

VSBR DEC/X11 SYSTEM EXERCISER MODULE MACV11 30A(1052) 12-OCT-78 17:08 PAGE 2  
XVSBB0.P11 12-OCT-78 12:23

SEQ 0001

.REM -

IDENTIFICATION

PRODUCT CODE: AC-E944R-MC  
PRODUCT NAME: CXVSPRO VSV01 MODULE  
PRODUCT DATE: SEPTEMBER 1978  
MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1976,1978 DIGITAL EQUIPMENT CORPORATION

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 ERRCED, THE MODULE WILL BE DROPPED.

2. REQUIREMENTS

HARDWARE: VTV01 DISPLAY CONTROLLER WITH VRV01 MONITOR.

STORAGE:: VSP 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.

5. CONFIGURATION REQUIREMENTS

-----  
DEFAULT PARAMETERS:

DEVADR: 172600, VECTCF: 360, BR1: 4, DEVCONT: 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 ABILITY OF THE CHARACTER CONTROL/STATUS REGISTER TO FUNCTION PROPERLY.  
IF ANY ERRORS ARE DETECTED, THE MODULE WILL BE DROPPED.

THE 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.

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 HORIZONTAL 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 (MCNC MODE)

TESTS ALL LOCATIONS OF THE L.U.T. WITH ALL DIFFERENT NUMBERS. TESTS ALL SHADeS OF A FCUR PIT 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.

COMPIINED 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 COMPIINED COLORS AND INTENSITY LEVELS.

L.U.T. LEVELS

00	=	1	2	3	4	5	6	7	8	9
10	=	60	63	74	77	88	91	94	97	99

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. ENVIRONMENT

#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

VSBP DEC/X11 SYSTEM EXERCISER MODULE  
XVSBP0.P11 12-OCT-78 12:23

MACY11 30A(1052) 12-OCT-78 17:08 PAGE 8

SEQ 0007

```
214 000000*  
215 000000: 051526 041102 040 BEGIN: ;*****  
216 000005* 172600 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE  
217 000010* 000360 ADDR: 172600+0 ;1ST DEVICE ADDR  
218 000012* 000000 VECTOR: 36040 ;DEVICE VECTOR.  
219 000014* 000001 REG: .BYTE PFTV4+0 ;AND REG LEVEL.  
220 000016* 000000 FVID1: +1 ;DEVICE INDICATOR 1.  
221 000020* 000000 SR1: OPEN ;SWITCH REGISTER 1.  
222 000022* 000000 SR2: OPEN ;SWITCH REGISTER 2.  
223 000024* 000000 SR3: OPEN ;SWITCH REGISTER 3.  
224 000026* 000000 SR4: OPEN ;SWITCH REGISTER 4.  
225 000030* 140000 STAT: 140000 ;STATUS WORD.  
226 000031* 000310 INIT: START ;MODULE START ADDR.  
227 000034* 000000 SP0INT: MCDSP ;MODULE STACK POINTER.  
228 000036* 000001 PASCNT: 0 ;PASS COUNTER.  
229 000038* 000001 ICNT: 1 ;# OF ITERATIONS PER PASS=1  
230 000042* 000000 LOCNT: 0 ;LOC TO COUNT ITERATIONS  
231 000044* 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS  
232 000046* 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS  
233 000048* 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS  
234 000050* 000000 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS  
235 000052* 000000 SVSCNT: 0 ;# OF SYS ERRORS ACCUMULATED  
236 000054* 000000 RANDM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED  
237 000056* 000000 CONFIG: 0 ;RESERVED FOR MONITOR USE  
238 000060* 000000 REGS1: 0 ;RESERVED FOR MONITOR USE  
239 000062* 000000 SVR0: OPEN ;LOC TO SAVE R0.  
240 000064* 000000 SVR1: OPEN ;LOC TO SAVE R1.  
241 000066* 000000 SVR2: OPEN ;LOC TO SAVE R2.  
242 000068* 000000 SVR3: OPEN ;LOC TO SAVE R3.  
243 000070* 000000 SVR4: OPEN ;LOC TO SAVE R4.  
244 000072* 000000 SVR5: OPEN ;LOC TO SAVE R5.  
245 000076* 000000 CSPA1: OPEN ;ADDR OF CURRENT CSR.  
246 000100* 000000 SADR: OPEN ;ADDR OF GOOD DATA, OR  
247 000102* 000000 ACSR1: OPEN ;CONTENTS OF CSR.  
248 000124* 000000 WASADR: OPEN ;ADDR OF BAD DATA, OR  
249 000166* 000000 ASR1: OPEN ;STATUS REG CONTENTS.  
250 000167* 000000 ERPTYP: 0 ;TYPE OF ERROR  
251 000170* 000000 ASR: OPEN ;EXPECTED DATA.  
252 000172* 000000 AWAS: OPEN ;ACTUAL DATA.  
253 000112* 000310* RSTRT: RESTRT ;RESTART ADDRESS AFTER END OF PASS  
254 000114* 000000 WDTO: OPEN ;WORDS TO MEMORY PER ITERATION  
255 000116* 000000 WDFP: OPEN ;WORDS FROM MEMORY PER ITERATION  
256 000120* 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION  
257 000122* 000000 IDNUM: 76 ;MODULE IDENTIFICATION NUMBER=76  
258 000040* :DEPT SPSIZ ;MODULE STACK STARTS HERE.  
259 .NLIST
```

VSBP DEC/Y11 SYSTEM EXERCISER MODULE  
XVSBP0.P11 12-OCT-78 12:23

MACY11 30A(1052) 12-OCT-78 17:08 PAGE 9

SEQ 0008

```
270 .WORD 0  
271 .LIST  
272 .FNDR  
273 000224*  
274 ;*****  
275 000224* 000000  
276 STEMP: OPEN  
277 ;VTV01 BUS ADDRESSES  
278 VCCSR: OPEN ;DEVICE ADDRESSES  
279 VCHRLC: C  
280 VCHRRI: C  
281 VCPDSC: C  
282 VCPUSH: C  
283  
284 000246* 000000 VCCSR: 0  
285 000242* 000000 VGPC: C  
286 000244* 000000 VGBUF: C  
287 00024F* 000000 VCCOLR: C  
288  
289 ;VTV01 INTERRUPT VECTORS  
290 VCIINT: OPEN ;INTR. VECTOR  
291 VCIINT1: OPEN  
292  
293 ;MISC TEMP LOCATIONS  
294 000250* 000000 DVISAV: OPEN  
295 000256* 000000 TEMP1: 0  
296 000262* 000000 TEMP2: 0  
297 000268* 000000 TEMP3: 0  
298 00026E* 000137 TEMP4: OPEN ;LAST VALID CHARACTER CODE  
299 000254* 000000 TEMP1: 0  
300 000256* 000000 TEMP2: 0  
301 000262* 000000 TEMP3: 0  
302 000268* 000000 TEMP4: 0  
303 00026E* 000137 LSTCHAR: 137  
304 000270* 000040 XHAIR: 256.  
305 000272* 000040 VHAIR: 256.  
306 000274* 000400 WIDTH: 64.  
307 000276* 001100 VHOI: 256.  
308 000300* 000031 TOTLIC: 1600.  
309 000302* 000176 NUMPIX: 4096.  
310 000304* 000000 FATAL: 0  
311 000306* 000000  
312  
313 000310* RESTRRT:  
314 000310* 012767 012000 177602 START: MOV #12000,INTR ;12000 INTERRUPTS/ITERATION  
315 000310* 012767 012000 177572 MOV #150,WDPR ;150 WORDS FROM MEM/ITERATION  
316 000310* 012767 012000 177572 MOV #150,WDPR ;LOAD ADDRESS POINTER  
317 000310* 012767 012000 177572 ADD #150,WDPR ;LOAD ADDRESS VALUE  
318 000330* 012767 012000 177452 ADD #150,WDPR ;LOAD THE ADDRESSES  
319 000334* 012767 012000 177452 CMP #VCIINT,RO ;TEST FOR END  
320 000336* 022100 000250* 10S: BNE 10S  
321 000342* 001374 ADD #2,VCHRL0 ;ADJUST THE ADDRESSES  
322 000344* 022100 000002 177656 ADD #3,VCHRRI  
323 000346* 022100 000004 177646 ADD #4,VCPDSC  
324 000366* 062767 000005 177642 ADD #5,VCPUSH
```

