

1

.REM.

IDENTIFICATION

PRODUCT CODE: AC-E978K-MC

PRODUCT NAME: CXTMBK0 TM02,03/TU16,TE16 MO

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, 1979 DIGITAL EQUIPMENT CORPORATION

1.

ABSTRACT

THE TMB IS AN IOMODX MODULE THAT CAN EXERCISE UP TO 8 TM02 OR TM03 CONTROLLERS WITH UP TO 8 TUI6 OR TE16 (*OR* UP TO 4 TU77) TAPE SLAVES ON EACH CONTROLLER. EACH SLAVE WILL BE EXERCISED BY DOING A WRITE, READ REVERSE, READ FORWARD AND A IN-COKE COMPARE. THIS SEQUENCE OF FUNCTIONS WILL BE DEFINED AS A CYCLE. AN "END OF PASS" WILL BE REACHED AFTER 500 CYCLES. WHEN "END OF PASS" IS REACHED, A "TAPE MARK" FUNCTION IS EXECUTED BEFORE CONTINUING. WHEN THE MODULE DETECTS A "SOFT ERROR", A SOFT ERROR COUNTER WILL BE INCREMENTED. IF THERE ARE 3 CONSECUTIVE SOFT ERRORS, IT WILL BE CONSIDERED A HARD ERROR AND AN ERROR MESSAGE WILL BE TYPED OUT.

TM03 COMPATABILITY IS THE TMB DEFAULT MODE FOR DENSITY SWITCHING. IN OTHER WORDS, TMB WILL SWITCH BETWEEN 1600 AND 800 BPI ONLY IF THE SLAVE IS AT BOT. IF THE MODULE IS TO TEST A TM02, BIT 5 OF SR1 SHOULD BE SET TO A 1 TO ALLOW DENSITY SWITCHING AFTER EACH WRITE-READ REV-READ FWD CYCLE, REGARDLESS OF TAPE POSITION.

WHEN A "SOFT ERROR" IS DETECTED, THE ERROR WILL NOT BE REPORTED UNTIL END-OF-TAPE IS REACHED. THE EXCEPTIONS ARE:

- (1) IF SR1 BIT2 IS SET, ALL SOFT ERRORS ARE REPORTED WHEN THEY OCCUR
- (2) IF THE REPLY LIMIT IS EXCEEDED ON SOFT ERRORS, AN ERROR MESSAGE WILL BE PRINTED TO INDICATE THIS. TWO COUNTERS ARE KEPT, ONE CONTAINING THE NUMBER OF "SOFT ERRORS" THAT OCCURRED DURING THE LAST PASS ON THE TAPE AND THE OTHER KEEPING TRACK OF THE TOTAL NUMBER OF SOFT ERRORS ON ALL PASSES OF THAT TAPE. A SOFT ERROR IS INDICATED BY ANY OF THE SOFT ERROR BITS SET IN MTER (15,10,7,6) AND NO HARD ERROR BITS SET.

THE COUNTER CONTAINING THE NUMBER OF ERRORS PER PASS WILL BE ZEROED EVERY TIME THE TAPE IS REWOUND. IF 20 "SOFT ERRORS" ARE DETECTED DURING ONE PASS OF TAPE, THE TUI6 SLAVE WILL BE DROPPED.

2. REQUIREMENTS

HARDWARE: AT LEAST 1 TUI6 SLAVE ON A TM02 CONTROLLER
OR
AT LEAST 1 TE16 SLAVE ON A TM03 CONTROLLER

STORAGE: TMB REQUIRES:

- 1. DECIMAL WORDS: 1989
- 2. OCTAL WORDS: 03705
- 3. OCTAL BYTES: 7612

3. PASS DEFINITION

ONE PASS CONSISTS OF 500 CYCLES OF A WRITE, READ REVERSE, READ FORWARD, AND DATA CHECK OPERATIONS.

4. EXECUTION TIME

ONE PASS OF TMB TAKES APPROXIMATELY 1 MINUTE

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS: DEVADR: 172440 VECTOR: 224 BR1:5
REQUIRED PARAMETERS: NONE

6. DEVICE/OPTION SETUP

POWER UP ALL DRIVES (SLAVES) AND MAKE READY AT LOAD POINT

7. MODULE OPERATION TEST SEQUENCE

- A. SET UP REGISTER ADDRESSES
- B. DO A RH RESET
- C. SETUP ALL SRI OPTIONS
- D. CHECK FOR ALL AVAILABLE DRIVES
- E. CHECK FOR, AND REWIND ALL SLAVES
- F. DO A WRITE, READ REVERSE, READ FORWARD, DATA CHECK
ON EACH SLAVE ON THIS DRIVE
- G. REPORT ANY ERRORS
- H. IF DONE ALL SLAVES ON THIS DRIVE, POINT TO NEXT DRIVE
- I. IF NOT TIME FOR EOP, GO TO F
- J. DO EOP CALL AND GO TO F

8. OPERATION OPTIONS

SRI

BIT0 = 0: IF RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, THE
FUNCTION IS ABORTED AND TESTING IS CONTINUED.

BIT0 = 1: IF RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, THE ERROR
IS CONSIDERED FATAL AND THE SLAVE IS DROPPED.

BIT1 = 0: MODULE WILL EXERCISE ALL SLAVES FOUND ON THE SYSTEM.

BIT1 = 1: OPERATOR MUST SELECT THE SLAVES IN ADDITION TO
THE DRIVES (DVID1) TO BE PEXERCISED. SEE SECTION 10,
FOR DETAILS.

BIT2 = 0: REPORT A SOFT ERROR AS AN ERROR ONLY IF THE
RETRY LIMIT HAS BEEN EXCEEDED.

BIT2 = 1: REPORT ALL SOFT ERRORS AS THEY OCCUR.

BIT3 = 0: USE PROGRAM DEFAULT RETRY LIMITS.

BIT3 = 1: USE ALTERNATE (ENGINEERING) RETRY LIMITS. THESE THREE
LOCATIONS, WLM, RRLM, PRLM, MAY BE CHANGED AND USED
IN CONJUNCTION WITH THIS SRI SETTING.

BIT4 = 0: PRINT SOFT ERROR SUMMARY AT EOT.

BIT4 = 1: DO NOT PRINT SOFT ERROR SUMMARY.

BIT5 = 0: ALTERNATE DENSITY ONLY EVERY CYCLE THROUGH BOT
(FOR TMB3 COMPATABILITY)

BIT5 = 1: ALTERNATE DENSITY EVERY WRITE, RD-REV, RD-FWD, DATCK CYCLE.

BIT6 = 0: IFST 1600 B.P.I.

BIT6 = 1: DO NOT TEST 1600 B.P.I.

BIT7 = 0: TEST 800 B.P.I.

BIT7 = 1: DO NOT TEST 800 B.P.I.

9. NON-STANDARD PRINTOUTS

ERROR MESSAGES DUMP THE CONTENTS OF THE 14 REGISTERS (16 FOR RH/70)
IN THE FOLLOWING ORDER:

CS1 WC BA FC CS2 DS ER AS
CK DR MR DT SN TC RAE CS3
CYCLECNT

THE CYCLE COUNTER LOCATION IN THE ERROR MESSAGE CONTAINS
THE CYCLE COUNT AT THE TIME OF THE ERROR. THIS COUNT IS INITIALIZED
EVERY TIME THE PROGRAM STARTS WRITING AT THE BEGINNING OF TAPE.
THIS SHOULD AID IN NOTING ANY BAD SPOTS ON A TAPE.

