

.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 DIGITALS 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 "IDMOD" 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 FDP-11/05 TAKES APPROXIMATELY TWO MINUTES TO COMPLETE ONE PASS.

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 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 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	=	1	23	34	47	10	14	20	40
10	=	60	63	74	17	33	25	52	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

```

214 000000- IOMOD <VSRB>,177600,360,4,1,76
215 000000- MODULE 140000,VSRB,172600,360,4,1,76
216 ; TITLE VSRB DEC/X11 SYSTEM EXERCISER MODULE
217 ; DXCGM VERSION 6 23-MAY-78
218 ; *****LIST*****
219 ;*****LIST*****
220 000000- BEGIN:
221 000000- XFLAG: ASCII /VSRB / ;MODULE NAME
222 000000- ADDR: 172600+0 USED TO KEEP TRACK OF WBUF USAGE
223 000000- VECTOR: 360+0 ;1ST DEVICE ADDR.
224 000010- 000360 ;1ST DEVICE VECTOR.
225 000010- 000360 ;1ST BR LEVEL.
226 000013- 000 ;2ND BR LEVEL.
227 000013- 000000 ;3RD BR LEVEL.
228 000016- 000000 ;DEVICE INDICATOR 1.
229 000020- 000000 ;SWITCH REGISTER 1
230 000022- 000000 ;SWITCH REGISTER 2
231 000024- 000000 ;SWITCH REGISTER 3
232 000026- 140000 ;SWITCH REGISTER 4
233 000030- 000310- *****LIST*****
234 000032- 000224- *****LIST*****
235 000034- 000000- *****LIST*****
236 000037- 000000- *****LIST*****
237 000040- 000000- *****LIST*****
238 000042- 000000- *****LIST*****
239 000044- 000000- *****LIST*****
240 000046- 000000- *****LIST*****
241 000048- 000000- *****LIST*****
242 000050- 000000- *****LIST*****
243 000052- 000000- *****LIST*****
244 000054- 000000- *****LIST*****
245 000056- 000000- *****LIST*****
246 000058- 000000- *****LIST*****
247 000060- 000000- *****LIST*****
248 000062- 000000- *****LIST*****
249 000064- 000000- *****LIST*****
250 000066- 000000- *****LIST*****
251 000068- 000000- *****LIST*****
252 000070- 000000- *****LIST*****
253 000072- 000000- *****LIST*****
254 000074- 000000- *****LIST*****
255 000076- 000000- *****LIST*****
256 000078- 000000- *****LIST*****
257 000080- 000000- *****LIST*****
258 000082- 000000- *****LIST*****
259 000084- 000000- *****LIST*****
260 000086- 000000- *****LIST*****
261 000088- 000000- *****LIST*****
262 000090- 000000- *****LIST*****
263 000092- 000000- *****LIST*****
264 000094- 000000- *****LIST*****
265 000096- 000000- *****LIST*****
266 000098- 000000- *****LIST*****
267 000100- 000000- *****LIST*****
268 000102- 000000- *****LIST*****
269 000104- 000000- *****LIST*****

```

```

270 ;WORD 0
271 ;LIST
272 ;FNDR
273 *****LIST*****
274 *****LIST*****
275 *****LIST*****
276 *****LIST*****
277 *****LIST*****
278 *****LIST*****
279 *****LIST*****
280 *****LIST*****
281 *****LIST*****
282 *****LIST*****
283 *****LIST*****
284 *****LIST*****
285 *****LIST*****
286 *****LIST*****
287 *****LIST*****
288 *****LIST*****
289 *****LIST*****
290 *****LIST*****
291 *****LIST*****
292 *****LIST*****
293 *****LIST*****
294 *****LIST*****
295 *****LIST*****
296 *****LIST*****
297 *****LIST*****
298 *****LIST*****
299 *****LIST*****
300 *****LIST*****
301 *****LIST*****
302 *****LIST*****
303 *****LIST*****
304 *****LIST*****
305 *****LIST*****
306 *****LIST*****
307 *****LIST*****
308 *****LIST*****
309 *****LIST*****
310 *****LIST*****
311 *****LIST*****
312 *****LIST*****
313 *****LIST*****
314 *****LIST*****
315 *****LIST*****
316 *****LIST*****
317 *****LIST*****
318 *****LIST*****
319 *****LIST*****
320 *****LIST*****
321 *****LIST*****
322 *****LIST*****
323 *****LIST*****
324 *****LIST*****