VSSB DEC/Y11 SYSTEM EXERCISER MODULE MACV11 30A(1052) 12-OCT-78 17:08 PAGE 10  
XVSBB0.P11 12-OCT-78 12:23

SEQ 0009

```
325 000324* 062767 000023 177636      ADD    #20,VCCSR  
326 000410* 062767 000024 177626      ADD    #22,VGPC  
327 00041F* 062767 000030 177622      ADD    #24,VGBUF  
328 000435* 062767 177609 177619      ADD    #30,VGCLR  
329 000435* 062767 177609 177619      MOV    VECTOR,6CINT      ;LOAD BASIC INTR. VECTOR  
330 000440* 062767 000035 177604      MOV    VINT,6CINTI  
331 000446* 062767 177345 177600      ADD    #VINTI  
332 000446* 062767 177345 177600      MOV    DIVID1,DIVSAV  
333 000454* 062767 177642 177412      CLR    FATAL  
334 000454* 062767 177642 177412      MCV    VCCSR,CSRA      ;CLEAR DROP MODULE INDICATOR  
335 000460* 062767 177642 177412      MCV    VCCSR,CSRA      ;LOAD BUS ADDRESS IN CASE OF ERRORS
```

VSSB DEC/Y11 SYSTEM EXERCISER MODULE MACV11 30A(1052) 12-OCT-78 17:08 PAGE 11  
XVSBB0.P11 12-OCT-78 12:23

SEQ 0010

```
336 000546* 012767 002000 177410      ;TEST THE CHAR CURSOR DISABLE BIT  
337 000546* 012767 002000 177410      CURST: MOV    #BIT15,ASTAT      ;LOAD EXPECTED VALUE  
338 000547* 012767 177404 177524      MOV    ASTAT,6VCCSR      ;LOAD THE REGISTER  
339 000562* 012767 177520 177372      MOV    #VCCSR,ACSR      ;READ THE REGISTER  
340 000562* 012767 140000 177364      BIC    #140000,ACSP      ;MASK TO BITS 15-14  
341 000562* 012767 177362 177356      CMP    ASTAT,ACSR      ;COMPARE FOR SAME  
342 000562* 001410 000025 177352      BEQ    000562*  
343 000562* 012767 000025 177352      MOV    #25,FRTYP      ;BIT STUCK  
344 000534* 104405 000000* 000000      *****  
345 000534* 104405 000000* 000000      HRDERS,BEGIN,NULL      ;"CHAR CURSOR DISABLE" FAILED TO SET  
346 000542* 005267 177540      *****  
347 000542* 005267 177540      INC    FATAL      ;SET DROP MODULE FLAG  
348 000546* 012767 004000 177330      ;TEST THE Y CROSS HAIR ENABLE BIT  
349 000546* 012767 177334 177444      ACROSS: MOV    #BIT15,ASTAT      ;LOAD EXPECTED VALUE  
350 000546* 012767 177440 177312      MOV    ASTAT,6VCCSR      ;LOAD THE REGISTER  
351 000562* 012767 177312 177276      MOV    #VCCSR,ACSR      ;READ THE REGISTER  
352 000562* 012767 177276 177276      BIC    #140000,ACSP      ;MASK TO BITS 15-14  
353 000562* 001410 000025 177272      CMP    ASTAT,ACSP      ;COMPARE FOR SAME  
354 000562* 012767 000025 177272      BEQ    000562*  
355 000562* 012767 000025 177272      MOV    #25,FRTYP      ;BIT STUCK  
356 000614* 104405 000000* 000000      *****  
357 000614* 104405 000000* 000000      HRDERS,BEGIN,NULL      ;"Y CROSS HAIR ENABLE" FAILED TO SET  
358 000622* 005267 177460      *****  
359 000622* 005267 177460      INC    FATAL      ;SET DROP MODULE FLAG  
360 000626* 012767 010000 177250      ;TEST THE X CROSS HAIR ENABLE BIT  
361 000626* 012767 177254 177364      ACROSS: MOV    #BIT15,ASTAT      ;LOAD EXPECTED VALUE  
362 000642* 012767 177360 177232      MOV    ASTAT,6VCCSR      ;LOAD THE REGISTER  
363 000642* 012767 177232 177216      MOV    #VCCSR,ACSR      ;READ THE REGISTER  
364 000642* 012767 177216 177216      BIC    #140000,ACSP      ;MASK TO BITS 15-14  
365 000642* 001410 000025 177212      CMP    ASTAT,ACSP      ;COMPARE FOR SAME  
366 000642* 012767 000025 177212      BEQ    000642*  
367 000642* 012767 000025 177212      MOV    #25,FRTYP      ;BIT STUCK  
368 000674* 104405 000000* 000000      *****  
369 000674* 104405 000000* 000000      HRDERS,BEGIN,NULL      ;"X CROSS HAIR ENABLE" FAILED TO SET  
370 0007C2* 005267 177400      *****  
371 0007C2* 005267 177400      INC    FATAL      ;SET DROP MODULE FLAG  
372 000766* 012767 015000 177170      ;TEST THAT "CLEAR LOW BYTE" DOES NOT CLEAR HIGH BYTE  
373 000766* 012767 015000 177170      CLRLOW: MOV    #15000,ASTAT      ;LOAD EXPECTED READ/WRITE RESULT  
374 000722* 105077 177300 177304      MOV    ASTAT,6VCCSR      ;LOAD THE REGISTER BITS  
375 000722* 105077 177304 177304      CLRP    #VCCSR      ;CLEAR LOW BYTE  
376 000726* 012767 177274 177146      MOV    #VCCSR,ACSR      ;READ THE RESULT OF "CLRP"  
377 000734* 012767 140000 177140      BIC    #BIT15,1BIT14,ACSR      ;MASK TO TOP TWO BITS  
378 000734* 012767 177140 177132      CMP    ASTAT,ACSP      ;COMPARE FOR SAME  
379 000734* 012767 177132 177132      BEQ    000734*  
380 000734* 012767 177132 177126      MOV    #25,FRTYP      ;BIT STUCK  
381 000760* 104405 000000* 000000      *****  
382 000760* 104405 000000* 000000      HRDERS,BEGIN,NULL      ;CLR LOW BYTE CHANGED THE HIGH BYTE OF THE CHAR STATUS R  
383 000766* 005267 177314      *****  
384 000766* 005267 177314      INC    FATAL      ;SET DROP MODULE FLAG  
385 000772* 005767 177310      ;TEST IF A LOGICAL ERROR OCCURRED  
386 000772* 005767 177310      FATAL: TST    FATAL      ;TEST IF AN ERROR OCCURRED ?  
387 000772* 005767 177310      BEQ    DVNX      ;PBR IF NOT
```

