.REM _

IDENTIFICATION

PRODUCT CODE: AC-E944B-MC

PRODUCT NAME: CXVSBB0 VSV01 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

"VSB" IS AN "IOMOD" THAT EXERCISES ONE VSV01 DISPLAY SYSTEM
INCLUDED IN THIS MODULE ARE LOGIC TESTS OF THE CHARACTER, SYNC AND
UP TO SIX BIT-MAP CONTROL LOGIC'S. A LOGIC ERROR IN THE SYNC OR CHARACTER
SECTIONS IS CONSIDERED A FATAL ERROR AND WILL RESULT IN THE MODULE
BEING "DROPPED". A LOGIC ERROR IN THE BIT MAP SECTION IS ALSO CONSIDERED
A "FATAL" ERROR. THE MAP IN ERROR WILL BE REMOVED FROM TESTING.
IF ALL SELECTED BIT-MAPS HAVE ERRORED, THE MODULE WILL BE DROPPED.

2. REQUIREMENTS

HARDWARE: VTV01 DISPLAY CONTROLLER WITH VRV01 MONITOR.

STORAGE:: VSB REQUIRES:
1. DECIMAL WORDS: 968
2. OCTAL WORDS: 1710
3. OCTAL BYTES: 3620

3. PASS DEFINITION

ONE PASS OF VSB MODULE CONSISTS OF ONE ITERATION OF THE THREE
CHARACTER SUB-PICTURES, AND TWO INTERATIONS OF THE FOUR BIT-MAP
SUB-PICTURES WHICH RESULTS IN:

12 THOUSAND PROGRAM INTERRUPTS

4. EXECUTION TIME

VSB RUNNING ALONE ON PDP-11/05 TAKES
APPROXIMATELY TWO MINUTES TO COMPLETE ONE PASS.

SEQ 0003

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 172600, VECTOR: 360, BR1: 4, DEVCNT: 1, SR1: 0

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

THE VTV01/VRV01 MUST HAVE THE POWER ON.

7. MODULE OPERATION

THE MODULE WILL BEGIN BY TESTING THE ABLILTY OF THE CHARACTER
CONTROL/STATUS REGISTER TO FUNCTION PROPERLY.
IF ANY ERRORS ARE DETECTED, THE MODULE WILL BE DROPPED.

TTHE MODULE WILL NOW DISPLAY THE CHARACTER SUB-TEST PATTERNS.
AFTER THE "CURSOR MOTION" SUB-TEST PATTERN. THE MODULE WILL NOW
CHECK THE VALUE OF "DVID1". IF 0, THE MODULE WILL REPORT
"END OF PASS". IF DVID1 IS NON-ZERO, THE MODULE WILL TEST EACH BIT MAP SELECTED
BY "DVID1". UPON COMPLETION OF ALL BIT MAPS, THE MODULE WILL REPORT "END OF PASS".
IF A BIT MAP SELECTED HAS A LOGIC ERROR, THAT BIT MAP WILL BE REMOVED
FROM TESTING FOR THAT PASS. IF ALL SELECTED HAVE ERRORS, THE MODULE
WILL BE DROPPED.

9.0 TEST PATTERN DESCRIPTION (CHARACTER GENERATOR AND SYNC TEST)

DYNAMIC X CROSS-HAIR POSITION

THE PATTERN CONSISTS OF A SINGLE VERTICAL LINE EXTENDING FROM TOP TO THE BOTTOM OF THE SCREEN. THE X POSITION OF THE LINE BEGINS AT THE LEFT EDGE AND MOVES TOWARD THE RIGHT EDGE OF THE SCREEN. THE MOVEMENT SHOULD BE SMOOTH WITH NO JUMP IN POSITION.

DYNAMIC Y CROSS-HAIR POSITION

THE PATTERN CONSISTS OF A SINGLE HORIZONTAL LINE EXTENDING FROM LEFT TO RIGHT EDGE OF THE SCREEN. THE Y POSITION OF THE LINE BEGINS AT THE TOP AND MOVES TO THE BOTTOM OF THE SCREEN. THE MOVEMENT SHOULD BE SMOOTH WITH NO JUMP IN POSITION.

ROTATING CHARACTER SET

THIS TEST SHOWS THAT ALL CHARACTERS CAN BE INTERMIXED WITHOUT PROBLEMS. THE PATTERN CONSISTS OF A FULL LINE OF SEQUENTIAL CHARACTERS. THE NEXT LINE STARTS WITH THE NEXT STARTING CHARACTER, THIS SEQUENCE IS REPEATED UNTIL THE ENTIRE CHARACTER SET HAS BEEN DISPLAYED IN THE FIRST COLUMN.

SEQ 0005

EXPAND FUNCTION (BIT MAP TESTS)

THE PATTERN CONSISTS OF AN INTENSIFIED BIT MAP IN THE UPPER LEFT CORNER. AFTER A DELAY, THE "EXPAND" MODE STATUS BIT IS SET. THE RESULTING PICTURE SHOULD EXPAND TO COVER THE ENTIRE VERTICAL SCREEN AREA. THE HORIZONTAL SCREEN AREA SHOULD EXPAND BY THE SAME VERTICAL SIZE.

ORGIN FUNCTION

THE PATTERN CONSISTS OF AN INTENSIFIED BIT MAP IN THE UPPER LEFT CORNER. THE "ORGIN" REGISTER BITS ARE UPDATED AND THE BOX SHOULD MOVE TO THE RIGHT. AFTER FOUR POSITION CHANGES, THE MAP WILL NOW BE ORGINED TO THE NEXT LOWER LEVEL. THE PROCESS IS REPEATED UNTIL ALL FOUR HORIZINTAL AND FOUR VERTICAL ORGINS HAVE BEEN LOADED. ORGIN LOCATIONS ON THE FAR RIGHT AND ACROSS THE BOTTOM WILL BE CUT IN HALF.

DYNAMIC INTENSITY LEVEL (MONO MODE)

TESTS ALL LOCATIONS OF THE L.U.T. WITH ALL DIFFERENT NUMBERS.
TESTS ALL SHADES OF A FOUR BIT DIGITAL TO ANALOG CONVERTER.
THE PATTERN CONSISTS OF AN EXPANDED BIT MAP WITH SIXTEEN DIFFERENT VERTICAL INTENSITY LEVEL BANDS DISPLAYED. AFTER A DELAY, THE L.U.T. ADDRESSES ARE CHANGED AND THE RESULT IS THE INTENSITY BANDS MOVE TO THE RIGHT.

COMBINED COLOR TEST

THE PATTERN CONSISTS OF SIXTEEN DIFFERENT VERTICAL COLOR BANDS.
THE FIRST THREE ARE THE DIFFERENT GREEN COLOR LEVELS.
THE SECOND THREE ARE THE DIFFERENT RED COLOR LEVELS.
THE THIRD ARE THE DIFFERENT BLUE COLOR LEVELS.
THE REMAINING SEVEN ARE COMBINED COLORS AND INTENSITY LEVELS.