10. SELECTING SLAVES

WHEN BIT 1 IS SET IN SR1, THE OPERATOR MUST SELECT THE
SLAVES TO BE TESTED. THE SLAVE TABLE THAT MUST
BE MODIFIED STARTS AT LOCATION 252 (SEL).
THESE WORDS WILL INDICATE THE SLAVES WANTED FOR EACH DRIVE.
TO SELECT THE SLAVES FOR A PARTICULAR DRIVE, A "1"
MUST BE SET IN THE CORRESPONDING BIT POSITION FOR THE
DESIRED SLAVE. AN EXAMPLE: IF SLAVES "0" AND
"2" ARE TO BE SELECTED ON DRIVE 6, A 5 WOULD BE PLACED IN
LOCATION SEL (APC 252).
THE NEXT 7 LOCATIONS AFTER "SEL" WOULD INDICATE THE SLAVES
DESIRIED FOR DRIVES 1 THROUGH 7 RESPECTIVELY.
THERE ARE 8 WORDS IN THIS TABLE, ONE FOR EACH DRIVE. ALSO,
REMEMBER YOU MUST PESET DVID1 IF YOU HAVE MORE THAN 1 DRIVE.

```

000000* IOMDX <TMK > 172440,224,5,0,0,500,130,BUFIN,256,,1224,
000000* MODULF 150000,TMK ,172440,224,5,0,0,500,130,BUFIN,256,,1224,
; TITLE TMK DEC/X11 SYSTEM EXERCISE MODULE
; DEXCOM VERSION 6 23-MAY-78
; LIST RTN
;*****
000000* REGIN;
000000* 046524 045502 040 MODNAM: ASCII /TMK / ;MODULF NAME,
000005* 000 XFLAG: BYTE OPEN ;USED TO KEEP TRACK OF KBUFF USAGE
000006* 172440 ADDR: 172440+0 ;1ST DEVICE ADDR.
000010* 000224 VECTOR: 224+0 ;1ST DEVICE VECTOR,
000012* 240 BR1: BYTE PPTY5+0 ;1ST BR LEVEL,
000013* 000 BP2: BYTE PPTY0+0 ;2ND BR LEVEL,
000014* 000001 DVID1: 0+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* 150000 STAT: 150000 ;STATUS WORD,
000030* 002104 INIT: START ;MODULE START ADDR,
000032* 000252 SPOIN: MODSP ;MODULE STACK POINTER,
000034* 000000 PASCNT: 0 ;PASS COUNTER,
000036* 000500 ICONT: 500 ;# OF ITERATIONS PER PASS=500
000040* 000000 ICONT1: 0 ;LOC TO COUNT ITERATIONS
000042* 000000 SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044* 000000 HRCNT1: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046* 000000 SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050* 000000 HRPAS1: 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* 000000 CONF1: ;RESERVED FOR MONITOR USE
000060* 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000062* 000000 RES2: 0 ;RESERVED FOR MONITOR USE
000064* 000000 SVR0: OPEN ;LOC TO SAVE R0,
000066* 000000 SVR1: OPEN ;LOC TO SAVE R1,
000070* 000000 SVR2: OPEN ;LOC TO SAVE R2,
000072* 000000 SVR3: OPEN ;LOC TO SAVE R3,
000074* 000000 SVR4: OPEN ;LOC TO SAVE R4,
000076* 000000 SVR5: OPEN ;LOC TO SAVE R5,
000080* 000000 SVR6: OPEN ;LOC TO SAVE R6,
000082* 000000 CSRA: OPEN ;ADDR OF CURRENT CSR,
000084* 000000 SBADR: ;ADDR OF GOOD DATA, OR
000086* 000000 ACSR: OPEN ;CONTENTS OF CSR,
000088* 000000 WASADP: ;ADDR OF BAD DATA, OR
000090* 000000 ASTAT: OPEN ;STATUS REG CONTENTS,
000092* 000000 ERR1YP: ;TYPE OF ERROR
000094* 000000 ASD: OPEN ;EXPECTED DATA,
000096* 000000 AWAS: OPEN ;ACTUAL DATA,
000098* 002414 PSTRT: PFSTRT ;RESTART ADDRESS AFTER END OF PASS
000100* 000000 WDOT: OPEN ;WORDS TO MEMORY PER ITERATION
000102* 000000 WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
000104* 000000 INTR: OPEN ;# OF INTERRUPTS PER ITERATION
000106* 000130 IDNUM: 130 ;MODULE IDENTIFICATION NUMBER=130
000108* 001016 RBUFVA: BUFIN ;READ BUFFER VIRTUAL ADDRESS
000110* 000000 RBUFFA: OPEN ;READ BUFFER PHYSICAL ADDRESS
    
```

```

000130* 000000 RBUFFA: OPEN ;READ BUFFER EA BITS
000132* 000400 RBUFSZ: 256 ;SIZE OF THE READ BUFFER
000134* 000000 WBUFFA: OPEN ;WRITE BUFFER PHYSICAL ADDRESS
000136* 000000 WBUFEA: OPEN ;WRITE BUFFER EA BITS
000140* 002000 WBUFRQ: 1024 ;WRITE BUFFER SIZE REQUESTED
000142* 000000 WBUFSZ: OPEN ;WRITE BUFFER SIZE AVAILABLE
000144* 000000 CDWDC1: OPEN ;CDATA/DATCK ENRROR COUNT
000146* 000000 CDWDC2: OPEN ;CDATA/DATCK WORD COUNT
000150* 000000 FREE: OPEN ;RESERVED FOR FUTURE USE
;*****
000152* ;REPT SPSI7 ;MODULE STACK STARTS HERE,
;LIST
;WORD 0
;LIST
;ENDH
MODSP;
;*****
    
```

```

285 000252* 000000    TABLE:
286 000252* 000000    PHCS1: 0
287 000254* 000000    ;CONTROL STATUS REG 1 *
288 000256* 000000    NTWC: 0    ;WORD COUNT REG *
289 000260* 000000    RBBA: 0    ;BUS ADDRESS *
290 000262* 000000    NTFC: 0    ;FRAME COUNT *
291 000264* 000000    PHCS2: 0    ;CONTROL STATUS 2 *
292 000266* 000000    MDS: 0     ;DEVICE STATUS *
293 000270* 000000    MTEP: 0    ;ERROR REG. *
294 000272* 000000    MTAS: 0    ;ATTENTION SUMMARY *
295 000274* 000000    MTCY: 0    ;CHECK CHARACTERS *
296 000276* 000000    MTDB: 0    ;DATA BUFFER *
297 000300* 000000    MTRR: 0    ;MAINTENANCE REG *
298 000302* 000000    MTDT: 0    ;DRIVE TYPE *
299 000304* 000000    MTSN: 0
300 000306* 000000    MTRC: 0
301 000310* 000000    PHRAE: 0
302 000312* 002066*    PHCS3: 0
303 000314* 177777*    CYCCNT
304                                177777
305 000316* 000000    SEL: OPEN  ; DRIVE 0
306 000320* 000000    OPLN  ; DRIVE 1
307 000322* 000000    OPEN  ; DRIVE 2
308 000324* 000000    OPLN  ; DRIVE 3
309 000326* 000000    OPEN  ; DRIVE 4
310 000330* 000000    OPEN  ; DRIVE 5
311 000332* 000000    OPEN  ; DRIVE 6
312 000334* 000000    OPEN  ; DRIVE 7
313 000336* 000000    SLACT: 0
314
315 000340* 000000    PDEA: 0
316 000342* 000000    WREA: 0
317 000344* 000000    CLK: 0
318 000346* 000000    CNT: 0
319 000350* 000000    DENSTY: 0    ; HOLDS DENSITY COMMAND
320 000352* 000000    FUNC: 0
321 000354* 000000    ZERO: 0
322 000356* 000000    DVICE: 0    ;COPY OF DVID1
323 000360* 000000    DRIVE: 0    ;WORKING COPY OF DVID1
324 000362* 000000    DRYVE: 0    ;RE-ERRANT DRIVE COUNTER
325 000364* 000000    WB: 0
326 000366* 000000    WBSAV: 0
327 000370* 000000    RB: 0
328 000372* 000000    RBSAV: 0
329 000374* 000000    SLAVNO: 0
330 000376* 000000    FOINT: 0
331 000400* 000000    HOLD: 0
332 000402* 000000    HLK: 0
333 000404* 000000    SOFTLC: 0    ;SOFT ERROR TABLE LOCATION
334 000406* 000000    SOFTTO: 0    ;SOFT ERROR TOTAL PER SLAVE
335 000410* 000000    BUFEND: OPEN ; ADDRESS OF THE END OF THE READ BUFFER
336 000412* 000000    BUFEPA: OPEN ; END OF READ BUFFER PHYSICAL ADDRESS
337 000414* 000000    BUFEFA: OPEN ; END OF READ BUFFER EA BITS
338 000416* 000200*    SOFTER: ,BLK# 128, ;SOFT ERROR COUNTER TABLE
339 000418* 000400*    BUFIN: ,BLK# 256, ; INPUT/READ BUFFER
340 000416* 000000*    TABLSF: ,BLK# 6    ;PUT SOFT ERROR NUMBER HERE IN ASCII.

