; GET NEXT POINTER
1503 004460 021227 177777          CMP     (R2),#-1        ; END OF TABLE ?
1504 004464 001002          BNE     1$
1505 004466 012702 007266          MOV     #CHXLST,R2      ; YES RESET IT
1506 004472 005237 007474          1$: INC     CARX          ; GET NEXT COORD
1507 004476 023737 007474 007446  CMP     CARX,XMAX        ; END OF ROW ?
1508 004504 003757          BLE     T0607        ; NOT YET
1509 004506 062701 000002          ADD     #2,R1          ; GET NEW LINE STARTER
1510 004512 021127 177777          CMP     (R1),#-1        ; END OF TABLE
1511 004516 001002          BNE     2$
1512 004520 012701 007266          MOV     #CHXLST,R1      ; YES RESET IT
1513 004524 005237 007476          2$: INC     CARY          ; UPDATE ROW
1514 004530 023737 007476 007500  CMP     CARY,YMAX        ; END OF DISPLAY ?
1515 004536 003737          BLE     T0606        ; NOT YET
1516
1517          ; WAIT AROUND
1518
1519 004540 004737 012040          JSR      PC,WAITS          ; WAIT 2 SECONDS
1520 004544 000002          2
1521 004546 005337 007506          DEC     REPCNT          ; ALL DONE
1522 004552 001406          BEQ     T0615          ; YES
1523
1524          ;
1525          ;
1526          ;
1527 004554 062700 000002          ADD     #2,R0          ; NEW STARTER
1528 004560 021027 177777          CMP     (R0),#-1        ; ALL DONE ?
1529 004564 001321          BNE     T0605          ; NO
1530 004566 000701          BR     T0604          ; YES RESET TABLE MIXTURE
1531 004570 104404          T0615: TRAP+4
1532 004572 004372          T0604
1533 004574 032777 000100 002700  BIT     #100,ASWR        ; PRESELECT TEST
1534 004602 001402          BEQ     1$
1535 004604 000137 001326          JMP     START2          ; YES GO DO IT
1536 004610          1$:
1537 004610 005237 007302          INC     DEVFLG          ; UPDATE UNIT COUNT
1538 004614 023737 007302 007304  CMP     DEVFLG,DEVCNT    ; ALL UNITS DONE
1539 004622 002402          BLT     2$
1540 004624 000137 013062          JMP     ENDIT          ; NOT YET
1541 004630 013700 007302          2$: MOV     DEVFLG,R0      ; YES, SIGNAL END OF PASS
1542 004634 005200          INC     R0          ; GET CURRENT UNIT
1543 004636 006300          ASL     R0          ; AS A WORD OFFSET
1544 004640 016037 007306 007436  MOV     CSRO(R0),DEVCSR  ; GET NEW CSR ADDRESS
1545 004646 016037 007310 007440  MOV     DBUFO(R0),DEVBUF ; AND DATA BUFFER
1546 004654 000137 001326          JMP     START2

```



```
1548 .SBTTL TEST7
1549
1550 : THIS TEST DISPLAYS A ROTATING CHARACTER PATTERN IN ALL PICTURE
1551 : LOCATIONS. IN ADDITION THE COLOUR, IS CHANGED FOR EACH COLUMN.
1552
1553 : THIS IS AN EXERCISER ONLY, NOT PART OF THE DIAGNOSTIC ROUTINES
1554
1555 004660 012737 000007 007454 TEST7: MOV #7,TESTNO ; SET TEST NUMBER
1556 004666 104402 TRAP+2
1557 004670 004704 T0701
1558 004672 004737 011624 JSR PC,TESTR ; PRINT TESTNUMBER
1559 004676 004737 014734 JSR PC,TYP0UT
1560 004702 006454 TS7MSG
1561 004704 012737 000200 007506 T0701: MOV #200,REPCNT
1562 004712 005737 007462 TST PROM ; IS THERE A PROM ON BOARD
1563 004716 001037 BNE T0704 ; YES DON'T LOAD CHSR
1564
1565 : RAM CHARACTER SET NEEDS LOADING
1566
1567 004720 012700 007624 MOV #P02CHR,R0 ; GET START OF DATA TO LOAD
1568 004724 005037 007550 CLR CHAR ; START AT CHARACTER ZERO
1569 004730 005001 T0702: CLR R1 ; 8 LINES PER CHARACTER
1570 004732 013702 007550 T0703: MOV CHAR,R2 ; GET CHAR CODE
1571 004736 006302 ASL R2
1572 004740 006302 ASL R2 ; SET IN RIGHT PLACE
1573 004742 006302 ASL R2
1574 004744 050102 BIS R1,R2 ; ADD IN LINE NUMBER
1575 004746 052702 100000 BIS #100000,R2 ; SELECT CHSR
1576 004752 010277 002460 MOV R2,@DEVCSR ; LOAD CSR
1577 004756 112003 MOVMB (R0)+,R3 ; GET DATA
1578 004760 000303 SWAB R3
1579 004762 105003 CLRB R3 ; ENSURE 8 BITS LOADED
1580 004764 000303 SWAB R3
1581 004766 010377 002446 MOV R3,@DEVBUF ; LOAD DATA OUT
1582 004772 005201 INC R1 ; UPDATE LINE NUMBER
1583 004774 020127 000007 CMP R1,#7 ; ALL LINES DONE
1584 005000 003754 BLE T0703 ; NO
1585 005002 005237 007550 INC CHAR ; UPDATE CHAR COUNT
1586 005006 023727 007550 000177 CMP CHAR,#177 ; ALL DONE
1587 005014 003745 BLE T0702 ; NOT YET
1588
1589 : NOW START PUTTING UP PICTURE
1590
1591 005016 T0704: MOV #40000,VIDEON ; ENABLE VIDEO
1592 005016 012737 040000 007450 MOV #0,BASE1 ; START AT CHARAACTER 0
1593 005024 012737 000000 007460 T0705: CLR CARY ; START AT FIRST ROW
1594 005032 005037 007476 MOV BASE1,BASE ; SET THIS PICTURE START
1595 005036 013737 007460 007456 T0706: MOV BASE,CHAR ; SET START FOR THIS ROW
1596 005044 013737 007456 007550 T0707: CLR CARY ; START OF COLUMN
1597 005052 005037 007474 CLR FLASH ; NO FLASH
1598 005056 005037 007472 CLR BLINK ; NO BLINK
1599 005062 005037 007470 MOV #1,BAKGND ; RED ON
1600 005066 012737 000001 007466 T0710: MOV #1,FORGND ; RED
1601 005074 012737 000001 007464 T0711: JSR PC,LOELEM ; DISPLAY THE CHARACTER
1602 005102 004737 012522 INC CHAR ; SELECT NEXT ONE
1603 005106 005237 007550
```

|      |        |        |        |        |        |               |   |                          |
|------|--------|--------|--------|--------|--------|---------------|---|--------------------------|
| 1604 | 005112 | 042737 | 177600 | 007550 | BIC    | #177600,CHAR  | : | IN RANGE 0-177           |
| 1605 | 005120 | 005237 | 007474 |        | INC    | CARX          | : | GET NEXT COORD           |
| 1606 | 005124 | 023737 | 007474 | 007446 | CMP    | CARX,XMAX     | : | END OF ROW ?             |
| 1607 | 005132 | 003432 |        |        | BLE    | T0712         | : | NOT YET                  |
| 1608 | 005134 | 005237 | 007456 |        | INC    | BASE          | : | ELSE SET NEW ROW STARTER |
| 1609 | 005140 | 042737 | 177600 | 007456 | BIC    | #177600,BASE  | : | IN RANGE 0-177           |
| 1610 | 005146 | 005237 | 007476 |        | INC    | CARY          | : | UPDATE ROW               |
| 1611 | 005152 | 023737 | 007476 | 007500 | CMP    | CARY,YMAX     | : | END OF DISPLAY ?         |
| 1612 | 005160 | 003731 |        |        | BLE    | T0706         | : | NOT YET                  |
| 1613 | 005162 | 005237 | 007460 |        | INC    | BASE1         | : | SET NEW PICTURE STARTER  |
| 1614 | 005166 | 042737 | 177600 | 007460 | BIC    | #177600,BASE1 | : | IN RANGE 0-177           |
| 1615 |        |        |        |        |        |               |   |                          |
| 1616 |        |        |        |        | :      | WAIT AROUND   |   |                          |
| 1617 |        |        |        |        | :      |               |   |                          |
| 1618 | 005174 | 004737 | 012040 |        | JSR    | PC,WAITS      | : | WAIT 1 SECONDS           |
| 1619 | 005200 | 000001 |        |        | 1      |               |   |                          |
| 1620 | 005202 | 005337 | 007506 |        | DEC    | REPCNT        |   |                          |
| 1621 | 005206 | 001422 |        |        | BEQ    | T0715         |   |                          |
| 1622 | 005210 | 000137 | 005032 |        | JMP    | T0705         | : | YES                      |
| 1623 | 005214 | 000137 | 005254 |        | JMP    | T0715         | : | NO EXIT TEST             |
| 1624 |        |        |        |        | :      |               |   |                          |
| 1625 | 005220 | 006337 | 007464 |        | T0712: | ASL           | : | GET NEW FOREGROUND       |
| 1626 | 005224 | 023727 | 007464 | 000004 | CMP    | FORGND,#4     | : | ALL DONE                 |
| 1627 | 005232 | 003723 |        |        | BLE    | T0711         | : | NO                       |
| 1628 | 005234 | 005237 | 007466 |        | INC    | BAKGND        | : | GET NEXT BACK GROUND     |
| 1629 | 005240 | 023727 | 007466 | 000007 | CMP    | BAKGND,#7     | : | ALL DONE                 |
| 1630 | 005246 | 003712 |        |        | BLE    | T0710         | : | NO                       |
| 1631 | 005250 | 000137 | 005056 |        | JMP    | T0707         | : | YES                      |
| 1632 | 005254 | 104404 |        |        | T0715: | TRAP+4        |   |                          |
| 1633 | 005256 | 004704 |        |        | T0701  |               |   |                          |
| 1634 | 005260 | 000137 | 001326 |        | JMP    | START2        | : | YES GO DO IT             |



```

1636                                     .SBTTL TEST10
1637                                     :
1638                                     : THIS TEST PRINTS THE CURRENT ACCUMULATION OF ERRORS
1639                                     :
1640 005264 012737 000010 007454 TEST10: MOV #10,TESTNO ; SET TEST NUMBER
1641 005272 104402 TRAP+2
1642 005274 005310 T1001
1643 005276 004737 011624 JSR PC,TESTR ; PRINT TEST NUMBER
1644 005302 004737 014734 JSR PC,TYPOUT
1645 005306 006462 TS10MS
1646 005310 004737 014700 T1001: JSR PC,CRLF ; NEW LINE
1647 005314 004737 014734 JSR PC,TYPOUT ; SAY
1648 005320 007205 EMSG9 ; ERROR COUNT =
1649 005322 013700 007442 MOV ERD11,R0 ; PRINT HI WORD
1650 005326 004737 015024 JSR PC,PROCT
1651 005332 013700 007574 MOV ERDIS,R0 ; AND LO WORD
1652 005336 004737 015024 JSR PC,PROCT
1653 005342 005037 007444 CLR ERRFLG
1654 005346 005037 007574 CLR ERDIS
1655 005352 005037 007442 CLR ERD11
1656 005356 000137 001200 JMP RSTART ; RESTART
  
```

```

1658 .SBTTL TEST11
1659
1660 : THIS TEST DISPLAYS A PICTURE SUITABLE FOR TESTING ADAPTERS.
1661 : 7 BLOCKS OF 4 ROWS EACH OF THE FOLLOWING COLOURS, WITH 1,
1662 : 2, AND 4 MHZ BARS
1663
1664 005362 012737 000011 007454 TEST11: MOV #11,TESTNO ; SET TEST NUMBER
1665 005370 104402 TRAP+2
1666 005372 005400 5$
1667 005374 004737 011624 JSR PC,TESTR ; PRINT TEST NUMBER
1668
1669 005400 005005 5$: CLR R5 ; FIRST PICTURE ELEMENT
1670
1671 005402 010500 10$: MOV R5,R0 ; GET COLOUR
1672 005404 000300 SWAB R0
1673 005406 005200 INC R0 ; FIRST COLOUR RED, NOT BLACK
1674 005410 042700 177770 BIC #177770,R0 ; FOREGROUND
1675 005414 010501 MOV R5,R1 ; GET CHARACTER
1676 005416 042701 177700 BIC #177700,R1
1677 005422 010577 002010 MOV R5,@DEVCSR ; SET POSITION
1678 005426 020127 000014 CMP R1,#12. ; 1/2 MHZ?
1679 005432 103013 BCC 20$ ; NO..
1680 005434 042701 000076 BIC #76,R1 ; EVEN?
1681 005440 001406 BEQ 12$ ; YES...
1682 005442 010002 MOV R0,R2 ; SET F/B SAME..
1683 005444 006302 ASL R2 ; COPY TO BACKGROUND
1684 005446 006302 ASL R2
1685 005450 006302 ASL R2
1686 005452 050200 BIS R2,R0 ; COPY BACK..
1687 005454 000422 BR 77$
1688
1689 005456 005000 12$: CLR R0
1690 005460 000420 BR 77$ ; LOAD IT..
1691
1692 005462 020127 000030 20$: CMP R1,#24. ; 1 MHZ?
1693 005466 103001 BCC 40$ ; NO..
1694 005470 000414 BR 77$
1695
1696 005472 020127 000044 40$: CMP R1,#36. ; 2 MHZ?
1697 005476 103003 BCC 60$ ; NO..
1698 005500 052700 011400 BIS #19.*400,R0 ; LOAD 2 MHZ..
1699 005504 000406 BR 77$
1700
1701 005506 020127 000060 60$: CMP R1,#48. ; 4 MHZ?
1702 005512 103005 BCC 100$ ; NO, DISCARD..
1703 005514 052700 011000 BIS #18.*400,R0
1704 005520 000400 BR 77$
1705
1706 005522 010077 001712 77$: MOV R0,@DEVBUF ; LOAD CHARACTER
1707
1708 005526 005205 100$: INC R5
1709 005530 020527 004000 CMP R5,#4000 ; DONE?
1710 005534 001322 BNE 10$ ; NO..
1711
1712 005536 005737 007462 TST PROM ; LOAD CHSR?
1713 005542 001015 BNE 120$ ; NO..

```



```

1714
1715 005544 012700 007624 ;
1716 005550 012701 002000 MOV #P02CHR,R0 ; VAS...
1717 005554 012777 140000 001654 MOV #2000,R1
1718 MOV #140000,@DEVCSR
1719 005562 112077 001652 ;102$: MOVB (R0)+,@DEVBUF ; LOAD BYTE INTO CHSR
1720 005566 005277 001644 INC @DEVCSR
1721 005572 005301 DEC R1
1722 005574 001372 BNE 102$
1723
1724 005576 012777 040000 001632 ;120$: MOV #40000,@DEVCSR ; TURN ON VIDEO...
1725 005604 000137 001200 JMP RSTART

```