L.U.T.	LEVELS							
00 - 10	1 60	2 63	3 74	4 17	10 33	14 25	20 52	40 77

8. OPERATION OPTIONS

LOCATION "DVID1" CAN BE MODIFIED TO EXTEND THE NUMBER OF BIT-MAPS TESTED.

9. NON STANDARD PRINTOUTS

ALL PRINTOUTS HAVE STANDARD MEANINGS AS REPRESENTED IN
DEC/X11 DOCUMENTATION.

10. ENVIROMENT

#1
11/10 WITH 16K CF MEMORY
RK-11-D DISK CONTROLLER WITH 1 DRIVE
VSV01 DISPLAY SYSTEM

#2
11/45 WITH 24K CF MEMORY (16K CORE + 8K MOS)
KT-11-D MEMORY MANAGEMENT
RK-11-D DISK CONTROLLER WITH 1 DRIVE
VSV01 DISPLAY SYSTEM

#3
11/40 WITH 28K CF MEMORY
RK-11-D DISK CONTROLLER WITH 1 DRIVE.
VSV01 DISPLAY SYSTEM
KW11-L LINE CLOCK
TC-11 DECTAPE CONTROLLER WITH 1 DRIVE

213 - .LIST SEQ,BIN

```
                                        IOMOD   <VSBB.>172600,360,4,,,1,76
                                        MODULE  140000,VSBB,172600,360,4,,,1,76
                                        .TITLE  VSBB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION  6       23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 051526  041102      040 MODNAM: .ASCII  /VSBB / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   0PEN                ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 172600                  ADDR:   172600+0                    ;1ST DEVICE ADDR.
000010' 000360                  VECTOR: 360+0                       ;1ST DEVICE VECTOR.
000012'     200                 BR1:    .BYTE   PRIV4+0             ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRIV+0              ;2ND BR LEVEL.
000014' 000001                  CVID1:  +1                          ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                        ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                        ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                        ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                        ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 140000                  STAT:   140000                      ;STATUS WORD.
000030' 000310'                 INIT:   START                       ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP                       ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                           ;PASS COUNTER.
000036' 000001                  ICONT:  1                           ;# OF ITERATIONS PER PASS=1
000040' 000000                  ICOUNT: 0                           ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                           ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                           ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                           ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                           ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                           ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                           ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                             ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                           ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0                           ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                        ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                        ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                        ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                        ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                        ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                        ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                        ;LOC TO SAVE R6.
000100' 000000                  CSPA:   OPEN                        ;ADDR OF CURRENT CSR.
000102'                         SBADR:                              ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                        ;CONTENTS OF CSR.
000104'                         WASADR:                             ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                        ;STATUS REG CONTENTS.
000106'                         ERRTYP:                             ;TYPE OF ERROR
000106' 000000                  ASR:    OPEN                        ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                        ;ACTUAL DATA.
000112' 000310'                 RSTRT:  RESTRT                      ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                        ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                        ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                        ;# OF INTERRUPTS PER ITERATION
000122' 000076                  IDNUM:  76                          ;MODULE IDENTIFICATION NUMBER=76
                                        .REPT   SPSIZ               ;MODULE STACK STARTS HERE.
000040
                                        .NLIST
```



```
                                        .WORD   0
                                        .LIST
                                        .ENDR
000224'                         MODSP:
                                ;*****************************************************************************
000224' 000000                  STEMP:  OPEN
                                ;VTV01 BUS ADDRESSES
000226' 000000                  VCCSR:  OPEN                        ;DEVICE ADDRESSES
000230' 000000                  VCHRLC: 0
000232' 000000                  VCHRHI: 0
000234' 000000                  VCPOS:  0
000236' 000000                  VCPOSH: 0
000240' 000000                  VGCSR:  0
000242' 000000                  VGPC:   0
000244' 000000                  VGBUF:  0
000246' 000000                  VGCOLR: 0
                                ;VTV01 INTERRUPT VECTORS
000250' 000000                  VCINT:  OPEN                        ;INTR. VECTOR
000252' 000000                  VCINT1: OPEN
                                ;MISC TEMP LOCATIONS
000254' 000000                  DVISAV: OPEN
000256' 000000                  TEMP1:  0
000260' 000000                  TEMP2:  0
000262' 000000                  TEMP3:  0
000264' 000000                  TEMP0:  OPEN
000266' 000137                  LASTCH: 137                         ;LAST VALID CHARACTER CODE
000270' 000040                  STCHAR: 40
000272' 000400                  XHAIR:  256.
000274' 000400                  YHAIR:  256.
000276' 000100                  WIDTH:  64.                         ;CHAR SCREEN WIDTH
000300' 000031                  VHG:    25.                         ;VERTICAL LINE SIZE
000302' 003100                  TOTALC: 1600.
000304' 010000                  NUMPIX: 4096.
000306' 000000                  FATAL:  0
000310'                         RESTRT:
000310' 012767  012000  177602  START:  MOV     #12000,INTR         ;12000 INTERRUPTS/ITERATION
000316' 012767  000226  177572          MOV     #150.,WDFR          ;150 WORDS FROM MEM/ITERATION
000324' 012700  000226'                 MOV     #VCCSR,R0           ;LOAD ADDRESS POINTER
000330' 010761  177452                  MOV     ADDR,R1             ;LOAD ADDRESS VALUE
000334' 010120                  10$:    MOV     R1,(R0)+            ;LOAD THE ADDRESSES
000336' 027700  000250'                 CMP     #VCINT,R0           ;TEST FOR END
000342' 001374                          BNE     10$
000344' 062767  000002  177656          ADD     #2,VCHRLO           ;ADJUST THE ADDRESSES