```

```

341 002024* 000000
342 002026* 000000    TABLS1: ,BLK# 6    ;HAS TO BE ZERO TO END ASCII MESSAGE
343 002034* 000000
344 002036* 000000    TABLS2: ,BLK# 6
345 002044* 000000
346 002046* 000000    TABLS3: ,BLK# 6
347 002054* 000000
348
349
350 002056* 000000    ;DO NOT CHANGE THE ORDER OF THE NEXT 4 LOCATIONS
351 002060* 000000    PA18: 0
352 002062* 000000    XMEN: 0
353 002064* 000000    PA22: 0
354 002066* 000000    EA22: 0
355    CYCCNT: 0
356 002070* 000000    CLRDRY: 0    ;DRIVE READY TIMER
357
358 002072* 000
359 002073* 000
360 002074* 000
361 002075* 007
362 002076* 017
363 002077* 002
364 002100* 000
365 002101* 000
366 002102* 000
367 002103* 000
368
369    ;RIT 0 IS EOT BIT
370    ;RIT 1 IS REWIND HIT
371    ;RIT 2 IS HARD ERROR BIT
372    ;RIT 3 IS ALL DRIVES DONE BIT

```

```

373 002104* 012767 002000 176004 START: MOV #1024,,WDFR ;1024 WORDS FROM MEM/ITERATION
374 002112* 012767 000400 175774 MOV #256,,WDTO ;256 WORDS TO MEM/ITERATION
375 002120* 012767 000007 175772 MOV #7,INTR ;7 INTERRUPTS PER/ITERATION
376 002126* 004567 004144 JSR R5,CLTAR ;CLEAN SOFT ERROR TABLE AND SLAVE TABLE
377 ; IF OPERATOR DID NOT SELECT SLAVES
378 ; SELECT DENSITY TO START AT 1600 IF POSSIBLE. IF 1600 BPI IS
379 ; INHIBITED, THEN 800 BPI--IF 800 BPI IS ALSO INHIBITED, DROP THE
380 ; MODULE
381 002132* 032767 000100 175656 BIT #BIT6,SR1 ;IF 1600 BPI IS ENABLED THEN
382 002140* 001004 BNE 206 ;BEGIN
383 002142* 012767 002300 176200 MOV #2300,DENSITY ; SET 1600 BPI
384 002150* 000415 BR 216 ;END
385 002152* ;ELSE
386 002152* 032767 000200 175636 206: FIT #BIT7,SP1 ; IF 800 BPI IS ENABLED THEN
387 002160* 001004 BNE 226 ; BEGIN
388 002162* 012767 001300 176160 MOV #1300,DENSITY ; SET 800 BPI
389 002170* 000405 BR 236 ; END
390 002172* 226: ; ELSE (* NEITHER ENABLED -- THAT'S IT *)
391 ; BEGIN
392 002172* 104403 000000 007040* MSGS,BEGIN,NODENS ;ASCII MESSAGE CALL WITH COMMON HEADER
393 002200* 000167 000632 JMP FINI ; DROP MODULE
394 ; END
395 002204* 236: ;
396 002204* 216: ;
397 002204* 005067 176136 CIR CNT ;ZERO PASS COUNTER
398 002210* 005067 177652 CLR CYCNT ;CLEAR CYCLE TAPE MARK COUNTER
399 002214* 105067 177663 CLRB FLAG ;CLEAR ALL FLAGS
400 002220* 012767 000001 176150 MOV #1,POINT ;SET SLAVE POINTER
401 002226* 005067 176142 CLP SLAVNO ;CLEAR DROP POINTER
402 002232* 016767 175556 MOV DVID1,DVICE ;GET SELECTED DRIVES
403 002240* 016767 176112 MOV DVICE,DRIVE ;SAVE DRIVES
404 002246* 012767 177777 MOV #-1,DRIVE ;SET DEVICE COUNTER
405 002254* 004767 004344 JSR PC,SETUP ;SETUP REG ADDRESSES
406 002260* 004767 003144 JSR PC,PEZET ;RESET ALL REGISTERS
407 002264* 004567 004156 JSR R5,SWITCH ; FIND SR1 OPTIONS
408 002270* 122737 000006 000041 CMPB #6,0041 ; IF TM IS LOAD MEDIUM THEN
409 002276* 001021 RFE ;BEGIN
410 002300* 113700 000040 MOVB #40,R0 ; GET LOAD-DEVICE NUMBER
411 002304* 012701 000001 MOV #1,R1 ; INITIALIZE DRIVE POINTER
412 002310* 105700 16: TSTR R0 ; WHILE NOT POINTING AT LOAD-DRIVE DO
413 002312* 001403 REQ 126 ; BEGIN
414 002314* 006301 ASL R1 ; POINT TO NEXT DRIVE
415 002316* 105300 DECB R0 ; COUNT SHIFTS
416 002320* 000773 BR 16 ; END
417 002322* 130167 176030 126: BITB R1,DVICE ; IF LOAD-DRIVE SELECTED THEN
418 002326* 001405 REQ 136 ; BEGIN
419 002330* 140167 176022 RCR R1,DVICE ; DROP THE DEVICE
420 002334* 104403 000000 007014* MSGS,BEGIN,LDRIVE ;ASCII MESSAGE CALL WITH COMMON HEADER
421 ; END
422 002342* 136: ;END
423 002342* 005767 176010 TST DVICE ;IF NO DRIVES ARE SELECTED THEN
424 002346* 001002 BNE 36 ;BEGIN
425 002350* 000167 000462 JMP FINI ; DROP MODULE
426 ;END
427
428