```

```

326 000374 062767 000020 177636 ADD #20,VGCSR
327 000302 062767 000022 177632 ADD #20,VGPC
328 000410 062767 000024 177626 ADD #24,VGBUF
329 000410 062767 000030 177622 ADD #30,VGCLR
330 000410 062767 177520 177622 MOV #VCCSS,VCIINT ;LOAD BASIC INTR. VECTOR
331 000432 062767 177612 177612 MOV VCINT,VCIINT1
332 000440 062767 000002 177604 ADD #2,VCIINT1
333 000440 062767 177342 177600 MOV #VIDL,DVISAV
334 000460 062767 177542 177412 MOV #25,ERRTYP ;CLEAR DROP MODULE INDICATOR
335 000460 062767 177542 177412 MOV #VCCSR,CSRA ;LOAD BUS ADDRESS IN CASE OF ERRORS

```

```

336 000466 062767 002000 177410 ;TEST THE CHAR CURSOR DISABLE BIT
337 000474 062767 177404 177524 CURTST: MOV #BIT10,ASTAT ;LOAD EXPECTED VALUE
338 000474 062767 177404 177524 MOV ASTAT,@VCCSR ;LOAD THE REGISTER
339 000510 062767 140000 177372 MOV @VCCSR,ACSR ;READ THE REGISTER
340 000510 062767 140000 177364 BIC #140000,ACSR ;MASK TO BITS 15-14
341 000510 062767 177362 177366 CMP ASTAT,ACSR ;COMPARE FOR SAME
342 000524 001410 000025 177352 BEQ VCCRSS
343 000524 062767 000025 177352 MOV #25,ERRTYP ;BIT STUCK
344 000534 104405 000000 000000 ;***** ;CHAR CURSOR DISABLE" FAILED TO SET
345 000534 104405 000000 000000 HRDEFS,REGIN,NULL ;*****
346 000542 005267 177540 INC FATAL ;SET DROP MODULE FLAG
347
348
349 ;TEST THE Y CROSS HAIR ENABLE BIT
350 000546 062767 000000 177330 VYCRSS: MOV #BIT11,ASTAT ;LOAD EXPECTED VALUE
351 000546 062767 177324 177444 MOV ASTAT,@VCCSR ;LOAD THE REGISTER
352 000560 062767 177440 177312 MOV @VCCSR,ACSR ;READ THE REGISTER
353 000570 062767 140000 177364 BIC #140000,ACSR ;MASK TO BITS 15-14
354 000570 062767 177362 177276 CMP ASTAT,ACSR ;COMPARE FOR SAME
355 000600 062767 000025 177272 BEQ XCRSS
356 000600 062767 000025 177272 MOV #25,ERRTYP ;BIT STUCK
357 000614 104405 000000 000000 ;***** ;"Y CROSS HAIR ENABLE" FAILED TO SET
358 000614 104405 000000 000000 HRDEFS,REGIN,NULL ;*****
359 000622 005267 177460 INC FATAL ;SET DROP MODULE FLAG
360
361
362 ;TEST THE X CROSS HAIR ENABLE BIT
363 000626 062767 010000 177250 VXCRSS: MOV #BIT12,ASTAT ;LOAD EXPECTED VALUE
364 000634 062767 177244 177364 MOV ASTAT,@VCCSR ;LOAD THE REGISTER
365 000642 062767 177360 177232 MOV @VCCSR,ACSR ;READ THE REGISTER
366 000650 062767 140000 177224 BIC #140000,ACSR ;MASK TO BITS 15-14
367 000650 062767 177222 177216 CMP ASTAT,ACSR ;COMPARE FOR SAME
368 000664 001410 BEQ CLRLW
369 000664 062767 000025 177212 MOV #25,ERRTYP ;BIT STUCK
370 000674 104405 000000 000000 ;***** ;"X CROSS HAIR ENABLE" FAILED TO SET
371 000674 104405 000000 000000 HRDEFS,REGIN,NULL ;*****
372 000702 005267 177400 INC FATAL ;SET DROP MODULE FLAG
373
374
375 ;TEST THAT "CLEAR LOW BYTE" DOES NOT CLEAR HIGH BYTE
376 000706 062767 016000 177170 CLRLOW: MOV #16000,ASTAT ;LOAD EXPECTED READ/WRITE RESULT
377 000714 062767 177164 177364 MOV ASTAT,@VCCSR ;LOAD THE REGISTER BITS
378 000722 105077 177300 CLR @VCCSR ;CLEAR LOW BYTE