000352' 062767  000003  177652          ADD     #3,VCHRHI
000360' 062767  000004  177646          ADD     #4,VCPOS
000366' 062767  000005  177642          ADD     #5,VCPOSH
```

```
000374' 062767  000020  177636          ADD     #20,VGCSR
000402' 062767  000022  177632          ADD     #22,VGPC
000410' 062767  000024  177626          ADD     #24,VGBUF
000416' 062767  000030  177622          ADD     #30,VGCOLR
000424' 016767  177360  177616          MOV     VECTOR,VCINT            ;LOAD BASIC INTR. VECTOR
000432' 016767  177612  177612          MOV     VCINT,VCINT1
000440' 062767  000002  177604          ADD     #2,VCINT1
000446' 016767  177342  177600          MOV     @VIDI,DVISAV
000454' 005067  177626                  CLR     FATAL                   ;CLEAR DROP MODULE INDICATOR
000460' 012767  177542  177412          MOV     VCCSR,CSRA              ;LOAD BUS ADDRESS IN CASE OF ERRORS
```



```
                                ;TEST THE CHAR CURSOR DISABLE BIT
000466' 012767  002000  177410  CURTST: MOV     #BIT10,ASTAT            ;LOAD EXPECTED VALUE
000474' 016777  177404  177524          MOV     ASTAT,@VCCSR            ;LOAD THE REGISTER
000502' 017767  177520  177372          MOV     @VCCSR,ACSR             ;READ THE REGISTER
000510' 042767  140000  177364          BIC     #140000,ACSR            ;MASK TO BITS 15-14
000516' 026767  177362  177356          CMP     ASTAT,ACSR              ;COMPARE FOR SAME
000524' 001410                          BEQ     VCROSS
000526' 012767  000025  177352          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
000534' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;'CHAR CURSOR DISABLE' FAILED TO SET
                                        ;*****************************************************************
000542' 005267  177540                  INC     FATAL                   ;SET DROP MODULE FLAG

                                ;TEST THE V CROSS HAIR ENABLE BIT
000546' 012767  004000  177330  VCROSS: MOV     #BIT11,ASTAT            ;LOAD EXPECTED VALUE
000554' 016777  177324  177444          MOV     ASTAT,@VCCSR            ;LOAD THE REGISTER
000562' 017767  177440  177312          MOV     @VCCSR,ACSR             ;READ THE REGISTER
000570' 042767  140000  177304          BIC     #140000,ACSR            ;MASK TO BITS 15-14
000576' 026767  177302  177276          CMP     ASTAT,ACSR              ;COMPARE FOR SAME
000604' 001410                          BEQ     XCROSS
000606' 012767  000025  177272          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
000614' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;'V CROSS HAIR ENABLE' FAILED TO SET
                                        ;*****************************************************************
000622' 005267  177460                  INC     FATAL                   ;SET DROP MODULE FLAG

                                ;TEST THE X CROSS HAIR ENABLE BIT
000626' 012767  010000  177250  XCROSS: MOV     #BIT12,ASTAT            ;LOAD EXPECTED VALUE
000634' 016777  177244  177364          MOV     ASTAT,@VCCSR            ;LOAD THE REGISTER
000642' 017767  177360  177232          MOV     @VCCSR,ACSR             ;READ THE REGISTER
000650' 042767  140000  177224          BIC     #140000,ACSR            ;MASK TO BITS 15-14
000656' 026767  177222  177216          CMP     ASTAT,ACSR              ;COMPARE FOR SAME
000664' 001410                          BEQ     CLRLOW
000666' 012767  000025  177212          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
000674' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;'X CROSS HAIR ENABLE' FAILED TO SET
                                        ;*****************************************************************
000702' 005267  177400                  INC     FATAL                   ;SET DROP MODULE FLAG

                                ;TEST THAT 'CLEAR LOW BYTE' DOES NOT CLEAR HIGH BYTE
000706' 012767  016000  177170  CLRLOW: MOV     #16000,ASTAT            ;LOAD EXPECTED READ/WRITE RESULT
000714' 016777  177164  177304          MOV     ASTAT,@VCCSR            ;LOAD THE REGISTER BITS
000722' 105077  177300                  CLRB    @VCCSR                  ;CLEAR LOW BYTE
000726' 017767  177274  177146          MOV     @VCCSR,ACSR             ;READ THE RESULT OF 'CLRB'
000734' 042767  140000  177140          BIC     #BIT15!BIT14,ACSR       ;MASK TO TOP TWO BITS
000742' 026767  177136  177132          CMP     ASTAT,ACSR              ;COMPARE FOR SAME
000750' 001410                          BEQ     FATAL0
000752' 012767  000025  177126          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
000760' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CLR LOW BYTE CHANGED THE HIGH BYTE OF THE CHAR STATUS R
                                        ;*****************************************************************
000766' 005267  177314                  INC     FATAL                   ;SET DROP MODULE FLAG


                                ;TEST IF A LOGICAL ERROR OCCURRED
000772' 005767  177310  FATAL0: TST     FATAL                   ;TEST IF AN ERROR OCCURRED ?
000776' 001402                          BEQ     DYNX                    ;BR IF NOT
```

```
   392  001000' 104410  000000'                ENDS,BEGIN              ;FATAL LOGIC ERROR DETECTED - DROP MODULE
```



```
   393                                          ;DYNAMIC TESTING OF THE X CROSS HAIR POSITION REGISTERS
   394
   395  001004' 012777  010000  177214  DYNX:   MOV     #BIT12,@VCCSR           ;ENABLE X CROSS HAIRS
   396  001012' 005067  177066                  CLR     ASTAT                   ;CLEAR EXPECTED VALUE
   397  001016' 005077  177206                  CLR     @VCHRLO                 ;CLEAR POS.
   398  001022' 116777  177056  177200  1$:     MOVB    ASTAT,@VCHRLO           ;LOAD THE LOW BYTE (X POSITION)
   399  001030' 004567  002104                  JSR     R5,DELAY
   400  001034' 000001                          1
   401  001036' 005267  177042                  INC     ASTAT                   ;UPDATE X POSITION
   402  001042' 026767  177036  177222          CMP     ASTAT,XHAIR             ;TEST IF MORE LINES?
   403  001050' 001364                          BNE     1$
   404
   405                                          ;DYNAMIC TESTING OF THE Y CROSS HAIR POSITION REGISTERS
   406
   407  001052' 012777  004000  177146  DYNY:   MOV     #BIT11,@VCCSR           ;ENABLE Y CROSS HAIRS
   408  001060' 005067  177020                  CLR     ASTAT
   409  001064' 005077  177140                  CLR     @VCHRLO                 ;CLEAR POSITION
   410  001070' 116777  177010  177134  1$:     MOVB    ASTAT,@VCHRHI
   411  001076' 004567  002034                  JSR     R5,DELAY
   412  001102' 000001                          1
   413  001104' 105267  176774                  INCB    ASTAT
   414  001110' 126767  177160  176766          CMPB    YHAIR,ASTAT
   415  001116' 001364                          BNE     1$
   416  001120' 005077  177102                  CLR     @VCCSR                  ;DISABLE CROSS HAIRS
   417
   418                                          ;ROTATING CHARACTERS ACROSS ALL COLS. <ALL CHARACTERS>
   419  001124'                                 ROTCHR:
   420  001124' 004767  001440                  JSR     PC,HOME
   421  001130' 012767  000040  177132          MOV     #40,STCHAR              ;SET-UP STARTING CHARACTER
   422
   423  001136' 004767  001426                  JSR     PC,HOME