392 001000\* 104410 000000\*

ENDS,REGIN

;FATAL LOGIC ERROR DETECTED - DROP MODULE

```

393 ;DYNAMIC TESTING OF THE X CROSS HAIR POSITION REGISTERS
394 001004* 012777 010000 177214 DYNX: MOV #R112,@VCCSR ;ENABLE X CROSS HAIRS
395 001012* 005067 177066 CLR ASTAT ;CLEAR EXPECTED VALUE
396 001017* 005077 177206 CLR @VCHRL0 ;CLEAR POS
397 001022* 116777 177056 1S: MOVR ASTAT,@VCHRLC ;LOAD THE LOW BYTE (X POSITION)
398 001030* 004567 022104 JSR P5,DELAY
399 001034* 004567 177042 I
400 001035* 004567 177042 INC ASTAT ;UPDATE X POSITION
401 001036* 004567 177042 CMP ASTAT,XHAIR ;TEST IF MORE LINES?
402 001037* 004567 177042 BNE 1S
403 001050* 001364 177036 177222
404
405 ;DYNAMIC TESTING OF THE Y CROSS HAIR POSITION REGISTERS
406 001052* 012777 004000 177146 DYNY: MOV #R111,@VCCSR ;ENABLE Y CROSS HAIRS
407 001060* 005067 177020 CLR ASTAT ;CLEAR POSITION
408 001064* 005077 177140 CLR @VCHRL0 ;CLEAR POSITION
409 001070* 116777 177010 1S: MOVR ASTAT,@VCHRH1
410 001075* 004567 002036 JSR R5,DELAY
411 001076* 004567 177010 I
412 001102* 000001 INC ASTAT ;UPDATING Y POSITION
413 001104* 105457 177160 176766 INC ASTAT,ASTAT ;TEST FOR MORE LINES?
414 001116* 001364 177160 BNE 1S
415 001117* 005077 177102 CLR @VCCSR ;DISABLE CROSS HAIRS
416 001120* 005077 177102
417 ;ROTATING CHARACTERS ACROSS ALL COLS. <ALL CHARACTERS>
418 ROTCHR: JSR PC,HOME ;SET-UP STARTING CHARACTER
419 001130* 012767 000440 177132 MOV #46,STCHAR
420 001131* 012767 000440 JSR PC,HOME ;LOAD TEMP
421 001132* 012767 000440 MOV VHO,TENPO ;LOAD R1-TO CHARACTER
422 001142* 016767 177132 177114 1S: MOV STCHAR,R1 ;LOAD A BUFFER STARTING WITH
423 001143* 016767 177132 JSR R5,LIC ;THAT CHARACTER AND WIDTH <BYTE>
424 001150* 016767 177114 INC WIDTH ;DISPLAY A FULL LINE FROM THE BUFFER
425 001154* 004567 001514 JSR PC,YPRNT ;DONE?
426 001160* 004567 002176* JSR R5,TEMPO ;BR IF YES
427 001161* 004567 002176* BNE 2S
428 001162* 004767 001450 JSR PC,HOME ;LOAD TEMP
429 001163* 004767 001450 DEC TEMPO ;LOAD R1-TO CHARACTER
430 001164* 004767 001450 BNE 2S ;LOAD A BUFFER STARTING WITH
431 001165* 005367 177072 JSR R5,DELAY ;THAT CHARACTER AND WIDTH <BYTE>
432 001172* 001010 177072 INC STCHAR ;DISPLAY A FULL LINE FROM THE BUFFER
433 001174* 004567 001740 JSR R5,TEMPO ;DONE?
434 001202* 004767 001362 BNE 2S
435 001206* 016767 177066 177050 JSR PC,HOME ;LOAD TEMP
436 001206* 016767 177066 MOV VHO,TENPO ;LOAD R1-TO CHARACTER
437 001214* 005267 177050 INC STCHAR ;LOAD A BUFFER STARTING WITH
438 001226* 026767 177042 2S: CMP LASTCH,STCHAR ;THAT CHARACTER AND WIDTH <BYTE>
439 001226* 001350 BNE 1S ;TEST FOR FINAL CHARACTER
440
441 001230* 004567 001704 JSR R5,DELAY ;BRANCH IF NOT COMPLETED
442 001231* 004567 001704 JSR PC,HOME
443 001237* 004767 001326

```

VSSR DEC/X11 SYSTEM EXERCISER MODULE MACV11 30A(1052) 12-OCT-78 17:08 PAGE 14  
XVSBB0.F11 12-OCT-78 12:23

SEQ 0013

```

444 001242* 016767 176540 177006 NEXTON: MOV ADDR,TEMP1      ;GET BASE ADDRESS
445 001250* 042767 177700 176770 ADD #20,TEMP1      ;UPDATE TO THE MAP ADDRESS
446 001256* 005447 177700 BIC #177700,DVISAV    ;MASK TO LOWER SIX BITS
447 001262* 005267 176766 MAPRTA: BIC #FFFF,MAPRTA    ;IF NO MAPS SELECTED
448 001268* 015767 000001 176762 CLR TSPTR        ;CLEAR MAP INDICATOR
449 001274* 015267 176766 MOV #BIT0,TEMP3      ;LOAD TESTING BIT
450 001280* 015267 176756 176746 MAPRTB: BIT TEMP3,DVISAV    ;TEST IF THIS BIT MAP IS SELECTED
451 001302* 001402 176756 REG MAPRET      ;REG IF NOT
452 001314* 000167 000064 JNP TSTMAP      ;SELECTED - TEST THIS MAP
453 001314* 005767 176766 MAPRET: TST FATAL      ;TEST FOR FATAL ERROR
454 001320* 001403 176734 REG IS      ;REG IF NONE
455 001336* 0062767 000020 176724 BIC TEMP2,DVISAV    ;DROP THAT BIT MAP
456 001336* 005267 176716 ADD #20,TEMP1      ;UPDATE TO NEXT BUS ADDRESS
457 001336* 005267 176716 INC TEMP2      ;UPDATE MAP INDICATOR
458 001342* 006367 176714 ASL TEMP3      ;MOVE THE TEST BIT
459 001342* 0022767 000100 176706 CMP #100,TEMP3    ;TEST IF ANY MORE POSSIBLE MAPS
460 001354* 001351 176672 RNE MAPFTR      ;BR IF MORE
461 001354* 005767 176672 TST DVISAV      ;TEST IF ANY MAPS LEFT ?
462 001364* 0010102 176672 RNE MAPPTA    ;BR IF SOME
463 001370* 104410 000000* ENDS, BEGIN      ;TEST IF SOME
464 001370* 000167 176710 MAPRTA: ENDITS,BEGIN      ;DROP THE MODULE BECAUSE ALL MAPS HAVE ERRED
465 001370* 104413 000000* JMP START      ;SIGNAL END OF ITERATION
466 001374* 000167 176710 JMP START      ;MONITOR SHALL TEST END OF PASS