```
1727 .SBTTL ASCII STRINGS
1728
1729 .NLIST BEX
1730
1731 005610 046533 031467 037466 SET56: .ASCII $[M736?/M737?..a$
1732 005627 133 027124 027126 SETENC: .ASCII /[T.V. ENCODED?(Y OR N)..a/
1733 005660 042133 041505 046511 NUMDEV: .ASCII /[DECIMAL NUMBER OF UNITS(1)..a/
1734 005716 042533 052116 051105 CHRCOD: .ASCII /[CENTER IN OCTAL THE CHARACTER/
1735 005753 133 047503 042504 .ASCII /[CODE TO BE DISPLAYED....a/
1736 006005 133 052126 031526 GOMSG: .ASCII /[VTV30K DIAGNOSTIC/
1737 006027 133 052123 051101 .ASCII /[START ADDRESS 1000, RESTART 1200/
1738 006070 050133 042514 051501 .ASCII /[PLEASE READ THE LISTING BEFORE PROCEEDING!a/
1739 006144 052133 042510 042040 GOMS1: .ASCII /[THE DEVICES TO BE TESTED MUST/
1740 006202 044133 053101 020105 .ASCII /[HAVE THEIR ADDRESSES IN ASCENDING ORDER, AND/
1741 006257 133 040510 042526 .ASCII /[HAVE THE SAME CHARACTERISTICSa/
1742 006317 133 042523 042514 WMSG: .ASCII /[SELECT SWITCH REGISTER OPTIONSa/
1743 006360 041440 051123 100 TS0MSG: .ASCII / CSRa/
1744 006365 040 046106 020104 TS1MSG: .ASCII / FLD VIDEO BELL TSTa/
1745 006411 040 047503 040114 TS2MSG: .ASCII / COLa/
1746 006416 041440 051125 047523 TS3MSG: .ASCII / CURSORa/
1747 006426 043040 041055 100 TS4MSG: .ASCII / F-Ba/
1748 006433 040 044103 051123 TS5MSG: .ASCII / CHSRa/
1749 006441 040 044520 020103 TS6MSG: .ASCII / PIC STOREa/
1750 006454 042440 042530 040122 TS7MSG: .ASCII / EXERa/
1751 006462 042533 051122 051440 TS10MS: .ASCII /[ERR SUMa/
1752 006473 133 051524 040124 TESMSG: .ASCII /[TSTa/
1753 006500 042533 042116 047440 PASMSG: .ASCII /[END OF PASSa/
1754 006516 052133 040522 020120 ILVMSG: .ASCII /[TRAP TO ILLEGAL VECTOR a/
1755 006550 043133 047522 020115 FRMSG: .ASCII /[FROM ADDRESS a/
1756 006570 052133 050131 020105 REDMES: .ASCII /[TYPE CNTRL-C TO CONTINUE a/
1757 006623 133 053523 020122 SWMSG: .ASCII /[SWR = a/
1758 006635 133 020044 040040 MODADM: .ASCII /[S a/
1759 006642 027440 040040 MODSPA: .ASCII ? / a?
1760 006646 025133 020040 100 MODPRM: .ASCII /[* a/
1761 006653 133 044506 051522 BMSG: .ASCII /[FIRST BUS ADDRESS IS .....a/
1762 006707 133 047111 040526 ODMSG: .ASCII /[INVALID ADDRESS a/
1763 006732 047133 047117 042440 NMSG: .ASCII /[NON EXISTANT ADDRESS a/
1764 006762 042133 043105 052501 NODEFM: .ASCII /[DEFAULT SETTINGS ARE NOT ALLOWED/
1765 007023 133 046120 040505 .ASCII /[PLEASE RE-ENTER THE VALUE .....a/
1766 007064 042533 040043 EMSG1: .ASCII /[Ea/
1767 007070 020040 052101 050040 EMSG2: .ASCII / AT PC a/
1768 007102 043533 047517 020104 EMSG3: .ASCII /[GOOD : a/
1769 007113 040 020040 040502 EMSG4: .ASCII / BAD :a/
1770 007124 042133 052101 020101 EMSG5: .ASCII /[DATA : a/
1771 007135 040 020040 042101 EMSG6: .ASCII / ADDRESS :a/
1772 007153 133 052123 052101 EMSG7: .ASCII /[STATUS :a/
1773 007166 041533 046101 042514 EMSG8: .ASCII /[CALLED FROM :a/
1774 007205 040 020040 051105 EMSG9: .ASCII / ERROR COUNT = a/
1775 007230 .EVEN
1776 007230 000000 000000 000000 OCTNUM: .WORD 0,0,0
1777 007236 100 000 .BYTE 100,0
1778 007240 040502 042523 030061 BASE11: .ASCII /BASE10a/
1779 007250 .EVEN
1780 007250 000000 000000 000000 DECMSG: .WORD 0,0,0
1781 007256 000000 000000 000000 OCTMSG: .WORD 0,0,0,0
1782 .EVEN
```



VTV30K DIAGNOSTIC MACY11 30A(1052) 08-JAN-81 10:58 PAGE 31-1<sup>F</sup> 4  
CVTCA.SRC 08-JAN-81 10:36 ASCII STRINGS

SEQ 0044

1783

.LIST BEX

|      |        |        |        |        |                                  |
|------|--------|--------|--------|--------|----------------------------------|
| 1785 |        |        |        |        | .SBTTL PROGRAM VARIABLES         |
| 1786 |        |        |        |        |                                  |
| 1787 | 007266 | 000021 | 000042 | 000104 | CHXLST: .WORD 21,42,104,10,40,-1 |
|      | 007274 | 000010 | 000040 | 177777 |                                  |
| 1788 | 007302 | 000000 |        |        | DEVFLG: 0                        |
| 1789 | 007304 | 000001 |        |        | DEVCNT: 1                        |
| 1790 | 007306 | 174000 |        |        | CSR0: 174000                     |
| 1791 | 007310 | 174002 |        |        | DBUF0: 174002                    |
| 1792 | 007312 | 000000 |        |        | CSR1: 0                          |
| 1793 | 007314 | 000000 |        |        | DBUF1: 0                         |
| 1794 | 007316 | 000000 |        |        | CSR2: 0                          |
| 1795 | 007320 | 000000 |        |        | DBUF2: 0                         |
| 1796 | 007322 | 000000 |        |        | CSR3: 0                          |
| 1797 | 007324 | 000000 |        |        | DBUF3: 0                         |
| 1798 | 007326 | 000000 |        |        | CSR4: 0                          |
| 1799 | 007330 | 000000 |        |        | DBUF4: 0                         |
| 1800 | 007332 | 000000 |        |        | CSR5: 0                          |
| 1801 | 007334 | 000000 |        |        | DBUF5: 0                         |
| 1802 | 007336 | 000000 |        |        | CSR6: 0                          |
| 1803 | 007340 | 000000 |        |        | DBUF6: 0                         |
| 1804 | 007342 | 000000 |        |        | CSR7: 0                          |
| 1805 | 007344 | 000000 |        |        | DBUF7: 0                         |
| 1806 | 007346 | 000000 |        |        | CSR8: 0                          |
| 1807 | 007350 | 000000 |        |        | DBUF8: 0                         |
| 1808 | 007352 | 000000 |        |        | CSR9: 0                          |
| 1809 | 007354 | 000000 |        |        | DBUF9: 0                         |
| 1810 | 007356 | 000000 |        |        | CSR10: 0                         |
| 1811 | 007360 | 000000 |        |        | DBUF10: 0                        |
| 1812 | 007362 | 000000 |        |        | CSR11: 0                         |
| 1813 | 007364 | 000000 |        |        | DBUF11: 0                        |
| 1814 | 007366 | 000000 |        |        | CSR12: 0                         |
| 1815 | 007370 | 000000 |        |        | DBUF12: 0                        |
| 1816 | 007372 | 000000 |        |        | CSR13: 0                         |
| 1817 | 007374 | 000000 |        |        | DBUF13: 0                        |
| 1818 | 007376 | 000000 |        |        | CSR14: 0                         |
| 1819 | 007400 | 000000 |        |        | DBUF14: 0                        |
| 1820 | 007402 | 000000 |        |        | CSR15: 0                         |
| 1821 | 007404 | 000000 |        |        | DBUF15: 0                        |
| 1822 | 007406 | 000000 |        |        | CSR16: 0                         |
| 1823 | 007410 | 000000 |        |        | DBUF16: 0                        |
| 1824 | 007412 | 000000 |        |        | CSR17: 0                         |
| 1825 | 007414 | 000000 |        |        | DBUF17: 0                        |
| 1826 | 007416 | 000000 |        |        | CSR18: 0                         |
| 1827 | 007420 | 000000 |        |        | DBUF18: 0                        |
| 1828 | 007422 | 000000 |        |        | CSR19: 0                         |
| 1829 | 007424 | 000000 |        |        | DBUF19: 0                        |
| 1830 | 007426 | 000000 |        |        | CSR20: 0                         |
| 1831 | 007430 | 000000 |        |        | DBUF20: 0                        |
| 1832 | 007432 | 000000 | 000000 |        | .WORD 0,0                        |
| 1833 | 007436 | 000000 |        |        | DEVCSR: 0                        |
| 1834 | 007440 | 000000 |        |        | DEVBUF: 0                        |
| 1835 | 007442 | 000000 |        |        | ERRDI1: 0                        |
| 1836 | 007444 | 000000 |        |        | ERRFLG: 0                        |
| 1837 | 007446 | 000000 |        |        | XMAX: 0                          |
| 1838 | 007450 | 000000 |        |        | VIDEON: 0                        |
| 1839 | 007452 | 000000 |        |        | WAIT1: 0                         |



|      |        |        |         |             |
|------|--------|--------|---------|-------------|
| 1840 | 007454 | 000000 | TESTNO: | 0           |
| 1841 | 007456 | 000000 | BASE:   | 0           |
| 1842 | 007460 | 000000 | BASE1:  | 0           |
| 1843 | 007462 | 000000 | PROM:   | 0           |
| 1844 | 007464 | 000000 | FORGND: | 0           |
| 1845 | 007466 | 000000 | BAKGND: | 0           |
| 1846 | 007470 | 000000 | BLINK:  | 0           |
| 1847 | 007472 | 000000 | FLASH:  | 0           |
| 1848 | 007474 | 000000 | CARX:   | 0           |
| 1849 | 007476 | 000000 | CARY:   | 0           |
| 1850 | 007500 | 000000 | YMAX:   | 0           |
| 1851 | 007502 | 177570 | SWR:    | HSWR        |
| 1852 | 007504 | 000000 | SSWR:   | 0           |
| 1853 | 007506 | 000000 | REPCNT: | 0           |
| 1854 | 007510 | 000137 | JM600:  | JMP @#START |
| 1855 | 007514 | 000001 | FSTCNT: | 1           |
| 1856 | 007516 | 000000 | TRPARG: | 0           |
| 1857 | 007520 | 000000 | TRPSEL: | 0           |
| 1858 | 007522 | 000000 | TRPMEM: | 0           |
| 1859 | 007524 | 000000 | SAVPC:  | 0           |
| 1860 | 007526 | 000000 | SAVPC1: | 0           |
| 1861 | 007530 | 000000 | TYPOTA: | 0           |
| 1862 | 007532 | 000000 | RAND:   | 0           |
| 1863 | 007534 | 000000 | TYPD1:  | 0           |
| 1864 | 007536 | 000000 | MODADR: | 0           |
| 1865 | 007540 | 000000 | MODSAV: | 0           |
| 1866 | 007542 | 000000 | BASADD: | 0           |
| 1867 | 007544 | 000000 | TRPERR: | 0           |
| 1868 | 007546 | 000000 | PARITY: | 0           |
| 1869 | 007550 | 000000 | CHAR:   | 0           |
| 1870 | 007552 | 052525 | RANDN:  | 52525       |
| 1871 | 007554 | 000001 | RANSEL: | 1           |
| 1872 | 007556 | 000006 | RANMTA: | 6           |
| 1873 | 007560 | 052525 | RANST:  | 52525       |
| 1874 | 007562 | 000000 | GOOD:   | 0           |
| 1875 | 007564 | 000000 | BAD:    | 0           |
| 1876 | 007566 | 000000 | DATA:   | 0           |
| 1877 | 007570 | 000000 | STATUS: | 0           |
| 1878 | 007572 | 000000 | ADDRES: | 0           |
| 1879 | 007574 | 000000 | ERRDIS: | 0           |
| 1880 | 007576 | 000000 | ERRARG: | 0           |
| 1881 | 007600 | 000000 | CALLPC: | 0           |
| 1882 | 007602 | 000000 | CNVFLG: | 0           |
| 1883 | 007604 | 000000 | STRADD: | 0           |
| 1884 | 007606 | 000000 | STRLEN: | 0           |
| 1885 | 007610 | 000000 | LOWCHR: | 0           |
| 1886 | 007612 | 000000 | UPPCHR: | 0           |
| 1887 | 007614 | 000000 | RUBFLG: | 0           |
| 1888 | 007616 | 000000 | RANDC:  | 0           |
| 1889 | 007620 | 000000 | FSAVPW: | 0           |
| 1890 | 007622 | 000000 | PAL:    | 0           |

.NLIST BEX

.. THIS CHARACTER SET IS OVERLAYED BY THE RELEVANT CHARACTER SET  
.. FOR ENCODED OR UNENCODED DEVICES.

P02CHR:

|      |        |     |     |     |                                              |
|------|--------|-----|-----|-----|----------------------------------------------|
| 1892 |        |     |     |     |                                              |
| 1893 |        |     |     |     |                                              |
| 1894 |        |     |     |     |                                              |
| 1895 |        |     |     |     |                                              |
| 1896 | 007624 |     |     |     |                                              |
| 1897 | 007624 | 360 | 360 | 360 | .BYTE 360,360,360,360,360,360,360,360        |
| 1898 | 007634 | 010 | 010 | 010 | .BYTE 010,010,010,010,010,010,010,010        |
| 1899 | 007644 | 030 | 030 | 030 | .BYTE 030,030,030,030,030,030,030,030        |
| 1900 | 007654 | 074 | 074 | 074 | .BYTE 074,074,074,074,074,074,074,074        |
| 1901 | 007664 | 000 | 000 | 000 | .BYTE 000,000,000,000,377,000,000,000        |
| 1902 | 007674 | 000 | 000 | 000 | .BYTE 000,000,000,000,377,377,000,000        |
| 1903 | 007704 | 000 | 000 | 377 | .BYTE 000,000,377,377,377,377,000,000        |
| 1904 | 007714 | 010 | 010 | 010 | .BYTE 010,010,010,010,377,010,010,010        |
| 1905 | 007724 | 030 | 030 | 030 | .BYTE 030,030,030,377,377,030,030,030        |
| 1906 | 007734 | 074 | 074 | 377 | .BYTE 074,074,377,377,377,377,074,074 :10    |
| 1907 | 007744 | 001 | 002 | 004 | .BYTE 001,002,004,010,020,040,100,200        |
| 1908 | 007754 | 200 | 100 | 040 | .BYTE 200,100,040,020,010,004,002,001        |
| 1909 | 007764 | 000 | 014 | 022 | .BYTE 000,014,022,040,170,040,100,176        |
| 1910 | 007774 | 201 | 102 | 044 | .BYTE 201,102,044,030,030,044,102,201        |
| 1911 | 010004 | 074 | 176 | 377 | .BYTE 074,176,377,377,377,377,176,074        |
| 1912 | 010014 | 030 | 074 | 176 | .BYTE 030,074,176,377,377,176,074,034        |
| 1913 | 010024 | 010 | 010 | 024 | .BYTE 010,010,024,042,301,042,024,010        |
| 1914 | 010034 | 010 | 010 | 034 | .BYTE 010,010,034,076,377,076,034,010        |
| 1915 | 010044 | 252 | 252 | 252 | .BYTE 252,252,252,252,252,252,252,252        |
| 1916 | 010054 | 314 | 314 | 314 | .BYTE 314,314,314,314,314,314,314,314 :20    |
| 1917 | 010064 | 377 | 000 | 377 | .BYTE 377,000,377,000,377,000,377,000        |
| 1918 | 010074 | 377 | 377 | 000 | .BYTE 377,377,000,000,377,377,000,000        |
| 1919 | 010104 | 377 | 203 | 205 | .BYTE 377,203,205,211,221,241,301,377        |
| 1920 | 010114 | 377 | 201 | 201 | .BYTE 377,201,201,201,201,201,201,377        |
| 1921 | 010124 | 377 | 376 | 374 | .BYTE 377,376,374,370,360,340,300,200        |
| 1922 | 010134 | 377 | 177 | 077 | .BYTE 377,177,077,037,017,007,003,001        |
| 1923 | 010144 | 200 | 300 | 340 | .BYTE 200,300,340,360,370,374,376,377        |
| 1924 | 010154 | 001 | 003 | 007 | .BYTE 001,003,007,017,037,077,177,377        |
| 1925 | 010164 | 003 | 007 | 016 | .BYTE 003,007,016,034,070,160,340,300        |
| 1926 | 010174 | 300 | 340 | 260 | .BYTE 300,340,260,070,034,016,007,003 :30    |
| 1927 | 010204 | 303 | 347 | 176 | .BYTE 303,347,176,074,074,176,347,303        |
| 1928 | 010214 | 000 | 160 | 120 | .BYTE 000,160,120,160,000,000,000,000        |
| 1929 | 010224 | 000 | 000 | 000 | .BYTE 000,000,000,000,000,000,000,000        |
| 1930 | 010234 | 010 | 010 | 010 | .BYTE 010,010,010,010,010,000,010,000        |
| 1931 | 010244 | 024 | 024 | 024 | .BYTE 024,024,024,000,000,000,000,000        |
| 1932 | 010254 | 024 | 024 | 066 | .BYTE 024,024,066,000,066,024,024,000        |
| 1933 | 010264 | 010 | 036 | 040 | .BYTE 010,036,040,034,002,074,010,000        |
| 1934 | 010274 | 000 | 062 | 062 | .BYTE 0, 62, 62, 4, 10, 20, 46, 46           |
| 1935 | 010304 | 000 | 020 | 050 | .BYTE 0, 20, 50, 50, 20, 52, 44, 32          |
| 1936 | 010314 | 000 | 030 | 030 | .BYTE 0, 30, 30, 30, 0, 0, 0, 0 :40(')       |
| 1937 | 010324 | 000 | 010 | 020 | .BYTE 0, 10, 20, 40, 40, 40, 20, 10          |
| 1938 | 010334 | 000 | 010 | 004 | .BYTE 0, 10, 4, 2, 2, 2, 4, 10               |
| 1939 | 010344 | 000 | 052 | 034 | .BYTE 0, 52, 34, 76, 34, 52, 0, 0            |
| 1940 | 010354 | 000 | 000 | 010 | .BYTE 0, 0, 10, 10, 76, 10, 10, 0            |
| 1941 | 010364 | 000 | 000 | 000 | .BYTE 0, 0, 0, 0, 30, 30, 10, 20             |
| 1942 | 010374 | 000 | 000 | 000 | .BYTE 0, 0, 0, 0, 76, 0, 0, 0                |
| 1943 | 010404 | 000 | 000 | 000 | .BYTE 0, 0, 0, 0, 0, 0, 30, 30               |
| 1944 | 010414 | 000 | 002 | 002 | .BYTE 0, 2, 2, 4, 10, 20, 40, 40             |
| 1945 | 010424 | 000 | 034 | 042 | .BYTE 000,034,042,046,052,062,042,034        |
| 1946 | 010434 | 000 | 010 | 030 | .BYTE 000,010,030,010,010,010,010,034 :50(1) |
| 1947 | 010444 | 000 | 034 | 042 | .BYTE 000,034,042,002,034,040,040,076        |



|      |        |     |     |     |       |                                 |        |
|------|--------|-----|-----|-----|-------|---------------------------------|--------|
| 1948 | 010454 | 000 | 034 | 042 | .BYTE | 000.034.042.002.014.002.042.034 |        |
| 1949 | 010464 | 000 | 004 | 014 | .BYTE | 000.004.014.024.044.074.004.004 |        |
| 1950 | 010474 | 000 | 076 | 040 | .BYTE | 0.76.40.74.2.2.42.34            |        |
| 1951 | 010504 | 000 | 014 | 020 | .BYTE | 0.14.20.40.74.42.42.34          |        |
| 1952 | 010514 | 000 | 076 | 002 | .BYTE | 0.76.2.4.10.20.20.20            |        |
| 1953 | 010524 | 000 | 034 | 042 | .BYTE | 0.34.42.42.34.42.42.34          |        |
| 1954 | 010534 | 000 | 034 | 042 | .BYTE | 0.34.42.42.36.2.4.30            |        |
| 1955 | 010544 | 000 | 000 | 030 | .BYTE | 0.0.30.30.0.30.30.0             |        |
| 1956 | 010554 | 000 | 030 | 030 | .BYTE | 0.30.30.0.30.30.10.20           | :60(,) |
| 1957 | 010564 | 000 | 004 | 010 | .BYTE | 0.4.10.20.40.20.10.4            |        |
| 1958 | 010574 | 000 | 000 | 000 | .BYTE | 0.0.0.76.0.76.0.0               |        |
| 1959 | 010604 | 000 | 020 | 010 | .BYTE | 0.20.10.4.2.4.10.20             |        |
| 1960 | 010614 | 000 | 030 | 044 | .BYTE | 0.30.44.4.10.10.0.10            |        |
| 1961 | 010624 | 000 | 074 | 102 | .BYTE | 0.74.102.132.132.104.40.0       |        |
| 1962 | 010634 | 000 | 010 | 024 | .BYTE | 0.10.24.42.42.76.42.42          |        |
| 1963 | 010644 | 000 | 074 | 022 | .BYTE | 0.74.22.22.34.22.22.74          |        |
| 1964 | 010654 | 000 | 034 | 042 | .BYTE | 0.34.42.40.40.40.42.34          |        |
| 1965 | 010664 | 000 | 074 | 022 | .BYTE | 0.74.22.22.22.22.22.74          |        |
| 1966 | 010674 | 000 | 076 | 040 | .BYTE | 0.76.40.40.70.40.40.76          | :70(E) |
| 1967 | 010704 | 000 | 076 | 040 | .BYTE | 0.76.40.40.70.40.40.40          |        |
| 1968 | 010714 | 000 | 036 | 040 | .BYTE | 0.36.40.40.46.42.42.36          |        |
| 1969 | 010724 | 000 | 042 | 042 | .BYTE | 0.42.42.42.76.42.42.42          |        |
| 1970 | 010734 | 000 | 034 | 010 | .BYTE | 0.34.10.10.10.10.10.34          |        |
| 1971 | 010744 | 000 | 002 | 002 | .BYTE | 0.2.2.2.2.2.2.34                |        |
| 1972 | 010754 | 000 | 042 | 044 | .BYTE | 0.42.44.50.60.50.44.42          |        |
| 1973 | 010764 | 000 | 040 | 040 | .BYTE | 0.40.40.40.40.40.40.76          |        |
| 1974 | 010774 | 000 | 042 | 066 | .BYTE | 0.42.66.52.42.42.42.42          |        |
| 1975 | 011004 | 000 | 042 | 062 | .BYTE | 0.42.62.52.46.42.42.42          |        |
| 1976 | 011014 | 000 | 034 | 042 | .BYTE | 0.34.42.42.42.42.42.34          | :80(O) |
| 1977 | 011024 | 000 | 074 | 042 | .BYTE | 0.74.42.42.74.40.40.40          |        |
| 1978 | 011034 | 000 | 034 | 042 | .BYTE | 0.34.42.42.42.52.44.32          |        |
| 1979 | 011044 | 000 | 074 | 042 | .BYTE | 0.74.42.42.74.50.44.42          |        |
| 1980 | 011054 | 000 | 036 | 040 | .BYTE | 0.36.40.40.34.2.2.74            |        |
| 1981 | 011064 | 000 | 076 | 010 | .BYTE | 0.76.10.10.10.10.10.10          |        |
| 1982 | 011074 | 000 | 042 | 042 | .BYTE | 0.42.42.42.42.42.42.34          |        |
| 1983 | 011104 | 000 | 042 | 042 | .BYTE | 0.42.42.42.24.24.10.10          |        |
| 1984 | 011114 | 000 | 042 | 042 | .BYTE | 0.42.42.42.42.52.66.42          |        |
| 1985 | 011124 | 000 | 042 | 042 | .BYTE | 0.42.42.24.10.24.42.42          |        |
| 1986 | 011134 | 000 | 042 | 042 | .BYTE | 0.42.42.24.10.10.10.10          | :90(Y) |
| 1987 | 011144 | 000 | 076 | 002 | .BYTE | 0.76.2.4.10.20.40.76            |        |
| 1988 | 011154 | 000 | 070 | 040 | .BYTE | 000.070.040.040.040.040.070     |        |
| 1989 | 011164 | 000 | 040 | 040 | .BYTE | 000.040.040.020.010.004.002.002 |        |
| 1990 | 011174 | 000 | 016 | 002 | .BYTE | 000.016.002.002.002.002.002.016 |        |
| 1991 | 011204 | 030 | 074 | 176 | .BYTE | 030.074.176.377.030.030.030.030 |        |
| 1992 | 011214 | 010 | 014 | 016 | .BYTE | 010.014.016.377.377.016.014.010 |        |
| 1993 | 011224 | 030 | 030 | 030 | .BYTE | 030.030.030.030.377.176.074.030 |        |
| 1994 | 011234 | 020 | 060 | 160 | .BYTE | 020.060.160.377.377.160.060.020 |        |
| 1995 | 011244 | 000 | 377 | 377 | .BYTE | 000.377.377.377.377.377.377.377 | :100   |
| 1996 | 011254 | 000 | 000 | 377 | .BYTE | 000.000.377.377.377.377.377.377 |        |
| 1997 | 011264 | 000 | 000 | 000 | .BYTE | 000.000.000.377.377.377.377.377 |        |
| 1998 | 011274 | 000 | 000 | 000 | .BYTE | 000.000.000.000.377.377.377.377 |        |
| 1999 | 011304 | 000 | 000 | 000 | .BYTE | 000.000.000.000.000.377.377.377 |        |
| 2000 | 011314 | 000 | 000 | 000 | .BYTE | 000.000.000.000.000.000.377.377 |        |
| 2001 | 011324 | 000 | 000 | 000 | .BYTE | 000.000.000.000.000.000.000.377 |        |
| 2002 | 011334 | 376 | 376 | 376 | .BYTE | 376.376.376.376.376.376.376.376 |        |
| 2003 | 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 |        |     |     |     |       |                                 |      |



2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080

011624 004737 014734  
 011630 006473  
 011632 013700 007454  
 011636 004737 015024  
 011642 000207

```

.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

TESTR: JSR    PC,TYPOUT
        TESMSG
        MOV    TESTNO,R0
        JSR    PC,PROCT
        RTS    PC
        ;TEST NO
        ;PRINT OCTAL TEST NO
        ;EXIT
  
```

2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137

.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 SUCCEES 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