```

```

429 ;HERE THE PROGRAM SELECTS ALL THE DRIVES, REWINDS ALL THE
430 ;SLAVES AND CLEARS THEM TO THEIR INIT STATE BEFORE CONTINUING.
431
432 002354* 004567 002234 36: JSR R5,SELDRV ;SELECT A DRIVE
433 002360* 132767 000010 177515 BITB #BIT3,FLAG ;ALL DRIVES DONE?
434 002366* 001007 RNE 46 ;YES
435 002370* 032767 000002 175420 BIT #BIT1,SR1 ;DOES OPERATOR WANT TO
436 ;SELECT THE SLAVES
437 002376* 001003 RNE 46 ;YES, ASSUME SLAVE SELECT TABLE IS SETUP
438 002400* 004567 002350 JSR R5,SLAV ;NO, FIND ALL SLAVES ON THIS DRIVE
439 002404* 000763 BK 36 ;AND CHECK FOR MORE DRIVES
440 002406* 004567 003320 46: JSR R5,REWIND ;REWIND ALL SLAVES
441 002412* 000404 BR PEST ; CONTINUE
442 ;*****
443
444
445
446 002414* 005767 175414 RESIRT: TST PASCNT ; SUPPORT FOR CSS DT03'S
447 002420* 001001 RNE RFST ; CONTINUE
448 002422* 000630 BF START ; NO PASSES, BEGIN AT START
449 002424* 004567 003116 RESI: JSR R5,BUFF ; GET READ BUFFER INFORMATION
450 002430* 016767 175470 175752 MOV #BUFVA,BUFEND ; GET START OF READ BUFFER
451 002436* 062767 000776 175744 ADD #510,,BUFEND ; COMPUTE THE END OF THE BUFFER
452 002444* 104415 000000 000410* GETPA,BEGIN, BUFEND ;GET PHYSICAL ADDRESS FROM 16-BIT BUFEND
453 002452* 006367 175736 ASL BUFEA
454 002456* 006367 175732 ASL BUFEA
455 002462* 006367 175726 ASL BUFEA
456 002466* 006367 175722 ASL BUFEA
457
458 002472* NEXT:
459 002472* 104414 000000* GWRBFS, BEGIN ;GET WRITE BUFFER INFORMATION
460 002476* 016767 175440 175660 MOV #WRBFSZ,WR ;SAVE WRITE SIZE BUFFER
461 002504* 016767 175654 175654 MOV #WR,WRSAV ;SAVE IT FOR WC
462 002512* 006767 175646 175644 ADD #0,WR ;GET FRAM COUNT
463 002520* 005467 175642 NEG WRSAV ;GET 2'S COMP OF WORD COUNT
464 002524* 005467 175634 NEG WR ;GET 2'S COMP OF FRAM COUNT
465 002530* 016767 175402 175604 MOV #WRUFEA,WREA ;GET FA BITS
466 002536* 006367 175600 ASL WRFA ;SHIFT EA BITS INTO
467 002542* 006367 175574 ASL WREA ;BITS 0,9
468 002546* 006367 175570 ASL WREA
469 002552* 006367 175564 ASL WREA
470 002556* 004567 002032 JSR R5,SELDFV ;SELECT A DRIVE NUMBER
471 002562* 005767 175570 TST DVICE ;TEST FOR ANY DRIVES LEFT
472 002566* 001523 BEQ FINI ;NO DRIVES, GO DROP MODULE
473 002570* 132767 000010 177305 BITB #BIT3,FLAG ;CYCLED THROUGH ALL DRIVES?
474 002576* 001406 REQ SSL ;NO, CONTINUE
475 002600* 005767 175532 TST SLACT ;YES, ANY SLAVES LEFT?
476 002604* 001514 BEQ FINI ;NO, GO DROP MODULE
477 002606* 004767 003374 JSR PC,SWAP,DENSITY ;GO SEE ABOUT SWAPPING DENSITY
478 002612* 000727 BP NEXT ;ON TO NEXT DRIVE
479 002614* 004567 003022 SSL: JSR R5,GTSLAV ;GO GET SLAVE NUMBER
480 002620* 103724 BCS NEXT ;IF ALL SLAVES DONE, ON TO NEXT DRIVE
481 002622* 004567 002410 JSR R5,READY ;IS DVICE READY?
482 002626* 103403 BCS 16 ;YES, CONTINUE
483 002630* 004567 002440 JSR R5,NOTRDY ;NO, GO WAIT
484 002634* 103767 ;SLAVE NOT READY, GET NEXT SLAVE

```



```

485 002636* 105067 177230 1S: CLR B TRY1 ;CLEAR RETHY COUNTERS
486 002642* 105067 177225 CLR B TRY2 ;
487 002646* 105067 177222 CLR B TRY3 ;
488
489 002652* 004567 000566 GO: JSR R5,WRITE ;WRITE DATA
490 002656* 103471 RCS RFRPY1 ;TRY AGAIN IF ERRORS
491 002660* 132767 000001 177215 BITB #BIT0,FLAG ; EOT ?
492 002666* 001410 BNE 1S ; NO, CONTINUE
493 002670* 142767 000001 177205 FICB #BIT0,FLAG ; YES, CLEAR EOT FLAG
494 002676* 004767 001470 JSR PC,EOT ; GO DO EOT HOUSEKEEPING
495 002702* 004767 003300 JSR PC,SWAP,DENSITY ;GO SEE ABOUT SWAPPING DENSITY
496 002706* 000671 BR NEXT ; CONTINUE
497
498 002710* 004567 000576 GOA: JSR R5,RDREV ;READ REVERSE
499 002714* 103504 RCS RETRY2 ;TRY AGAIN IF ERRORS
500
501 002716* 004567 000636 GOF: JSR R5,RDFWD ;READ FORWARD
502 002722* 103544 RCS RFRPY3 ;TRY AGAIN IF ERRORS
503 002724* 104412 000000* 000126* CDATA$,BEGIN,RBUPFA ; REQUEST FOR MONITOR TO CHECK DATA
504 002732* 002734* .+2 ; IF ERROR, CONTINUE
505
506 002734* 005267 175406 1S: INC CNT ;COUNT A CYCLE
507 002740* 005267 177122 INC CYCCNT ;COUNT TOTAL CYCLES PER TAPE
508 002744* 026767 175066 175374 CMP ICONT,CNT ;TIME FOR END OF PASS?
509 002752* 001422 BEO PASS ;YES
510 002754* 104413 000000* ENDT$,BEGIN ; SIGNAL END OF ITERATION,
511 ; MONITOR SHALL TEST END OF PASS
512 002760* 022767 000007 175406 CONTNU: CMP #7,SLAVRO ;ALL SLAVES DONE ON THIS DRIVE?
513 002766* 001007 BNE 1S ;NO, ON TO NEXT ONE
514 002770* 005067 175400 CLR SLAVNO ;YES, CLEAR UNIT NUMBER
515 002774* 012767 000001 175374 MOV #1,POINT ;RESET POINTER
516 003002* 000167 177464 JMP NEXT ;ON TO NEXT DRIVE
517 003006* 005267 175362 1S: INC SLAVRO ;INCREMENT UNIT NUMBER
518 003012* 006367 175360 ASL POINT ;MOVE POINTER
519 003016* 000676 BR SSL ;GO TRY ANOTHER SLAVE
520
521 003020* 004567 000402 PASS: JSR R5,TPMK ;GO DO A TAPE MARK
522 003024* 000240 NOP ;ERROR RETURN
523 003026* 005067 175314 CLR CNT ;YES CLEAR COUNT
524 003032* 104413 000000* ENDT$,BEGIN ; SIGNAL END OF ITERATION,
525 ; MONITOR SHALL TEST END OF PASS
526
527 003036* FINI:
528 003036* 104410 000000* ENDS$,BEGIN ; DROP THE MODULE
529 ;
530 -----
531
532 ;ENTERED BY AN ERROR WHILE WRITING
533 PTRY1: INCB TRY1 ; COUNT A RETRY
534 003042* 105267 177024 BITB #BIT2,FLAG ; HARD ERROR ?
535 003046* 132767 000001 177027 BNE 1S ; YES, DON'T RETRY
536 003054* 001021 CMBR LIM1,TRY1 ;LIMIT MET?
537 003056* 126767 177016 177006 BNE GO ;NO, TRY AGAIN
538 003064* 001272 MSG$,BEGIN,WRIT ;ASCII MESSAGE CALL WITH COMMON HEADER
539 003066* 104403 000000* 007072* BIT #BIT2,SR1 ; NEED TO PRINT ERROR MESSAGE ?
540 003074* 032767 000004 174714