```

VSSR DEC/Y11 SYSTEM EXERCISER MODULE MACV11 30A(1052) 12-OCT-78 17:08 PAGE 15  
XVSSBY0.F11 12-OCT-78 12:23

SEQ 0014

```

468 ;NOW LOAD THE BUS ADDRESSES FROM "TEMP1" AND START TESTING
469 001400* 016767 176652 176632 TSTM: MCV TEMP1,VGCSR      ;LOAD BUS ADDRESSES
470 001406* 016767 176644 176626 MCV TEMP1,VGPC      ;LOAD BUS ADDRESSES
471 001414* 016767 176636 176622 MCV TEMP1,VGBUF      ;LOAD BUS ADDRESSES
472 001420* 016767 176639 176626 MCV TEMP1,VCCOLR      ;LOAD BUS ADDRESSES
473 001436* 0022767 000002 176604 ADD #2,VGCR      ;LOAD BUS ADDRESSES
474 001436* 0062767 000002 176600 ADD #2,VGCR      ;LOAD BUS ADDRESSES
475 001442* 0062767 C0010 176624 ADD #10,VCCOLR      ;LOAD BUS ADDRESS INCASE OF ERROR
476 001442* 0062767 C0010 176640 ADD #10,VCCSP,CSPA      ;LOAD BUS ADDRESS INCASE OF ERROR
477 001463* 016767 176562 176620 CLF FATAL      ;CLEAR DPOP MODULE FLAG
478 001460* 005267 176622 CLF FATAL      ;CLEAR DPOP MODULE FLAG
479
480 ;TEST THAT "EXPAND" BIT CAN BE SET
481 001462* 012767 004000 176412 TSTEXP: MOV #BIT1,ACSR      ;LOAD THE EXPECTED
482 001472* 016777 176406 176540 MOV ASTAT,AVGCSR      ;LOAD THE REGISTER
483 001500* 017677 176534 176374 MOV AVGCSR,ACSR      ;READ THE REGISTER
484 001506* 042767 176360 176366 BIC #171360,ACSR      ;MASK TO BITS
485 001514* 028767 176364 176360 CMP ASTAT,ACSR      ;COMPARE THE EXPECTED TO RECD
486 001524* 001410 REQ TSTMOR      ;REQ TSTMOR
487 001524* 012767 000025 176354 MOV #23,ERRTYP      ;BIT STUCK
488 001532* 104405 000000* 000000 HDEFS,BEGIN,NULL      ;"EXPAND" BIT FAILED TO SET
489 001540* 005267 176542 INC FATAL      ;SET DROP MODULE FLAG
490
491 ;TEST THAT "MONO" BIT CAN BE SET
492 001544* 012767 002000 176332 TSTMOR: MOV #BIT10,ASTAT      ;LOAD THE EXPECTED
493 001552* 016777 176326 176450 MOV ASTAT,AVGCSR      ;LOAD THE REGISTER
494 001560* 017677 176454 176314 MOV AVGCSR,ACSR      ;READ THE REGISTER
495 001566* 042767 171360 176306 BIC #171360,ACSR      ;MASK TO BITS
496 001574* 028767 176304 176300 CMP ASTAT,ACSR      ;COMPARE THE EXPECTED TO RECD
497 001582* 001410 BEQ MAPTST      ;MAPTST
498 001604* 012767 000025 176274 MOV #23,ERRTYP      ;BIT STUCK
499 001612* 104405 000000* 000000 HDEFS,BEGIN,NULL      ;"MONO" BIT FAILED TO SET
500 001620* 005267 176462 INC FATAL      ;SET DROP MODULE FLAG
501
502 ;TEST THAT "MAP ENABLE" BIT CAN BE SET
503 001624* 012767 000400 176252 MAPTST: MOV #BIT8,ASTAT      ;LOAD THE EXPECTED
504 001632* 016777 176244 176400 MOV ASTAT,AVGCSR      ;LOAD THE REGISTER
505 001640* 017677 176314 176234 MOV AVGCSR,ACSR      ;READ THE REGISTER
506 001646* 042767 171360 176226 BIC #171360,ACSR      ;MASK TO BITS
507 001654* 028767 176224 176220 CMP ASTAT,ACSR      ;COMPARE THE EXPECTED TO RECD
508 001662* 001410 BEQ TSTMOR      ;TSTMOR
509 001664* 012767 000025 176214 MOV #23,ERRTYP      ;BIT STUCK
510 001672* 104405 000000* 000000 HDEFS,BEGIN,NULL      ;MAP ENABLE FAILED TO SET
511 001700* 005267 176402 INC FATAL      ;SET DROP MODULE FLAG