379 000722 105077 177274 MOV @VCCSR,ACSR ;READ THE RESULT OF "CLRB"
380 000734 062767 140000 177140 BIC #BIT15,BIT14,ACSR ;MASK TO TOP TWO BITS
381 000742 062767 177136 177132 CMP ASTAT,ACSR ;COMPARE FOR SAME
382 000750 001410 BEQ FATAL
383 000750 062767 000025 177126 MOV #25,ERRTYP ;BIT STUCK
384 000760 104405 000000 000000 ;***** ;CLR LOW BYTE CHANGED THE HIGH BYTE OF THE CHAR STATUS R
385 000760 104405 000000 000000 HRDEFS,REGIN,NULL ;*****
386 000766 005267 177314 INC FATAL ;SET DROP MODULE FLAG
387
388
389 ;TEST IF A LOGICAL ERROR OCCURRED
390 000772 005767 177310 FATAL: FATAL
391 000776 001402 REG DVANX ;TEST IF AN ERROR OCCURRED ?
;BR 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
395 001004* 012777 010000 177214 DYNX:  MOV    #BIT12,@VCCSR    ;ENABLE X CROSS HAIRS
396 001012* 005067 177066          CLR    ASTAT        ;CLEAR EXPECTED VALUE
397 001018* 005077 177206          CLR    @VCHRLD     ;CLEAR POS
398 001020* 116777 177056 177200 1S:  MOV    ASTAT,@VCHRLC ;LOAD THE LOW BYTE (X POSITION)
399 001030* 004567 002104          JSF   RS,DELAY
400 001034* 000001          I
401 001036* 005267 177042          INC    ASTAT        ;UPDATE X POSITION
402 001040* 026767 177036 177222 CMP    ASTAT,XHAIR ;TEST IF MORE LINES?
403 001050* 001364          BNE   IS
404
405                                     ;DYNAMIC TESTING OF THE Y CROSS HAIR POSITION REGISTERS
406
407 0010E2* 012777 004000 177146 DYNV:  MOV    #BIT11,@VCCSR    ;ENABLE Y CROSS HAIRS
408 001060* 005067 177020          CLR    ASTAT        ;CLEAR POSITION
409 001064* 005077 177140          CLR    @VCHRLD     ;CLEAR POSITION
410 001070* 116777 177010 177134 1S:  MOV    ASTAT,@VCHRH1
411 001076* 004567 00203A          JSF   RS,DELAY
412 001102* 000001          I
413 001104* 105267 176774          INCR  VHAIR,ASTAT
414 001110* 105267 177160 176766 CMP    VHAIR,ASTAT
415 001116* 001364          IS
416 001120* 005077 177102          CLR    @VCCSR      ;DISABLE CROSS HAIRS
417
418                                     ;ROTATING CHARACTERS ACROSS ALL COLS. <ALL CHARACTERS>
419 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 1S:  MOV    VHO,TEMPO ;LOAD TEMP
425 001150* 016767 177114          MOV    STCHAR,R1 ;LOAD R1=TO CHARACTER
426 001154* 004567 001514          JSR   RS,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   2S ;BR IF YES
433 001174* 004567 001740          JSR   RS,DELAY
434 001200* 000144          I
435 001202* 004767 001362          JSR   PC,HOME
436 001206* 016767 177056 177050 2S:  MOV    VHO,TEMPO ;UPDATE THE STARTING CHARACTER
437 001214* 005267 177050          INC   STCHAR ;TEST FOR FINAL CHARACTER
438 001220* 026767 177042          CMP   LASTCH,STCHAR
439 001226* 001350          BNE   IS ;BRANCH IF NOT COMPLETED
440
441 001230* 004567 001704          JSR   RS,DELAY
442 001234* 000144          I
443 001236* 004767 001326          JSR   PC,HOME