   424  001142' 016767  177132  177114          MOV     VHO,TEMPO               ;LOAD TEMP
   425  001150' 016701  177114          1$:     MOV     STCHAR,R1               ;LOAD R1=TO CHARACTER
   426  001154' 004567  001514                  JSR     R5,LIC                  ;LOAD A BUFFER STARTING WITH
   427  001160' 000276'                         WIDTH                           ; THAT CHARACTER AND WIDTH <BYTE>
   428
   429  001162' 004767  001450                  JSR     PC,XPRNT                ;DISPLAY A FULL LINE FROM THE BUFFER
   430
   431  001166' 005367  177072                  DEC     TEMPO                   ;DONE ?
   432  001172' 001010                          BNE     2$                      ;BR IF YES
   433  001174' 004567  001740                  JSR     R5,DELAY
   434  001200' 000144                          100.
   435  001202' 004767  001362                  JSR     PC,HOME
   436  001206' 016767  177066  177050          MOV     VHO,TEMPO
   437  001214' 005267  177050          2$:     INC     STCHAR                  ;UPDATE THE STARTING CHARACTER
   438  001220' 026767  177042  177042          CMP     LASTCH,STCHAR           ;TEST FOR FINAL CHARACTER
   439  001226' 001350                          BNE     1$                      ;BRANCH IF NOT COMPLETED
   440
   441  001230' 004567  001704                  JSR     R5,DELAY
   442  001234' 000144                          100.
   443  001236' 004767  001326                  JSR     PC,HOME
```

```
001242' 016767  176540  177006  NEXTON: MOV     ADDR,TEMP1              ;GET BASE ADDRESS
001250' 062767  000020  177000          ADD     #20,TEMP1               ;UPDATE TO THE MAP ADDRESS
001256' 042767  177700  176770          BIC     #177700,DVISAV          ;MASK TO LOWER SIX BITS
001264' 001441                          BEQ     MAPRTA                  ;BR IF NO MAPS SELECTED
001266' 005067  176766                  CLR     TEMP2                   ;CLEAR MAP INDICATOR
001272' 012767  000001  176762          MOV     #BIT0,TEMP3             ;LOAD TESTING BIT
001300' 036767  176756  176746  MAPRTB: BIT     TEMP3,DVISAV            ;TEST IF THIS BIT MAP IS SELECTED
001306' 001402                          BEQ     MAPRET                  ;BR IF NOT
001310' 000167  000064                  JMP     TSTMAP                  ;SELECTED - TEST THIS MAP
001314' 005767  176766          MAPRET: TST     FATAL                   ;TEST FOR FATAL ERROR
001320' 001403                          BEQ     1$                      ;BR IF NONE
001322' 046767  176734  176724          BIC     TEMP3,DVISAV            ;DROP THAT BIT MAP
001330' 062767  000020  176720  1$:     ADD     #20,TEMP1               ;UPDATE TO NEXT BUS ADDRESS
001336' 005267  176716                  INC     TEMP2                   ;UPDATE MAP INDICATOR
001342' 006367  176714                  ASL     TEMP3                   ;MOVE THE TEST BIT
001346' 022767  000100  176706          CMP     #100,TEMP3              ;TEST IF ANY MORE POSSIBLE MAPS
001354' 001351                          BNE     MAPRTB                  ;BR IF MORE
001356' 005767  176672                  TST     DVISAV                  ;TEST IF ANY MAPS LEFT ?
001362' 001002                          BNE     MAPPTA                  ;BR IF SOME
001364' 104410  000000'                 ENDS,BEGIN              ;DROP THE MODULE BECAUSE ALL MAPS HAVE ERRORED
001370'                         MAPRTA:
001370' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
001374' 000167  176710                  JMP     START
```



```
                                ;NOW LOAD THE BUS ADDRESSES FROM "TEMP1" AND START TESTING
001400' 016767  176652  176632  TSTMAP: MOV     TEMP1,VGCSR             ;LOAD BUS ADDRESSES
001406' 016767  176644  176626          MOV     TEMP1,VGPC
001414' 016767  176636  176622          MOV     TEMP1,VGBUF
001422' 016767  176630  176616          MOV     TEMP1,VGCOLR
001430' 062767  000002  176604          ADD     #2,VGPC
001436' 062767  000004  176600          ADD     #4,VGBUF
001444' 062767  000010  176574          ADD     #10,VGCOLR
001452' 016767  176562  176420          MOV     VGCSR,CSRA              ;LOAD BUS ADDRESS INCASE OF ERROR
001460' 005067  176622                  CLR     FATAL                   ;CLEAR DROP MODULE FLAG

                                ;TEST THAT "EXPAND" BIT CAN BE SET
001464' 012767  004000  176412  TSTEXP: MOV     #BIT11,ASTAT            ;LOAD THE EXPECTED
001472' 016777  176406  176540          MOV     ASTAT,@VGCSR            ;LOAD THE REGISTER
001500' 017767  176534  176374          MOV     @VGCSR,ACSR             ;READ THE REGISTER
001506' 042767  171360  176366          BIC     #171360,ACSR            ;MASK TO BITS
001514' 026767  176364  176360          CMP     ASTAT,ACSR              ;COMPARE THE EXPECTED TO RECVD
001522' 001410                          BEQ     TSTMON
001524' 012767  000025  176354          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
001532' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"EXPAND" BIT FAILED TO SET
                                        ;*****************************************************************
001540' 005267  176542                  INC     FATAL                   ;SET DROP MODULE FLAG

                                ;TEST THAT "MONO" BIT CAN BE SET
001544' 012767  002000  176332  TSTMON: MOV     #BIT10,ASTAT            ;LOAD THE EXPECTED
001552' 016777  176326  176460          MOV     ASTAT,@VGCSR            ;LOAD THE REGISTER
001560' 017767  176454  176314          MOV     @VGCSR,ACSR             ;READ THE REGISTER
001566' 042767  171360  176306          BIC     #171360,ACSR            ;MASK TO BITS
001574' 026767  176304  176300          CMP     ASTAT,ACSR              ;COMPARE THE EXPECTED TO RECVD
001602' 001410                          BEQ     MAPTST
001604' 012767  000025  176274          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
001612' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"MONO" BIT FAILED TO SET
                                        ;*****************************************************************
001620' 005267  176462                  INC     FATAL                   ;SET DROP MODULE FLAG

                                ;TEST THAT "MAP ENABLE" BIT CAN BE SET
001624' 012767  000400  176252  MAPTST: MOV     #BIT8,ASTAT             ;LOAD THE EXPECTED
001632' 016777  176246  176400          MOV     ASTAT,@VGCSR            ;LOAD THE REGISTER
001640' 017767  176374  176234          MOV     @VGCSR,ACSR             ;READ THE REGISTER
001646' 042767  171360  176226          BIC     #171360,ACSR            ;MASK TO BITS
001654' 026767  176224  176220          CMP     ASTAT,ACSR              ;COMPARE THE EXPECTED TO RECVD
001662' 001410                          BEQ     TSTORG
001664' 012767  000025  176214          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
001672' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;MAP ENABLE FAILED TO SET
                                        ;*****************************************************************
001700' 005267  176402                  INC     FATAL                   ;SET DROP MODULE FLAG
```

```
                                ;TEST THAT "ORGIN POINT" BITS CAN BE SET
001704' 012767  000002  176172  TSTORG: MOV     #BIT1,ASTAT             ;LOAD THE EXPECTED
001712' 013777  176166  176320  1$:     MOV     ASTAT,@VGCSR            ;LOAD THE REGISTER