```

```

541 003102* 001406 REQ 1S ; NO, CONTINUE
542 003104* 012767 000001 174774 MOV #1,ERRTY ;DATA ERROR
543 ;*****
544 003112* 104406 000000* 000252* SOFF$,BEGIN,TABLE ; YES, PRINT ERROR MESSAGE
545 ;*****
546 003120* 004567 002324 1S: JSR R5,CLEAR ;CLEAR OUT ERRORS
547 003124* 000507 BR NEXTA ;YES
548
549 ;ENTERED BY AN ERROR WHILE READING IN REVERSE
550 003126* 105267 176741 RETRY2: INCB TRY2 ;COUNT A RETRY
551 003132* 132767 000004 176743 BITB #BIT2,FLAG ; HARD ERROR ?
552 003140* 001021 BNE 1S ; YES, DON'T RETRY
553 003142* 126767 176733 176723 CMBR LIM2,TRY2 ;LIMIT MET?
554 003150* 001020 BNE 2S ;NO
555 003152* 104403 000000* 007106* MSG$,BEGIN,RDR ;ASCII MESSAGE CALL WITH COMMON HEADER
556 003160* 032767 000004 174630 BIT #BIT2,SR1 ; NEED TO PRINT ERROR MESSAGE ?
557 003166* 001406 BEO 1S ; NO, CONTINUE
558 003170* 012767 000001 174710 MOV #1,ERRTY ;DATA ERROR
559 ;*****
560 003176* 104406 000000* 000252* SOFF$,BEGIN,TABLE ; YES, PRINT ERROR MESSAGE
561 ;*****
562 003204* 004567 002240 1S: JSP R5,CLEAR ;CLEAR OUT ERRORS
563 003210* 000455 BR NEXTA ; CONTINUE ON
564 003212* 004567 002232 2S: JSR R5,CLEAR ;CLEAR OUT ERRORS
565 003216* 004567 000016 JSR R5,FWDSPA ; SPACE FORWARD
566 003222* 103401 FCS 36 ; ERROR RETURN
567 003224* 000402 BR 4S ; TRY AGAIN
568 003226* 004567 002216 3S: JSR R5,CLEAR ;CLEAR OUT ERRORS
569 003232* 000626 BR GOA ; TRY AGAIN
570
571 ;ENTERED BY AN ERROR WHILE READING FORWARD
572 003234* 105267 176634 PTRY3: INCB TRY3 ;COUNT A RETRY
573 003240* 132767 000004 176635 BITB #BIT2,FLAG ; HARD ERROR ?
574 003246* 001021 BNE 1S ; YES, DON'T RETRY
575 003250* 126767 176626 176616 CMBR LIM3,TRY3 ;LIMIT MET?
576 003256* 001020 BNE 2S ;NO
577 003260* 104403 000000* 007106* MSG$,BEGIN,RDF ;ASCII MESSAGE CALL WITH COMMON HEADER
578 003266* 032767 000004 174522 BIT #BIT2,SR1 ; NEED TO PRINT ERROR MESSAGE ?
579 003274* 001406 BEO 1S ; NO, CONTINUE
580 003276* 012767 000001 174602 MOV #1,ERRTY ;DATA ERROR
581 ;*****
582 003304* 104406 000000* 000252* SOFF$,BEGIN,TABLE ; YES, PRINT ERROR MESSAGE
583 ;*****
584 003312* 004567 002132 1S: JSR R5,CLEAR ;CLEAR OUT ERRORS
585 003316* 000412 BR NEXTA ; CONTINUE ON
586 003320* 004567 002124 2S: JSP R5,CLEAR ;CLEAR OUT ERRORS
587 003324* 004567 000034 JSR R5,PKSPAC ; DO A BACKSPACE
588 003330* 103401 FCS 38 ; ERROR RETURN
589 003332* 000402 BR 4S ;TRY AGAIN
590 003334* 004567 002110 3S: JSP R5,CLEAR ;CLEAR OUT ERRORS
591 003340* 000167 177352 4S: JMP GOB ; TRY AGAIN
592
593
594 003344* 032767 000001 174444 NEXTA: BIT #BIT0,SR1 ;DROP SLAVE?
595 003352* 001602 REQ CONTNU ;NO, GO DO ANOTHER DRIVE
596 003354* 004567 001574 NEXTB: JSR R5,DRSV ;YES

```

597 003360* 000167 177374 ; JMP CONTNU ;GO DO ANOTHER DRIVE
598 ;
599 ;
600 ;
601 ;
602 003364* 016777 174772 174670 BKSPAC: MOV DRYVE,@RHCS2 ;LOAD SELECTED UNIT
603 003372* 012777 177777 174660 MOV #17777,@MTFC ;LOAD FRAME COUNT
604 003400* 012767 000133 174744 MOV #133,FUNC ;LOAD BACKSPACE FUNCTION
605 003406* 000561 ; BR GOGO1 ;DO IT
606 003410* 012777 177777 174642 RWDSFA: MOV #17777,@MTFC ; LOAD FRAME COUNT
607 003416* 012767 000131 174726 MOV #131,FUNC ; LOAD SPACE FORWARD FUNCTION
608 003424* 000552 ; BR GOGO1 ; DO IT
609 003426* 016777 174730 174626 TPMA: MOV DRYVE,@RHCS2 ;LOAD SELECTED UNIT
610 003434* 012767 000127 174710 MOV #127,FUNC ;SET TAPE MARK AND
611 003442* 000543 ; BR GOGO1 ;DO IT
612 003444* 016777 174712 174610 WRITE: MOV DRYVE,@RHCS2 ;SET DRIVE NUMBER UP
613 003452* 012767 000161 174672 MOV #161,FUNC ;LOAD WRITE
614 003460* 016777 174702 174566 MOV WRSVA,@MTWC ;LOAD WORD COUNT
615 003466* 016777 174672 174564 MOV WR,@MTFC ;LOAD FRAME COUNT
616 003474* 016777 174434 174554 MOV WHUFPA,@RHBA ; LOAD BUSS ADDRESS
617 003502* 016767 174634 176350 MOV WREA,XMEM ;SET EA BITS
618 003510* 000455 ; BR GOGO ;DO IT
619 003512* 016777 174644 174542 RUREVI: MOV DRYVE,@RHCS2 ;SET DRIVE NUMBER UP
620 003520* 012767 000177 174624 MOV #177,FUNC ;LOAD READ REVERSE
621 003526* 016777 174640 174520 MOV RRSVA,@MTWC ;LOAD WORD COUNT
622 003534* 016777 174630 174516 MOV RB,@MTFC ;LOAD FRAME COUNT
623 003542* 016777 174644 174506 MOV RUFPA,@RHBA ; LOAD BUS ADDRESS
624 003550* 016767 174640 176302 MOV RUFFEA,XMEM ;SET EA BITS
625 003556* 000432 ; BR GOGO ;DO IT
626 003560* 016777 174576 174474 RDFWD: MOV DRYVE,@RHCS2 ;SET DRIVE NUMBER UP
627 003566* 012767 000171 174556 MOV #171,FUNC ;LOAD READ FORWARD
628 003574* 016777 174572 174552 MOV RRSVA,@MTWC ;LOAD WORD COUNT
629 003602* 016777 174562 174450 MOV RB,@MTFC ;LOAD FRAME COUNT
630 003610* 016777 174312 174440 MOV RBUFPA,@RHBA ; LOAD BUSS ADDRESS
631 003616* 016767 174516 176234 MOV RREA,XMEM ;SET EA BITS
632 003624* 000407 ; BR GOGO ;DO IT
633 003626* 012767 000125 174516 ERAS: MOV #125,FUNC ;SET ERASE COMMAND
634 003634* 016777 174522 174420 MOV DRYVE,@RHCS2 ;SET DRIVE NUMBER
635 003642* 000443 ; BR GOGO1 ; DO IT
636 ;
637 003644* 032767 001000 174204 GOGO: BIT #ADDR22,RES1 ;11/70 SUPPORTED?
638 003652* 001434 BEQ 18 ;NO, SKIP 11/70 SETUP: NO GOOD
639 003654* 017767 174376 176174 MOV @RHBA,PA18 ;CONVERT 18 BIT
640 003662* 006267 176172 ASR XMEM ;ADDRESS TO 22 BITS
641 003666* 006267 176166 ASR XMEM ;
642 003672* 006267 176162 ASR XMEM ;
643 003676* 006267 176156 ASR XMEM ;
644 003702* 104416 000000* 002056* MAP22S, BEGIN,PA18 ; GET 22-BIT ADDR FROM 18-BIT ADDR
645 003710* 016777 176146 174340 MOV PA22,@RHBA ;SETUP RM FOR 22 BITS
646 003716* 016777 176142 174362 MOV PA22,@RHBAE ;LOAD THE EA BITS
647 003724* 042767 000034 176132 RIC #34,EA22 ;CLEAR UNUSED BITS
648 003732* 000367 176126 SWAB EA22 ;LINEUP EA BITS FOR RHCS1
649 003736* 016767 176122 176114 MOV EA22,XMEM ;SETUP BITS 8,9 FOR CS1
650 003744* 056767 176110* 174400* 18: HIS XMEM,FUNC ;LOAD EA BITS INTO COMMAND WORD
651 ;
652 003752* 056777 174372 174324 GOGO1: RTS DENSTY,@MTC ; SET THE DENSITY