```

```

444 001242* 016767 176540 177006 NEXTON: MOV ADDR,TEMP1 ;GET BASE ADDRESS
445 001250* 062767 000020 177000 ADD #20,TEMP1 ;UPDATE TO THE MAP ADDRESS
446 001256* 042767 177700 176770 RLC #177700,DVISAV ;MASK TO LOWER SIX BITS
447 001484* 016767 176540 177000 MOV MAPPTA ;IF NO MAPS SELECTED
448 001266* 005067 176766 CLS TEMP3 ;CLEAR MAP INDICATOR
449 001272* 012767 000001 176762 MOV #BIT0,TEMP3 ;LOAD TESTING BIT
450 001366* 036767 176756 176746 MAPRTB: BIT TEMP3,DVISAV ;TEST IF THIS BIT MAP IS SELECTED
451 001366* 036767 176756 176746 REC MAPRET ;IF NOT
452 001310* 000167 000064 JMP TSTMAP ;SELECTED - TEST THIS MAP
453 001314* 005767 176766 MAPRET: TST FATAL ;TEST FOR FATAL ERROR
454 001320* 001463 REC 1S ;IF NONE
455 001322* 046767 176734 176724 RLC TEMP3,DVISAV ;DROP THAT BIT MAP
456 001336* 062767 000020 176720 ADD #20,TEMP1 ;UPDATE TO NEXT BUS ADDRESS
457 001336* 005267 176716 1S: INC TEMP3 ;UPDATE MAP INDICATOR
458 001342* 006367 176714 ASE TEMP3 ;MOVE THE TEST BIT
459 001346* 061767 000100 176706 CME #D00,TEMP3 ;TEST IF ANY MORE POSSIBLE MAPS
460 001364* 001351 RNE MAPRTB ;IF MORE
461 001356* 005767 176722 TST DVISAV ;TEST IF ANY MAPS LEFT ?
462 001362* 001067 BNE MAPPTA ;IF SOME
463 001374* 104410 000000* ENDS,REGIN ;DROP THE MODULE BECAUSE ALL MAPS HAVE ERRORED
464 001370* 104413 000000* MAPRTA: ENDS,REGIN ;SIGNAL END OF ITERATION.
465 001374* 006167 176710 JMP START ;MONITOR SHALL TEST END OF PASS