001720' 017767  176314  176154          MOV     @VGCSR,ACSR             ;READ THE REGISTER
001726' 042767  171360  176146          BIC     #171360,ACSR            ;MASK TO BITS
001734' 026767  176144  176140          CMP     ASTAT,ACSR              ;COMPARE THE EXPECTED TO RECVD
001742' 001410                          BEQ     2$
001744' 012767  000025  176134          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
001752' 104405  000000' 000000          ERDES$,BEGIN,NULL       ;"ORGIN POINT" BIT FAILED TO SET
                                        ;*****************************************************************
001760' 005267  176322                  INC     FATAL                   ;SET DROP MODULE FLAG
001764' 005267  176114          2$:     INC     ASTAT                   ;UPDATE EXPECTED VALUE OF "ORGIN POINT"
001770' 022767  000020  176106          CMP     #20,ASTAT               ;TEST IF VALID VALUE FOR "ORGIN"
001776' 001345                          BNE     1$                      ;BR IF MORE TO TEST

                                ;TEST THAT CLEAR LOW BYTE DOES NOT CLEAR HIGH BYTE
002000' 012767  006400  176076  TSTLOW: MOV     #BIT11!BIT10!BIT8,ASTAT
002006' 012777  006417  176224          MOV     #BIT11!BIT10!BIT8+17,@VGCSR
002014' 105077  176220                  CLRB    @VGCSR                  ;CLEAR LOW BYTE
002020' 017767  176214  176054          MOV     @VGCSR,ACSR             ;READ THE REGISTER
002026' 042767  170360  176046          BIC     #170360,ACSR            ;MASK TO BITS
002034' 026767  176044  176040          CMP     ASTAT,ACSR              ;TEST IF SAME
002042' 001410                          BEQ     TSTHGH
002044' 012767  000025  176034          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002052' 104405  000000' 000000          ERDES$,BEGIN,NULL       ;CLEARING LOW BYTE OF STATUS CHANGED THE HIGH BYTE
                                        ;*****************************************************************
002060' 005267  176222                  INC     FATAL                   ;SET DROP MODULE FLAG

                                ;TEST THAT CLEAR HIGH BYTE DOES NOT CLEAR LOW BYTE
002064' 012767  000017  176012  TSTHGH: MOV     #17,ASTAT               ;LOAD EXPECTED READ VALUE
002072' 012777  006417  176140          MOV     #BIT11!BIT10!BIT8+17,@VGCSR
002100' 016700  176134                  MOV     VGCSR,R0                ;LOAD STATUS REGISTER
002104' 005200                          INC     R0                      ;MAKE ADDRESS
002106' 105010                          CLRB    (R0)                    ;OF HIGH BYTE
002110' 017767  176124  175764          MOV     @VGCSR,ACSR             ;CLEAR HIGH BYTE
002116' 042767  170360  175756          BIC     #170360,ACSR            ;READ REGISTER
002124' 026767  175754  175750          CMP     ASTAT,ACSR              ;MASK OUT BITS
002132' 001410                          BEQ     RDWRT                   ;COMPARE EXPECT TO RCVD
002134' 012767  000025  175744          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002142' 104405  000000' 000000          ERDES$,BEGIN,NULL       ;CLEARING THE HIGH BYTE OF STATUS CHANGED THE LOW BYTE
                                        ;*****************************************************************
002150' 005267  176132                  INC     FATAL                   ;SET DROP MODULE FLAG
```



```
                                ;READ-WRITE TEST OF THE LOW 3 BITS OF THE MAP P.C.
002154' 005067  175724  RDWRT:  CLR     ASTAT                   ;CLEAR EXPECTED READ VALUE
002160' 005067  176040          CLR     STEMP                   ;CLEAR WRITE VALUE
002164'                 1$:
002164' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002170' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002174' 013777  176024  176040          MOV     STEMP,@VGPC             ;LOAD BIT MAP P.C.
002202' 017767  176032  175672          MOV     @VGCSR,ACSR             ;READ STATUS (BIT MAP)
002210' 042767  177617  175664          BIC     #177617,ACSR            ;MASK TO UNUSED BITS
002216' 026767  175662  175656          CMP     ASTAT,ACSR              ;TEST IF SAME
002224' 001410                          BEQ     2$
002226' 012767  000025  175652          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
002234' 104405  000000' 000000          ERDES$,BEGIN,NULL       ;BIT MAP P.C. DID NOT LOAD CORRECTLY
                                        ;*****************************************************************
002242' 005267  176040                  INC     FATAL                   ;SET DROP MODULE FLAG
002246' 062767  000220  175630  2$:     ADD     #20,ASTAT               ;UPDATE EXPECTED READ VALUE
002254' 005267  175744                  INC     STEMP                   ;UPDATE EXPECTED WRITE VALUE
002260' 022767  000010  175736          CMP     #10,STEMP               ;TEST FOR FIRST NON-VALID
002266' 001336                          BNE     1$                      ;BRANCH IF STILL OK VALUES.

002270' 005767  176012                  TST     FATAL                   ;TEST IF AN ERROR OCCURRED
002274' 001402                          BEQ     VISEXP                  ;BR IF NOT
002276' 000167  177012                  JMP     MAPRET                  ;RETURN NOW - DONT BOTHER WITH DISPLAY STUFF
```

```
                                ;TEST THE "EXPAND" FUNCTION
002302' 004767  000444          VISEXP: JSR     PC,CLRMAP
002306' 012777  000017  175732          MOV     #17,@VGCOLR             ;LOAD MAX INTENSITY INTO LOCATION 0

002314' 012777  002400  175716          MOV     #BIT10!BIT8,@VGCSR      ;EXPAND OFF AND ENABLE BIT MAP
002322' 004567  000612                  JSR     R5,DELAY
002326' 000144                          100.

002330' 052777  004000  175702          BIS     #BIT11,@VGCSR           ;SET THE "EXPAND" BIT
002336' 004567  000576                  JSR     R5,DELAY
002342' 000144                          100.

                                ;TEST THE "ORGIN" FUNCTION

002344' 004767  000402          VISORG: JSR     PC,CLRMAP               ;CLEAR THE MAP

002350' 012777  000017  175670          MOV     #17,@VGCOLR             ;LOAD MAX INTENSITY INTO TABLE LOC. 0
002356' 012777  002400  175654          MOV     #BIT10!BIT8,@VGCSR      ;ENABLE THE BIT MAP
002364' 005067  175634                  CLR     STEMP

002370' 116777  175630  175642  1$:     MOVB    STEMP,@VGCSR            ;LOAD ORGIN REGISTER
002376' 004567  000536                  JSR     R5,DELAY
002402' 000144                          100.
002404' 105267  175614                  INCB    STEMP                   ;UPDATE THE ORGIN POSITION
002410' 122767  000020  175606          CMPB    #20,STEMP               ;TEST FOR A NON-VALID ORGIN
002416' 001364                          BNE     1$                      ;BR IF A VALID ORGIN

                                ;INTENSITY LEVEL TEST-MONOCROME

002420' 004767  000326          VISINT: JSR     PC,CLRMAP               ;CLEAR THE BIT MAP
002424' 004767  000644                  JSR     PC,LODSEQ               ;LOAD TWO WORDS PER LOOK TABLE LOCATION
002430' 005067  000012                  CLR     2$                      ;CLEAR STARTING INTENSITY LEVEL
002434' 012777  006400  175576          MOV     #BIT11!BIT10!BIT8,@VGCSR        ;SET MONOCROME + ENABLE MAP
002442' 004567  000734          1$:     JSR     R5,LODSTA               ;LOAD SEQUENCE
002446' 000000                  2$:     0                               ;INDEX INTO DATA VALVE TABLE