```

2138 011644 004737 013740 SYNCUP: JSR PC,SAVREG ; SAVE GOODIES
2139 011650 005005 CLR R5 ; SETS TO 1 WHEN HIGH LEVEL DETECTED
2140 011652 005004 CLR R4 ; SET SYNC INDETERMINATE
2141 011654 012703 037777 MOV #037777,R3 ; SETUP TIMEOUT
2142 ;
2143 011660 012702 000040 MOV #40,R2 ; SET FILTER TIME CONSTANT
2144 ;
2145 011664 017701 175546 10$: MOV @DEVCSR,R1 ; GET CONTENTS
2146 011670 042701 157777 BIC #157777,R1 ; ISOLATE VIDEO-ON
2147 011674 162701 010000 SUB #010000,R1 ; NORMALISE AROUND ZERO.
2148 011700 100405 BMI 20$ ; UP OR DOWN?
2149 011702 020427 040000 CMP R4,#040000 ; AT END STOP?
2150 011706 001406 BEQ 30$ ; YES..
2151 011710 060104 ADD R1,R4 ; ELSE, UPDATE.
2152 011712 000404 BR 30$
2153 ;
2154 011714 020427 140000 20$: CMP R4,#140000 ; AT END STOP?
2155 011720 001401 BEQ 30$ ; YES..
2156 011722 060104 ADD R1,R4
2157 ;
2158 011724 005704 30$: TST R4 ; SIGN?
2159 011726 100413 BMI 35$ ; ZERO...
2160 011730 020427 030000 CMP R4,#030000 ; VALID ONE?
2161 011734 103417 BCS 40$ ; NO...
2162 011736 012702 000040 MOV #40,R2 ; RESET TIME CONSTANT
2163 011742 020327 037764 CMP R3,#037764 ; READY TO TEST?
2164 011746 103012 BCC 40$ ; NO...
2165 011750 012705 000001 MOV #1,R5 ; SET FLAG..
2166 011754 000407 BR 40$
2167 ;
2168 011756 020427 160000 35$: CMP R4,#160000 ; VALID ZERO?
2169 011762 103004 BCC 40$ ; NO..
2170 011764 005705 TST R5 ; HIGH LEVEL YET?
2171 011766 001017 BNE 50$ ; YES, A-OK!
2172 011770 012702 000040 MOV #40,R2 ; RESET TIME CONSTANT
2173 ;
2174 011774 005302 40$: DEC R2 ; TIME CONSTANT TALLY..
2175 011776 001005 BNE 45$ ; OK SO FAR IF BRANCH..
2176 ;
2177 ; ERROR-
2178 ; CHECK 'LEVB CLOCK H' STABILITY
2179 ; 'LEVB VA17 H'
2179 012000 004737 016666 JSR PC,ERROR ; ELSE NOISY SYNC SIGNAL!
2180 012004 000070 70 ;
2181 012006 000137 012032 JMP NOSYNC ; ERROR EXIT
2182 ;
2183 012012 005303 45$: DEC R3 ; TIMEOUT FOR SYNC?
2184 012014 001323 BNE 10$ ; NO....
2185 ;
2186 ; ERROR-
2187 ; CHECK 'LEVB CLOCK H' (8 MHZ)
2188 ; 'LEVB CLOCK L'
2189 ; 'LEVA VA02 L'
2189 012016 004737 016666 JSR PC,ERROR ; SYNC STOPPED!
2190 012022 000067 67 ;
2191 012024 000402 BR NOSYNC ; ERROR RETURN
2192 ;
2193 ;

```

VTV30K DIAGNOSTIC  
CVVTCA.SRC 08-JAN-81 10:36 MACY11 30A(1052) 08-JAN-81 10:58 PAGE 35-2<sup>B</sup> 5  
SYNC UP THE PROGRAM

SEQ 0053

|      |        |        |        |         |     |           |                         |
|------|--------|--------|--------|---------|-----|-----------|-------------------------|
| 2194 | 012026 | 062716 | 000002 | 50\$:   | ADD | #2,(SP)   | ; UPDATE TO GOOD RETURN |
| 2195 |        |        |        | :       |     |           |                         |
| 2196 | 012032 | 004737 | 013776 | NOSYNC: | JSR | PC,RSTREG | ; RECOVER REGISTERS.    |
| 2197 | 012036 | 000207 |        |         | RTS | PC        | ; ..AND EXIT            |



2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254

.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

WAITS: JSR PC,SAVREG ; SAVE REGISTERS  
MOV @ (SP),R1 ; GET SECOND COUNT  
ADD #2,(SP) ; PROTECT RETURN  
MOV TESTNO,R2 ; GET TEST NUMBER  
TSTB TABLE2(R2) ; IS IT STALL-ABLE?  
BEQ 1\$ ; NO IF BRANCH  
TST REACT ; HAS USER OVERRIDDEN TIMER?  
BEQ 1\$ ; NO..  
MOV REACT,R1 ; ELSE RESET TIME.  
1\$: MOV WAIT1,R2 ; GET TIME FOR ONE SECOND.

|      |        |        |        |     |      |                                         |
|------|--------|--------|--------|-----|------|-----------------------------------------|
| 2255 |        |        |        |     |      |                                         |
| 2256 | 012104 | 105737 | 177560 | 2s: | TSTB | TKS ; ANY CONSOLE INPUT?                |
| 2257 | 012110 | 100012 |        |     | BPL  | 3\$ ; NO...                             |
| 2258 | 012112 | 004737 | 014206 |     | JSR  | PC,READ ; ELSE GET CHARACTER..          |
| 2259 | 012116 | 020027 | 000040 |     | CMP  | R0,#' ; EXIT ON SPACE                   |
| 2260 | 012122 | 001414 |        |     | BEQ  | 7\$                                     |
| 2261 | 012124 | 020027 | 000007 |     | CMP  | R0,#'G&37 ; 'BELL'?                     |
| 2262 | 012130 | 001002 |        |     | BNE  | 3\$ ; NO..                              |
| 2263 | 012132 | 004737 | 013546 |     | JSR  | PC,MONIT ; ..ELSE CALL UP MONITOR.      |
| 2264 |        |        |        |     |      |                                         |
| 2265 | 012136 | 004737 | 011644 | 3s: | JSR  | PC,SYNCUP ; SYNC UP                     |
| 2266 | 012142 | 000240 |        |     | NOP  | ; IGNORE ERRORS                         |
| 2267 | 012144 | 005302 |        |     | DEC  | R2 ; ONE SECOND?                        |
| 2268 | 012146 | 001356 |        |     | BNE  | 2\$ ; NOT YET..                         |
| 2269 | 012150 | 005301 |        |     | DEC  | R1 ; MORE SECONDS?                      |
| 2270 | 012152 | 001352 |        |     | BNE  | 1\$ ; YES, KEEP LOOPING                 |
| 2271 |        |        |        |     |      |                                         |
| 2272 | 012154 | 004737 | 013776 | 7s: | JSR  | PC,RSTREG ; EXIT AFTER RECOVERING REGS. |
| 2273 | 012160 | 000207 |        |     | RTS  | PC                                      |



2275  
2276  
2277  
2278  
2279  
2280  
2281  
2282  
2283  
2284  
2285  
2286  
2287  
2288  
2289  
2290  
2291  
2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312  
2313  
2314  
2315  
2316  
2317  
2318  
2319  
2320  
2321  
2322  
2323  
2324  
2325  
2326  
2327  
2328  
2329  
2330

.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

```

2331
2332 012162 004737 013740      SETDEV: JSR      PC,SAVREG      ; SAVE OLD ENVIRONMENT
2333
2334 012166 004737 014734      1s:      JSR      PC,TYPOUT      ; ASK MODULE TYPE?
2335 012172 005610
2336 012174 004737 014206      SET56
2337 012200 012701 012374      JSR      PC,READ      ; READ REPLY...A/B/C/D/G
2338
2339 012204 121100              MOV      #DEVLSST,R1      ; GET CHECK TABLE ADDRESS
2340 012206 001404              3s:      CMPB     (R1),R0      ; THIS MODULE?
2341 012210 005721              BEQ      10$      ; YES...
2342 012212 005711              TST      (R1)+      ; ELSE SKIP ON TO NEXT
2343 012214 001373              TST      (R1)      ; END OF TABLE?
2344 012216 000763              BNE      3$      ; NO, TRY NEXT
2345
2346 012220 005037 007622      BR       1$      ; ELSE RESTART.
2347 012224 032711 001000      10$:     CLR      PAL      ; PRESET TO NON-TV-ENCODED.
2348 012230 001407              BIT      #1000,(R1)      ; IS IT 525?
2349 012232 012737 000033 007500 BEQ      12$      ; NO IF BRANCH
2350 012240 012737 000074 007452 MOV      #27.,YMAX      ; MAXIMUM ROW NO FOR 525
2351 012246 000406              MOV      #60.,WAIT1      ; 60 FRAMES/SEC
2352
2353 012250 012737 000037 007500 12$:     MOV      #31.,YMAX      ; 32 ROWS IN 625
2354 012256 012737 000062 007452 MOV      #50.,WAIT1      ; 50 FRAMES/SEC
2355
2356 012264 012737 000077 007446 13$:     MOV      #63.,XMAX      ; 48 VISIBLE,16 INVISIBLE CELLS.
2357 012272 012737 000001 007462 MOV      #1,PROM      ; PRESET TO PROM CHAR SET.
2358 012300 032711 000400      BIT      #400,(R1)      ; PROM?
2359 012304 001030              BNE      20$      ; YES...
2360 012306 005037 007462      CLR      PROM      ; ELSE RESET TO RAM CHAR STORE.
2361 012312 004737 014734      JSR      PC,TYPOUT      ; ASK IF ENCODED FOR TV!
2362 012316 005627
2363 012320 004737 014206      SETENC
2364 012324 012701 017242      JSR      PC,READ      ; GET REPLY..
2365 012330 120027 000131      MOV      #NORCH,R1      ; DEFAULT TO NORMAL CHARACTER SET
2366 012334 001005              CMPB     R0,#'Y      ; IF YES, THEN WE ARE ENCODED.
2367 012336 012737 000001 007622 BNE      15$      ; SET ENCODED (ONLY RELEVANT FOR RAM)
2368 012344 012701 021242      MOV      #1,PAL      ; GET THIS CHAR SET
2369
2370 012350 012700 007624      MOV      #P02CHR,R0      ; GET DESTINATION.
2371 012354 012702 001000      MOV      #1000,R2      ; 512 WORDS=128 CHARS.
2372
2373 012360 012120              16$:     MOV      (R1)+,(R0)+      ; TRANSFER 2 BYTES.
2374 012362 005302              DEC      R2
2375 012364 001375              BNE      16$      ; ANY MORE?
2376
2377 012366 004737 013776      20$:     JSR      PC,RSTREG      ; RECOVER REGISTERS.
2378 012372 000207      RTS      PC
2379
2380
2381
2382
2383 012374 000470      ; THE FOLLOWING TABLE HAS AN ASCII CHACK BYTE IN EACH WORD,
2384 012376 001471      ; TOGETHER WITH 400 IF IT IS A PROM UNIT, AND 1000 FOR 525.
2385 012400 000060      DEVLST: '8+400+0000      ; INTERFACE, -KA
2386 012402 001061      '9+400+1000      ; -KB
2387
2388      '0+000+0000      ; -KC
2389
2390      '1+000+1000      ; -KD

```



VTV30K DIAGNOSTIC  
CVTCA.SRC 08-JAN-81 10:36 MACY11 30A(1052) 08-JAN-81 10:58 PAGE 37-2<sup>G 5</sup>  
SELECT 525/625

SEQ 0058

2387 012404 000462  
2388 012406 000000

'2+400+0000  
0

;

-KG

2390  
2391  
2392  
2393  
2394  
2395  
2396  
2397  
2398  
2399  
2400  
2401  
2402  
2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419  
2420  
2421  
2422  
2423  
2424  
2425  
2426  
2427  
2428  
2429  
2430  
2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445

.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

|         |      |           |                                |
|---------|------|-----------|--------------------------------|
| LOADFS: | JSR  | PC,SAVREG |                                |
|         | CLR  | CARY      | ; CLEAR OUR Y COORD            |
|         | CLR  | CARX      | ; AND OUR X COORD              |
|         | MOV  | CHAR,R2   | ; PRESET THE PICT TO THIS CHAR |
|         | SWAB | R2        | ; HIGH ORDER BYTE              |
|         | BIS  | BLINK,R2  | ; LOAD BLINK AND FLASH BITS    |
|         | BIS  | FLASH,R2  |                                |

|        |        |        |
|--------|--------|--------|
| 012410 | 004737 | 013740 |
| 012414 | 005037 | 007476 |
| 012420 | 005037 | 007474 |
| 012424 | 013702 | 007550 |
| 012430 | 000302 |        |
| 012432 | 053702 | 007470 |
| 012436 | 053702 | 007472 |



|      |        |        |        |     |           |                            |
|------|--------|--------|--------|-----|-----------|----------------------------|
| 2446 | 012442 | 053702 | 007464 | BIS | FORGND,R2 | : LOAD FOREGROUND COLOUR   |
| 2447 | 012446 | 013701 | 007466 | MOV | BAKGND,R1 | : GET BACKGROUND COLOUR    |
| 2448 | 012452 | 006301 |        | ASL | R1        | : TO CORRECT BITS          |
| 2449 | 012454 | 006301 |        | ASL | R1        |                            |
| 2450 | 012456 | 006301 |        | ASL | R1        |                            |
| 2451 | 012460 | 050102 |        | BIS | R1,R2     | : AND LOAD INTO DATUM WORD |
| 2452 | 012462 | 013700 | 007436 | MOV | DEVCSR,R0 | : GET THE CSR ADDRESS      |
| 2453 | 012466 | 013701 | 007440 | MOV | DEVBUF,R1 | : ALSO THE DATA BUFFER     |
| 2454 | 012472 | 013703 | 007450 | MOV | VIDEON,R3 | : GET THE VIDEO ON BIT..   |
| 2455 | 012476 | 012704 | 004000 | MOV | #4000,R4  | : PICTURE STORE COUNT      |
| 2456 |        |        |        |     |           |                            |
| 2457 | 012502 | 010310 |        | MOV | R3,(R0)   | : LOAD CSR WITH ADDRESS    |
| 2458 | 012504 | 010211 |        | MOV | R2,(R1)   | : LOAD PICTURE ELEMENT     |
| 2459 | 012506 | 005203 |        | INC | R3        | : NEXT PICTURE ELEMENT     |
| 2460 | 012510 | 005304 |        | DEC | R4        | : ANY MORE?                |
| 2461 | 012512 | 001373 |        | BNE | 2\$       | : YES...                   |
| 2462 | 012514 | 004737 | 013776 | JSR | PC,RSTREG | : RESTORE THE REGISTERS    |
| 2463 | 012520 | 000207 |        | RTS | PC        | : AND EXIT.                |

2465  
2466  
2467  
2468  
2469  
2470  
2471  
2472  
2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480  
2481  
2482  
2483  
2484  
2485  
2486  
2487  
2488  
2489  
2490  
2491  
2492  
2493  
2494  
2495  
2496  
2497  
2498  
2499  
2500  
2501  
2502  
2503  
2504  
2505  
2506  
2507  
2508  
2509  
2510  
2511  
2512  
2513  
2514  
2515  
2516  
2517  
2518  
2519  
2520

.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

LOELEM: JSR PC,SAVREG  
 MOV CARX,R2 ; SET UP OUR SAR REGISTER  
 MOV CARY,R1



|      |        |        |        |      |            |                                   |
|------|--------|--------|--------|------|------------|-----------------------------------|
| 2521 | 012536 | 000301 |        | SWAB | R1         | ; GET Y ADDRESS IN TOP BYTE       |
| 2522 | 012540 | 105001 |        | CLRB | R1         |                                   |
| 2523 | 012542 | 006201 |        | ASR  | R1         |                                   |
| 2524 | 012544 | 006201 |        | ASR  | R1         | ; SHIFT IT DOWN                   |
| 2525 | 012546 | 042701 | 174000 | BIC  | #174000,R1 | ; CLEAR RUBBISH                   |
| 2526 | 012552 | 053701 | 007450 | BIS  | VIDEON,R1  | ; SET VIDEO ON                    |
| 2527 | 012556 | 050102 |        | BIS  | R1,R2      | ; THEN SET IT UP                  |
| 2528 | 012560 | 010277 | 174652 | MOV  | R2,@DEVCSR | ; SET OUR X AND Y ADDRESSES       |
| 2529 | 012564 | 013702 | 007464 | MOV  | FORGND,R2  | ; START WITH FOREGROUND           |
| 2530 | 012570 | 013701 | 007466 | MOV  | BAKGND,R1  | ; GET BACKGROUND COLOUR           |
| 2531 | 012574 | 006301 |        | ASL  | R1         |                                   |
| 2532 | 012576 | 006301 |        | ASL  | R1         | ; SET IT IN CORRECT POSITION      |
| 2533 | 012600 | 006301 |        | ASL  | R1         |                                   |
| 2534 | 012602 | 050102 |        | BIS  | R1,R2      | ; ADD TO COMPOSITE                |
| 2535 | 012604 | 053702 | 007470 | BIS  | BLINK,R2   | ; SET BLINK IF REQUIRED           |
| 2536 | 012610 | 053702 | 007472 | BIS  | FLASH,R2   | ; AND FLASH                       |
| 2537 | 012614 | 013701 | 007550 | MOV  | CHAR,R1    | ; GET CHARACTER                   |
| 2538 | 012620 | 000301 |        | SWAB | R1         | ; THE SET IN THE CORRECT POSITION |
| 2539 | 012622 | 105001 |        | CLRB | R1         |                                   |
| 2540 | 012624 | 050102 |        | BIS  | R1,R2      | ; ADD IT IN                       |
| 2541 | 012626 | 010277 | 174606 | MOV  | R2,@DEVBUF | ; OUT                             |
| 2542 | 012632 | 004737 | 013776 | JSR  | PC,RSTREG  |                                   |
| 2543 | 012636 | 000207 |        | RTS  | PC         |                                   |

2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591  
2592  
2593  
2594  
2595  
2596  
2597

.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

2598 012640 005037 012702  
 2599 012644 012737 012670 000004  
 2600 012652 012737 000340 000006

SILLSI: CLR LSIFLG ;SET UP FLAG FOR NON-LSI  
 MOV #1\$,4 ;INSTALL TRAP THRU 4 VECTOR  
 MOV #340,6 ;AND CORRESPONDING PRIORITY



|      |        |        |               |      |           |              |  |                                |
|------|--------|--------|---------------|------|-----------|--------------|--|--------------------------------|
| 2601 |        |        |               | :    |           |              |  |                                |
| 2602 | 012660 | 005737 | 177776        |      | TST       | PSW          |  | ; TRY ADDRESSING THE PSW --    |
| 2603 | 012664 | 000240 |               |      | NOP       |              |  | ; IF NOT THERE, TRAP THRU 4    |
| 2604 | 012666 | 000404 |               |      | BR        | 2\$          |  | ; WILL OCCUR, ELSE BRANCH.     |
| 2605 |        |        |               | :    |           |              |  |                                |
| 2606 | 012670 | 022626 |               | 1\$: | CMP       | (SP)+, (SP)+ |  | ; PERFORM A DUMMY RTI          |
| 2607 | 012672 | 012737 | 177777 012702 |      | MOV       | #-1, LSIFLG  |  | ; PROCESSOR HAS ONE INT.LEVEL  |
| 2608 |        |        |               | :    |           |              |  |                                |
| 2609 | 012700 | 000207 |               | 2\$: | RTS       | PC           |  | ; RETURN TO MAINLINE CALL.     |
| 2610 |        |        |               | :    |           |              |  |                                |
| 2611 |        |        |               | :    |           |              |  |                                |
| 2612 |        |        |               | :    |           |              |  |                                |
| 2613 | 012702 | 000000 |               |      | LSIFLG: 0 |              |  | ; 0 => MULTIPLE INT.PRIORITIES |
| 2614 |        |        |               |      |           |              |  | ; -1=> SINGLE INT. PRIORITY    |
| 2615 |        |        |               | :    |           |              |  |                                |
| 2616 |        |        |               | :    |           |              |  |                                |
| 2617 |        |        |               | :    |           |              |  |                                |

2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672

.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



2674  
2675  
2676  
2677  
2678  
2679  
2680  
2681  
2682  
2683  
2684  
2685  
2686  
2687  
2688  
2689  
2690  
2691  
2692  
2693  
2694  
2695  
2696  
2697  
2698  
2699  
2700  
2701  
2702  
2703  
2704  
2705  
2706  
2707  
2708  
2709  
2710  
2711  
2712  
2713  
2714  
2715  
2716  
2717  
2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729

.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

2730  
2731  
2732  
2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753  
2754  
2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769

012714 005000  
012716 012701 000002  
012722 020027 000014  
012726 001002  
012730 022020  
012732 000410  
012734 020027 000040  
012740 001002  
012742 022020  
012744 000403  
012746 010120  
012750 012720 000004  
012754 062701 000004  
012760 020027 000774  
012764 002756  
012766 012737 013326 000034  
012774 012737 000340 000036  
013002 012737 013156 000030  
013010 012737 000340 000032  
013016 012737 013226 000020  
013024 012737 000340 000022  
013032 013737 007510 000200  
013040 013737 007512 000202  
013046 012737 013142 000046  
013054 005037 000052  
013060 000207

.....

COMPLETION CODES:  
NONE

POSSIBLE ERROR CODES:  
NONE

VECTOR: CLR R0 ;FILL 0-572 WITH IOT TRAPS  
 MOV #2,R1  
 FILL: CMP R0,#14 ;ODT TRAP?  
 BNE 1\$  
 CMP (R0)+,(R0)+ ;YES BUMP R0  
 BR FILL1  
 1\$: CMP R0,#40 ;XXDP RETURN ADDRESS  
 BNE 2\$ ;NO  
 CMP (R0)+,(R0)+ ;YES BUMP R0  
 BR FILL1  
 2\$: MOV R1,(R0)+ ;'+2'  
 MOV #4,(R0)+ ;'IOT'  
 FILL1: ADD #4,R1  
 CMP R0,#774  
 BLT FILL  
 MOV #TRAPSV,34 ;TRAP (LOOP CONTROL)  
 MOV #340,36  
 MOV #FADR,30 ; PLUG EMT FOR READING  
 MOV #340,32 ; THE PROCESSOR STATUS  
 MOV #ILLVEC,20 ;PLUG 20 FOR IOTS  
 MOV #340,22  
 MOV JM600,200 ;SET UP JMP START IN LOC 200  
 MOV JM600+2,202  
 MOV #SENDAD,46  
 CLR 52 ;POINT TO RETURN TO XXDP  
 RTS PC ;CLEAR 52  
 ;THEN EXIT



2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789  
2790  
2791  
2792  
2793  
2794  
2795  
2796  
2797  
2798  
2799  
2800  
2801  
2802  
2803  
2804  
2805  
2806  
2807  
2808  
2809  
2810  
2811  
2812  
2813  
2814  
2815  
2816  
2817  
2818  
2819  
2820  
2821  
2822  
2823  
2824  
2825  
2826

.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  
ASSUMED TO CONTAIN THE XXDP MONITOR RETURN ADDRESS

POSSIBLE ERROR CODES:

NONE