```

```

519          ;TEST THAT "ORIGIN POINT" BITS CAN BE SET
520  001712* 012767 000002 176120  TSTORG: MOV #P11#R10#BITS8,ASTAT      ;LOAD THE EXPECTED
521  001720* 012767 176166 176320  15:    MOV ASTAT,@VGCSR           ;LOAD THE REGISTER
522  001726* 012767 176314 176154  MOV @VGCSR,ACSR            ;READ THE REGISTER
523  001726* 012767 171360 176146  RIC #171366,ACSR            ;MASK TO BITS
524  001734* 026767 176144 176140  CMP ASTAT,ACSR            ;COMPARE THE EXPECTED TO RECVD
525  001742* 001410 000025 176134  BEQ 25
526  001744* 012767 000025 176134  MOV #25,ERRRTYP           ;BIT STUCK
527  001752* 104405 000000* 000000  *****HRDERS,RECIN,NULL***** ;"ORIGIN POINT" BIT FAILED TO SET
528  001760* 005267 176322 000000* 000000  25: INC FATAL                ;SET DROP MODULE FLAG
529  001764* 025267 176134 000020 176106  INC ASTAT                ;UPDATE EXPECTED VALUE OF "ORIGIN POINT"
530  001770* 022767 000020 176106  CMP #20,ASTAT              ;TEST IF VALID VALUE FOR "ORIGIN POINT"
531  001776* 001345 000025 176134  BNE 15                   ;BR IF MORE TO TEST
532
533          ;TEST THAT CLEAR LOW BYTE DOES NOT CLEAR HIGH BYTE
534  002000* 012767 006400 176076  TSTLOW: MOV #R11#R10#BITS8,ASTAT      ;TEST
535  002006* 012777 006417 176224  MOV #R11#R10#BITS8+17,@VGCSR   ;LOAD EXPECTED READ VALUE
536  002114* 012767 176226 000000* 000000  CLR @VGCSR             ;CLEAR LOW BYTE
537  002120* 012767 176214 176054  MOV @VGCSR,ACSR            ;READ THE REGISTER
538  002126* 012767 176300 176046  RIC #176306,ACSR            ;MASK TO BITS
539  002132* 026767 176044 176040  CMP ASTAT,ACSR            ;TEST IF SAME
540  002044* 012767 000025 176034  BEQ 25
541  002044* 012767 000025 176034  MOV #25,ERRRTYP           ;BIT STUCK
542  002052* 104405 000000* 000000  *****HRDERS,RECIN,NULL***** ;CLEARING LOW BYTE OF STATUS CHANGED THE HIGH BYTE
543  0020F0* 005267 176222 000000* 000000  INC FATAL                ;SET DROP MODULE FLAG
544
545          ;TEST THAT CLEAR HIGH BYTE DOES NOT CLEAR LOW BYTE
546  002064* 012767 000017 176012  TSTHGH: MOV #R11#R10#BITS8+17,@VGCSR   ;LOAD EXPECTED READ VALUE
547  002072* 012777 006417 176140  MOV #R11#R10#BITS8+17,@VGCSR   ;LOAD STATUS REGISTER
548  002116* 012767 176134 000000* 000000  MOV @VGCSR,RO             ;MAKE ADDRESS
549  002124* 012767 176134 000000* 000000  INC RO                  ;OF HIGH BYTE
550  002130* 012767 176134 000000* 000000  CLR RO                  ;CLEAR HIGH BYTE
551  002136* 012767 176300 175764  MOV @VGCSR,ACSR            ;READ THE REGISTER
552  002142* 026767 175756 175750  RIC #175756,ACSR            ;MASK TO BITS
553  002142* 012767 000025 175744  CMP ASTAT,ACSR            ;COMPARE EXPECT TO RCVD
554  002132* 026767 175754 175750  BEQ RDRTY
555  002134* 012767 000025 175744  MOV #25,ERRRTYP           ;BIT STUCK
556  002142* 104405 000000* 000000  *****HRDERS,RECIN,NULL***** ;CLEARING THE HIGH BYTE OF STATUS CHANGED THE LOW BYTE
557  002150* 005267 176132 000000* 000000  INC FATAL                ;SET DROP MODULE FLAG

```

```

564          ;READ-WRITE TEST OF THE LOW 3 BITS OF THE MAP P.C.
565  002154* 005067 175724  RDWRIT: CLR ASTAT               ;CLEAR EXPECTED READ VALUE
566  002160* 005067 176240  STEMP                 ;CLEAR WRITE VALUE
567  002164* 104407 000000* 000000* 15: BPEAKS,RECIN           ;TEMPORARY RETURN TO MONITOR
568  002170* 104407 000000* 000000* BPEAKS,RECIN           ;THEN CONTINUE AT NEXT INSTRUCTION.
569  002174* 012777 176024 176040  MOV STEMP,@VGPC          ;LOAD BIT MAP P.C.
570  002200* 012767 176024 176040  MOV @VGPC,ACSR            ;READ STATUS (BIT MAP)
571  002210* 026767 176062 176056  CMP #176062,ACSR            ;MASK TO UNUSED BITS
572  002216* 026767 176062 176056  BEQ 25
573  002224* 001410 000025 175652  MOV #25,ERRRTYP           ;BIT STUCK
574  002234* 104405 000000* 000000  *****HRDERS,RECIN,NULL***** ;BIT MAP P.C. DID NOT LOAD CORRECTLY
575  002242* 005267 176040 000000* 000000* 25: INC FATAL                ;SET DROP MODULE FLAG
576  002248* 005267 176034 175630  ADD #10,ASTAT              ;UPDATE EXPECTED WRITE VALUE
577  002254* 026767 176034 175630  INC STEP,ASTAT            ;TEST FOR FIRST NON-VALID
578  002260* 022767 000010 175736  CMP #10,STEP,ASTAT            ;TEST FOR FIRST NON-VALID
579  002266* 001336 000025 175652  BNE 15                   ;BRANCH IF STILL OK VALUES.
580
581  002270* 005767 176012  TST FATAL                ;TEST IF AN ERROR OCCURRED
582  002274* 001410 000000* 000000  REG V15EXP
583  002276* 00167 177012  JMP MAPRET              ;RETURN NOW - DONT BOTHER WITH DISPLAY STUFF

```

VSSB DFC/X11 SYSTEM EXERCISER MODULE  
XVSBB0.P11 12-OCT-78 12:23

MACY11 30A(1052) 12-OCT-78 17:08 PAGE 18

SEQ 0017

590 ;TEST THE "EXPAND" FUNCTION  
591 0023027 004767 000444 175732  
592 VISEXP: JSR PC,CLRMAP  
593 MOV #17,@VGCOLR ;LOAD MAX INTENSITY INTO LOCATION 0  
594 0023147 012777 002400 175716  
595 MOV #BIT10!BIT8,@VGCSR ;EXPAND OFF AND ENABLE BIT MAP  
596 0023227 004567 000612  
597 JSR R5,DELAY 100.  
598 0023307 052777 004000 175702  
599 BIS #BIT11,@VGCSR ;SET THE "EXPAND" BIT  
600 JSR R5,DELAY 100.  
601 ;TEST THE "ORGIN" FUNCTION  
602 0023447 004767 000402  
603 VISORG: JSR PC,CLRMAP ;CLEAR THE MAP  
604 0023567 012777 000017 175670  
605 MOV #17,@VGCOLR ;LOAD MAX INTENSITY INTO TABLE LOC. 0  
606 0023567 00567 175654  
607 CLR STEM  
608 0023647 00567 175634  
609 0023707 116777 175630 175642 1S:  
610 MOVB STEM, @VGCSR ;LOAD ORGIN REGISTER  
611 JSR R5,DELAY 100.  
612 INC# STEM  
613 0024047 105267 175614  
614 0024197 122767 000020 175606  
615 BNE 1S ;TEST FOR A NON-VALID ORGIN  
616 ;TEST FOR A VALID ORGIN  
617 ;INTENSITY LEVEL TEST-MONOCROME  
618 0024297 004767 000326  
619 VISIONT: JSR PC,CLRMAP ;CLEAR THE BIT MAP  
620 0024247 004767 000644  
621 CLR 2S ;LOAD TWO WORDS PER LOOKUP TABLE LOCATION  
622 0024307 00567 000012  
623 MOVB #BIT11!BIT10!BIT8,@VGCSR ;CLEAR STARTING INTENSITY LEVEL  
624 0024347 002777 000400 175576 1S:  
625 JSR R5,LODSTA ;SET MONOCROME + ENABLE MAP  
626 0024457 003000 2S: ;LOAD SEQUENCE  
627 TBLFO ;INDEX INTO DATA VALVE TABLE  
628 0024527 004567 000462  
629 JSR R5,DELAY 100.  
630 ADD #1,2S ;UPDATE INDEX INTO TABLE FOR LOCATION 0'S VALUE  
631 0024747 001367 000020 177760  
632 BNE 1S ;TEST IF LAST VALUE FOR MONOCROME  
633 ;TEST IF MORE LEVELS FOR LOCATION 0  
634 0024767 004767 00025A  
635 0025027 004767 000265  
636 0025067 00567 00031C  
637 0025127 005277 175522  
638 0025127 004567 000660  
639 0025227 003152  
640 0025227 003152 004100 175504  
641 0025347 004567 000400  
642 0025407 000144  
643 0025427 002767 000001 177752  
644 0025567 002767 000020 177744  
645 0025567 001357  
646 0025607 005077 175454  
647 CLR @VGCSR  
648 ;DISABLE MAP  
649 0025647 000167 176524  
650 JMP MAPPET