002450' 003060'                         TABLE0

002452' 004567  000462                  JSR     R5,DELAY
002456' 000144                          100.
002460' 062767  000001  177760          ADD     #1,2$                   ;UPDATE INDEX INTO TABLE FOR LOCATION 0'S VALUE
002466' 022767  000020  177752          CMP     #20,2$                  ;TEST IF LAST VLUE FOR MONOCROME
002474' 001362                          BNE     1$                      ;BR IF MORE LEVELS FOR LOCATION 0
```



```
                                ;COLOR LEVEL TEST - COLORCROME
002476' 004767  000250          VISCLT: JSR     PC,CLRMAP               ;ENSURE CLEAR MAP
002502' 004767  000566                  JSR     PC,LODSEQ               ;LOAD TWO WORDS PER LOOKUP TABLE LOC.
002506' 005067  000010                  CLR     2$                      ;CLEAR INDEX POINTER
002512' 005077  175522                  CLR     @VGCSR                  ;CLEAR STATUS
002516' 004567  000660          1$:     JSR     R5,LODSTA               ;LOAD DATA INTO TABLE
002522' 000000                  2$:     0
002524' 003100'                         TABLER
002526' 012777  004400  175504          MOV     #BIT11!BIT8,@VGCSR      ;ENABLE MAP
002534' 004567  000400                  JSR     R5,DELAY
002540' 000144                          100.
002542' 062767  000001  177752          ADD     #1,2$                   ;UPDATE LOC 0 INDEX INTO TABLE
002550' 022767  000020  177744          CMP     #20,2$                  ;TEST IF FINISHED FOR LOC 0
002556' 001357                          BNE     1$                      ;
002560' 005077  175454                  CLR     @VGCSR                  ;DISABLE MAP


002564' 000167  176524                  JMP     MAPRET
```

```
                        ;
002570' 004567  000344          HOME:   JSR     R5,DELAY
002574' 000001                  1
002576' 012767  000035  000672          MOV     #35,BUFFER              ;LOAD HOME CHAR
002604' 112767  177777  000665          MOVB    #-1,BUFFER+1            ;LOAD TERMINATOR
002612' 004567  000020                  JSR     PC,XPRNT
002616'                         1$:
002616' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002622' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002626' 005777  175374                  TST     @VCCSR                  ;WAIT FOR READY
002632' 100371                          BPL     1$                      ;WAIT
002634' 000207                          RTS     PC
                        ;
                        ;
                                        .SBTTL  DISPLAY BUFFER ON VTV01 SCREEN SUB-ROUTINE
                        ;
002636' 012700  003476'         XPRNT:  MOV     #BUFFER,R0              ;LOAD BUFFER POINTER
002642'                         1$:
002642' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002646' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002652' 005777  175350                  TST     @VCCSR                  ;TEST IF READY IS SET
002656' 100371                          BPL     1$                      ;WAIT TILL DONE
002660' 112077  175342                  MOVB    (R0)+,@VCCSR            ;LOAD THE CHARACTER
002664' 122710  000377                  CMPB    #377,(R0)               ;TEST FOR TERM
002670' 001364                          BNE     1$                      ;BR IF NOT TERM.
002672' 000207                          RTS     PC                      ;EXIT
                        ;
                        ;
                        ;LOAD A INCREMENTING CHARACTER ACROSS THE SCREEN WIDTH
                        ;ONLY 40 THRU 177 ARE LEGAL CHARACTERS
                        ;
002674' 012700  003476'         LIC:    MOV     #BUFFER,R0      ;SET-UP BUFFER POINTER
002700  013502                  MOV     @(5)+,R2        ;SET-UP WIDTH
002702  110126                  1$:     MOVB    R1,(0)+         ;SAVE A CHARACTER IN THE BUFFER
002704  005201                  INC     R1              ;UPDATE THE CHARACTER
002706  026701  175354          CMP     LASTCH,R1       ;TEST FOR
002712  001002                  BNE     2$              ;BRANCH IF NOT
002714  012701  000040          MOV     #40,R1          ;MAKE A LEGAL CHARACTER
002720                          2$:
002720  104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002724  104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002730  005302                  DEC     R2              ;DECREMENT COUNT
002732  001363                  BNE     1$              ;BRANCH IF NOT COMPLETED
002734  112720  000015          MOVB    #15,(R0)+       ;LOAD "CR"
002740  112720  000012          MOVB    #12,(R0)+       ;LOAD "LF"
002744  112710  000377          MOVB    #377,(R0)       ;LOAD TERM
002750  000205                  RTS     R5              ;EXIT
```



```
                        ;SUBROUTINE TO CLEAR THE BIT MAP
                        ;
002752' 012767  000100  000074  CLRMAP: MOV     #BIT6,10$               ;LOAD BASE DELAY
002760' 005067  000072                  CLR     11$                     ;CLEAR COUNTER
002764' 004567  000150                  JSR     R5,DELAY
002770'                         1
002770' 012777  001000  175240          MOV     #BIT9,@VGCSR            ;SET "CLEAR MAP" BIT
003000                          1$:
003000' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003004' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003010' 105777  175224                  TSTB    @VGCSR                  ;TEST READY FLAG
003014' 100416                          BMI     2$                      ;BR IF READY SET
003015' 005267  000034                  INC     11$                     ;UPDATE COUNTER
003022' 001366                          BNE     1$                      ;BR IF NO OVERFLOW
003024' 005367  000024                  DEC     10$                     ;DECREMENT BASE DELAY
003030' 001363                          BNE     1$                      ;BR IF NO OVERFLOW
003032' 012767  000011  175046          MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                        ;*****************************************************************
003040' 104405  000000' 000000          ERDES$,BEGIN,NULL       ;READY FAILED TO SET AFTER A GROSS
                        ;*****************************************************************
003046' 005267  175234                  INC     FATAL                   ;SET DROP MODULE FLAG
                                                                        ;TIME DELAY
003052' 000207                  2$:     RTS     PC                      ;EXIT
                        ;
003054' 000000                  10$:    0                       ;BASE DELAY
003056' 000000                  11$:    0                       ;COUNT LOCATION
                        ;
                        ;
                                        .SBTTL  TABLE OF VALUES THAT WILL BE LOADED INTO THE COLOR TABLE (L.U.T.)
                        ;
003060'    000     001     002  TABLE0: .BYTE   0,1,2,3,4,5,6,7,10,11,12,13,14,15,16,17
003063'    003     004     005
003066'    006     007     010
003071'    011     012     013
003074'    014     015     016
003077'    017
003100'    001     002     003  TABLE0: .BYTE   1,2,3,4,10,14,20,40,60,63,74,17,33,25,52,77
003103'    004     010     014
003106'    020     040     060
003111'    063     074     017
003114'    033     025     052
003117'    077
003120'    001     002     003          .BYTE   1,2,3,4,10,14,20,40,60,63,74,17,33,25,52,77
003123'    004     010     014