```

```

468 ;NOW LOAD THE BUS ADDRESSES FROM "TEMP1" AND START TESTING
469 TSTMAP: MOV TEMP1,VGCSR ;LOAD BUS ADDRESSES
470 MOV TEMP1,VGPCR
471 MOV TEMP1,VGBUF
472 MOV TEMP1,VGCOLR
473 MOV TEMP1,VGCCOLR
474 ADD #2,VGPCR
475 ADD #4,VGBUF
476 ADD #1,VGCCOLR
477 MOV VCCSP,CSRA ;LOAD BUS ADDRESS IN CASE OF ERROR
478 CLF ;CLEAR DROP MODULE FLAG
479
480 ;TEST THAT "EXPAND" BIT CAN BE SET
481 TSTEXP: MOV #BIT11,ASTAT ;LOAD THE EXPECTED
482 MOV ASTAT,@VGCSR ;LOAD THE REGISTER
483 MOV @VCCSP,ACSR ;READ THE REGISTER
484 BIC #11360,ACSR ;MASK TO BITS
485 CMP ASTAT,ACSR ;COMPARE THE EXPECTED TO RCVD
486 BEQ TSTMON ;BIT STUCK
487 MOV #25,ERRTYP ;BIT STUCK
488 ***** ;*****
489 HRDEFS,REGIN,NULL ;"EXPAND" BIT FAILED TO SET
490 ***** ;*****
491 INC FATAL ;SET DROP MODULE FLAG
492
493 ;TEST THAT "MONO" BIT CAN BE SET
494 TSTMCR: MOV #BIT10,ASTAT ;LOAD THE EXPECTED
495 MOV ASTAT,@VGCSR ;LOAD THE REGISTER
496 MOV @VCCSP,ACSR ;READ THE REGISTER
497 BIC #11360,ACSR ;MASK TO BITS
498 CMP ASTAT,ACSR ;COMPARE THE EXPECTED TO RCVD
499 BEQ TSTMON ;BIT STUCK
500 MOV #25,ERRTYP ;BIT STUCK
501 ***** ;*****
502 HRDEFS,REGIN,NULL ;"MONO" BIT FAILED TO SET
503 ***** ;*****
504 INC FATAL ;SET DROP MODULE FLAG
505
506 ;TEST THAT "MAP ENABLE" BIT CAN BE SET
507 MAPTST: MOV #BIT0,ASTAT ;LOAD THE EXPECTED
508 MOV ASTAT,@VGCSR ;LOAD THE REGISTER
509 MOV @VCCSP,ACSR ;READ THE REGISTER
510 BIC #11360,ACSR ;MASK TO BITS
511 CMP ASTAT,ACSR ;COMPARE THE EXPECTED TO RCVD
512 BEQ TSTMON ;BIT STUCK
513 MOV #25,ERRTYP ;BIT STUCK
514 ***** ;*****
515 HRDEFS,REGIN,NULL ;MAP ENABLE FAILED TO SET
516 ***** ;*****
517 INC FATAL ;SET DROP MODULE FLAG
518

```

```

510 001704 012767 000002 176172 ;TEST THAT "ORGIN POINT" BITS CAN BE SET
511 001712 012777 176166 176320 ;STORE: MOV #R17,ASTAT ;LOAD THE EXPECTED
512 001720 012767 176314 176154 1S: MOV #ASTAT,@VCCSR ;LOAD THE REGISTER
513 001724 042767 171360 176146 R1C #171360,ACSR ;READ THE REGISTER
514 001734 022767 176144 176140 CMP #ASTAT,ACSR ;MASK TO BITS
515 001744 012767 000025 176134 BEQ #25,ERRTYP ;COMPARE THE EXPECTED TO RCVD
516 001752 104405 000000 000000 ;***** ;BIT STUCK
517 001760 005267 176322 ;***** ;ORGIN POINT" BIT FAILED TO SET
518 001764 042767 176114 176114 ;***** ;ORGIN POINT" BIT FAILED TO SET
519 001774 022767 000020 176106 2S: INC #20,ASTAT ;SET DROP MODULE FLAG
520 001776 001345 ;UPDATE EXPECTED VALUE OF "ORGIN POINT"
521 ;TEST IF VALID VALUE FOR "ORGIN"
522 ;OR IF MORE TO TEST
523
524 ;TEST THAT CLEAR LOW BYTE DOES NOT CLEAR HIGH BYTE
525 ;STL0: MOV #R17,RIT10 ;LOAD EXPECTED READ VALUE
526 002000 012767 006400 176076 ;***** ;LOAD STATUS REGISTER
527 002006 012777 006417 176224 ;***** ;LOAD STATUS REGISTER
528 002014 105077 176220 ;***** ;READ THE REGISTER
529 002020 012767 176214 176054 CLR #0,@VCCSR ;CLEAR LOW BYTE
530 002024 042767 170360 176046 R1C #170360,ACSR ;READ THE REGISTER
531 002034 022767 176044 176040 CMP #ASTAT,ACSR ;MASK TO BITS
532 002044 001410 000025 176034 BEQ #25,ERRTYP ;TEST IF SAME
533 002044 012767 000025 176034 ;***** ;BIT STUCK
534 002052 104405 000000 000000 ;***** ;CLEARING LOW BYTE OF STATUS CHANGED THE HIGH BYTE
535 002056 042767 176222 ;***** ;CLEARING LOW BYTE OF STATUS CHANGED THE HIGH BYTE
536 002060 005267 176222 ;***** ;SET DROP MODULE FLAG
537
538 ;TEST THAT CLEAR HIGH BYTE DOES NOT CLEAR LOW BYTE
539 ;STHG: MOV #R17,ASTAT ;LOAD EXPECTED READ VALUE
540 002064 012767 006017 176012 ;***** ;LOAD STATUS REGISTER
541 002072 012777 006417 176140 ;***** ;LOAD STATUS REGISTER
542 002076 012767 176134 ;***** ;MAKE ADDRESS
543 002104 005200 ;***** ;HIGH BYTE
544 002106 105010 176124 175764 CLR #0,(R0) ;CLEAR HIGH BYTE
545 002110 012767 176124 175756 MOV #0,@VCCSR,ACSR ;READ REGISTER
546 002114 042767 175754 175750 R1C #170360,ACSR ;MASK OUT BITS
547 002124 022767 175754 175750 CMP #ASTAT,ACSR ;COMPARE EXPECT TO RCVD
548 002134 001410 000025 175744 BEQ #R0,R0 ;COMPARE EXPECT TO RCVD
549 002134 012767 000025 175744 ;***** ;BIT STUCK
550 002142 104405 000000 000000 ;***** ;CLEARING THE HIGH BYTE OF STATUS CHANGED THE LOW BYTE
551 002142 104405 000000 000000 ;***** ;CLEARING THE HIGH BYTE OF STATUS CHANGED THE LOW BYTE
552 002150 005267 176132 ;***** ;SET DROP MODULE FLAG
553