653 003760* 016777 174366 174264 MOV FUNC,@RHCS1 ;EXECUTE THE COMMAND
654 003766* 104400 000000* EXIT, BEGIN ;EXIT TO MONITOR, MODULE WAIT FOR INTERRUPT,
655 ;
656 ;
657 ;
658 003772* 132777 000002 174264 RFWIND: BITB #BIT1,@MTDS ;AT BOT ALREADY?
659 004000* 001013 BNE 18 ;NR IF YES
660 004002* 012777 000007 174242 MOV #7,@RHCS1 ;EXECUTE THE RFWIND
661 ;
662 004010* 104407 000000* 004040* 28: BPEAKS,BEGIN ;TEMPORARY RETURN TO MONITOR,
663 004014* 104407 000000* ; RREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION,
664 004020* 032777 020000 174236 BIT #BIT13,@MTDS ;WAIT UNTIL SLAVE BEGINS ACTION ON COMMAND
665 004026* 001770 BREQ 28 ;
666 004030* 000205 RTS P5 ;RETURN
667 ;
668 004032* ;
669 ;
670 004032* 000004 000000* 004040* ;
671 ;
672 ;
673 004040* 005777 174206 ERKORS: TST @RHCS1 ;ATTENTION OR ERROR?
674 004044* 100401 BHI 18 ;YES
675 004046* 000547 BR 128 ;NO GET OUT
676 004050* 042777 001000 174210 18: RIC #BIT9,@MTER ;CLEAR PCOF
677 004056* 032777 002000* 174200 HIT #BIT10,@MTDS ;REACHED EOT?
678 004064* 001404 REQ 28 ;NO
679 004066* 152767 000001 176007 BISP #BIT0,FLAG ; YES, SET EOT FLAG
680 004074* 000534 BR 128 ; GET OUT
681 004076* 004567 001310 28: JSR RS,ERSUPJ ;LOAD ERROR INFO.
682 004102* 142767 000004 175773 KICR #BIT2,FLAG ; CLEAR HARD ERROR FLAG
683 004110* 032777 075477 174150 BIT #75477,@MTER ; ANY HARD ERROR BITS SET ?
684 004116* 001016 BNE 38 ; YES, GO REPORT THEM
685 004120* 032777 102300 174140 BIT #102300,@MTER ;IS THIS A SOFT ERROR?
686 004126* 001517 PEQ 128 ;NO CONTINUE
687 004130* 004567 022366 JSR RS,SFTCNT ;GO ADD ERROR TO SOFT ERROR TABLE
688 ;
689 004134* 032767 000004 173654 ;(COUNT ERROR ON THIS PASS & FOR TOTAL ERRORS.
690 004142* 001477 BIT #BIT2,SR1 ; REPORT SOFT ERRORS ?
691 004144* 104403 000000* 007046* BFO 118 ; NO, CONTINUE
692 004152* 000432 MSGS,BEGIN,SOFT ;ASCII MESSAGE CALL WITH COMMON HEADER
693 004154* 152767 000004 175721 38: BISP #BIT2,FLAG ; CONTINUE
694 004162* 032777 020000 174062 BIT #BIT13,@RHCS1 ; SET HARD ERROR FLAG
695 004170* 001035 BNE 76 ;MASSBUS CONTROL PARITY ERROR?
696 004172* 032777 000400 174062 BIT #BIT8,@RHCS2 ;YES
697 004200* 001035 BNE 88 ;MASSBUS DATA PARITY ERROR?
698 004202* 032777 040000 174054 BIT #BIT14,@MTDS ;YES
699 004210* 001404 REQ 48 ;ANY DRIVE ERROR?
700 004212* 104403 000000* 007034* MSGS,BEGIN,HARD ;NO, CONTINUE
701 004220* 000430 BR 98 ;ASCII MESSAGE CALL WITH COMMON HEADER
702 004222* 032777 040000 174022 48: BIT #BIT14,@RHCS1 ;GO DUMP REGISTERS
703 004230* 001011 BNE 68 ;TRANSFER ERROR?
704 004232* 005777 174026 TST @MTDS ;YES
705 004236* 100441 BHI 118 ;ATTENTION ACTIVE?
706 004240* 005067 CLR ERRTYP ;YES
707 ;
708 004244* 104406 000000* 000252* SOFS,BEGIN,TABLE ;UNKNOWN ERROR TYPE

```

709 ;*****
710 004252* 000433 BR 115 ;CONTINUE
711 004254* 65: MSGNS,BEGIN,TPER ;ASCII MESSAGE CALL WITH COMMON HEADER
712 004254* 104403 000000* 007020* BR 95 ;DUMP REGISTERS
713 004262* 000407
714
715 004264* 76: MSGNS,BEGIN,MCPERR ;ASCII MESSAGE CALL WITH COMMON HEADER
716 004264* 104403 000000* 007021* BR 96 ;DUMP REGISTERS
717 004272* 000403
718 004274* 88: MSGNS,BEGIN,MDPERR ;ASCII MESSAGE CALL WITH COMMON HEADER
719 004274* 104403 000000* 007030*
720
721 004302* 016700 173766 98: MOV MTDB,R0 ;SAVE DATA BUFFER ADDRESS
722 004306* 105777 173750 TSTB @RHCS2 ;READABLE?
723 004312* 100403 BHI 106 ;YES, CONTINUE
724 004314* 012767 000354* 173752 MOV #ZERO,MTDB ;NO, PRINT 0 INSTEAD
725
726 004322* 012767 000030 173556 106: MOV #30,ERRTPY ;XFER ERROR
727 ;*****
728 004330* 104405 000000* 000252* HRDFRS,BEGIN,TABLE ;
729 ;*****
730 004336* 010067 173732 MOV R0,MTDB ;RESTORE DATA BUFFER
731 004342* 004567 001102 118: JSR R5,CLEAR ;CLEAR OUT ERRORS.
732 004346* 042777 040000 173676 BIC #BIT14,@RHCS1 ;CLEAR TR1
733 004354* 056777 174014 173722 BIS SLAVNO,@MTTC ;SET UNIT NUMBER
734 004362* 000261 SEC
735 004364* 000205 PTS R5 ;ERROR RETURN
736 004366* 000241 128: CLC ;NO ERROR
737 004370* 000205 RIS R5 ;RETURN OK
738 ;
739 -----
740
741 ;WHEN END OF TAPE HAS BEEN REACHED, REWIND ALL SLAVES
742 ;TYPE OUT NUMBER OF SOFT ERRORS FOR EACH SLAVE
743 ;FOR THIS PASS AND THE TOTAL NUMBER OF SOFT ERRORS
744 ;FOR EACH SLAVE.
745 ;-----
746
747 004372* 004767 001032 EOI: JSR PC,REZET ;DRIVE & CONTROLLER RESET
748 004376* 004567 001130 JSR R5,REWIND ; START REWIND ALL SLAVES
749 004402* 032767 000020 173406 BIT #BIT4,SR1 ; PRINT OUT SUMMARY ?
750 004410* 001100 BNE 48 ; NO, GET OUT
751 004412* 016767 173740 173740 MOV DVICE,DRIVE ;RESET DRIVE POINTER
752 004420* 012767 177777 173734 MOV #=-1,DRIVE
753 004426* 005067 173742 18: CLR SLAVNO ;CLEAR SLAVE #
754 004432* 012767 000001 173736 MOV #1,POINT ;SETUP SLAVE POINTER
755 004440* 004567 000150 JSR R5,SELDRV ; GET VALID DRIVE NUMBER
756 004444* 132767 000010 175431 BITB #BIT3,FLAG ; ALL DRIVES DONE ?
757 004452* 001057 BNE 48 ; YES, GET OUT
758 004454* 016700 173702 28: MOV DRYVE,R0 ;GET DRIVE # INTO R0
759 004460* 006300 ASL R0 ; MAKE IT A WORD ADDRESS
760 004462* 136700 173710 000316* BITR POINT,SEL(R0) ;DOES THIS SLAVE EXIST?
761 004470* 001437 BEQ 36 ;NO SLAVE GET NEXT
762 004472* 004567 001724 JSR R5,CALTAR ;CALCULATE OFFSET TO ERROR TABLE
763 ;*****
764 ;CONVERT DRYVE TO ASCII AND