```

2827
2828
2829 013062 012706 001000      :
2830 013066                                : ENDIT: MOV #START,SP      ; RESET THE STACK
2831 013102 032777 000100 174372 : PSWSET #340             ; AND PROCESSOR PRIORITY
2832 013110 001402                                : BIT #100,@SWR           ; IS THERE A PRESELECT ON
2833 013112 000137 001326      : BEQ 1$                  ; NO
2834                                : JMP START2              ; YES GO SELECT THE TEST
2835                                : NO PRESELECT ON SO SIGNAL END OF PASS
2836                                :
2837 013116 104402      1$: TRAP+2      ; BYPASS MESSAGE
2838 013120 001306      START1
2839 013122 004737 014734 2$: JSR PC,TYPOUT ; END OF PASS
2840 013126 006500      PASMSG
2841 013130 013700 000042      MOV @#42,R0 ; GET RETURN ADDRESS TO XXDP
2842 013134 001002      BNE $ENDAD ; IF IT IS ZERO THERE IS NO
2843 013136 000137 001306      JMP START1 ; MONITOR SO RESTART DIAG
2844
2845
2846                                :
2847 013142 004710      $ENDAD: JSR PC,(R0) ; CALLED VIA XXDP SO RETURN
2848 013144 000240      NOP ; THERE
2849 013146 000240      NOP
2850 013150 000240      NOP ; ALLOW A SLIGHT PAUSE
2851 013152 000137 001306      JMP START1 ; THEN RESTART THE DIAGNOSTIC
2852
2853
2854

```



2856  
2857  
2858  
2859  
2860  
2861  
2862  
2863  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879  
2880  
2881  
2882  
2883  
2884  
2885  
2886  
2887  
2888  
2889  
2890  
2891  
2892  
2893  
2894  
2895  
2896

.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  
013164 000002

FADR: MOV 2(SP),FSAVPW ; READ PRIORITY  
RTI ; RETURN TO CALLER  
;

2898  
2899  
2900  
2901  
2902  
2903  
2904  
2905  
2906  
2907  
2908  
2909  
2910  
2911  
2912  
2913  
2914  
2915  
2916  
2917  
2918  
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935  
2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953

.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

|        |        |        |        |        |      |           |                         |
|--------|--------|--------|--------|--------|------|-----------|-------------------------|
| 013166 | 032777 | 000040 | 174306 | BELL:  | BIT  | #40,ASWR  |                         |
| 013174 | 001401 |        |        |        | BEQ  | BELL1     |                         |
| 013176 | 000207 |        |        |        | RTS  | PC        |                         |
| 013200 | 004737 | 013276 |        | BELL1: | JSR  | PC,TPREDY | ;WAIT FOR PRINTER READY |
| 013204 | 112737 | 000007 | 177566 |        | MOVB | #7,TPB    |                         |
| 013212 | 004737 | 013276 |        |        | JSR  | PC,TPREDY | ;WAIT FOR PRINTER READY |

VTV30K DIAGNOSTIC MACY11 30A(1052) 08-JAN-81 10:58 PAGE 45-1<sup>H 6</sup>  
CVTCA.SRC 08-JAN-81 10:36 RING TTY BELL

SEQ 0072

2954 013216 112737 000000 177566  
2955 013224 000207

MOVB #0,TPB  
RTS PC

;PRINT NULL  
;GO OUT



2957  
2958  
2959  
2960  
2961  
2962  
2963  
2964  
2965  
2966  
2967  
2968  
2969  
2970  
2971  
2972  
2973  
2974  
2975  
2976  
2977  
2978  
2979  
2980  
2981  
2982  
2983  
2984  
2985  
2986  
2987  
2988  
2989  
2990  
2991  
2992  
2993  
2994  
2995  
2996  
2997  
2998  
2999  
3000  
3001  
3002  
3003  
3004  
3005  
3006  
3007  
3008  
3009  
3010  
3011  
3012

.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

|      |        |        |        |         |                               |
|------|--------|--------|--------|---------|-------------------------------|
| 3013 |        |        |        |         |                               |
| 3014 |        |        |        |         |                               |
| 3015 |        |        |        |         |                               |
| 3016 |        |        |        |         |                               |
| 3017 |        |        |        |         |                               |
| 3018 |        |        |        |         |                               |
| 3019 | 013226 | 004737 | 014734 | ILLVEC: | JSR PC,TYPEOUT                |
| 3020 | 013232 | 006516 |        |         | ILVMSG                        |
| 3021 | 013234 | 012600 |        |         | MOV (SP)+,R0                  |
| 3022 | 013236 | 162700 | 000004 |         | SUB #4,R0                     |
| 3023 | 013242 | 004737 | 015024 |         | JSR PC,PROCT ;PRINT VECTOR    |
| 3024 | 013246 | 004737 | 014734 |         | JSR PC,TYPEOUT                |
| 3025 | 013252 | 006550 |        |         | FRMSG ;PRINT MAINLINE ADDRESS |
| 3026 | 013254 | 005726 |        |         | TST (SP)+                     |
| 3027 | 013256 | 012600 |        |         | MOV (SP)+,R0                  |
| 3028 | 013260 | 004737 | 015024 |         | JSR PC,PROCT                  |
| 3029 | 013264 | 005726 |        |         | TST (SP)+                     |
| 3030 | 013266 | 004737 | 013546 |         | JSR PC,MONIT                  |
| 3031 | 013272 | 000137 | 001200 |         | JMP RSTART                    |

3033  
3034  
3035  
3036  
3037  
3038  
3039  
3040  
3041  
3042  
3043  
3044  
3045  
3046  
3047  
3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082

.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

TPREDY: TSTB TPS ;ROUTINE TO WAIT FOR PRINTER READY  
 BPL TPREDY  
 RTS PC

013276 105737 177564  
 013302 100375  
 013304 000207



3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099  
3100  
3101  
3102  
3103  
3104  
3105  
3106  
3107  
3108  
3109  
3110  
3111  
3112  
3113  
3114  
3115  
3116  
3117  
3118  
3119  
3120  
3121  
3122  
3123  
3124  
3125  
3126  
3127  
3128  
3129  
3130  
3131  
3132  
3133  
3134  
3135  
3136  
3137  
3138  
3139

.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

VTV30K DIAGNOSTIC  
CVTCA.SRC 08-JAN-81 10:36 MACY11 30A(1052) 08-JAN-81 10:58 PAGE 48-1 M 6  
SET ITERATION COUNT

SEQ 0077

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

3145  
3146  
3147  
3148  
3149  
3150  
3151  
3152  
3153  
3154  
3155  
3156  
3157  
3158  
3159  
3160  
3161  
3162  
3163  
3164  
3165  
3166  
3167  
3168  
3169  
3170  
3171  
3172  
3173  
3174  
3175  
3176  
3177  
3178  
3179  
3180  
3181  
3182  
3183  
3184  
3185  
3186  
3187  
3188  
3189  
3190  
3191  
3192  
3193  
3194  
3195  
3196  
3197  
3198  
3199  
3200

.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 ARGUMENT 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:



```

3201                                     ON EXECUTION OF ANY TRAP INSTRUCTION
3202
3203
3204
3205                                     IMPLICIT INPUT PARAMETERS:
3206                                     THE SWR HAS BEEN SET UP
3207
3208
3209
3210                                     OUTPUT PARAMETERS:
3211
3212                                     EXIT TO THE CONTENTS OF THE ADDRESS FOLLOWING
3213                                     THE TRAP INSTRUCTION IF A MATCH WAS FOUND, ELSE THE
3214                                     PROGRAM WILL CONTINUE FROM THE ADDRESS AFTER THAT.
3215
3216
3217                                     IMPLICIT OUTPUT PARAMETERS:
3218                                     NONE
3219
3220
3221                                     COMPLETION CODES:
3222                                     NONE
3223
3224
3225                                     POSSIBLE ERROR CODES:
3226                                     NONE
3227
3228
3229
3230
3231
3232
3233 013326 004737 013740 TRAPSV: JSR PC,SAVREG ;SAVE REGS
3234 013332 011600          MOV (SP),R0
3235 013334 016037 177776 007516 MOV -2(R0),TRPARG ;GET TRAP CALL
3236 013342 105737 177560      TSTB TKS ;LOOK FOR MONITOR CALL
3237 013346 100015          BPL 3$
3238 013350 004737 014206      JSR PC,READ
3239 013354 120027 000017      CMPB R0,#17 ;IS IT CNTRL-O ?
3240 013360 001003          BNE 1$ ;NO
3241 013362 004737 015450      JSR PC,MODIFY ;YES CALL MODIFIER ROUTINE
3242 013366 000405          BR 3$
3243 013370 120027 000007      1$: CMPB R0,#7 ;IS IT CTRL-G?
3244 013374 001002          BNE 3$
3245 013376 004737 013546      JSR PC,MONIT ;YES, GO TO SWR MONITOR
3246 013402 017737 174074 007520 3$: MOV @SWR,TRPSEL
3247 013410 132737 000020 007521 BITB #20,TRPSEL+1 ;IS IT PRESERVE SCOPE
3248 013416 001012          BNE TRPSCP ;YES
3249 013420 013737 007520 007522 MOV TRPSEL,TRPMEM ;NO SO SAVE SWITCH SETTING
3250 013426 042737 170777 007522 BIC #170777,TRPMEM ;GET IT SO WE CAN COMPARE
3251 013434 132737 000016 007521 BITB #16,TRPSEL+1 ;ANY SCOPE LEVELS SET
3252 013442 001412          BEQ TRPLP ;NO
3253 013444 013700 007516 TRPSCP: MOV TRPARG,R0 ;YES
3254 013450 006000          ROR R0 ;GET TO POSITION FOR COMPARE
3255 013452 006000          ROR R0 ;FOR 10 & ABOVE
3256 013454 000300          SWAB R0 ;ONLY FOR SCOPE BITS(9-11)
  
```

|      |        |        |        |         |      |             |                           |
|------|--------|--------|--------|---------|------|-------------|---------------------------|
| 3257 | 013456 | 042700 | 170777 |         | BIC  | #170777,R0  | :ARGUMENT SCOPE BITS      |
| 3258 | 013462 | 020037 | 007522 |         | CMP  | R0,TRPMEM   | :AS SELECTED ?            |
| 3259 | 013466 | 001422 |        |         | BEQ  | TRPBAK      | :YES, GO BACK             |
| 3260 | 013470 | 013700 | 007520 | TRPLP:  | MOV  | TRPSEL,R0   | :NO, TEST LOOP            |
| 3261 | 013474 | 006100 |        |         | ROL  | R0          | :FOR BITS 7&8             |
| 3262 | 013476 | 006100 |        |         | ROL  | R0          |                           |
| 3263 | 013500 | 000300 |        |         | SWAB | R0          |                           |
| 3264 | 013502 | 142700 | 000371 |         | BICB | #371,R0     | :CUT OUT SW06             |
| 3265 | 013506 | 142737 | 000370 | 007516  | BICB | #370,TRPARG | :ONLY SWITCHES 7 OR 8 NOW |
| 3266 | 013514 | 130037 | 007516 |         | BITB | R0,TRPARG   | :ANY SELECTED ?           |
| 3267 | 013520 | 001005 |        |         | BNE  | TRPBAK      | :YES                      |
| 3268 | 013522 | 004737 | 013776 |         | JSR  | PC,RSTREG   |                           |
| 3269 | 013526 | 062716 | 000002 |         | ADD  | #2,(SP)     | :NO SCOPE OR LOOP,SO RUN  |
| 3270 | 013532 | 000002 |        |         | RTI  |             | :PAST ARGUMENT AND RETURN |
| 3271 | 013534 | 004737 | 013776 | TRPBAK: | JSR  | PC,RSTREG   | :RESTORE REGS             |
| 3272 | 013540 | 017616 | 000000 |         | MOV  | @(SP),(SP)  | :RETURN ADDRESS TO STACK  |
| 3273 | 013544 | 000002 |        |         | RTI  |             | :EXIT, LOOPING            |

3275  
3276  
3277  
3278  
3279  
3280  
3281  
3282  
3283  
3284  
3285  
3286  
3287  
3288  
3289  
3290  
3291  
3292  
3293  
3294  
3295  
3296  
3297  
3298  
3299  
3300  
3301  
3302  
3303  
3304  
3305  
3306  
3307  
3308  
3309  
3310  
3311  
3312  
3313  
3314  
3315  
3316  
3317  
3318  
3319  
3320  
3321  
3322  
3323  
3324  
3325  
3326  
3327  
3328  
3329  
3330

.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



```

3331
3332
3333
3334 013546 010046
3335 013550 023727 007502 177570
3336 013556 001430
3337 013560 004737 014734
3338 013564 006623
3339 013566 013700 007504
3340 013572 004737 015024
3341 013576 012700 000076
3342 013602 004737 015012
3343 013606 004737 014034
3344 013612 005737 007532
3345 013616 001413
3346 013620 010037 007504
3347 013624 032737 000100 007504
3348 013632 001405
3349 013634 000137 001326
3350 013640 004737 013656
3351 013644 000775
3352 013646 004737 014700
3353 013652 012600
3354 013654 000207

      :
      :
      :
MONIT: MOV    R0, -(SP)          ;SAVE R0
        CMP    SWR, #HSWR       ;HARDWARE SWR?
        BEQ    MONITA          ;IF YES, GO TO END
        JSR    PC, TYP OUT      ;SWR=
        SWRMSG
        MOV    SSWR, R0
        JSR    PC, PROCT
        MOV    #76, R0
        JSR    PC, PCHR         ;PRINT '>'
        JSR    PC, OCTIN        ;GET NEW SETTING
        TST    RAND             ;ANY INPUT?
        BEQ    MONITX
        MOV    R0, SSWR         ;YES UPDATE SSWR
        BIT    #100, SSWR       ;TEST SELECTED?
        BEQ    MONITX
        JMP    START2          ;YES GO DO IT
MONITA: JSR    PC, TYP CTC       ;CTRL-C TO CONTINUE
        BR     MONITA
MONITX: JSR    PC, CRLF
        MOV    (SP)+, R0        ;RESTORE R0
        RTS    PC
  
```

3356  
3357  
3358  
3359  
3360  
3361  
3362  
3363  
3364  
3365  
3366  
3367  
3368  
3369  
3370  
3371  
3372  
3373  
3374  
3375  
3376  
3377  
3378  
3379  
3380  
3381  
3382  
3383  
3384  
3385  
3386  
3387  
3388  
3389  
3390  
3391  
3392  
3393  
3394  
3395  
3396  
3397  
3398  
3399  
3400  
3401  
3402  
3403  
3404  
3405  
3406  
3407  
3408  
3409  
3410  
3411

.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

|         |        |           |                                |
|---------|--------|-----------|--------------------------------|
| TYPCTC: | JSR    | PC,TYPCTC | ;PRINTS TYPE CTRL/C WHEN READY |
|         | REDMES |           |                                |
|         | JSR    | PC,READ   | ;CTRL/C ENTERED                |
|         | CMP    | R0,#3     |                                |
|         | BNE    | QEXIT2    | ;NO                            |

|        |        |        |
|--------|--------|--------|
| 013656 | 004737 | 014734 |
| 013662 | 006570 |        |
| 013664 | 004737 | 014206 |
| 013670 | 020027 | 000003 |
| 013674 | 001005 |        |

VTV30K DIAGNOSTIC  
CVVTC.A.SRC

08-JAN-81

MACY11 30A(1052)  
10:36

08-JAN-81 10:58 PAGE 51-1  
WAIT FOR CNTRL-C

G 7

SEQ 0084

3412 013676 004737 014700  
3413 013702 062716 000002  
3414 013706 000207  
3415 013710 020027 000017  
3416 013714 001003  
3417 013716 004737 015450  
3418 013722 000755  
3419 013724 020027 000007  
3420 013730 001366  
3421 013732 004737 013546  
3422 013736 000747  
3423  
3424

JSR PC,CRLF  
ADD #2,(SP) ;YES SO JUMP OVER NO FIND RETURN  
QEXIT1: RTS PC ;GO BACK  
QEXIT2: CMP R0,#17 ;CNTRL-O ?  
BNE QEXIT3 ;NO  
JSR PC,MODIFY ;YES CALL MODIFY PROGRAM  
BR TYPCTC ;THEN GO BACK TO START  
QEXIT3: CMP R0,#7 ;CTRL-G?  
BNE QEXIT1  
JSR PC,MONIT ;GO TO SWR MONITOR  
BR TYPCTC ;LOOK FOR CTRL-C AGAIN



3426  
3427  
3428  
3429  
3430  
3431  
3432  
3433  
3434  
3435  
3436  
3437  
3438  
3439  
3440  
3441  
3442  
3443  
3444  
3445  
3446  
3447  
3448  
3449  
3450  
3451  
3452  
3453  
3454  
3455  
3456  
3457  
3458  
3459  
3460  
3461  
3462  
3463  
3464  
3465  
3466  
3467  
3468  
3469  
3470  
3471  
3472  
3473  
3474  
3475  
3476  
3477  
3478  
3479  
3480  
3481

.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

SAVREG: MOV (SP)+,SAVPC ;SAVE PC FOR RETURN FROM THIS ROUTINE  
MOV (SP)+,SAVPC1  
MOV R5,-(SP)  
MOV R4,-(SP)  
MOV R3,-(SP)

013740 012637 007524  
013744 012637 007526  
013750 010546  
013752 010446  
013754 010346

|      |        |        |     |              |                   |
|------|--------|--------|-----|--------------|-------------------|
| 3482 | 013756 | 010246 | MOV | R2,-(SP)     |                   |
| 3483 | 013760 | 010146 | MOV | R1,-(SP)     |                   |
| 3484 | 013762 | 010046 | MOV | R0,-(SP)     |                   |
| 3485 | 013764 | 013746 | MOV | SAVPC1,-(SP) |                   |
| 3486 | 013770 | 013746 | MOV | SAVPC,-(SP)  | ;PUT PC READY FOR |
| 3487 | 013774 | 000207 | RTS | PC           | ;RETURN           |
| 3488 |        |        |     |              |                   |
| 3489 |        |        |     |              |                   |
| 3490 |        |        |     |              |                   |

3492  
3493  
3494  
3495  
3496  
3497  
3498  
3499  
3500  
3501  
3502  
3503  
3504  
3505  
3506  
3507  
3508  
3509  
3510  
3511  
3512  
3513  
3514  
3515  
3516  
3517  
3518  
3519  
3520  
3521  
3522  
3523  
3524  
3525  
3526  
3527  
3528  
3529  
3530  
3531  
3532  
3533  
3534  
3535  
3536  
3537  
3538  
3539  
3540  
3541  
3542  
3543  
3544  
3545  
3546  
3547

|        |        |        |
|--------|--------|--------|
| 013776 | 012637 | 007524 |
| 014002 | 012637 | 007526 |
| 014006 | 012600 |        |
| 014010 | 012601 |        |
| 014012 | 012602 |        |
| 014014 | 012603 |        |
| 014016 | 012604 |        |
| 014020 | 012605 |        |
| 014022 | 013746 | 007526 |
| 014026 | 013746 | 007524 |