```

```

564 ;READ-WRITE TEST OF THE LOW 3 BITS OF THE MAP P.C.
565 ;RDWRT: CLR #ASTAT ;CLEAR EXPECTED READ VALUE
566 002154 005067 175724 ;CLEAR WRITE VALUE
567 002160 005067 176040 1S: CLR #STEMP
568 002164 104407 000000 ;TEMPORARY RETURN TO MONITOR
569 002164 104407 000000 ;THEN CONTINUE AT NEXT INSTRUCTION.
570 002170 012777 176024 176040 ;LOAD BIT MAP P.C.
571 002174 012777 176024 176040 MOV #STEMP,@VCCSR ;LOAD BIT MAP P.C.
572 002180 012767 176034 175672 ;READ STATUS (BIT MAP)
573 002186 042767 176114 175664 R1C #176114,ACSR ;MASK TO UNUSED BITS
574 002194 022767 175662 175656 CMP #ASTAT,ACSR ;TEST IF SAME
575 002204 001410 000025 175652 BEQ #25,ERRTYP ;TEST IF SAME
576 002204 012767 000025 175652 ;***** ;BIT STUCK
577 002234 104405 000000 000000 ;***** ;BIT MAP P.C. DID NOT LOAD CORRECTLY
578 002234 104405 000000 000000 ;***** ;BIT MAP P.C. DID NOT LOAD CORRECTLY
579 002242 005267 176040 ;***** ;BIT MAP P.C. DID NOT LOAD CORRECTLY
580 002246 005267 006320 175630 2S: INC #20,ASTAT ;SET DROP MODULE FLAG
581 002254 005267 175744 ;UPDATE EXPECTED READ VALUE
582 002254 005267 175744 ;UPDATE EXPECTED WRITE VALUE
583 002260 022767 000010 175736 CMP #10,STEMP ;TEST FOR FIRST NON-VALID
584 002266 001336 ;BRANCH IF STILL OK VALUES.
585
586 002270 005767 176012 ;TEST IF AN ERROR OCCURRED
587 002274 001402 ;RR IF NOT
588 002276 001617 177012 JMP #MAPRET ;RETURN NOW - DONT BOTHER WITH DISPLAY STUFF