```

```

765 ;STORE AT TABLSF
766 004476* 104420 000000* 000362* OTOA6,BEGIN,DRIVE,TABLSF
767 004504* 002016*
768 ;*****
769 ;*****
770 ;CONVERT SLAVNO TO ASCII AND
771 ;STORE AT TABLS1
772 004506* 104420 000000* 000374* OTOA6,BEGIN,SLAVNO,TABLS1
773 004514* 002026*
774 ;*****
775 SOFTFR(R0),SOFTLC ;TYPE NUMBER
776 ;*****
777 ;CONVERT SOFTLC TO ASCII AND
778 ;STORE AT TABLS2
779 004524* 104420 000000* 000404* OTOA6,BEGIN,SOFTLC,TABLS2
780 004532* 002036*
781 ;*****
782 CLR SOFTFR(R0) ;CLEAR SOFT ERROR COUNTER
783 004540* 002700 000002 ADD #2,R0 ;UPDATE ERROR POINTER
784 004544* 016007 000416* 173634 MOV SOFTFR(R0),SOFTTO ;TYPE TOTAL
785 ;*****
786 ;CONVERT SOFTTO TO ASCII AND
787 ;STORE AT TABLS3
788 004552* 104420 000000* 000406* OTOA6,BEGIN,SOFTTO,TABLS3
789 004560* 007046*
790 ;*****
791 MSGNS,BEGIN,TOTALE ;ASCII MESSAGE CALL WITH COMMON HEADER
792 004570* 022767 000007 173576 38: CMP #7,SLAVNO ;ALL SLAVES TESTED?
793 004576* 001713 BFC 16 ; YES, GO GET ANOTHER DRIVE
794 004600* 005267 173570 INC SLAVNO ;NO GET NEXT SLAVE
795 004604* 006367 173566 ASL POINT ;MOVE SLAVE POINTER
796 004610* 000721 HR 26 ;TEST FOR NEXT SLAVE
797 004612* 000207 KIS PC ; RETURN
798 ;
799 -----
800
801 ;SELECT A LEGAL DRIVE
802 ;THIS SUBROUTINE USES R0 AND R1
803
804 004614* 005267 173542 SELDRV: INC DRYVE ;GET NEW DRIVE NUMBER
805 004620* 142767 000010 175255 BICR #BIT3,FLAG ;CLEAR END OF DRIVE FLAG
806 004626* 022767 000010 173526 CMP #8,DRIVE ;CHECKED ALL DRIVES?
807 004634* 001435 REQ 35 ;YES GET OUT
808 004636* 006267 173516 ASR DRYVE ;IS THIS DRIVE PRESENT
809 004642* 103364 BCC SELDRV ;NO TRY ANOTHER DRIVE
810 004644* 016777 173512 173410 MOV DRYVE,@RHCS2 ;LOAD DRIVE NUMBER
811 004652* 032777 040000 173420 BIT #40000,@MTDT ;IS IT A LEGAL DRIVE?
812 004660* 001034 RNE 48 ;YES
813 004662* 012700 000001 MOV #1,R0 ;NO SETUP DRIVE POINTER
814 004666* 016701 173470 MOV DRYVE,R1 ;GET DRIVE NUMBER
815 004672* 005701 18: TST R1 ;SETUP POINTER IO #
816 004674* 001012 BNE 26 ;DRIVE NUMBER IN BIT POSITION
817 004676* 040067 173454 BIC R0,DVICE ;CLEAR ILLEGAL DRIVE
818 ;*****
819 ;CONVERT DRYVE TO ASCII AND
820 ;STORE AT TABLSF

```

```

821 004702* 104420 000000* 000362*      OTOAs,REGIN,DRIVE,TABLSF
822 004710* 002016*
823
824 004712* 104403 000000* 007114*      ;*****
825 004720* 000735 000000* 007114*      MSGNs,BFGIN,DD ;ASCII MESSAGE CALL WITH COMMON HEADER
826 004722* 005301 26:      BR SELDRV ;GET NEXT DRIVE
827 004724* 006300*      DEC R1 ;DECREMENT DRIVE NUMBER
828 004726* 000761 36:      ASL R0 ;MOVE DRIVE POINTER
829 004730* 152767 000010 175145 36:      BR 18 ;TRY AGAIN
830 004736* 012767 177777 173416 36:      RISP #BIT3,FLAG ;SET END OF DRIVE FLAG
831 004744* 016767 173466 173406 48:      MOV #=-1,DRIVE ;RESET DRIVE COUNTER
832 004752* 000205 48:      RTS R5 ;RESET DRIVE REGISTER
833 ;
834 ;-----
835
836 ;THIS ROUTINE BUILDS SLAVE TABLE FOR SELECTED DRIVE
837 ;IF NO SLAVES ARE PRESENT, DRIVE IS DROPPED.
838 004754* 016777 173402 173300 SLAVI: MOV DRYVE,@RHCS2 ;LOAD UNIT SELECT
839 004762* 016700 173374 173374 18:      MOV DRYVE,R0 ;SET INDEX
840 004766* 006300 173306 173306 18:      ASL R0 ; MAKE IT A WORD ADDRESS
841 004770* 016777 173400 173306 18:      MOV SLAVNO,@MTTC ;SELECT SLAVE
842 004776* 032777 010000 173260 18:      BIT #10000,@MTDS ;CHECK FOR MOL
843 005004* 001406 173364 000316*      BEQ 25 ;NO
844 005006* 156760 173364 000316*      BISR POINT,SEL(R0) ;SEL SLAVE BIT
845 005014* 005267 173316 173316 18:      INC SLACT ;INCREMENT SLAVE COUNTER
846 005020* 000403 173350 000316* 26:      BR 35 ;CONTINUE
847 005022* 146760 173350 000316* 26:      RICH POINT,SEL(R0) ;CLEAR BIT
848 005030* 022767 000007 173336 38:      CMP #7,SLAVNO ;SLAVE LIMIT MET?
849 005036* 001405 173336 173326 18:      HFO 45 ;YES
850 005040* 005267 173330 173326 18:      INC SLAVNO ;NO
851 005044* 006367 173326 173326 18:      ASL POINT ;MOVE POINTER
852 005050* 000747 173326 173326 18:      BR 16 ;GO CHECK ANOTHER SLAVE
853 005052* 105760 000316* 48:      TSTB SEL(R0) ;ANY SLAVES SELECTED?
854 005056* 001030 173306 173306 18:      BNE 75 ;YES, GET OUT
855 005060* 016767 173276 173312 18:      MOV DRYVE,HOLD ;NO
856 005066* 012767 000001 173306 58:      MOV #1,HLY ;CALCULATE WHICH DRIVE TO DROP
857 005074* 005767 173306 173306 18:      IST HOLD ; FOUND IT YET ?
858 005100* 001405 173274 173274 18:      BFO 65 ; YES, GET OUT
859 005102* 006367 173274 173266 18:      ASL HLK ; NO, MOVE THE POINTER
860 005106* 005367 173266 173266 18:      DFC HOLD ; LOOK AT THE NEXT DRIVE
861 005112* 000770 173262 173234 68:      BR 56 ; SEE IF THIS IS IT
862 005114* 046767 173262 173234 68:      BIC HLK,DRIVE ;ELIMINATE DRIVE FROM EXERCISE
863 ;*****
864 ;CONVERT DRIVE TO ASCII AND
865 ;STORE AT TABLSF
866 005122* 104420 000000* 000362*      OTOAs,REGIN,DRIVE,TABLSF
867 005130* 002016*
868
869 005132* 104403 000000* 007114*      ;*****
870 005140* 012767 000001 173230 76:      MSGNs,BEGIN,DD ;ASCII MESSAGE CALL WITH COMMON HEADER
871 005146* 005067 173222 173222 18:      MOV #1,POINT ;RESET POINTER
872 005152* 000205 173222 173222 18:      CLR SLAVNO ;CLEAR SLAVE COUNT
873 005154* 000205 173222 173222 18:      RTS R5 ;RETURN
874 ;
875 ;-----
876 ;DROP SLAVE FROM TEST