003126'    020     040     060
003131'    063     074     017
003134'    033     025     052
003137'    077
                        ;
                        ;
                                        .EVEN
```

```
                                        .SBTTL  DELAY SUBROUTINE -- WAIT FOR "VERTICAL SYNC INTERRUPT"
003140' 012567  000064'         DELAY:  MOV     (R5)+,10$               ;LOAD THE ARG.
003144' 012777  003170' 175076  4$:     MOV     #3$,@VCINT              ;LOAD INTR. RETURN VECTOR
003152' 005077  175074          CLR     @VCINT1
003156' 052777  020000  175042  6$:     BIS     #BIT13,@VCCSR           ;ENABLE INTR.
003164' 104400  000000'         EXIT$,BEGIN                             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003170' 042777  020000  175030  3$:     BIC     #BIT13,@VCCSR   ;DISABLE INTR.
                                        ;-------------------------------------------------------------------
003176' 000004  000000' 003204'         PIRQ$,BEGIN,5$          ; QUEUE UP TO CONTINUE AT 5$ AND RTI
                                        ;-------------------------------------------------------------------
003204' 005367  000020          5$:     DEC     10$                     ;FINISHED DELAY ?
003210' 001362                  BNE     6$                      ;BR IF NOT
003212' 042777  020000  175006  2$:     BIC     #BIT13,@VCCSR           ;CLEAR INTR. ENABLE
003220' 016777  175026  175022          MOV     VCINT1,@VCINT           ;RESTORE INTR. VECTOR
003226' 000205                  RTS     R5                      ;EXIT

003230' 000000                  10$:    0

                                        ;RANDOM NUMBER GENERATOR

003232' 066767  000034  000030  $RAND:  ADD     $LONUM,$HINUM           ;ADJUST DATA
003240' 005567  000026          ADC     $LONUM
003244' 066767  000020  000020          ADD     $HINUM,$LONUM
003252' 005567  000012          ADC     $HINUM
003256' 104407  000000'         BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003262' 104407  000000'         BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003266' 000207                  RTS     PC                      ;EXIT

003270' 176543                  $HINUM: 176543
003272' 123456                  $LONUM: 123456
```



```
                                        ;LOAD MAP WITH TWO WORDS OF THE SAME VALUE
                                        ;       INTO THE ENTIRE BITMAP STARTING AT LOC. 0

003274' 010046                  LODSEG: MOV     R0,-(SP)                ;SAVE R0
003276' 010146                  MOV     R1,-(SP)                ;SAVE R1
003300' 005077  174736          CLR     @VGPC                   ;ENSURE CLEAR PC
003304' 016761  174774          MOV     NUMPIX,R1               ;LOAD COUNTER
003310' 006201                  ASR     R1                      ;ADJUST COUNT

003312' 005000                  1$:     CLR     R0                      ;START WITH PIXEL VALUE OF 0
003314'                         2$:
003314' 104407  000000'         BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003320' 104407  000000'         BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003324' 105777  174710          TSTB    @VGCSR                  ;ENSURE MAP IS READY
003330' 100371                  BPL     2$
003332' 010077  174706          MOV     R0,@VGBUF               ;LOAD A PIXEL WORD
003336'                         3$:
003336' 104407  000000'         BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003342' 104407  000000'         BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003346' 105777  174666          TSTB    @VGCSR                  ;ENSURE MAP IS READY
003352' 100371                  BPL     3$
003354' 010077  174664          MOV     R0,@VGBUF               ;LOAD 2ND PIXEL WORD

003360' 005301                  DEC     R1                      ;FINISHED ALL PIXELS ?
003362' 100404                  BMI     4$                      ;BR IF FINISHED
003364' 062700  010421          ADD     #10421,R0               ;UPDATE PIXEL DATA TO NEXT L.U.T. ADDRESS
003370' 103351                  BCC     2$                      ;BR IF MORE ADDRESSES TO LOAD
003372' 000747                  BR      1$                      ;BR TO RESET PIXEL DATA

003374' 012601                  4$:     MOV     (SP)+,R1                ;RESTORE R1
003376' 012600                  MOV     (SP)+,R0                ;RESTORE R0
003400' 000207                  RTS     PC                      ;EXIT
```

```
                        .SBTTL  SUBROUTINE TO LOAD THE L.U.T. WITH THE VALUE OF A TABLE

003402' 010046          LODSTA: MOV     R0,-(SP)                ;SAVE R0
003404' 012567  000056          MOV     (R5)+,10$               ;GET STARTING LUT ADDRESS
003410' 012500                  MOV     (R5)+,R0                ;GET TABLE ADDRESS
003412' 012767  000020  000050  MOV     #16.,11$                ;LOAD COUNTER

003420' 116767  000042  000045          MOVB    10$,12$+1               ;LOAD HIGH BYTE WITH LUT ADDRESS
003426'                 1$:
003426' 104407  000000'         BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003432' 104407  000000'         BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003436' 112067  000030          MOVB    (R0)+,12$               ;LOAD VALUE INTO LOW BYTE
003442' 016777  000024  174576  MOV     12$,@VGCOLR             ;LOAD THE TABLE
003450' 105267  000017          INCB    12$+1                   ;UPDATE THE LUT POINTER
003454' 005367  000010          DEC     11$                     ;FINISHED ALL L.U.T. ADDRESSES
003460' 001362                  BNE     1$                      ;BR IF NOT
003462' 012600                  MOV     (SP)+,R0                ;RESTORE R0
003464' 000205                  RTS     R5                      ;EXIT

003466' 000000          10$:    0
003470' 000020          11$:    16.                             ;COUNTER OF L.U.T.
003472' 000000          12$:    0                               ;TEMP LOCATION

003474' 000000          SET:    0

003476' 000000          BUFFER: OPEN
003500' 000050                  .BLKW   40.

000001                  .END
```



```
ACSR    000102R         257#  339*  340*  341   352*  353*  354   365*  366*  367   379*  380*  381
                        483*  484*  485   496*  497*  498   509*  510*  511   522*  523*  524   539*
                        540*  541   555*  556*  557   572*  573*  574
ADDR    000076R         223#  318   444
ADDR22= 001000          275#
ASR     000104R         261#
ASTAT   000104R         259#  337*  338   341   350*  351   354   363*  364   367   376*  377   381
                        396*  398   401*  402   408*  410   413*  414   481*  482   485   494*  495
                        498   507*  508   511   520*  521   524   531*  532   536*  541   550*  557
                        566*  574   581*
AWAS    000110R         262#
BEGIN   000000R         220#  345   358   371   385   392   463   465   489   502   515   528   545
                        561   569   570   578   657   658   667   668   688   689   705   706   715
                        753   756   773   774   791   792   797   798   822   823
BIT0    = 000001        275#  449
BIT1    = 000002        275#  520
BIT10   = 002000        275#  337   494   536   537   551   594   607   622
BIT11   = 004000        275#  350   407   481   536   537   551   598   622   640
BIT12   = 010000        275#  363   395
BIT13   = 020000        275#  752   754   760