```

.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

RSTREG: MOV      (SP)+,SAVPC
        MOV      (SP)+,SAVPC1
        MOV      (SP)+,R0
        MOV      (SP)+,R1
        MOV      (SP)+,R2
        MOV      (SP)+,R3
        MOV      (SP)+,R4
        MOV      (SP)+,R5
        MOV      SAVPC1,-(SP)
        MOV      SAVPC,-(SP)      ;PUT PC READY FOR
  
```



VTV30K DIAGNOSTIC  
CVVTC.A.SRC 08-JAN-81 10:36 MACY11 30A(1052) 08-JAN-81 10:58 PAGE 53-1<sup>K</sup> 7  
RESTORE REGISTERS

SEQ 0088

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

|      |        |        |        |         |      |                 |                                 |
|------|--------|--------|--------|---------|------|-----------------|---------------------------------|
| 3606 |        |        |        | :       |      |                 |                                 |
| 3607 |        |        |        | :       |      |                 |                                 |
| 3608 |        |        |        | :       |      |                 |                                 |
| 3609 | 014034 | 004737 | 013740 | OCTIN:  | JSR  | PC, SAVREG      | : SAVE THE REGISTERS            |
| 3610 | 014040 | 005000 |        | TYPOTB: | CLR  | R0              | : CLEAR R0                      |
| 3611 | 014042 | 005037 | 007532 |         | CLR  | RAND            | : CLEAR FLAG WORD               |
| 3612 | 014046 | 012737 | 007256 |         | MOV  | #OCTMSG, STRADD | : SET START ADDRESS OF STRING   |
| 3613 | 014054 | 012737 | 000006 | 007604  | MOV  | #6, STRLEN      | : AND ITS SIZE                  |
| 3614 | 014062 | 012737 | 000060 | 007606  | MOV  | #60, LOWCHR     | : MINIMUM CHAR                  |
| 3615 | 014070 | 012737 | 000067 | 007610  | MOV  | #67, UPPCHR     | : MAXIMUM CHAR                  |
| 3616 | 014076 | 005037 | 007602 | 007612  | CLR  | CNVFLG          | : DON'T CONVERT TO UPPER CASE   |
| 3617 | 014102 | 004737 | 014232 |         | JSR  | PC, GETSTR      | : GET A STRING                  |
| 3618 | 014106 | 013701 | 007606 |         | MOV  | STRLEN, R1      | : ZERO LENGTH ?                 |
| 3619 | 014112 | 001420 |        |         | BEQ  | TYPOTD          | : YES                           |
| 3620 | 014114 | 012702 | 007256 |         | MOV  | #OCTMSG, R2     | : NO GET START ADDRES OF STRING |
| 3621 | 014120 | 112203 |        | TYPOTC: | MOVB | (R2)+, R3       | : GET A CHARACTER               |
| 3622 | 014122 | 162703 | 000060 |         | SUB  | #60, R3         | : PUT IN RANGE                  |
| 3623 | 014126 | 000241 |        |         | CLC  |                 |                                 |
| 3624 | 014130 | 006300 |        |         | ASL  | R0              | : SHIFT OUT RESULT              |
| 3625 | 014132 | 103417 |        |         | BCS  | TYPOTE          | : C BIT SET ERROR               |
| 3626 | 014134 | 006300 |        |         | ASL  | R0              | : TIMES 4                       |
| 3627 | 014136 | 103415 |        |         | BCS  | TYPOTE          | : C BIT ERROR                   |
| 3628 | 014140 | 006300 |        |         | ASL  | R0              | : TIMES 8                       |
| 3629 | 014142 | 050300 |        |         | BIS  | R3, R0          | : SET NEW BITS IN               |
| 3630 | 014144 | 005237 | 007532 |         | INC  | RAND            | : SET GOOD                      |
| 3631 | 014150 | 005301 |        |         | DEC  | R1              |                                 |
| 3632 | 014152 | 001362 |        |         | BNE  | TYPOTC          | : LOOP IF MORE TO COME          |
| 3633 | 014154 | 010037 | 007530 | TYPOTD: | MOV  | R0, TYPOTA      | : SAVE FINAL NUMBER             |
| 3634 | 014160 | 004737 | 013776 |         | JSR  | PC, RSTREG      | : RESTORE REGISTERS             |
| 3635 | 014164 | 013700 | 007530 |         | MOV  | TYPOTA, R0      | : GET RESULT                    |
| 3636 | 014170 | 000207 |        |         | RTS  | PC              |                                 |
| 3637 | 014172 | 012700 | 000041 | TYPOTE: | MOV  | #41, R0         | : OVERFLOW                      |
| 3638 | 014176 | 004737 | 015012 |         | JSR  | PC, PCHR        | : PRINT A !                     |
| 3639 | 014202 | 000137 | 014040 |         | JMP  | TYPOTB          | : THEN GO AGAIN                 |
| 3640 |        |        |        | :       |      |                 |                                 |



3642  
3643  
3644  
3645  
3646  
3647  
3648  
3649  
3650  
3651  
3652  
3653  
3654  
3655  
3656  
3657  
3658  
3659  
3660  
3661  
3662  
3663  
3664  
3665  
3666  
3667  
3668  
3669  
3670  
3671  
3672  
3673  
3674  
3675  
3676  
3677  
3678  
3679  
3680  
3681  
3682  
3683  
3684  
3685  
3686  
3687  
3688  
3689  
3690  
3691  
3692  
3693

.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

READ: TSTB TKS  
 BPL READ  
 MOV TKB,R0  
 BIC #177600,R0  
 JSR PC,PCHR  
 RTS PC

014206 105737 177560  
 014212 100375  
 014214 013700 177562  
 014220 042700 177600  
 014224 004737 015012  
 014230 000207

.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  
STLEN TO INDICATE ITS SIZE  
CNVFLG 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 STLEN 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  
STLEN 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:

STLEN THE NUMBERS OF CHARACTERS READ  
RANDC THE TERMINATING CHARACTER

3751  
3752  
3753  
3754  
3755  
3756  
3757  
3758  
3759  
3760  
3761  
3762  
3763  
3764  
3765  
3766  
3767  
3768  
3769  
3770  
3771 014232 004737 013740  
3772 014236 005037 007614  
3773 014242 013702 007604  
3774 014246 005001  
3775 014250 105737 177560  
3776 014254 100375  
3777 014256 013700 177562  
3778 014262 042700 177600  
3779  
3780  
3781  
3782 014266 020027 000015  
3783 014272 001002  
3784 014274 000137 014656  
3785 014300 020027 000012  
3786 014304 001002  
3787 014306 000137 014656  
3788 014312 020027 000033  
3789 014316 001002  
3790 014320 000137 014656  
3791 014324 020027 000177  
3792 014330 001002  
3793 014332 000137 014504  
3794 014336 020027 000022  
3795 014342 001002  
3796 014344 000137 014602  
3797 014350 020027 000025  
3798 014354 001002  
3799 014356 000137 014552  
3800  
3801  
3802  
3803  
3804  
3805 014362 020027 000140  
3806 014366 002405

: IMPLICIT OUTPUT PARAMETERS:

: NONE

: COMPELETION CODES:

: NONE

: POSSIBLE ERROR CONDITIONS:

: A '?' IS PRINTED IF AN INVALID CHARACTER IS ENTERED

GETSTR: JSR PC, SAVREG ; SAVE REGISTERS  
GETCH1: CLR RUBFLG ; CLEAR RUBOUT FLAG  
MOV STRADD, R2 ; GET STARTING POINTER  
CLR R1 ; SET INTIAL LENGTH  
GETCH2: TSTB TKS ; ANY TTY INPUT  
BPL GETCH2 ; NO WAIT FOR IT  
MOV TKB, R0 ; GET CHARACTER  
BIC #177600, R0 ; CLEAR RUBBISH

: CHECK FOR CONTROL CODES

1\$: CMP R0, #15 ; WAS IT <CR> ?  
BNE 1\$ ; NO  
JMP CHRFIN  
3\$: CMP R0, #12 ; WAS IT <LF>  
BNE 3\$ ; NO  
JMP CHRFIN  
5\$: CMP R0, #33 ; WAS IT <ESC>  
BNE 5\$ ; NO  
JMP CHRFIN  
7\$: CMP R0, #177 ; RUBOUT ?  
BNE 7\$ ; NO  
JMP RUBCHR ; YES  
9\$: CMP R0, #22 ; CNTRL-R  
BNE 9\$ ; NO  
JMP LINECH ; YES EXCO LINE  
CMP R0, #25 ; CNTRL-U  
BNE GETCH3 ; NO  
JMP LINDEL ; YES DELETE LINE

: IT WAS NOT A CONTROL CODE CHECK  
: FOR LEGALITY ?

GETCH3: CMP R0, #140 ; IS IT LOWER CASE  
BLT 3\$ ; NO



```

3807 014370 005737 007602      TST      CNVFLG      : YES DO WE CONVERT TO UPPER
3808 014374 001402              BEQ      3$             : NO
3809 014376 042700 000040      BIC      #40,R0      : YES STRIP 40 OFF
3810 014402 020037 007610      3$: CMP      R0,LOWCHR : IS THE CHAR TOO SMALL
3811 014406 002002              BGE      5$             : NO
3812 014410 000137 014636      JMP      ILLCHR      : YES
3813 014414 020037 007612      5$: CMP      R0,UPPCHR : IS THE CHAR TOO BIG ?
3814 014420 003402              BLE      GETCH4          : NO
3815 014422 000137 014636      JMP      ILLCHR      : YES ITS ILLEGAL
3816
3817      : WE HAVE A LEGAL CHARACTER
3818
3819 014426 010003      GETCH4: MOV      R0,R3      : SAVE CHAR
3820 014430 005737 007614      TST      RUBFLG      : IS RUBOUT SET
3821 014434 001406      BEQ      3$             : NO
3822 014436 012700 000057      MOV      #57,R0      : YES PRINT A '/'
3823 014442 004737 015012      JSR      PC,PCHR      :
3824 014446 005037 007614      CLR      RUBFLG      : CLEAR RUBOUT FLAG
3825 014452 005201 007606      3$: INC      R1             : UPDATE CHAR COUNT
3826 014454 020137 007606      CMP      R1,STRLEN : END OF STRING SEEN
3827 014460 003403              BLE      5$             : NO
3828 014462 005000              CLR      R0             : YES, FORCE A NULL TERMINATOR
3829 014464 000137 014656      JMP      CHRFIN      : AND COMPLETE
3830 014470 010300 015012      5$: MOV      R3,R0      : RESTORE CHAR
3831 014472 004737 015012      JSR      PC,PCHR      : ECHO IT
3832 014476 110022              MOVB     R0,(R2)+        : SAVE IT IN BUFFER
3833 014500 000137 014250      JMP      GETCH2      : NO GET NEXT ONE
3834
3835      : RUBOUT WAS HIT
3836
3837 014504 005701      RUBCHR: TST      R1             : ANY CHARACTERS TO RUBOUT
3838 014506 001002              BNE      3$             : YES
3839 014510 000137 014572      JMP      LINDL1      : NO
3840 014514 005737 007614      3$: TST      RUBFLG      : IS RUBOUT SET ALREADY ?
3841 014520 001006              BNE      5$             : YES
3842 014522 012700 000057      MOV      #57,R0      : NO PRINT A '/'
3843 014526 004737 015012      JSR      PC,PCHR      :
3844 014532 005237 007614      INC      RUBFLG      : SET RUBOUT
3845 014536 114200 015012      5$: MOVB     -(R2),R0 : GET LAST CHAR ENTERED
3846 014540 004737 015012      JSR      PC,PCHR      : PRINT IT
3847 014544 005301              DEC      R1             : REDUCE COUNT
3848 014546 000137 014250      JMP      GETCH2      : GET ANOTHER CHAR
3849
3850      : RUBOUT LINE WAS HIT
3851
3852 014552 012700 000136      LINDEL: MOV      #136,R0 : PRINT A ^
3853 014556 004737 015012      JSR      PC,PCHR      :
3854 014562 012700 000125      MOV      #125,R0      : THEN U
3855 014566 004737 015012      JSR      PC,PCHR      :
3856 014572 004737 014700      LINDL1: JSR      PC,CRLF : START ON A NEWLINE
3857 014576 000137 014236      JMP      GETCH1      : A GO BACK TO BEGINNING
3858
3859      : CNTRL-R WAS HIT
3860
3861 014602 112712 000100      LINECH: MOVB     #'a,(R2) : PUT IN A TERMINATOR
3862 014606 004737 014700      JSR      PC,CRLF      : START ON A NEW LINE

```

VTV30K DIAGNOSTIC MACY11 30A(1052) 08-JAN-81 10:58 PAGE 56-3<sup>E 8</sup>  
CVVTCA.SRC 08-JAN-81 10:36 ENTERING A CHARACTER STRING

SEQ 0095

|      |        |        |        |          |     |                           |                        |
|------|--------|--------|--------|----------|-----|---------------------------|------------------------|
| 3863 | 014612 | 013737 | 007604 | 014630   | MOV | STRADD,3\$                | ; SET START ADDRESS    |
| 3864 | 014620 | 005037 | 007614 |          | CLR | RUBFLG                    | ; CLEAR RUBOUTS        |
| 3865 | 014624 | 004737 | 014734 |          | JSR | PC,TYPOUT                 | ; PRINT LINE           |
| 3866 | 014630 | 000000 |        | 3\$:     | 0   |                           |                        |
| 3867 | 014632 | 000137 | 014250 |          | JMP | GETCH2                    | ; AND GET ANOTHER CHAR |
| 3868 |        |        |        | :        |     |                           |                        |
| 3869 |        |        |        | :        |     | ILLEGAL CHARACTER ENTERED |                        |
| 3870 |        |        |        | :        |     |                           |                        |
| 3871 | 014636 | 004737 | 015012 | ILLCHR:  | JSR | PC,PCHR                   | ; ECHO IT              |
| 3872 | 014642 | 012700 | 000077 |          | MOV | #?,R0                     | ; PRINT A ?            |
| 3873 | 014646 | 004737 | 015012 |          | JSR | PC,PCHR                   |                        |
| 3874 | 014652 | 000137 | 014602 |          | JMP | LINECH                    | ; THE ECHO LINE        |
| 3875 |        |        |        | :        |     |                           |                        |
| 3876 |        |        |        | :        |     | A TERMINATOR WAS FOUND    |                        |
| 3877 |        |        |        | :        |     |                           |                        |
| 3878 | 014656 | 010037 | 007616 | CHRF IN: | MOV | R0,RANDC                  | ; SAVE TERMINATOR      |
| 3879 | 014662 | 163702 | 007604 |          | SUB | STRADD,R2                 | ; CALCULATE BYTE COUNT |
| 3880 | 014666 | 010237 | 007606 |          | MOV | R2,STLEN                  | ; SET IT FOR RETURN    |
| 3881 | 014672 | 004737 | 013776 |          | JSR | PC,RSTREG                 | ; RESTORE REGISTERS    |
| 3882 | 014676 | 000207 |        |          | RTS | PC                        | ; AND EXIT             |

3884  
3885  
3886  
3887  
3888  
3889  
3890  
3891  
3892  
3893  
3894  
3895  
3896  
3897  
3898  
3899  
3900  
3901  
3902  
3903  
3904  
3905  
3906  
3907  
3908  
3909  
3910  
3911  
3912  
3913  
3914  
3915  
3916  
3917  
3918  
3919  
3920  
3921  
3922  
3923  
3924  
3925  
3926  
3927  
3928  
3929  
3930  
3931  
3932  
3933  
3934  
3935  
3936  
3937  
3938  
3939

.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

```
CRLF:  MOV  R0,-(SP)
        MOV  #15,R0          ;PRINT A RETURN - LINE FEED
        JSR  PC,PCHR
        CLR  R0              ;DUMMY
        JSR  PC,PCHR
        MOV  #12,R0
        JSR  PC,PCHR
        MOV  (SP)+,R0
        RTS  PC
```

```
014700 010046
014702 012700 000015
014706 004737 015012
014712 005000
014714 004737 015012
014720 012700 000012
014724 004737 015012
014730 012600
014732 000207
```



VTV30K DIAGNOSTIC  
CVTCA.SRC 08-JAN-81 10:36 MACY11 30A(1052) 08-JAN-81 10:58 PAGE 57-1<sup>G</sup> 8  
CARRIAGE RETURN LINE FEED

SEQ 0097

3940

3942  
3943  
3944  
3945  
3946  
3947  
3948  
3949  
3950  
3951  
3952  
3953  
3954  
3955  
3956  
3957  
3958  
3959  
3960  
3961  
3962  
3963  
3964  
3965  
3966  
3967  
3968  
3969  
3970  
3971  
3972  
3973  
3974  
3975  
3976  
3977  
3978  
3979  
3980  
3981  
3982  
3983  
3984  
3985  
3986  
3987  
3988  
3989  
3990  
3991  
3992  
3993  
3994  
3995  
3996  
3997

.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 (C) WILL GENERATE A <CR>,<LF>  
100 (a) 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

|      |        |        |        |         |      |            |
|------|--------|--------|--------|---------|------|------------|
| 3998 |        |        |        | :       |      |            |
| 3999 |        |        |        | :       |      |            |
| 4000 |        |        |        | :       |      |            |
| 4001 | 014734 | 004737 | 013740 | TYPOUT: | JSR  | PC, SAVREG |
| 4002 | 014740 | 017601 | 000000 |         | MOV  | @(SP), R1  |
| 4003 | 014744 | 062716 | 000002 |         | ADD  | #2, (SP)   |
| 4004 | 014750 | 111100 |        | PMSG1:  | MOVB | @R1, R0    |
| 4005 | 014752 | 022700 | 000100 |         | CMP  | #100, R0   |
| 4006 | 014756 | 001412 |        |         | BEQ  | PMSG4      |
| 4007 | 014760 | 022700 | 000133 |         | CMP  | #133, R0   |
| 4008 | 014764 | 001003 |        |         | BNE  | PMSG2      |
| 4009 | 014766 | 004737 | 014700 |         | JSR  | PC, CRLF   |
| 4010 | 014772 | 000402 |        |         | BR   | PMSG3      |
| 4011 | 014774 | 004737 | 015012 | PMSG2:  | JSR  | PC, PCHR   |
| 4012 | 015000 | 005201 |        | PMSG3:  | INC  | R1         |
| 4013 | 015002 | 000762 |        |         | BR   | PMSG1      |
| 4014 | 015004 | 004737 | 013776 | PMSG4:  | JSR  | PC, RSTREG |
| 4015 | 015010 | 000207 |        |         | RTS  | PC         |

;SAVE REGS  
 ;R1 POINTS AT STRING  
 ;JUMPS OVER ARGUMENT  
 ;PRINT THE MESSAGE POINTED  
 ;TO BY R1 UNTIL @, WHICH IS END.  
 ;[ MEANS CR-LF  
  
 ;RESTORE REGS



4017  
4018  
4019  
4020  
4021  
4022  
4023  
4024  
4025  
4026  
4027  
4028  
4029  
4030  
4031  
4032  
4033  
4034  
4035  
4036  
4037  
4038  
4039  
4040  
4041  
4042  
4043  
4044  
4045  
4046  
4047  
4048  
4049  
4050  
4051  
4052  
4053  
4054  
4055  
4056  
4057  
4058  
4059  
4060  
4061  
4062  
4063  
4064

015012 004737 013276  
 015016 010037 177566  
 015022 000207