```

```

589 ;TEST THE "EXPAND" FUNCTION
590 002302 004767 000444 VISEXP: JSR PC,CLRMAP ;LOAD MAX INTENSITY INTO LOCATION 0
591 00230E 012777 000017 MOV #17,@VGCCLR ;EXPAND OFF AND ENABLE BIT MAP
592 ;
593 002314 012777 002400 175716 MOV #BIT10IBIT8,@VGCSPR ;SET THE "EXPAND" BIT
594 002322 004567 000612 JSR RS,DELAY
595 100.
596 ;
597 002330 052777 004000 175702 BIS #BIT11,@VGCSPR ;SET THE "EXPAND" BIT
598 002336 004567 000578 JSR RS,DELAY
599 100.
600 ;TEST THE "ORGIN" FUNCTION
601 ;
602 ;
603 002344 004767 000402 VISORG: JSR PC,CLRMAP ;CLEAR THE MAP
604 ;
605 002350 012777 000017 MOV #17,@VGCCLR ;LOAD MAX INTENSITY INTO TABLE LOC. 0
606 002356 012777 002400 175654 MOV #BIT10IBIT8,@VGCSPR ;ENABLE THE BIT MAP
607 002364 005067 175634 CLR STMP
608 ;
609 002370 116777 175630 175642 1S: MOVW STMP,@VGCSPR ;LOAD ORGIN REGISTER
610 002376 004567 000536 JSR RS,DELAY
611 100.
612 002402 000144 INCA STMP ;UPDATE THE ORGIN POSITION
613 002404 105267 175614 175606 CMPR #20,STMP ;TEST FOR A NON-VALID ORGIN
614 002416 122767 000020 BNE 1S ;RR IF A VALID ORGIN
615 ;
616 ;INTENSITY LEVEL TEST-MONOCROME
617 ;
618 ;
619 002420 004767 000326 VISINT: JSR PC,CLRMAP ;CLEAR THE BIT MAP
620 002424 004767 000644 JSR PC,L0DSEQ ;LOAD TWO WORDS PER LOOK TABLE LOCATION
621 002430 005067 000017 CLP 2S ;CLEAR STARTING INTENSITY LEVEL
622 002434 012777 000400 MOV #BIT11IBIT10IBIT8,@VGCSPR ;SET MONOCROME + ENABLE MAP
623 002442 004567 000734 1S: JSR RS,L0DSTA ;LOAD SEQUENCE
624 002446 000000 2S: C ;INDEX INTO DATA VALVE TABLE
625 002450 003060 ;
626 ;
627 002452 004567 000462 JSR RS,DELAY
628 100.
629 002456 000144 ADD #1,2S ;UPDATE INDEX INTO TABLE FOR LOCATION 0'S VALUE
630 002460 052767 000001 177760 BNE 1S ;TEST IF LAST VALUE FOR MONOCROME
631 002474 001362 000020 BNE 1S ;RR IF MORE LEVELS FOR LOCATION 0

```

```

632 ;COLOR LEVEL TEST - COLORCROME
633 002476 004767 000250 VISCLT: JSR PC,CLRMAP ;ENSURE CLEAR MAP
634 002482 004767 000666 JSR PC,L0DSEQ ;LOAD TWO WORDS PER LOOKUP TABLE LOC.
635 002486 005067 000017 CLP 2S ;CLEAR INDEX POINTER
636 002494 004567 000666 CLP @VGCSPR ;CLEAR STATUS
637 002502 004567 000666 1S: JSR RS,L0DSTA ;LOAD DATA INTO TABLE
638 2S: 0 ;
639 002524 003100 ;
640 002528 012777 004400 175504 MOV #BIT11IBIT8,@VGCSPR ;ENABLE MAP
641 002534 004567 000400 JSR RS,DELAY
642 002540 000144 100.
643 002542 052767 000001 177752 ADD #1,2S ;UPDATE LOC 0 INDEX INTO TABLE
644 002546 052767 000020 177744 BNE 1S ;TEST IF FINISHED FOR LOC 0
645 002554 004567 175454 CLP @VGCSPR ;DISABLE MAP
646 002560 005077 ;
647 ;
648 ;
649 002564 000167 17524 JMP MAPPET