BIT14   = 040000        275#  380
BIT15   = 100000        275#  380
BIT2    = 000004        275#
BIT3    = 000010        275#
BIT4    = 000020        275#
BIT5    = 000040        275#
BIT6    = 000100        275#  699
BIT7    = 000200        275#
BIT8    = 000400        275#  507   536   537   551   594   607   622   640
BIT9    = 001000        275#  703
BREAK$= 104407          275#  569   570   657   658   667   668   688   689   705   706   773   774
                        791   792   797   798   822   823
BR1     000012R         225#
BR2     000013R         226#
BTOD$ = 104421          275#
BUFFER  003476R         653*  654*  665   680   838#
CDATA$= 104412          275#
CLRLOW  000706R         368   376#
CLRMAP  002752R         591   604   619   633   699#
CONFIG  000056R         245#
CSRA    000100R         255#  335*  477*
CURTST  000466R         337#
DATCK$= 104411          275#
DATER$= 104404          275#
DELAY   003140R         399   411   433   441   595   599   611   627   641   651   701   749#
DVID1   000014R         227#  343
DVISAV  002754R         298#  343*  446*  450   455*  461
DVNX    001004R         391   395#
DVNY    001052R         407#
ENDIT$= 104413          275#  465
END$  = 104410          275#  392   463
ERRTYP  000106R         260#  343*  356*  369*  383*  487*  500*  513*  526*  543*  559*  576*  713*
EXIT$ = 104400          275#  753
FATAL   000306R         311#  334*  347*  360*  373*  387*  390   453   478*  491*  504*  517*  530*
                        547#  563*  580*  586   717*
FATAL0  000772R         382   390#
```

```
GETPA$= 104415            275#
GWBUF$= 104414            276#
HOME    002570R           420     423     435     443     651#
HRDCNT  000044R           240#
HRDER$= 104405            275#    345     358     371     385     489     502     515     528     545     561     578     715
HRDPAS  000050R           242#
ICONT   000036R           237#
ICOUNT  000040R           238#
IDNUM   000122R           267#
INIT    000030R           234#
INTP    000120R           266#    315*
LASTCH  000266R           303#    438     684
LIC     002674R           426     680#
LODSEQ  003274R           620     634     783#
LODSTA  003402R           623     637     815#
MAPRET  001314R           451     453#    588     649
MAPRTA  001370R           447     462     464#
MAPRTB  001300R           450#    460
MAPTST  001624R           499     507#
MAP22$= 104416            275#
MODNAM  000000R           221#
MODSP   000224R           235     273#
MSGN$ = 104463            275#
MSGS$ = 104462            275#
MSGS  = 104401            275#
NEXTCN  001242R           444#
NULL  = 000000            275#    345     358     371     385     489     502     515     528     545     561     578     715
NUMPIX  003264R           310#    786
OPEN  = 000000            222     228     229     230     231     248     249     250     251     252     253     254     255
                          257     259     261     262     264     265     266     275#    276     280     293     294     298
                          302     838
OTOA$ = 104420            275#
PASCNT  000034R           236#
PIRQS = 000004            275#    756
POPSP = 005726            275#
POPSP2= 022626            275#
PRTY  = 000000            225     275#
PRTY0 = 000000            275#
PRTY1 = 000040            275#
PRTY2 = 000100            275#
PRTY3 = 000140            275#
PRTY4 = 000200            225     275#
PRTY5 = 000240            275#
PRTY6 = 000300            275#
PRTY7 = 000340            275#
PS    = 177776            275#
PSW   = 177776            275#
PUSH  = 005746            275#
PUSH2 = 034646            275#
RANDS = 104417            275#
RANNUM  000054R           244#
RDWRT   002144R           559     566#
RESTRT  000310R           263     314#
RES1    000056R           246#
RES2    000060R           247#
ROTCHR  001124R           419#
```



```
RSTRT   000112R           263#
SBADR   001152R           256#
SET     003474R           836#
SOFCNT  000042R           239#
SOFER$= 104406            275#
SOFPAS  000046R           241#
SPOINT  000032R           235#
SPSIZ = 000040              1#    268
SR1     000016R           228#
SR2     000020R           229#
SR3     000022R           230#
SR4     000024R           231#
START   000310R           234     315#    467
STAT    000026R           233#
STCHAR  000270R           304#    421*    425     437*    438
SVR0    000062R           248#
SVR1    000064R           249#
SVR2    000066R           251#
SVR3    000070R           251#
SVR4    000072R           252#
SVR5    000074R           253#
SVR6    000076R           254#
SYSCNT  000052R           243#
TABLER  003100R           639     732#
TABLEW  003060R           625     726#
TEMP0   000254R           302     424*    431*    436*
TEMP1   000256R           299#    424*    445*    456*    470     471     472     473
TEMP2   000260R           300#    448*    457*
TEMP3   000262R           301#    449*    450     455     458*    459
TOTALC  003362R           309#
TRPDFD= 000022            275#
TSTEXP  001464R           481#
TSTHGH  002264R           542     550#
TSTLOW  002000R           536#
TSTMAP  001400R           452     470#
TSTMON  001544R           486     494#
TSTORG  001744R           512     520#
VCCSR   000226R           240#    317     335     338*    339     351*    352     364*    365     377*    378*    379     395*
                          407*    416*    659     669     671*    752*    754*    760*
VCHRHI  000232R           282#    323*    410*
VCHRLO  000230R           281#    322*    397*    398*    409*
VCINT   000250R           293#    326     330*    331     750*    761*
VCINT1  000252R           294#    331*    332*    751*    761
VCPOS   000234R           283#    324*
VCPOSH  000236R           284#    325*
VECTOR  000010R           224#    330
VGBUF   000024R           288#    328*    472*    475*    795*    801*
VGCOLR  000246R           285#    329*    473*    476*    592*    606*    825*
VGCSR   000240R           286#    326*    470*    477     482*    483     495*    496     508*    509     521*    522     537*
                          538*    539     551*    552     555     572     594*    598*    607*    610*    622*    636*    640*
                          646*    703*    707     793     799
VGPC    000242R           287#    327*    471*    474*    571*    785*
VH0     000300R           308#    424     436
VISCLT  002476R           633#
VISEXP  002302R           587     591#
VISINT  002420R           619#
```

SEQ 0027

```
VISORG  002344R           604#
WASADR  000104R           258#
WDFR    000116R           265#    316*
WDTO    000114R           264#
WIDTH   000276R           307#    427
XCROSS  000626R           355     363#
XFLAG   000005R           222#
XHAIR   000272R           305#    402
XPRNT   002636R           429     655     665#
YCROSS  000546R           342     350#
YHAIR   000274R           306#    414
$HINUM  003270R           769*    771     772*    777#
$LONUM  003272R           769     770*    771*    778#
$RAND   003232R           769#
$TEMP   000224R           276#    567*    571     582*    583     608*    610     613*    614
.    = 003620R            839#

. ABS.  000000      000
        003620      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XVSBB0,XVSBB0/SOL/CRF:SYM=DDXCOM,XVSBB0
 RUN-TIME:  1  2  .3  SECONDS
 RUN-TIME RATIO: 20/4=4.2
 CORE USED:  7K  (13 PAGES)
```