```

.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

PCHR:  JSR    PC,TPREDY    ;PRINTER READY
        MOV    R0,TPB
        RTS    PC
  
```

4066  
4067  
4068  
4069  
4070  
4071  
4072  
4073  
4074  
4075  
4076  
4077  
4078  
4079  
4080  
4081  
4082  
4083  
4084  
4085  
4086  
4087  
4088  
4089  
4090  
4091  
4092  
4093  
4094  
4095  
4096  
4097  
4098  
4099  
4100  
4101  
4102  
4103  
4104  
4105  
4106  
4107  
4108  
4109  
4110  
4111  
4112  
4113  
4114  
4115  
4116  
4117  
4118  
4119  
4120  
4121

015024 004737 013740  
 015030 012701 007230  
 015034 112721 000040  
 015040 020127 007236  
 015044 001373  
 015046 000241  
 015050 010002  
 015052 042702 177770  
 015056 062702 000060

.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

PROCT: JSR PC,SAVREG ;SAVE REGS  
 MOV #OCTNUM,R1 ;POINTER TO MESSAGE  
 PROCT1: MOVB #40,(R1)+ ;FILL WITH SPACES  
 CMP R1,#OCTNUM+6 ;ALL DONE  
 BNE PROCT1 ;NO  
 PRCT1A: CLC ;CLEAR AT START  
 PROCT2: MOV R0,R2 ;SAVE CHARS  
 BIC #177770,R2 ;CLEAR ALL BUT BOTTOM 3 BITS  
 ADD #60,R2 ;NOW ASCII

|      |        |        |        |         |        |                 |                          |
|------|--------|--------|--------|---------|--------|-----------------|--------------------------|
| 4122 | 015062 | 110241 |        |         | MOVB   | R2,-(R1)        | :STORE IT IN MESSAGE     |
| 4123 | 015064 | 042700 | 000007 |         | BIC    | #7,R0           | :NOW CLEAR BOTTOM 3 BITS |
| 4124 | 015070 | 001404 |        |         | BEQ    | PROCT3          | :ALL DONE                |
| 4125 | 015072 | 006000 |        |         | ROR    | R0              | :ROTATE NEXT 3 BITS      |
| 4126 | 015074 | 006000 |        |         | ROR    | R0              |                          |
| 4127 | 015076 | 006000 |        |         | ROR    | R0              |                          |
| 4128 | 015100 | 000763 |        |         | BR     | PROCT2          | :GO DO NEXT              |
| 4129 | 015102 | 004737 | 014734 | PROCT3: | JSR    | PC,TYPOUT       | :TYPE MESSAGE            |
| 4130 | 015106 | 007230 |        |         | OCTNUM |                 | :MESSAGE ADD.            |
| 4131 | 015110 | 004737 | 013776 |         | JSR    | PC,RSTREG       | :RESTORE REGS            |
| 4132 | 015114 | 000207 |        |         | RTS    | PC              |                          |
| 4133 |        |        |        |         | :      |                 |                          |
| 4134 |        |        |        |         | :      |                 |                          |
| 4135 |        |        |        |         | :      |                 |                          |
| 4136 |        |        |        |         | :      |                 |                          |
| 4137 |        |        |        |         | :      |                 |                          |
| 4138 | 015116 | 004737 | 013740 | PRNT3:  | JSR    | PC,SAVREG       |                          |
| 4139 | 015122 | 042700 | 177000 |         | BIC    | #177000,R0      | :CLEAR UNWANTER          |
| 4140 | 015126 | 012737 | 002040 | 007230  | MOV    | #2040,OCTNUM    | :SET UP MESSAGE          |
| 4141 | 015134 | 012737 | 040040 | 007232  | MOV    | #40040,OCTNUM+2 | :WITH END CODE           |
| 4142 | 015142 | 012701 | 007233 |         | MOV    | #OCTNUM+3,R1    | :& POINTER               |
| 4143 | 015146 | 000737 |        |         | BR     | PRCT1A          | :JUMP INTO MAIN ROUTINE  |
| 4144 |        |        |        |         |        |                 | :WITH POINTERS SET UP    |



4146  
4147  
4148  
4149  
4150  
4151  
4152  
4153  
4154  
4155  
4156  
4157  
4158  
4159  
4160  
4161  
4162  
4163  
4164  
4165  
4166  
4167  
4168  
4169  
4170  
4171  
4172  
4173  
4174  
4175  
4176  
4177  
4178  
4179  
4180  
4181  
4182  
4183  
4184  
4185  
4186  
4187  
4188  
4189  
4190  
4191  
4192  
4193  
4194  
4195  
4196  
4197  
4198  
4199  
4200  
4201

.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

TYPDEC: JSR PC,SAVREG ;SAVE REGISTERS  
 TPDEC1: CLR R0  
 CLR RAND ;CLEAR INDICATOR

015150 004737 013740  
 015154 005000  
 015156 005037 007532

|      |        |        |        |        |             |                 |   |                        |
|------|--------|--------|--------|--------|-------------|-----------------|---|------------------------|
| 4202 | 015162 | 012737 | 007250 | 007604 | MOV         | #DECMMSG,STRADD | : | SET START ADDRESS      |
| 4203 | 015170 | 012737 | 000005 | 007606 | MOV         | #5,STRLEN       | : | ONLY 5 CHARS           |
| 4204 | 015176 | 012737 | 000060 | 007610 | MOV         | #60,LOWCHR      | : | MINIMUM CHAR           |
| 4205 | 015204 | 012737 | 000071 | 007612 | MOV         | #71,UPPCHR      | : | MAXIMUM CHAR           |
| 4206 | 015212 | 005037 | 007602 |        | CLR         | CNVFLG          | : | DON'T CONVERT TO UPPER |
| 4207 | 015216 | 004737 | 014232 |        | JSR         | PC,GETSTR       | : | GET A STRING           |
| 4208 | 015222 | 013704 | 007606 |        | MOV         | STRLEN,R4       | : | GET LENGHT READ IN     |
| 4209 | 015226 | 001433 |        |        | BEQ         | TPDEC5          | : | ZERO MEANS COMLPETE    |
| 4210 | 015230 | 012702 | 007250 |        | MOV         | #DECMMSG,R2     | : | POINT TO STRING        |
| 4211 | 015234 | 112201 |        |        | TPDEC2: MOV | (R2)+,R1        | : | GET A CHARACTER        |
| 4212 | 015236 | 000241 |        |        | CLC         |                 |   |                        |
| 4213 | 015240 | 006300 |        |        | ASL         | R0              | : | RESULT TIMES 2         |
| 4214 | 015242 | 103413 |        |        | BCS         | TPDEC3          | : | C BIT SET MEANS ERROR  |
| 4215 | 015244 | 010003 |        |        | MOV         | R0,R3           | : | ASVE IT                |
| 4216 | 015246 | 006300 |        |        | ASL         | R0              | : | TIMES 4                |
| 4217 | 015250 | 103410 |        |        | BCS         | TPDEC3          | : | C BIT MEANS ERROR      |
| 4218 | 015252 | 006300 |        |        | ASL         | R0              | : | TIMES 8                |
| 4219 | 015254 | 103406 |        |        | BCS         | TPDEC3          | : | C BIT MEANS OVERFLOW   |
| 4220 | 015256 | 060300 |        |        | ADD         | R3,R0           | : | ADD IN TO BE TIMES 12  |
| 4221 | 015260 | 103404 |        |        | BCS         | TPDEC3          | : | C BIT MEANS ERROR      |
| 4222 | 015262 | 042701 | 177760 |        | BIC         | #177760,R1      | : | CLEAR UNWANTED BITS    |
| 4223 | 015266 | 060100 |        |        | ADD         | R1,R0           | : | ADD IN NEW VALUE       |
| 4224 | 015270 | 103006 |        |        | BCC         | TPDEC4          | : | C BIT                  |
| 4225 | 015272 | 012700 | 000041 |        | TPDEC3: MOV | #41,R0          | : | OVERFLOW PRINT A !     |
| 4226 | 015276 | 004737 | 015012 |        | JSR         | PC,PCHR         |   |                        |
| 4227 | 015302 | 000137 | 015154 |        | JMP         | TPDEC1          | : | AND GO BAK TO START    |
| 4228 | 015306 | 005237 | 007532 |        | TPDEC4: INC | RAND            | : | SET CHAR SEEN          |
| 4229 | 015312 | 005304 |        |        | DEC         | R4              | : | REDUCE COUNT           |
| 4230 | 015314 | 001347 |        |        | BNE         | TPDEC2          | : | NON ZERO MEANS MORE    |
| 4231 | 015316 | 010037 | 007534 |        | TPDEC5: MOV | R0,TYPD1        | : | ALL DONE SAVE RESULT   |
| 4232 | 015322 | 004737 | 013776 |        | JSR         | PC,RSTREG       | : | RESTORE REGISTERS      |
| 4233 | 015326 | 013700 | 007534 |        | MOV         | TYPD1,R0        | : | RESTORE RESULT         |
| 4234 | 015332 | 000207 |        |        | RTS         | PC              | : | AND EXIT               |
| 4235 |        |        |        |        | :           |                 |   |                        |



4237  
4238  
4239  
4240  
4241  
4242  
4243  
4244  
4245  
4246  
4247  
4248  
4249  
4250  
4251  
4252  
4253  
4254  
4255  
4256  
4257  
4258  
4259  
4260  
4261  
4262  
4263  
4264  
4265  
4266  
4267  
4268  
4269  
4270  
4271  
4272  
4273  
4274  
4275  
4276  
4277  
4278  
4279  
4280  
4281  
4282  
4283  
4284  
4285  
4286  
4287  
4288  
4289  
4290  
4291  
4292

015334 005700  
015336 100005  
015340 005400  
015342 112737 000055 007240  
015350 000403  
015352 112737 000040 007240  
015360 004737 013740  
015364 012703 007241  
015370 112723 000040  
015374 022703 007246  
015400 001373  
015402 005001  
015404 020027 000012  
015410 103404  
015412 162700 000012  
015416 005201  
015420 000771  
015422 062700 000060  
015426 110043  
015430 010100

```
.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

BASM10: TST R0 ; GET SIGN OF R0
        BPL BASE10 ; POSITIVE..
        NEG R0 ; IF MINUS,MAKE POSITIVE..
        MOVB #55,BASE11 ; AND PRINT '-'
        BR BAS10A ; THEN CONTINUE..

BASE10: MOVB #' ,BASE11 ; PRINT LEADING SPACE

BASE10A: JSR PC,SAVREG ; NOW SAVE OLD REGS..
        MOV #BASE11+1,R3 ; SETUP OUTPUT POINTER..
        BASE1A: MOVB #' ,(R3)+ ; PRELOAD STRING WITH SPACES.
        CMP #BASE11+6,R3 ; ALL DONE?
        BNE BASE1A ; NO..

BASE1D: CLR R1 ; R1 IS RECEIVER...

BASE1B: CMP R0,#12 ; MORE THAN 10?
        BLO BASE1C ; YES IF BRANCH
        SUB #12,R0 ; DIV BY REPEAT SUB
        INC R1 ; TALLY COUNT
        BR BASE1B ; DO NEXT

BASE1C: ADD #'0,R0 ; MAKE INTO NUMBER
        MOVB R0,-(R3) ; SAVE IN STRING
        MOV R1,R0
```



VTV30K DIAGNOSTIC  
CVVTC.A.SRC 08-JAN-81 10:36 MACY11 30A(1052) 08-JAN-81 10:58 PAGE 62-1<sup>C 9</sup>  
PRINT A DECIMAL NUMBER

SEQ 0106

4293 015432 001363  
4294 015434 004737 013776  
4295 015440 004737 014734  
4296 015444 007240  
4297 015446 000207

BNE BASE1D  
JSR PC,RSTREG  
JSR PC,TYPOUT  
BASE11  
RTS PC

: RESTORE REGISTERS..  
: TYPE MESSAGE  
: THIS ONE...  
: AND EXIT

4299  
4300  
4301  
4302  
4303  
4304  
4305  
4306  
4307  
4308  
4309  
4310  
4311  
4312  
4313  
4314  
4315  
4316  
4317  
4318  
4319  
4320  
4321  
4322  
4323  
4324  
4325  
4326  
4327  
4328  
4329  
4330  
4331  
4332  
4333  
4334  
4335  
4336  
4337  
4338  
4339  
4340  
4341  
4342  
4343  
4344  
4345  
4346  
4347  
4348  
4349  
4350  
4351  
4352  
4353  
4354

.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

```

4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372 015450 004737 013740
4373 015454
4374 015464
4375 015500 004737 014734
4376 015504 006635
4377 015506 004737 014034
4378 015512 005737 007532
4379 015516 001506
4380 015520 010037 007536
4381 015524 032737 000C01 007536
4382 015532 001404
4383 015534 004737 014734
4384 015540 006707
4385 015542 000756
4386 015544 012737 016354 000004
4387 015552 012737 000340 000006
4388 015560 005037 007544
4389 015564 005777 171746
4390 015570 005737 007544
4391 015574 001341
4392 015576 004737 014700
4393 015602 013700 007536
4394 015606 004737 015024
4395 015612 012700 000040
4396 015616 004737 015012
4397 015622 017700 171710
4398 015626 004737 015024
4399 015632 004737 014734
4400 015636 006642
4401 015640 004737 014034
4402 015644 005737 007532
4403 015650 001402
4404 015652 010077 171660
4405 015656 013700 007616
4406 015662 001706
4407 015664 020027 000015
4408 015670 001703
4409 015672 020027 000012
4410 015676 001006

:
: IMPLICIT OUTPUT PARAMETERS:
:
: THE LOCATIONS SPECIFIED WILL HAVE BEEN
: MODIFIED
:
: COMPLETION CODES:
:
: NONE
:
: POSSIBLE ERROR CODES:
:
: NONE
:
: MODIFY: JSR PC, SAVREG ;SAVE REGISTERS
: PSWREA MODSAV
: PSWSET #340
: MODI1: JSR PC, TYP0UT ;PROMPT $ FOR AN ADDRESS
: MODADM ;TO HAVE A LOOK AT
: JSR PC, OCTIN ;READ REPLY
: TST RAND ;ANYTHING READ ?
: BEQ MODXIT ;NO, SO EXIT
: MOV R0, MODADR ;ELSE SAVE OUR ADDRESS
: MODI2: BIT #1, MODADR ;IS IT EVEN ?
: BEQ MODI3 ;YES WE CAN USE IT
: JSR PC, TYP0UT ;ELSE SAY IT IS AN ODD
: ODAMSG ;ADDRESS, AND REPROMPT
: BR MODI1
: MODI3: MOV #NOMTRP, 4 ;PLUG TRAP THRU 4
: MOV #340, 6 ;FOR NOM TESTS
: CLR TRPERR ;CLEAR NOM FLAG
: TST @MODADR ;TEST OUR ADDRESS
: TST TRPERR ;DOES IT EXIST
: BNE MODI1 ;NO TRY AGAIN
: JSR PC, CRLF ;START ON A NEW LINE
: MOV MODADR, R0 ;PRINT OUR ADDRESS
: JSR PC, PROCT
: MOV #40, R0 ;THEN A SPACE
: JSR PC, PCHR ;AS A SEPARATOR
: MOV @MODADR, R0 ;THEN PRINT THE CONTENTS
: JSR PC, PROCT
: JSR PC, TYP0UT ;PROMPT FOR THE NEW
: MODSPA ;CONTENTS '/'
: JSR PC, OCTIN ;READ REPLY
: TST RAND ;ANY THING GIVEN
: BEQ MODI4 ;NO, DON'T UPDATE
: MOV R0, @MODADR ;ELSE SET NEW VALUE
: MODI4: MOV RANDC, R0 ; GET TERMINATOR
: BEQ MODI1 ; NULL MEANS <CR>
: CMP R0, #15 ;WAS IT <CR> ?
: BEQ MODI1 ;YES GET NEXT ADDRESS
: CMP R0, #12 ;WAS IT <LF> ?
: BNE MODI5 ;NO

```



|      |        |        |        |        |             |           |                              |
|------|--------|--------|--------|--------|-------------|-----------|------------------------------|
| 4411 | 015700 | 004737 | 014700 |        | JSR         | PC,CRLF   | : NEWLINE                    |
| 4412 | 015704 | 062737 | 000002 | 007536 | ADD         | #2,MODADR | :UPDATE THE ADDRESS BY 2     |
| 4413 | 015712 | 000704 |        |        | BR          | MOD12     | : THEN TRY THIS ADDRESS      |
| 4414 | 015714 | 020027 | 000033 |        | MOD15: CMP  | R0,#33    | :WAS IT EXIT ?               |
| 4415 | 015720 | 001405 |        |        | BEQ         | MODXIT    | :YES DISSAPPEAR              |
| 4416 | 015722 | 012700 | 000077 |        | MOV         | #77,R0    | :ELSE GIVE A ?               |
| 4417 | 015726 | 004737 | 015012 |        | JSR         | PC,PCHR   | :AS AN ERROR                 |
| 4418 | 015732 | 000751 |        |        | BR          | MOD14     | :THEN TRY AGAIN              |
| 4419 | 015734 | 012737 | 000006 | 000004 | MODXIT: MOV | #6,4      | :PLUG BACK TRAP THRU 4       |
| 4420 | 015742 | 012737 | 000004 | 000006 | MOV         | #4,6      | :WITH WHAT IS WAS            |
| 4421 | 015750 | 004737 | 014700 |        | JSR         | PC,CRLF   | :PUT OURSELVES ON A NEW LINE |
| 4422 | 015754 |        |        |        | PSWSET      | MODSAV    |                              |
| 4423 | 015770 | 004737 | 013776 |        | JSR         | PC,RSTREG | :RESTORE REGISTERS           |
| 4424 | 015774 | 000207 |        |        | RTS         | PC        | :AND GO AWAY                 |
| 4425 |        |        |        |        |             |           |                              |

4427  
4428  
4429  
4430  
4431  
4432  
4433  
4434  
4435  
4436  
4437  
4438  
4439  
4440  
4441  
4442  
4443  
4444  
4445  
4446  
4447  
4448  
4449  
4450  
4451  
4452  
4453  
4454  
4455  
4456  
4457  
4458  
4459  
4460  
4461  
4462  
4463  
4464  
4465  
4466  
4467  
4468  
4469  
4470  
4471  
4472  
4473  
4474  
4475  
4476  
4477  
4478  
4479  
4480  
4481  
4482

.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.

```

4483
4484 015776 005037 007454      SELBUS: CLR      TESTNO
4485 016002 004737 013740      JSR      PC,SAVREG      ; SAVE REGISTERS
4486 016006 012737 016354 000004  MOV      #NXMTRP,4      ; PLUG TRAP THRU 4
4487 016014 012737 000340 000006  MOV      #340,6
4488 016022 004737 014734      1$: JSR      PC,TYPOUT      ; ASK FOR NUMBER OF DEVICES
4489 016026 005660      JSR      NUMDEV
4490 016030 004737 015150      JSR      PC,TYPDEC      ; READ DECIMAL REPLY
4491 016034 005737 007532      TST      RAND      ; NULL REPLY MEANS 1
4492 016040 001002      BNE      2$
4493 016042 012700 000001      MOV      #1,R0
4494 016046 020027 000024      2$: CMP      R0,#20.      ; IS THE NUMBER TOO BIG ?
4495 016052 003363      BGT      1$      ; YES
4496 016054 005700      TST      R0      ; OR TOO SMALL
4497 016056 003761      BLE      1$      ; OOPS, REPROMPT
4498 016060 010037 007304      MOV      R0,DEVCNT      ; ELSE SAVE COUNT AWAY
4499 016064 023727 007304 000001  CMP      DEVCNT,#1      ; ONE DEVICE SELECTED
4500 016072 001406      BEQ      BUSSE1      ; YES
4501 016074 004737 014734      JSR      PC,TYPOUT
4502 016100 006144      GOMSI
4503 016102 004737 013656      3$: JSR      PC,TYPCTC      ; NO- ALL DEVICES MUST BE
4504 016106 000775      BR      3$      ; SEQUENTIAL
4505
4506      ; NOW GET OUR FIRST ADDRESS
4507
4508 016110 004737 014734      BUSSE1: JSR      PC,TYPOUT      ; FIRST BUS ADDRESS =
4509 016114 006653      BMSG
4510 016116 004737 014034      1$: JSR      PC,OCTIN      ; READ OCTAL REPLY
4511 016122 005737 007532      TST      RAND      ; DON'T ALLOW DEFAULTS
4512 016126 001004      BNE      2$
4513 016130 004737 014734      JSR      PC,TYPOUT
4514 016134 006762      NODEFM
4515 016136 000767      BR      1$
4516 016140 032700 000005      2$: BIT      #5,R0      ; THESE BITS CLEAR
4517 016144 001406      BEQ      3$      ; YES
4518 016146 004737 014734      4$: JSR      PC,TYPOUT
4519 016152 006707      ODMSG
4520 016154 004737 015024      JSR      PC,PROCT
4521 016160 000753      BR      BUSSE1
4522 016162 032700 100000      3$: BIT      #100000,R0      ; IT MUST BE ON THE 10 PAGE
4523 016166 001767      BEQ      4$
4524 016170 010037 007542      BUSSE2: MOV      R0,BASADD      ; SAVE START ADDRESS
4525 016174 013701 007304      MOV      DEVCNT,R1      ; GET DEVICE COUNT
4526 016200 012702 007306      MOV      #CSRO,R2      ; AND WHERE DATA GOES
4527 016204 013737 007542 007566  MOV      BASADD,DATA
4528 016212 005037 007544      1$: CLR      TRPERR      ; CLEAR NXM FLAG
4529 016216 005777 171344      TST      @DATA      ; CHECKIT
4530 016222 005737 007544      TST      TRPERR      ; OK ?
4531 016226 001411      BEQ      2$      ; YES
4532
4533      ; ERROR-
4534      ; CHECK 'LEVB CLOCK H'
4535      ; 'LEVB VA02 H'
4536      ; BUS CYCLE TIMING ON LEVA.
4537 016230 013737 007566 007572  MOV      DATA,ADDRES
4538 016236 004737 016666      JSR      PC,ERROR      ; NO- THIS IS NAUGHTY

```



```

4539 016242 020077          77+A
4540 016244 104410        TRAP+10
4541 016246 016212          1$
4542 016250 000717        BR      BUSSE1
4543 016252 013722 007566  2$:  MOV    DATA,(R2)+
4544 016256 062737 000002 007566  ADD    #2,DATA
4545 016264 005037 007544  3$:  CLR    TRPERR
4546 016270 005777 171272    TST    @DATA
4547 016274 005737 007544    TST    TRPERR
4548 016300 001403        BEQ     4$
4549
4550      ; ERROR-
4551      ; CHECK E4 ON LEVA
4552      ; BUS CYCLE TIMING ON LEVA
4552 016302 004737 016666    JSR     PC,ERROR
4553 016306 040076
4554 016310 104410        76+D
4555 016312 016264        4$:  TRAP+10
4556 016314 013722 007566    3$
4557 016320 062737 000006 007566  MOV    DATA,(R2)+
4558 016326 005301        ADD    #6,DATA
4559 016330 001330        DEC    R1
4560 016332 012737 000006 000004  BNE    1$
4561 016340 012737 000004 000006  MOV    #6,4
4562 016346 004737 013776    MOV    #4,6
4563 016352 000207        JSR     PC,RSTREG
4564      RTS    PC
      ;
      ; SAVE DATA AWAY
      ; POINT TO DBUFF ADDRESS
      ; CLEAR NXM FLAG
      ; CHECK FOR NXM
      ; IS IT ?
      ; NO- ALL IS WELL
      ;
      ; OOPS- NAUGHTY
      ; LOOP ON
      ; SAVE DBUFF ADDRESS
      ; POINT TO NEXT CSR
      ; ALL CSRS DONE
      ; NO
      ; RESET TRAP POINTERS
      ; WITH THE IOT TRAP
      ; YES EXIT
  
```

VTV30K DIAGNOSTIC MACY11 30A(1052) 08-JAN-61 10:58 PAGE 65<sup>J</sup> 9  
CVVTCA.SRC 08-JAN-81 10:36 SELECT BUS ADDRESSES

SEQ 0113

|      |        |        |        |
|------|--------|--------|--------|
| 4566 | 016354 | 004737 | 014734 |
| 4567 | 016360 | 006732 |        |
| 4568 | 016362 | 005237 | 007544 |
| 4569 | 016366 | 000002 |        |

|         |       |             |
|---------|-------|-------------|
| NXMTRP: | JSR   | PC, TYP OUT |
|         | NXMSG |             |
|         | INC   | TRPERR      |
|         | RTI   |             |

4571  
4572  
4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585  
4586  
4587  
4588  
4589  
4590  
4591  
4592  
4593  
4594  
4595  
4596  
4597  
4598  
4599  
4600  
4601  
4602  
4603  
4604  
4605  
4606  
4607  
4608  
4609  
4610  
4611  
4612  
4613  
4614  
4615  
4616  
4617  
4618  
4619  
4620  
4621  
4622  
4623  
4624  
4625  
4626

.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<br>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