```



```

746
747
748
749
750 003140* 012567 000064
751 003144* 012777 003190 175076
752 003152* 005077 175074
753 003156* 057777 020000 175042
754 003164* 104400 000000
755 003170* 042777 020000 175030
756 003176* 000004 000000 003204
757
758 003204* 005367 000020
759 003210* 001362
760 003212* 042777 020000 175006
761 003220* 016777 175026 175022
762 003226* 000205
763
764
765 003236* 000000
766
767
768
769 003232* 066767 000034 000030
770 003240* 005567 000026
771 003244* 066767 000020 000020
772 003248* 005567 000022
773 003256* 104407 000000
774 003262* 104407 000000
775 003266* 000207
776
777 003270* 176543
778 003272* 123456
779

```

```

.SBTTL DELAY SUBROUTINE -- WAIT FOR "VERTICAL SYNC INTERRUPT"
DELAY: MOV (R5)+,10$ ;LOAD THE ARG
4$: MOV #3,0VCINT ;LOAD INTR. RETURN VECTOR
CLP 0VCINT1
6$: RTS,REGIN ;ENABLE INTR.
3$: BIC #BIT13,@VCCSR ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
;DISABLE INTR.
;-----
;PIRQS,REGIN,5$ ;QUEUE UP TO CONTINUE AT 5$ AND R11
;-----
5$: DEC 10$ ;FINISHED DELAY ?
BNE 6$ ;BR IF NOT
2$: BIC #BIT13,@VCCSR ;CLEAR INTR. ENABLE
MOV 0VCINT1,0VCINT ;RESTORE INTR. VECTOR
RTS R5 ;EXIT

10$: C
;RANDOM NUMBER GENERATOR
SRAND: ADD $LONUM,$SHINUM ;ADJUST DATA
ADC $LONUM
ADD $SHINUM,$LONUM
ADC $SHINUM
RPEAKS,REGIN ;TEMPORARY RETURN TO MONITOR....
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
RTS PC ;EXIT

SHINUM: 176543
LONUM: 123456

```

```

780 ;LOAD MAP WITH TWO WORDS OF THE SAME VALUE
781 ; INTO THE ENTIRE BITMAP STARTING AT LOC. 0
782
783 003274* 010046
784 003276* 010146
785 003300* 005677 174736
786 003304* 012971 174774
787 003310* 006201
788
789 003312* 065606
790 003314*
791 003314* 104407 000000
792 003320* 104407 000000
793 003324* 042777 174710
794 003330* 104371
795 003332* 016077 174706
796 003336*
797 003336* 104407 000000
798 003340* 104407 000000
799 003346* 105777 174666
800 003352* 104371
801 003354* 016077 174664
802
803 003360* 005301
804 003362* 104404
805 003364* 062706 010421
806 003370* 104351
807 003372* 000747
808
809 003374* 012601
810 003376* 012600
811 003400* 006207
812

```

```

LONSEC: MOV R0,-(SP) ;SAVE R0
MOV R1,-(SP) ;SAVE R1
CLP @VGPC ;ENSURE CLEAR PC
MOV #RUMPIX,R1 ;LOAD COUNTER
ASF R1 ;ADJUST COUNTER

1$: CLP R0 ;START WITH PIXEL VALUE OF 0
2$: BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
TSTB @VGCSP ;ENSURE MAP IS READY
RPL 2$
MVC R0,@VGBUF ;LOAD A PIXEL WORD

3$: BREAKS,REGIN ;TEMPORARY RETURN TO MONITOR....
BREAKS,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
TSTB @VGCSP ;ENSURE MAP IS READY
RPL 3$
MVC R0,@VGBUF ;LOAD 2ND PIXEL WORD

DEC R1 ;FINISHED ALL PIXELS ?
RMI 4$ ;BR IF FINISHED
ADD #10421,R0 ;UPDATE PIXEL DATA TO NEXT L.U.T. ADDRESS
RCC 2$ ;BR IF MORE ADDRESSES TO LOAD
RP 1$ ;RR TO RESET PIXEL DATA

4$: MOV (SP)+,R1 ;RESTORE R1
MOV (SP)+,R0 ;RESTORE R0
RTS PC ;EXIT

```