VSSB DFC/X11 SYSTEM EXERCISER MODULE  
XVSBB0.P11 12-OCT-78 12:23

MACY11 30A(1052) 12-OCT-78 17:08 PAGE 19

SEQ 0018

633 ;COLOR LEVEL TEST - COLORCROME  
634 VISCOLT: JSR PC,CLRMAP ;ENSURE CLEAR MAP  
635 0025027 004767 000265 ;LOAD TWO WORDS PER LOOKUP TABLE LOC.  
636 CLR 2S ;CLEAR INDEX POINTER  
637 0025127 005277 175522 ;CLEAR STATUS  
638 0025127 004567 000660 1S: JSR R5,LODSTA ;LOAD DATA INTO TABLE  
639 0025227 003152 ;TABLEP  
640 MOVB #BIT11!BIT8,@VGCSR ;ENABLE MAP  
641 JSR R5,DELAY 100.  
642 ADD #1,2S ;UPDATE LOC 0 INDEX INTO TABLE  
643 CMP #20,2S ;TEST IF FINISHED FOR LOC 0  
644 BNE 1S  
645 CLR @VGCSR  
646 ;DISABLE MAP  
647 0025647 000167 176524  
648 JMP MAPPET

```

650 002570* 004567 000344 HOME: JSR R5,DELAY
652 002574* 000001 1 MOV #35,BUFFER ;LOAD HOME CHAR
653 002576* 012767 000035 000672 MOVB #1,BUFFER+1 ;LOAD TERMINATOR
654 002593* 004724 000020 JSR PC,XPRINT
655 002616* 104467 000009* 1S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR
656 002622* 104467 000006* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
657 002626* 005777 175374 TST @VCCSR ;TEST FOR READY
658 002632* 100371 RTS 1S ;WAIT
659 002634* 000207 RTS PC
660 .SPTTL DISPLAY BUFFER ON VTV01 SCREEN SUR-ROUTINE
661 002636* 012700 003476* XPRNT: MOV #PUFFER,R0 ;LOAD BUFFER POINTER
662 002642* 000000 1S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR
663 002642* 104467 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
664 002652* 005777 175350 TST @VCCSR ;TEST TIL READY IS SET
665 002656* 100371 175350 RPL 1S ;WAIT TIL DONE
666 002664* 112014 003777 MOVP (R0),(R0) ;LOAD THE CHARACTER
667 002664* 122710 000377 CMPR #377,(R0) ;TEST FOR TERM
668 002670* 001364 BNE 1S ;BR IF NOT TERM.
669 002672* 000207 RTS PC ;EXIT
670
671
672 ;LOAD A INCREMENTING CHARACTER ACROSS THE SCREEN WIDTH
673 ;ONLY 40 THRU 177 ARE LEGAL CHARACTERS
674 002674* 012700 003476* LTC: MOV #BUFFER,R0 ;SET-UP BUFFER POINTER
675 002674* 003567 000000* MOV R0,(5)+$2 ;SET-UP WIDTH
676 002674* 012700 000000* 1S: MOVB R0,(0)$+ ;UPDATE THE CHARACTER
677 002674* 005761 175354 INC R1 ;TEST FOR
678 002674* 001002 175354 CMP LASTCH,R1 ;TEST FOR
679 002674* 012701 000040 BNE 2S ;BRANCH IF NOT
680 002674* 104467 000000* MOV #40,R1 ;MAKE A LEGAL CHARACTER
681 002674* 012702 000000* 2S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR
682 002674* 005762 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
683 002674* 012702 000000* DEC R0 ;DECREMENT COUNT
684 002674* 005762 000000* BNE 1S ;BRANCH IF NOT COMPLETED
685 002674* 112720 000015 MOVB #15,(R0)+ ;LOAD "CR"
686 002674* 112720 000012 MOVB #12,(R0)+ ;LOAD "LF"
687 002674* 112710 000377 MOVB #377,(R0) ;LOAD TERM
688 002675* 000205 RTS RE ;EXIT
689
690

```

```

691 ;SUBROUTINE TO CLEAR THE BIT MAP
692 002752* 012767 000100 000074 CLRMAP: MOV #PIT6,10$ ;LOAD BASE DELAY
693 002760* 005067 000072 CLR 11$ ;CLEAR COUNTER
694 002764* 004567 000150 JSR R5,DELAY
695 002772* 012777 001000 175240 1S: MOV #PIT9,@VCCSR ;SET "CLEAR MAP" BIT
696 003000* 104467 000000* BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR
697 003004* 104467 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
698 003016* 105777 175224 TST @VCCSR ;TEST READY FLAG
699 003016* 004016 000000* BMI 2S ;BR IF READY SET
700 003016* 005367 000034 INC 11$ ;UPDATE COUNTER
701 003022* 005367 000034 BNE 11$ ;TEST FOR OVERFLOW
702 003022* 005367 000024 DEC 10$ ;DECREMENT BASE DELAY
703 003030* 001363 000000 BNE 1S ;BR IF NO OVERFLW
704 003032* 012767 000011 175046 MOV #11,ERRTYP ;ILLEGAL INTERRUPT
705 003040* 104465 000000* 000000 *****END***** ;READY FAILED TO SET AFTER A GROSS
706 003046* 005267 175234 HADERS,BEGIN,NULL ;SET DROP MODULE FLAG
707 003046* 005267 175234 INC FATAL ;SET FATAL
708 003052* 000207 2S: RTS PC ;TIME DELAY
709 003054* 000000 10$; 0 ;BASE DELAY
710 003056* 000000 11$; 0 ;COUNT LOCATION
711 .SPTTL TABLE OF VALUES THAT WILL BE LOADED INTO THE COLOR TABLE (L.O.T.)
712 003060* 000 001 002 TABLEO: .BYTE 0,1,2,3,4,5,6,7,10,11,12,13,14,15,16,17
713 003063* 003 004 005
714 003066* 006 007 010
715 003070* 010 011 012
716 003074* 014 015 016
717 003077* 017
718 003100* 001 002 003 TABLEB: .BYTE 1,2,3,4,10,14,20,40,60,63,74,17,33,25,52,77
719 003103* 004 010 014
720 003106* 006 040 060
721 003111* 003 074 017
722 003114* 003 025 052
723 003117* 001 002 003
724 003120* 004 010 014
725 003123* 004 010 014
726 003126* 020 040 060
727 003131* 063 074 017
728 003134* 055 025 052
729 003137* 077
730 .EVEN