```

4627
4628
4629
4630 016370 042703 177770
4631 016374 004737 013740
4632 016400 006303
4633 016402 000173 016406
4634 016406 016426
4635 016410 016432
4636 016412 016440
4637 016414 016446
4638 016416 016454
4639 016420 016464
4640 016422 016522
4641 016424 016642
4642 016426 005000
4643 016430 000507
4644 016432 005000
4645 016434 005100
4646 016436 000504
4647 016440 012700 052525
4648 016444 000501
4649 016446 012700 125252
4650 016452 000476
4651 016454 000241
4652 016456 004737 016476
4653 016462 000472
4654 016464 000241
4655 016466 004737 016476
4656 016472 005100
4657 016474 000465
4658 016476 006037 016520
4659 016502 001003
4660 016504 012737 100000 016520
4661 016512 013700 016520
4662 016516 000207
4663 016520 000001
4664 016522 012737 000005 007554
4665 016530 004737 016542
4666 016534 013700 007552
4667 016540 000443
4668 016542 013702 007552
4669 016546 001002
4670 016550 013702 007560
4671 016554 032737 000777 007554
4672 016562 001003
4673 016564 012737 000001 007554
4674 016572 013703 007554
4675 016576 013702 007552
4676 016602 033702 007556
4677 016606 001405
4678 016610 005102
4679 016612 033702 007556
4680 016616 001401
4681 016620 000402
4682 016622 000241

:
:
:
GENER: BIC #177770,R3
JSR PC,SAVRÉG
ASL R3
JMP @GENSEL(R3)
GENSEL: GEN0 ;ALL ZERO WORD
GEN1 ;ALL ONE WORD
GEN52 ;52 PATTERN
GEN25 ;25 PATTERN
GENR1 ;ROTATE '1' EACH CALL
GENR0 ;ROTATE '0' EACH CALL
GENRAN ;RANDOM NUMBER
GENINC ;INCREMENTING COUNT
GEN0: CLR R0 ;0>R0
BR GENEX
GEN1: CLR R0 ;NOT0>R0
COM R0
BR GENEX
GEN52: MOV #52525,R0 ;5252>R0
BR GENEX
GEN25: MOV #125252,R0 ;125252>R0
BR GENEX
GENR1: CLC
JSR PC,GENROT ;SHIFT 1 > R0
BR GENEX
GENR0: CLC
JSR PC,GENROT ;SHIFT 0 > R0
COM R0
BR GENEX
GENROT: ROR GENISH ;ROTATE 1 PATTERN
BNE GENER1 ;= 0?
MOV #100000,GENISH ;YES, SET MSB
MOV GENISH,R0 ;PUT 1 IN R0
RTS PC ;AND EXIT
GENISH: 1
GENRAN: MOV #5,RANSEL ;SET SELECT VALUE TO 5
JSR PC,RANGEN ;GENERATE RANDOM NUMBER IN R0
MOV RANDN,R0
BR GENEX
RANGEN: MOV RANDN,R2
BNE RAN1 ;IS RANDOM = 0
MOV RANST,R2 ;YES, PUT RANDOM START VALUE IN
BIT #777,RANSEL ;NO: IS RANSEL SELECT VALUE = 0
BNE RAN2 ;NO
MOV #1,RANSEL ;YES: SET RANSEL = 1
RAN2: MOV RANSEL,R3
MOV RANDN,R2
BIT RANMTA,R2 ;GET R2 <0 AND 1>
BEQ RANCLC
COM R2
BIT RANMTA,R2
BEQ RANCLC
BR RANSEC
RANCLC: CLC

```

|      |        |        |        |             |           |                           |
|------|--------|--------|--------|-------------|-----------|---------------------------|
| 4683 | 016624 | 000401 |        | BR          | RAN4      |                           |
| 4684 | 016626 | 000261 |        | ANSEC: SEC  |           |                           |
| 4685 | 016630 | 006037 | 007552 | RAN4: ROR   | RANDN     | :ROTATE C TO B15          |
| 4686 | 016634 | 005303 |        | DEC         | R3        | :IS THIS NUMBER REQUIRED? |
| 4687 | 016636 | 001357 |        | BNE         | RAN2+4    | :NO, GET ANOTHER          |
| 4688 | 016640 | 000207 |        | RTS         | PC        | :YES, EXIT                |
| 4689 | 016642 | 013700 | 007562 | GENINC: MOV | GOOD,R0   | :INCREMENTS LOC. 'GOOD'   |
| 4690 | 016646 | 005200 |        | INC         | R0        |                           |
| 4691 | 016650 | 010037 | 007562 | GENEX: MOV  | R0,GOOD   |                           |
| 4692 | 016654 | 010066 | 000002 | MOV         | R0,2(SP)  | :LOAD R0 MAINROUTINE      |
| 4693 | 016660 | 004737 | 013776 | JSR         | PC,RSTREG |                           |
| 4694 | 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:

4696  
4697  
4698  
4699  
4700  
4701  
4702  
4703  
4704  
4705  
4706  
4707  
4708  
4709  
4710  
4711  
4712  
4713  
4714  
4715  
4716  
4717  
4718  
4719  
4720  
4721  
4722  
4723  
4724  
4725  
4726  
4727  
4728  
4729  
4730  
4731  
4732  
4733  
4734  
4735  
4736  
4737  
4738  
4739  
4740  
4741  
4742  
4743  
4744  
4745  
4746  
4747  
4748  
4749  
4750  
4751



4752  
4753  
4754  
4755  
4756  
4757  
4758  
4759  
4760  
4761  
4762  
4763  
4764  
4765  
4766  
4767  
4768  
4769  
4770  
4771  
4772  
4773  
4774  
4775  
4776  
4777  
4778  
4779  
4780  
4781  
4782  
4783  
4784  
4785  
4786  
4787  
4788  
4789  
4790  
4791  
4792  
4793  
4794  
4795  
4796  
4797  
4798  
4799  
4800  
4801  
4802  
4803  
4804  
4805  
4806  
4807

```

      NONE

      IMPLICIT OUTPUT PARAMETERS:

      THE APPROPRAITE ERROR MESSAGE WILL BE PRINTED
      IF PERMITTED.

      COMPLETION CODES:

      NONE

      POSSIBLE ERROR CODES:

      NONE

      20-FEB-80
ERROR:  JSR      PC,SAVREG
      MOV      #1,ERRFLG      ; SET ERROR FLAG
      INC      ERRDIS         ; INC COUNT
      BNE      1$             ; CHECK FOR OVERFLOW
      INC      ERRDI1         ; THEN INC OVERFLOW COUNT
1$:     BIT      #40000,@SWR   ;SUPPRESS PRINTOUT
      BNE      ERHALT                ;YES
      MOV      @(SP),ERRARG      ;NO
      JSR      PC,TYPOUT         ;TYPE MESSAGE
      MSG1
      MOV      TESTNO,R0
      SWAB     R0
      ASL      R0
      BISB     ERRARG,R0
      JSR      PC,PROCT
      JSR      PC,TYPOUT
      MSG2
      MOV      (SP),R0
      JSR      PC,PROCT                ;PRINT 6 DIGITS
      BITB     #200,ERRARG+1
      BEQ      ERRSV1
      JSR      PC,TYPOUT
      MSG3
      MOV      GOOD,R0
      JSR      PC,PROCT
      JSR      PC,TYPOUT
      MSG4
      MOV      BAD,R0
      JSR      PC,PROCT
ERRSV1: BITB     #100,ERRARG+1      ;D SET ?
      BEQ      ERRSV2              ;NO
      JSR      PC,TYPOUT
      MSG5
      MOV      DATA,R0
      JSR      PC,PROCT

```

VTV30K DIAGNOSTIC MACY11 30A(1052) 08-JAN-81 10:58 PAGE 67-2 C 10  
 CVTCA.SRC 08-JAN-81 10:36 PRINT ERROR MESSAGES

SEQ 0119

|      |        |        |        |        |         |      |              |             |
|------|--------|--------|--------|--------|---------|------|--------------|-------------|
| 4808 | 017064 | 132737 | 000040 | 007577 | ERRSV2: | BITB | #40,ERRARG+1 | :A SET ?    |
| 4809 | 017072 | 001407 |        |        |         | BEQ  | ERRSV3       | :NO         |
| 4810 | 017074 | 004737 | 014734 |        |         | JSR  | PC,TYPOUT    |             |
| 4811 | 017100 | 007135 |        |        |         | MSG6 |              |             |
| 4812 | 017102 | 013700 | 007572 |        |         | MOV  | ADDRES,R0    |             |
| 4813 | 017106 | 004737 | 015024 |        |         | JSR  | PC,PROCT     |             |
| 4814 | 017112 | 132737 | 000020 | 007577 | ERRSV3: | BITB | #20,ERRARG+1 | :S SET ?    |
| 4815 | 017120 | 001407 |        |        |         | BEQ  | ERRSV4       | :NO         |
| 4816 | 017122 | 004737 | 014734 |        |         | JSR  | PC,TYPOUT    |             |
| 4817 | 017126 | 007153 |        |        |         | MSG7 |              |             |
| 4818 | 017130 | 013700 | 007570 |        |         | MOV  | STATUS,R0    |             |
| 4819 | 017134 | 004737 | 015024 |        |         | JSR  | PC,PROCT     |             |
| 4820 | 017140 | 132737 | 000010 | 007577 | ERRSV4: | BITB | #10,ERRARG+1 | :C SET      |
| 4821 | 017146 | 001407 |        |        |         | BEQ  | ERRSV5       | :NO         |
| 4822 | 017150 | 004737 | 014734 |        |         | JSR  | PC,TYPOUT    |             |
| 4823 | 017154 | 007166 |        |        |         | MSG8 |              |             |
| 4824 | 017156 | 013700 | 007600 |        |         | MOV  | CALLPC,R0    |             |
| 4825 | 017162 | 004737 | 015024 |        |         | JSR  | PC,PROCT     |             |
| 4826 | 017166 | 004737 | 014700 |        | ERRSV5: | JSR  | PC,CRLF      |             |
| 4827 | 017172 | 004737 | 014734 |        |         | JSR  | PC,TYPOUT    |             |
| 4828 | 017176 | 007205 |        |        |         | MSG9 |              |             |
| 4829 | 017200 | 013700 | 007574 |        |         | MOV  | ERRDIS,R0    |             |
| 4830 | 017204 | 004737 | 015352 |        |         | JSR  | PC,BASE10    |             |
| 4831 | 017210 | 032777 | 100000 | 170264 | ERHALT: | BIT  | #100000,@SWR |             |
| 4832 | 017216 | 001004 |        |        |         | BNE  | NOHALT       |             |
| 4833 | 017220 | 013700 | 007574 |        |         | MOV  | ERRDIS,R0    |             |
| 4834 | 017224 | 004737 | 013546 |        |         | JSR  | PC,MONIT     | : GO TO SWR |
| 4835 | 017230 | 004737 | 013776 |        | NOHALT: | JSR  | PC,RSTREG    |             |
| 4836 | 017234 | 062716 | 000002 |        |         | ADD  | #2,(SP)      |             |
| 4837 | 017240 | 000207 |        |        |         | RTS  | PC           |             |



4839  
4840  
4841  
4842  
4843  
4844  
4845 017242 360 360 360  
4846 017252 010 010 010  
4847 017262 030 030 030  
4848 017272 074 074 074  
4849 017302 000 000 000  
4850 017312 000 000 000  
4851 017322 000 000 377  
4852 017332 010 010 010  
4853 017342 030 030 030  
4854 017352 074 074 377  
4855 017362 001 002 004  
4856 017372 200 100 040  
4857 017402 000 014 022  
4858 017412 201 102 044  
4859 017422 074 176 377  
4860 017432 030 074 176  
4861 017442 010 010 024  
4862 017452 010 010 034  
4863 017462 252 252 252  
4864 017472 314 314 314  
4865 017502 377 000 377  
4866 017512 377 377 000  
4867 017522 377 203 205  
4868 017532 377 201 201  
4869 017542 377 376 374  
4870 017552 377 177 077  
4871 017562 200 300 340  
4872 017572 001 003 007  
4873 017602 003 007 016  
4874 017612 300 340 160  
4875 017622 303 347 176  
4876 017632 000 160 120  
4877 017642 000 000 000  
4878 017652 010 010 010  
4879 017662 024 024 024  
4880 017672 024 024 066  
4881 017702 010 036 040  
4882 017712 000 062 062  
4883 017722 000 020 050  
4884 017732 000 030 030  
4885 017742 000 010 020  
4886 017752 000 010 004  
4887 017762 000 052 034  
4888 017772 000 000 010  
4889 020002 000 000 000  
4890 020012 000 000 000  
4891 020022 000 000 000  
4892 020032 000 002 002  
4893 020042 000 034 042  
4894 020052 000 010 030

.SBTTL NORMAL CHARACTER SET.

.NLIST BEX

:THIS IS THE VTV30K CHARACTER SET IN ITS MAIN  
:VERSION, WITH A POUND SIGN AT ADDRESS 13,8 A  
:DEGREE SIGN AT ADDRESS 32.

NORCH: .BYTE 360,360,360,360,360,360,360,360  
.BYTE 010,010,010,010,010,010,010,010  
.BYTE 030,030,030,030,030,030,030,030  
.BYTE 074,074,074,074,074,074,074,074  
.BYTE 000,000,000,000,377,000,000,000  
.BYTE 000,000,000,000,377,377,000,000  
.BYTE 000,000,377,377,377,377,000,000  
.BYTE 010,010,010,010,377,010,010,010  
.BYTE 030,030,030,377,377,030,030,030  
.BYTE 074,074,377,377,377,377,074,074 :10  
.BYTE 001,002,004,010,020,040,100,200  
.BYTE 200,100,040,020,010,004,002,001  
.BYTE 000,014,022,040,170,040,100,176  
.BYTE 201,102,044,030,030,044,102,201  
.BYTE 074,176,377,377,377,377,176,074  
.BYTE 030,074,176,377,377,176,074,030  
.BYTE 010,010,024,042,301,042,024,010  
.BYTE 010,010,034,076,377,076,034,010  
.BYTE 252,252,252,252,252,252,252,252 :20  
.BYTE 314,314,314,314,314,314,314,314  
.BYTE 377,000,377,000,377,000,377,000  
.BYTE 377,377,000,000,377,377,000,000  
.BYTE 377,203,205,211,221,241,301,377  
.BYTE 377,201,201,201,201,201,201,377  
.BYTE 377,376,374,370,360,340,300,200  
.BYTE 377,177,077,037,017,007,003,001  
.BYTE 200,300,340,360,370,374,376,377  
.BYTE 001,003,007,017,037,077,177,377  
.BYTE 003,007,016,034,070,160,340,300  
.BYTE 300,340,160,070,034,016,007,003 :30  
.BYTE 303,347,176,074,074,176,347,303  
.BYTE 000,160,120,160,000,000,000,000  
.BYTE 000,000,000,000,000,000,000,000  
.BYTE 010,010,010,010,010,000,010,000  
.BYTE 024,024,024,000,000,000,000,000  
.BYTE 024,024,066,000,066,024,024,000  
.BYTE 010,036,040,034,002,074,010,000  
.BYTE 0, 62, 62, 4, 10, 20, 46, 46  
.BYTE 0, 20, 50, 50, 20, 52, 44, 32 :40(')  
.BYTE 0, 30, 30, 30, 0, 0, 0, 0  
.BYTE 0, 10, 20, 40, 40, 40, 20, 10  
.BYTE 0, 10, 4, 2, 2, 2, 4, 10  
.BYTE 0, 52, 34, 76, 34, 52, 0, 0  
.BYTE 0, 0, 10, 10, 76, 10, 10, 0  
.BYTE 0, 0, 0, 0, 30, 30, 10, 20  
.BYTE 0, 0, 0, 0, 76, 0, 0, 0  
.BYTE 0, 0, 0, 0, 0, 0, 30, 30  
.BYTE 0, 2, 2, 4, 10, 20, 40, 40  
.BYTE 000,034,042,046,052,062,042,034 :50(1)  
.BYTE 000,010,030,010,010,010,010,034



|      |        |     |     |     |       |                                 |        |
|------|--------|-----|-----|-----|-------|---------------------------------|--------|
| 4895 | 020062 | 000 | 034 | 042 | .BYTE | 000,034,042,002,034,040,040,076 |        |
| 4896 | 020072 | 000 | 034 | 042 | .BYTE | 000,034,042,002,014,002,042,034 |        |
| 4897 | 020102 | 000 | 004 | 014 | .BYTE | 000,004,014,024,044,076,004,004 |        |
| 4898 | 020112 | 000 | 076 | 040 | .BYTE | 0,76,40,74,2,2,42,34            |        |
| 4899 | 020122 | 000 | 014 | 020 | .BYTE | 0,14,20,40,74,42,42,34          |        |
| 4900 | 020132 | 000 | 076 | 002 | .BYTE | 0,76,2,4,10,20,20,20            |        |
| 4901 | 020142 | 000 | 034 | 042 | .BYTE | 0,34,42,42,34,42,42,34          |        |
| 4902 | 020152 | 000 | 034 | 042 | .BYTE | 0,34,42,42,36,2,4,30            |        |
| 4903 | 020162 | 000 | 000 | 030 | .BYTE | 0,0,30,30,0,30,30,0             |        |
| 4904 | 020172 | 000 | 030 | 030 | .BYTE | 0,30,30,0,30,30,10,20           | :60(,) |
| 4905 | 020202 | 000 | 004 | 010 | .BYTE | 0,4,10,20,40,20,10,4            |        |
| 4906 | 020212 | 000 | 000 | 000 | .BYTE | 0,0,0,76,0,76,0,0               |        |
| 4907 | 020222 | 000 | 020 | 010 | .BYTE | 0,20,10,4,2,4,10,20             |        |
| 4908 | 020232 | 000 | 030 | 044 | .BYTE | 0,30,44,4,10,10,0,10            |        |
| 4909 | 020242 | 000 | 074 | 102 | .BYTE | 0,74,102,132,132,104,40,0       |        |
| 4910 | 020252 | 000 | 010 | 024 | .BYTE | 0,10,24,42,42,76,42,42          |        |
| 4911 | 020262 | 000 | 074 | 022 | .BYTE | 0,74,22,22,34,22,22,74          |        |
| 4912 | 020272 | 000 | 034 | 042 | .BYTE | 0,34,42,40,40,40,42,34          |        |
| 4913 | 020302 | 000 | 074 | 022 | .BYTE | 0,74,22,22,22,22,22,74          |        |
| 4914 | 020312 | 000 | 076 | 040 | .BYTE | 0,76,40,40,70,40,40,76          | :70(E) |
| 4915 | 020322 | 000 | 076 | 040 | .BYTE | 0,76,40,40,70,40,40,40          |        |
| 4916 | 020332 | 000 | 036 | 040 | .BYTE | 0,36,40,40,46,42,42,36          |        |
| 4917 | 020342 | 000 | 042 | 042 | .BYTE | 0,42,42,42,76,42,42,42          |        |
| 4918 | 020352 | 000 | 034 | 010 | .BYTE | 0,34,10,10,10,10,10,34          |        |
| 4919 | 020362 | 000 | 002 | 002 | .BYTE | 0,2,2,2,2,2,42,34               |        |
| 4920 | 020372 | 000 | 042 | 044 | .BYTE | 0,42,44,50,60,50,44,42          |        |
| 4921 | 020402 | 000 | 040 | 040 | .BYTE | 0,40,40,40,40,40,40,76          |        |
| 4922 | 020412 | 000 | 042 | 066 | .BYTE | 0,42,66,52,42,42,42,42          |        |
| 4923 | 020422 | 000 | 042 | 062 | .BYTE | 0,42,62,52,46,42,42,42          |        |
| 4924 | 020432 | 000 | 034 | 042 | .BYTE | 0,34,42,42,42,42,42,34          | :80(O) |
| 4925 | 020442 | 000 | 074 | 042 | .BYTE | 0,74,42,42,74,40,40,40          |        |
| 4926 | 020452 | 000 | 034 | 042 | .BYTE | 0,34,42,42,42,52,44,32          |        |
| 4927 | 020462 | 000 | 074 | 042 | .BYTE | 0,74,42,42,74,50,44,42          |        |
| 4928 | 020472 | 000 | 036 | 040 | .BYTE | 0,36,40,40,34,2,2,74            |        |
| 4929 | 020502 | 000 | 076 | 010 | .BYTE | 0,76,10,10,10,10,10,10          |        |
| 4930 | 020512 | 000 | 042 | 042 | .BYTE | 0,42,42,42,42,42,42,34          |        |
| 4931 | 020522 | 000 | 042 | 042 | .BYTE | 0,42,42,42,24,24,10,10          |        |
| 4932 | 020532 | 000 | 042 | 042 | .BYTE | 0,42,42,42,42,52,66,42          |        |
| 4933 | 020542 | 000 | 042 | 042 | .BYTE | 0,42,42,24,10,24,42,42          |        |
| 4934 | 020552 | 000 | 042 | 042 | .BYTE | 0,42,42,24,10,10,10,10          | :90(Y) |
| 4935 | 020562 | 000 | 076 | 002 | .BYTE | 0,76,2,4,10,20,40,76            |        |
| 4936 | 020572 | 000 | 070 | 040 | .BYTE | 000,070,040,040,040,040,040,070 |        |
| 4937 | 020602 | 000 | 040 | 040 | .BYTE | 000,040,040,020,010,004,002,002 |        |
| 4938 | 020612 | 000 | 016 | 002 | .BYTE | 000,016,002,002,002,002,002,016 |        |
| 4939 | 020622 | 030 | 074 | 176 | .BYTE | 030,074,176,377,030,030,030,030 |        |
| 4940 | 020632 | 010 | 014 | 016 | .BYTE | 010,014,016,377,377,016,014,010 |        |
| 4941 | 020642 | 030 | 030 | 030 | .BYTE | 030,030,030,030,377,176,074,030 |        |
| 4942 | 020652 | 020 | 060 | 160 | .BYTE | 020,060,160,377,377,160,060,020 |        |
| 4943 | 020662 | 000 | 377 | 377 | .BYTE | 000,377,377,377,377,377,377,377 |        |
| 4944 | 020672 | 000 | 000 | 377 | .BYTE | 000,000,377,377,377,377,377,377 | :100   |
| 4945 | 020702 | 000 | 000 | 000 | .BYTE | 000,000,000,377,377,377,377,377 |        |
| 4946 | 020712 | 000 | 000 | 000 | .BYTE | 000,000,000,000,377,377,377,377 |        |
| 4947 | 020722 | 000 | 000 | 000 | .BYTE | 000,000,000,000,000,377,377,377 |        |
| 4948 | 020732 | 000 | 000 | 000 | .BYTE | 000,000,000,000,000,000,377,377 |        |
| 4949 | 020742 | 000 | 000 | 000 | .BYTE | 000,000,000,000,000,000,000,377 |        |
| 4950 | 020752 | 376 | 376 | 376 | .BYTE | 376,376,376,376,376,376,376,376 |        |

|      |        |     |     |     |       |                                 |      |
|------|--------|-----|-----|-----|-------|---------------------------------|------|
| 4951 | 020762 | 374 | 374 | 374 | .BYTE | 374,374,374,374,374,374,374,374 |      |
| 4952 | 020772 | 370 | 370 | 370 | .BYTE | 370,370,370,370,370,370,370,370 |      |
| 4953 | 021002 | 360 | 360 | 360 | .BYTE | 360,360,360,360,360,360,360,360 |      |
| 4954 | 021012 | 340 | 340 | 340 | .BYTE | 340,340,340,340,340,340,340,340 | :110 |
| 4955 | 021022 | 300 | 300 | 300 | .BYTE | 300,300,300,300,300,300,300,300 |      |
| 4956 | 021032 | 200 | 200 | 200 | .BYTE | 200,200,200,200,200,200,200,200 |      |
| 4957 | 021042 | 200 | 200 | 200 | .BYTE | 200,200,200,200,200,200,200,377 |      |
| 4958 | 021052 | 377 | 001 | 001 | .BYTE | 377,001,001,001,001,001,001,001 |      |
| 4959 | 021062 | 001 | 001 | 001 | .BYTE | 001,001,001,001,001,001,001,377 |      |
| 4960 | 021072 | 377 | 200 | 200 | .BYTE | 377,200,200,200,200,200,200,200 |      |
| 4961 | 021102 | 000 | 000 | 000 | .BYTE | 000,000,000,000,000,000,000,200 |      |
| 4962 | 021112 | 200 | 000 | 000 | .BYTE | 200,000,000,000,000,000,000,000 |      |
| 4963 | 021122 | 001 | 000 | 000 | .BYTE | 001,000,000,000,000,000,000,000 |      |
| 4964 | 021132 | 000 | 000 | 000 | .BYTE | 000,000,000,000,000,000,000,001 | :120 |
| 4965 | 021142 | 000 | 010 | 024 | .BYTE | 000,010,024,042,343,000,000,000 |      |
| 4966 | 021152 | 010 | 010 | 030 | .BYTE | 010,010,030,040,100,040,030,010 |      |
| 4967 | 021162 | 000 | 074 | 146 | .BYTE | 000,074,146,303,201,000,000,000 |      |
| 4968 | 021172 | 030 | 060 | 140 | .BYTE | 030,060,140,100,100,140,060,030 |      |
| 4969 | 021202 | 000 | 000 | 000 | .BYTE | 000,000,000,000,017,010,010,010 |      |
| 4970 | 021212 | 010 | 010 | 010 | .BYTE | 010,010,010,010,370,000,000,000 |      |
| 4971 | 021222 | 010 | 010 | 010 | .BYTE | 010,010,010,010,017,000,000,000 |      |
| 4972 | 021232 | 000 | 000 | 000 | .BYTE | 000,000,000,000,370,010,010,010 |      |



4974  
 4975  
 4976  
 4977

.SBTTL T.V.-ENCODED CHARACTER SET.  
 : THIS CHARACTER SET IS SPECIALLY SELECTED TO REDUCE THE  
 : BANDWIDTH REQUIREMENTS FOR PAL ENCODING PURPOSES.

|      |        |     |     |     |
|------|--------|-----|-----|-----|
| 4978 | 021242 | 360 | 360 | 360 |
| 4979 | 021252 | 030 | 030 | 030 |
| 4980 | 021262 | 030 | 030 | 030 |
| 4981 | 021272 | 074 | 074 | 074 |
| 4982 | 021302 | 000 | 000 | 000 |
| 4983 | 021312 | 000 | 000 | 000 |
| 4984 | 021322 | 000 | 000 | 377 |
| 4985 | 021332 | 030 | 030 | 030 |
| 4986 | 021342 | 030 | 030 | 030 |
| 4987 | 021352 | 074 | 074 | 377 |
| 4988 | 021362 | 003 | 006 | 014 |
| 4989 | 021372 | 200 | 300 | 140 |
| 4990 | 021402 | 000 | 034 | 062 |
| 4991 | 021412 | 203 | 306 | 154 |
| 4992 | 021422 | 074 | 176 | 377 |
| 4993 | 021432 | 030 | 074 | 176 |
| 4994 | 021442 | 030 | 030 | 074 |
| 4995 | 021452 | 030 | 030 | 074 |
| 4996 | 021462 | 146 | 146 | 146 |
| 4997 | 021472 | 314 | 314 | 314 |
| 4998 | 021502 | 377 | 000 | 377 |
| 4999 | 021512 | 377 | 377 | 000 |
| 5000 | 021522 | 377 | 303 | 307 |
| 5001 | 021532 | 377 | 303 | 303 |
| 5002 | 021542 | 377 | 376 | 374 |
| 5003 | 021552 | 377 | 377 | 177 |
| 5004 | 021562 | 200 | 300 | 340 |
| 5005 | 021572 | 003 | 007 | 017 |
| 5006 | 021602 | 003 | 007 | 016 |
| 5007 | 021612 | 300 | 340 | 160 |
| 5008 | 021622 | 303 | 347 | 176 |
| 5009 | 021632 | 000 | 160 | 120 |
| 5010 | 021642 | 000 | 000 | 000 |
| 5011 | 021652 | 000 | 030 | 030 |
| 5012 | 021662 | 000 | 066 | 066 |
| 5013 | 021672 | 066 | 066 | 167 |
| 5014 | 021702 | 030 | 076 | 140 |
| 5015 | 021712 | 000 | 146 | 146 |
| 5016 | 021722 | 000 | 060 | 154 |
| 5017 | 021732 | 000 | 030 | 030 |
| 5018 | 021742 | 000 | 030 | 060 |
| 5019 | 021752 | 000 | 030 | 014 |
| 5020 | 021762 | 000 | 066 | 034 |
| 5021 | 021772 | 000 | 000 | 030 |
| 5022 | 022002 | 000 | 000 | 000 |
| 5023 | 022012 | 000 | 000 | 000 |
| 5024 | 022022 | 000 | 000 | 000 |
| 5025 | 022032 | 000 | 006 | 006 |
| 5026 | 022042 | 000 | 074 | 146 |
| 5027 | 022052 | 000 | 030 | 070 |
| 5028 | 022062 | 000 | 074 | 146 |
| 5029 | 022072 | 000 | 074 | 146 |

PALCH: .BYTE 360,360,360,360,360,360,360,360  
 .BYTE 030,030,030,030,030,030,030,030  
 .BYTE 030,030,030,030,030,030,030,030  
 .BYTE 074,074,074,074,074,074,074,074  
 .BYTE 000,000,000,000,377,000,000,000  
 .BYTE 000,000,000,000,377,377,000,000  
 .BYTE 000,000,377,377,377,377,000,000  
 .BYTE 030,030,030,030,377,030,030,030  
 .BYTE 030,030,030,377,377,030,030,030  
 .BYTE 074,074,377,377,377,377,074,074  
 .BYTE 003,006,014,030,060,140,300,200  
 .BYTE 200,300,140,060,030,014,006,003  
 .BYTE 000,034,062,060,170,060,140,176  
 .BYTE 203,306,154,070,070,154,306,203  
 .BYTE 074,176,377,377,377,377,176,074  
 .BYTE 030,074,176,377,377,176,074,030  
 .BYTE 030,030,074,146,303,146,074,030  
 .BYTE 030,030,074,176,377,176,074,030  
 .BYTE 146,146,146,146,146,146,146,146  
 .BYTE 314,314,314,314,314,314,314,314  
 .BYTE 377,000,377,000,377,000,377,000  
 .BYTE 377,377,000,000,377,377,000,000  
 .BYTE 377,303,307,317,333,363,343,377  
 .BYTE 377,303,303,303,303,303,303,377  
 .BYTE 377,376,374,370,360,340,300,200  
 .BYTE 377,377,177,077,037,017,007,003  
 .BYTE 200,300,340,360,370,374,376,377  
 .BYTE 003,007,017,037,077,177,377,377  
 .BYTE 003,007,016,034,070,160,340,300  
 .BYTE 300,340,160,070,034,016,007,003  
 .BYTE 303,347,176,074,074,176,347,303  
 .BYTE 000,160,120,160,000,000,000,000  
 .BYTE 000,000,000,000,000,000,000,000  
 .BYTE 000,030,030,030,030,030,000,030  
 .BYTE 000,066,066,000,000,000,000,000  
 .BYTE 066,066,167,000,167,066,066,000  
 .BYTE 030,076,140,074,006,076,030,000  
 .BYTE 000,146,146,014,030,060,146,146  
 .BYTE 000,060,154,154,060,156,154,076  
 .BYTE 000,030,030,030,000,000,000,000  
 .BYTE 000,030,060,140,140,140,060,030  
 .BYTE 000,030,014,006,006,006,014,030  
 .BYTE 000,066,034,076,034,066,000,000  
 .BYTE 000,000,030,030,076,030,030,000  
 .BYTE 000,000,000,000,000,030,030,060  
 .BYTE 000,000,000,000,076,000,000,000  
 .BYTE 000,000,000,000,000,000,030,030  
 .BYTE 000,006,006,014,030,060,140,140  
 .BYTE 000,074,146,146,146,146,146,074  
 .BYTE 000,030,070,030,030,030,030,074  
 .BYTE 000,074,146,006,036,060,140,176  
 .BYTE 000,074,146,006,030,006,146,074

: 9

: 20

: 40

: 50



|      |        |     |     |     |       |                                 |      |
|------|--------|-----|-----|-----|-------|---------------------------------|------|
| 5030 | 022102 | 000 | 014 | 034 | .BYTE | 000,014,034,074,154,176,014,014 |      |
| 5031 | 022112 | 000 | 176 | 140 | .BYTE | 000,176,140,174,006,006,146,074 |      |
| 5032 | 022122 | 000 | 034 | 060 | .BYTE | 000,034,060,140,174,146,146,074 |      |
| 5033 | 022132 | 000 | 176 | 006 | .BYTE | 000,176,006,014,030,060,060,060 |      |
| 5034 | 022142 | 000 | 074 | 146 | .BYTE | 000,074,146,146,030,146,146,074 |      |
| 5035 | 022152 | 000 | 074 | 146 | .BYTE | 000,074,146,146,036,006,014,070 |      |
| 5036 | 022162 | 000 | 000 | 030 | .BYTE | 000,000,030,030,000,030,030,000 |      |
| 5037 | 022172 | 000 | 000 | 030 | .BYTE | 000,000,030,030,000,030,030,060 |      |
| 5038 | 022202 | 000 | 014 | 030 | .BYTE | 000,014,030,060,140,060,030,014 | : 60 |
| 5039 | 022212 | 000 | 000 | 000 | .BYTE | 000,000,000,076,000,076,000,000 |      |
| 5040 | 022222 | 000 | 060 | 030 | .BYTE | 000,060,030,014,006,014,030,060 |      |
| 5041 | 022232 | 000 | 074 | 146 | .BYTE | 000,074,146,006,034,030,000,030 |      |
| 5042 | 022242 | 000 | 030 | 146 | .BYTE | 000,030,146,156,156,146,060,000 |      |
| 5043 | 022252 | 000 | 030 | 066 | .BYTE | 000,030,066,146,146,176,146,146 |      |
| 5044 | 022262 | 000 | 174 | 146 | .BYTE | 000,174,146,146,170,146,146,174 |      |
| 5045 | 022272 | 000 | 074 | 146 | .BYTE | 000,074,146,140,140,140,146,074 |      |
| 5046 | 022302 | 000 | 174 | 146 | .BYTE | 000,174,146,146,146,146,146,174 |      |
| 5047 | 022312 | 000 | 176 | 140 | .BYTE | 000,176,140,140,170,140,140,176 |      |
| 5048 | 022322 | 000 | 176 | 140 | .BYTE | 000,176,140,140,170,140,140,140 | : 70 |
| 5049 | 022332 | 000 | 076 | 140 | .BYTE | 000,076,140,140,156,146,146,076 |      |
| 5050 | 022342 | 000 | 146 | 146 | .BYTE | 000,146,146,146,176,146,146,146 |      |
| 5051 | 022352 | 000 | 074 | 030 | .BYTE | 000,074,030,030,030,030,030,074 |      |
| 5052 | 022362 | 000 | 006 | 006 | .BYTE | 000,006,006,006,006,006,146,074 |      |
| 5053 | 022372 | 000 | 146 | 154 | .BYTE | 000,146,154,170,160,170,154,146 |      |
| 5054 | 022402 | 000 | 140 | 140 | .BYTE | 000,140,140,140,140,140,140,176 |      |
| 5055 | 022412 | 000 | 146 | 176 | .BYTE | 000,146,176,176,146,146,146,146 |      |
| 5056 | 022422 | 000 | 146 | 146 | .BYTE | 000,146,146,166,156,146,146,146 |      |
| 5057 | 022432 | 000 | 074 | 146 | .BYTE | 000,074,146,146,146,146,146,074 |      |
| 5058 | 022442 | 000 | 174 | 146 | .BYTE | 000,174,146,146,174,140,140,140 |      |
| 5059 | 022452 | 000 | 074 | 146 | .BYTE | 000,074,146,146,146,156,156,166 |      |
| 5060 | 022462 | 000 | 174 | 146 | .BYTE | 000,174,146,146,174,170,154,146 |      |
| 5061 | 022472 | 000 | 076 | 140 | .BYTE | 000,076,140,140,074,006,006,174 |      |
| 5062 | 022502 | 000 | 176 | 030 | .BYTE | 000,176,030,030,030,030,030,030 |      |
| 5063 | 022512 | 000 | 146 | 146 | .BYTE | 000,146,146,146,146,146,146,074 |      |
| 5064 | 022522 | 000 | 146 | 146 | .BYTE | 000,146,146,146,146,074,074,030 |      |
| 5065 | 022532 | 000 | 146 | 146 | .BYTE | 000,146,146,146,146,176,176,146 |      |
| 5066 | 022542 | 000 | 146 | 146 | .BYTE | 000,146,146,074,030,074,146,146 |      |
| 5067 | 022552 | 000 | 146 | 146 | .BYTE | 000,146,146,074,030,030,030,030 |      |
| 5068 | 022562 | 000 | 176 | 006 | .BYTE | 000,176,006,014,030,060,140,176 |      |
| 5069 | 022572 | 000 | 170 | 140 | .BYTE | 000,170,140,140,140,140,140,170 |      |
| 5070 | 022602 | 000 | 140 | 140 | .BYTE | 000,140,140,060,030,014,006,006 |      |
| 5071 | 022612 | 000 | 036 | 006 | .BYTE | 000,036,006,006,006,006,006,036 |      |
| 5072 | 022622 | 030 | 074 | 176 | .BYTE | 030,074,176,377,030,030,030,030 |      |
| 5073 | 022632 | 030 | 034 | 036 | .BYTE | 030,034,036,377,377,036,034,030 |      |
| 5074 | 022642 | 030 | 030 | 030 | .BYTE | 030,030,030,030,377,176,074,030 |      |
| 5075 | 022652 | 030 | 070 | 170 | .BYTE | 030,070,170,377,377,170,070,030 |      |
| 5076 | 022662 | 000 | 377 | 377 | .BYTE | 000,377,377,377,377,377,377,377 |      |
| 5077 | 022672 | 000 | 000 | 377 | .BYTE | 000,000,377,377,377,377,377,377 |      |
| 5078 | 022702 | 000 | 000 | 000 | .BYTE | 000,000,000,377,377,377,377,377 |      |
| 5079 | 022712 | 000 | 000 | 000 | .BYTE | 000,000,000,000,377,377,377,377 |      |
| 5080 | 022722 | 000 | 000 | 000 | .BYTE | 000,000,000,000,000,377,377,377 |      |
| 5081 | 022732 | 000 | 000 | 000 | .BYTE | 000,000,000,000,000,000,377,377 |      |
| 5082 | 022742 | 000 | 000 | 000 | .BYTE | 000,000,000,000,000,000,000,377 |      |
| 5083 | 022752 | 374 | 374 | 374 | .BYTE | 374,374,374,374,374,374,374,374 |      |
| 5084 | 022762 | 374 | 374 | 374 | .BYTE | 374,374,374,374,374,374,374,374 |      |
| 5085 | 022772 | 370 | 370 | 370 | .BYTE | 370,370,370,370,370,370,370,370 |      |

|      |        |     |     |     |                                       |       |
|------|--------|-----|-----|-----|---------------------------------------|-------|
| 5086 | 023002 | 360 | 360 | 360 | .BYTE 360,360,360,360,360,360,360,360 |       |
| 5087 | 023012 | 340 | 340 | 340 | .BYTE 340,340,340,340,340,340,340,340 |       |
| 5088 | 023022 | 300 | 300 | 300 | .BYTE 300,300,300,300,300,300,300,300 | : 110 |
| 5089 | 023032 | 300 | 300 | 300 | .BYTE 300,300,300,300,300,300,300,300 |       |
| 5090 | 023042 | 300 | 300 | 300 | .BYTE 300,300,300,300,300,300,300,377 |       |
| 5091 | 023052 | 377 | 003 | 003 | .BYTE 377,003,003,003,003,003,003,003 |       |
| 5092 | 023062 | 003 | 003 | 003 | .BYTE 003,003,003,003,003,003,003,377 |       |
| 5093 | 023072 | 377 | 300 | 300 | .BYTE 377,300,300,300,300,300,300,300 |       |
| 5094 | 023102 | 000 | 000 | 000 | .BYTE 000,000,000,000,000,000,000,300 |       |
| 5095 | 023112 | 300 | 000 | 000 | .BYTE 300,000,000,000,000,000,000,000 |       |
| 5096 | 023122 | 003 | 000 | 000 | .BYTE 003,000,000,000,000,000,000,000 |       |
| 5097 | 023132 | 000 | 000 | 000 | .BYTE 000,000,000,000,000,000,000,003 |       |
| 5098 | 023142 | 000 | 030 | 074 | .BYTE 000,030,074,146,347,000,000,000 |       |
| 5099 | 023152 | 030 | 030 | 070 | .BYTE 030,030,070,060,140,060,070,030 |       |
| 5100 | 023162 | 000 | 074 | 146 | .BYTE 000,074,146,303,303,000,000,000 |       |
| 5101 | 023172 | 030 | 070 | 160 | .BYTE 030,070,160,140,140,160,070,030 |       |
| 5102 | 023202 | 000 | 000 | 000 | .BYTE 000,000,000,000,037,030,030,030 |       |
| 5103 | 023212 | 030 | 030 | 030 | .BYTE 030,030,030,030,370,000,000,000 |       |
| 5104 | 023222 | 030 | 030 | 030 | .BYTE 030,030,030,030,037,000,000,000 |       |
| 5105 | 023232 | 000 | 000 | 000 | .BYTE 000,000,000,000,370,030,030,030 |       |
| 5106 | 001000 |     |     |     | .END START                            |       |



|               |                |               |               |               |
|---------------|----------------|---------------|---------------|---------------|
| A = 020000    | DBUF0 007310   | FORGND 007464 | MONITX 013646 | S = 010000    |
| ADDRES 007572 | DBUF1 007314   | FRMSG 006550  | NODEFM 006762 | SAVPC 007524  |
| BAD 007564    | DBUF10 007360  | FSAVPW 007620 | NOHALT 017230 | SAVPC1 007526 |
| BAKND 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 016473 | ODAMSG 006707 | START1 001306 |
| BASE10 015352 | DBUF2 007320   | GENRO 016464  | PAL 007622    | START2 001326 |
| BASE11 007240 | DBUF20 007430  | GENR1 016454  | PALCH 021242  | START3 001362 |
| BAS10 015334  | DBUF3 007324   | GENSEL 016406 | PARITY 007546 | STATUS 007570 |
| BAS10A 015360 | DBUF4 007330   | GENO 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  | SYNUP 011644  |
| CALLPC 007600 | DECMG 007250   | GETCH3 014362 | PRCT1A 015046 | TABLE 001366  |
| CARX 007474   | DEVBUF 007440  | GETCH4 014426 | PRNT3 015116  | TABLE1 001412 |
| CARY 007476   | DEV CNT 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 |
| CSRO 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 | TRPLP 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 006415 | 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 | SENDAD 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)