```

```

746          .SBTTL  DELAY SUBROUTINE -- WAIT FOR "VERTICAL SYNC INTERRUPT"
748          003144* 012567  000194* 175076  4$:    MOV    (RS)+,R0      ;LOAD THE ARG
750          003144* 012567  000194* 175076  4$:    MOV    #35,$0VCINT   ;LOAD INTR. RETURN VECTOR
751          003152* 005077  175074               CLR    $0VCINT1
752          003156* 022777  020000  175042  6$:    BIS    #BIT13,$0VCCSR ;ENABLE INTR.
753          003164* 104400  000000  175030  3$:    EXIT   #BIT13,$0VCCSR ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
754          003170* 042777  020000  175030  3$:    BIC    #BIT13,$0VCCSR ;DISABLE INTR.
755          003176* 000004  000000* 003204* 10$:   TIRQ$,REGIN,5$  ;QUEUE UP TO CONTINUE AT 5$ AND RTI
756          003204* 005367  000020               5$:   DEC    10$      ;FINISHED DELAY ?
757          003210* 001362  000020               BNE    6$      ;BR IF NOT
758          003212* 042777  020000  175006  2$:    BIC    #BIT13,$0VCCSR ;CLEAR INTR. ENABLE
759          003220* 01F777  175026  175022               MOV    VCINT1,$0VCINT ;RESTORE INTR. VECTOR
760          003226* 000205               RTS    R5      ;EXIT
761
762
763
764
765          003230* 000000               10$:   C
766
767          ;RANDOM NUMBER GENERATOR
768          003232* 066767  000034  000030  SRAND: ADD    $LONUM,$SHINUM ;ADJUST DATA
769          003240* 005567  000026               ADC    $LONUM
770          003244* 066767  000050  000020               ADD    $SHINUM,$LONUM
771          003244* 066767  000050               ADC    $SHINUM
772          003242* 005567  000012               ALU    $SHINUM
773          003256* 104407  000000               BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR
774          003266* 104407  000000               BREAKS,PCIN ;THEN CONTINUE AT NEXT INSTRUCTION.
775          003266* 000207               RTS    PC      ;EXIT
776
777          003270* 176543               SHINUM: 176543
778          003272* 123456               $LONUM: 123456

```

```

780          ;LCAD MAP WITH TWO WORDS OF THE SAME VALUE
781          ;INTO THE ENTIRE BITMAP STARTING AT LOC. 0
782
783          0033274* 010146
784          0033276* 010146
785          003300* 005777  174736
786          003304* 016761  174734
787          003310* 006201
788
789          003312* 000000
790          003314* 104407  000000* 15$:   CLP    R0      ;SAVE R0
791          003314* 104407  000000* 15$:   MOV    R1,-(SP) ;SAVE R1
792          003320* 104407  000000* 15$:   CLD    $VGPC
793          003324* 105777  174719               MOV    R0,RUMPIX,R1 ;ENSURE CLEAR PC
794          003330* 105777  174706               ASR    R1      ;LOAD COUNTER
795          003332* 105777  174706               BPL    2$      ;ADJUST COUNT
796
797          003334* 000000
798          003334* 104407  000000* 25$:   CLP    R0      ;START WITH PIXEL VALUE OF 0
799          003334* 104407  000000* 25$:   BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...
800          003334* 104407  000000* 25$:   BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
801          003334* 105777  174666               TSTP   $VGCSRP ;ENSURE MAP IS READY
802          003334* 105777  174666               BPL    3$      ;LOAD A PIXEL WORD
803          003334* 105777  174664               MCV    R0,$VGCBUF ;LCAD 2ND PIXEL WORD
804          003342* 104407  000000* 35$:   BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR...
805          003342* 104407  000000* 35$:   BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
806          003352* 105777  174666               TSTP   $VGCSR ;ENSURE MAP IS READY
807          003352* 105777  174666               BPL    3$      ;LOAD A PIXEL WORD
808          003354* 105777  174664               MOV    R0,$VGCBUF ;LCAD 2ND PIXEL WORD
809          003360* 005301               DEC    R1      ;FINISHED ALL PIXELS ?
810          003362* 104407               RMW    4$      ;BR IF FINISHED
811          003364* 005760  010421               ADD    #104421,R0 ;UPDATE PIXEL DATA TO NEXT L.U.T. ADDRESS
812          003364* 005760  010421               BCC    2$      ;BR IF MORE ADDRESSES TO LOAD
813          003372* 005747               RP    1$      ;BR TO RESET PIXEL DATA
814
815          003374* 012601               4$:    MOV    (SP)+,R1 ;RESTORE R1
816          003376* 012600               MOV    -(SP),R0 ;RESTORE R0
817          003400* 000207               RTS    PC      ;EXIT

```

VSSB DEC/X11 SYSTEM EXERCISER MODULE  
XVSBB0.P11 12-OCT-78 12:23

MACV11 30A(1052) 12-OCT-78 17:08 PAGE 24  
SUBROUTINE TO LOAD THE L.U.T. WITH THE VALUE OF A TABLE  
.SBTTL SUBROUTINE TO LOAD THE L.U.T. WITH THE VALUE OF A TABLE  
LODSTA: MOV R0E-(SP) ;SAVE R0  
MOV (R5)+,R0 ;GET STARTING LUT ADDRESS  
MOV #16.,11S ;GET TABLE ADDRESS  
MOV R0 ;LOAD COUNTER  
MOVB 10\$,12\$+1 ;LCAD HIGH BYTE WITH LUT ADDRESS  
15: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR  
MOV (R0)+,12\$ ;THEN CONTINUE AT NEXT INSTRUCTION.  
MOV 12\$+1,VGCOLR ;LOAD VALUE INTO LOW BYTE  
INCB 11\$+1 ;LOAD THE TABLE  
DEC 11\$ ;UPDATE THE LUT POINTER  
BNE 11\$ ;FINISHED ALL L.U.T. ADDRESSES  
MOV (SP)+,R0 ;BR IF NOT  
RTS R5 ;RESTORE R0  
;EXIT  
10\$: C ;COUNTER OF L.U.T.  
11\$: 16. ;TEMP LOCATION  
12\$: 0  
SET: 0  
BUFER: OPEN .BLKW 40.  
.END

SEQ 0023

VSSP DEC/X11 SYSTEM EXERCISE MODULE  
XVSPP0.P11 12-OCT-78 12:23

MACV11 30A(1052) 12-OCT-78 17:08 PAGE 26  
CROSS REFERENCE TABLE -- USER SYMBOLS

	ACSR 000102R	257#	339*	340*	341	352*	353*	354	365*	366*	367	379*	380*	381	382*
ADDR	000100R	483*	484*	485	496*	497*	498	509*	510*	511	522*	523*	524	539*	539*
ADDR22=	000100R	540*	541	555*	566*	557	572*	573*	574						
ASDR	000100R	542*	543	518	444										
ASTAT	000104R	259#	337*	338	341	350*	351	354	363*	364	367	376*	377	381	381
AWAS	000110R	396*	398	401*	402	408*	410	413*	414	481*	482	485	494*	495	495
BEGIN	000000R	220#	345	358	371	385	392	463	465	489	502	515	528	545	545
BIT0	= 000001	753	756	773	774	781	782	797	798	822	823				
BIT1	= 000002	275#	520												
BIT10	= 000006	275#	520												
BIT11	= 000007	275#	520												
BIT12	= 000008	275#	520												
BIT13	= 010000	275#	563	395	407	536	537	551	594	607	622				
BIT14	= 040000	275#	563	395	407	536	537	551	594	607	622				
BIT15	= 100000	275#	563	395	407	536	537	551	594	607	622				
BIT16	= 000009	275#	563	395	407	536	537	551	594	607	622				
BIT17	= 000014	275#	563	395	407	536	537	551	594	607	622				
BIT18	= 000013	275#	563	395	407	536	537	551	594	607	622				
BIT19	= 000026	275#	563	395	407	536	537	551	594	607	622				
BREAKS	= 104007	275#	563	395	407	536	537	551	594	607	622				
BR1	000012R	225#													
BR2	000013R	226#													
BTODS	= 104421	275#													
BUFER	000347FR	653*	654*	665	680	838#									
CALIBR	0000762R	275#	563	395	407	536	537	551	594	607	622				
CLRMAP	0002752R	275#	563	395	407	536	537	551	594	607	622				
CONFIG	000056R	245#	604	619	633	699#									
CSRA	000100R	255*	335*	477*											
CURST	000466R	275#													
DATERS	= 104404	275#													
DELY1	0003140R	275#	411	433	441	595	599	611	627	641	651	701	749#		
DV1D1	= 000014R	275#	333	446*	450	455*	461								
DVISAV	000254R	275#	333*	446*	450	455*	461								
DYNX	001004R	391	395#												
DVN	001052R	407#													
ENDS	= 104410	275#	465												
ERRTYPE	= 000106R	275#	395	463											
EXITS	= 104400	275#	323*	356*	369*	383*	487*	500*	513*	526*	543*	559*	576*	713*	
FATAL	= 000306R	311#	334*	347*	360*	373*	387*	390	453	478*	491*	504*	517*	530*	
FATAL0	000772R	382	563*	580*	586	591*									
		390#													

SEQ 0024

VSBR DEC/Y11 SYSTEM EXERCISER MODULE      MACY11 30A(1052) 12-OCT-78 17:08 PAGE 27  
 XVSBR0.P11 12-OCT-78 12:23      CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0025

GETPASS=	104415	275*
GMBUF\$=	102570R	420
HOME	000044R	420*
HRCNT	104405	275*
HRDPAS	000050R	242*
ICOUNT	000036R	337*
IDNIM	000122R	267*
INIT	000030R	344*
INTR	000120R	266*
LASTCH	000266R	303*
LIC	002674R	426
LOGEQ	003462R	623
LOGSTA	003463R	637
MAPRET	001314R	451
MAPRTA	001370R	447
MAPRTB	001300R	450*
MAPST	001624R	499
MAP2SS=	104416	275*
MUSYM	000052R	325*
MDSP	000052R	273*
MSCNS	= 104402	275*
MSCS\$	= 104402	275*
MSCS	= 104401	275*
NEXTON	001242P	444*
NULL	000000	275*
NUMPIX	000330R	312*
OPEN	= 000000	352
OTOAS	= 104420	275*
PASCNT	000034R	236*
POROS	= 000004	756
POPS	000022	275*
POSPD2	000020	275*
PRTV	000000	275*
PRTVO	= 000000	275*
PRTV1	= 000040	275*
PRTV2	= 000100	275*
PRTV3	= 000140	275*
PRTV4	= 000200	275*
PRTV5	= 000240	275*
PRTV6	= 000300	275*
PRTV7	= 000340	275*
PS	1777776	275*
PSW	1777776	275*
PUSH	= 000044	275*
PUSH2	= 000044	275*
PANS	= 104417	275*
PANNUM	000054P	444*
RDWRT	0021E4P	558
RESTART	000310R	263
RES1	000058R	246*
RES1	000060R	247*
ROTCHR	001124R	419*

VSBR DEC/Y11 SYSTEM EXERCISER MODULE      MACY11 30A(1052) 12-OCT-78 17:08 PAGE 28  
 XVSBR0.P11 12-OCT-78 12:23      CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0026

RSTRT	000112P	263*
SBDR	000170R	268*
SC1	003474R	268*
SDFCNT	000042R	339*
SDFERS	= 104406	275*
SDFPAS	000045R	241*
SPONT	000032R	235*
SPSIZ	= 000040	1*
SPSIZ	= 000040	268
SPS1	= 000040R	220*
SPS2	= 000040R	220*
SPS3	= 000022R	230*
SPS4	= 000024R	231*
START	000310R	234
STAT	000026P	233*
STARTR	000026P	233*
SVR0	000064R	240*
SVR1	000064R	240*
SVR2	000064R	240*
SVR3	000075P	251*
SVR4	000072R	252*
SVR5	000074R	253*
SVR6	000079R	254*
SVRCNT	000060P	249*
TAREF	003166R	272*
TAREFO	003060P	732#
TEMP	00264P	252
TEMP1	001266P	302*
TEMP2	000260P	446*
TEMP3	000260P	300*
TEMP4	000260P	449*
TOPLOC	000320P	309*
TRDFFN	= 000022	275*
TSTEXP	001464P	291*
TSTHIGH	002064P	542
TSTLOW	002060P	536#
TSTMAP	001400R	452
TSTMAP	001540R	486
VGCDRG	000336R	294#
VCCSR	000226P	380#
VCHRHI	000232R	282*
VCHRLO	000230R	291*
VCIINT	000250R	293*
VCLINT	000250R	294*
VCPDGS	0003234P	333*
VCPDOS	000336R	394#
VECTOR	000010R	324*
VGBUF	000244R	288*
VGCOLR	000246P	289*
VGCSR	000240R	286#
VGPC	000242R	567*
VHO	000300R	308*
VISCLT	002476P	633*
VISEXP	002302R	587
VISINT	002420R	619#
		591*

VSB80 DEC/X11 SYSTEM EXERCISER MODULE MACY11 30A(1052) 12-OCT-78 17:08 PAGE 29  
XVSBB0.P11 12-OCT-78 12:23 CROSS REFERENCE TABLE -- USER SYMBOLS  
SEQ 0027

VISORG 002344R	504#															
WASADR 000104R	558#															
WFDF 000116R	565#															
VIDEO 000117R	566#	316*														
WIDTH 000374R	567#															
XCROSS 000626R	568#	427														
XFLAG 000065R	569#	363#														
XHAIR 000272P	570#	402														
XPRMT 002636R	571#	658	665#													
XCROSS 000546R	572#	358#														
YHDM 000357R	573#	444														
SHNDM 003370R	574#	774														
SLONUM 003372R	575#	772*	772*	777#												
SRAND 003232R	576#	776	771*	771*	778#											
STEMP 000224R	577#	567*	571	582*	583	608*	610	613*	614							
.	= 003620R	578#														

\* ABS. 000000 000  
003620 001

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0  
XVSBB0,XVSBB0/SOL/CLK:SYM=DDXCOM,XVSBB0  
RUN-TIME: 1 2 3 SECONDS  
RUN-TIME RATIO: 20/4=4.2  
CORE USED: 7K (13 PAGES)