```

```

877 005154* 016700 173202 173202 18:      DRSV: MOV DRYVE,R0 ;SET INDEX
878 005160* 006300 173202 173202 18:      ASL R0 ; MAKE IT A WORD ADDRESS
879 005162* 146760 173210 000316*      RICH POINT,SEL(R0) ;DROP SLAVE
880 005170* 005367 173142 173142 18:      DEC SLACT ;SUBTRACT ONE FROM SLAVE COUNT
881 ;*****
882 ;CONVERT DRIVE TO ASCII AND
883 ;STORE AT TABLSF
884 005174* 104420 000000* 000362*      OTOAs,REGIN,DRIVE,TABLSF
885 005202* 002016*
886
887 ;*****
888 ;CONVERT SLAVNO TO ASCII AND
889 ;STORE AT TABLS1
890 005204* 104420 000000* 000374*      OTOAs,BEGIN,SLAVNO,TABLS1
891 005212* 002026*
892
893 005214* 104403 000000* 007052*      ;*****
894 005222* 005767 173110 173110 18:      MSGNs,REGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
895 005226* 001002 173110 173110 18:      TST SLACT ;ANYBODY STILL RUNNING?
896 005230* 104410 000000* 173110 18:      BNE 16 ;BR IF YES
897 005234* 000205 173110 173110 18:      ENDS,REGIN ;
898 ;
899 ;-----
900
901 005236* 016777 173120 173016 18:      MOV DRYVE,@RHCS2 ;SELECT DRIVE
902 005244* 032777 010000 173012 18:      BIT #BIT12,@MTDS ;IF MOL SET THEN
903 005252* 001406 173012 173012 18:      BEQ 18 ; BEGIN
904 005254* 032777 000200 173002 18:      BIT #BIT7,@MTDS ; IF DRY (DRIVE READY) SET THEN
905 005262* 001402 173002 173002 18:      BEQ 18 ; BEGIN
906 005264* 000261 173002 173002 18:      SEC ; SET DRIVE READY FOR RETURN
907 005266* 000205 173002 173002 18:      RTS R5 ; RETURN==DRIVE READY
908 ;
909 ; END
910 005270* 000241 173002 173002 18:      CLC ;DRIVE NOT READY
911 005272* 000205 173002 173002 18:      RTS R5 ;RETURN NOT READY
912 ;
913 ;-----
914
915 005274* 012767 077777 173042 18:      NOTRDY: MOV #77777,CLK ;SET TIMER
916 005302* 004567 000142 173042 18:      JSR R5,CLEAR ;CLEAR OUT ANY DRIVE ERRORS
917 005306* 004567 177724 177724 18:      JSR R5,READY ;SLAVE READY?
918 005312* 103423 177724 177724 18:      BCS 25 ;YES, RETURN
919 005314* 104407 000000* 177724 18:      BREAKs,REGIN ;TEMPORARY RETURN TO MONITOR....
920 005320* 104407 000000* 177724 18:      BREAKs,REGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
921 005324* 005367 173014 173014 18:      DEC CLK ;WAIT SOME MORE?
922 005330* 001366 173014 173014 18:      BNE 16 ;YES, CONTINUE
923 005332* 004567 000054 172542 18:      JSR R5,ERSUR1 ;NO, LOAD ERROR INFO.
924 005336* 012767 000005 172542 18:      MOV #6,ERRTYP ;DEVICE NOT READY
925 ;*****
926 ;*****
927 005344* 104405 000000* 000252*      HDRS$,REGIN,TABLE ;
928 005352* 004567 177576 177576 18:      JSR R5,DRSV ;DROP SLAVE
929 005356* 000261 177576 177576 18:      SEC ;
930 005360* 000205 177576 177576 18:      RTS R5 ;RETURN, NOT READY
931 005362* 000241 177576 177576 18:      CLC ;READY
932 005364* 000205 177576 177576 18:      RTS R5 ;RETURN

```

```
933
934 005366* 014167 172514 ERSUB2: MOV -(R1),ASB ;LOAD DATA
935 005372* 010167 172514 MOV R1,SBADR ;LOAD ADDRESS OF DATA WRITTEN
936 005376* 014267 172506 MOV -(P2),AWAS ;LOAD DATA
937 005402* 010267 172476 MOV R2,WASADR ;ADDRESS OF DATA HEAD
938 005406* 005721 TST (R1)+ ;RESET R1
939 005410* 005722 TST (R2)+ ;RESET R2
940
941
942 005412* 016767 172634 172460 ERSUB1: MOV RHCS1,CSFA ;LOAD ADDRESS OF CSR
943 005420* 017767 172626 172454 MOV RHCS1,ACSR ;LOAD CONTENTS OF CSR
944 005426* 000205 RTS R5
945 ;
946
947
948 ;CLEAR ALL RH CONTROLLER AND DEVICE REGISTERS
949 005430* 012777 000000 172624 FEZET: MOV #BIT5,RRHCS2 ;RH INIT
950 005436* 104407 000000* ;BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
951 005442* 104407 000000* ;BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
952 005446* 000207 RTS PC ;RETURN
953 ;
954
955
956
957 005450* 016777 172706 172604 CLEAR: MOV DRYVE,RRHCS2 ;LOAD DRIVE NUMBER
958 005456* 012777 000011 172566 MOV #11,RRHCS1 ;ISSUE A DRIVE CLEAR
959 005464* 005067 174400 CLR CLRPRDY ;CLR DRIVE READY TIMER
960 005470* 005267 174374 16: INC CLRPRDY ;COUNT A BREAK CYCLE
961 005474* 104407 000000* ;BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
962 005500* 104407 000000* ;BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
963 005504* 032777 000200 172552 BIT #BIT7,@MTDS ;READY ?
964 005512* 001006 BNE Z6 ;YES - LEAVE
965 005514* 005767 174350 TST CLRPRDY ;TIMED OUT ?
966 005520* 001363 BNE Z6 ;NO TRY AGAIN
967 005522* 104403 000000* 007142* MSGNS,BEGIN,DNR ;ASCII MESSAGE CALL WITH COMMON HEADER
968 005530* 052777 040000 172514 28: BIS #BIT14,RRHCS1 ;CLEAR TRF
969 005536* 056777 172632 172540 BIS SLAVNO,@MTTC ;SET SLAVE UNIT NUMBER
970 005544* 000205 RTS R5 ;RETURN
971 ;
972
973 ;THIS ROUTINE GETS THE BUFFER FROM THE MONITOR THEN PERFORMS
974 ;THE PROPER MANIPULATION
975
976
977
978 005546*
979 005546* 104415 000000* 000124* BUFF: GETPAS,BEGIN, RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
980 005554* 016767 172352 172606 MOV RBUFV2,RB
981 005562* 016767 172602 172602 MOV RB,RESAV
982 005570* 066767 172574 172572 ADD RB,RB
983 005576* 005467 172570 NEG RBSAV
984 005602* 005467 172562 NFG PP
985 005606* 005067 172526 CLR RDEA
986 005612* 056767 172312 172520 BIS #BUFEA,RDEA
987 005620* 006367 172514 ASL RDEA ; SHIFT TO POSITIONS
988 005624* 006367 172510 ASL RDEA ; R AND 9
```

```
989 005630* 006367 172504 ASL RDEA ;
990 005634* 006367 172504 ASL RDEA ;
991 005640* 000205 RTS R5
992 ;
993
994
995 ;THIS ROUTINE GETS THE SLAVES SELECTED FROM THE "SLAV" ROUTINE
996 005642* 016700 172514 GTSLAV: MOV DRYVE,R0 ;SET INDEX
997 005646* 006300 ASL R0 ; MAKE IT A WORD ADDRESS
998 005650* 136760 172522 000316* 18: RTRB POINT,SEL(R0) ;TEST FOR SLAVE
999 005656* 001405 BEQ Z6 ;NOT SELECTED, LOOK FOR MORE
1000 005660* 016777 172510 172416 MOV SLAVNO,@MTTC ;SET SLAVE NUMBER
1001 005666* 000241 CLC ;NORMAL RETURN: CLEAR CARRY
1002 005670* 000205 RTS R5 ;RETURN, SLAVE FOUND
1003 005672* 022767 000007 172474 25: CMP #7,SLAVNO ;ALL SLAVES DONE?
1004 005700* 001405 REQ Z6 ;YES, GET OUT
1005 005702* 005267 172466 INC SLAVNO ;NO, COUNT A SLAVE
1006 005706* 006367 172464 ASL POINT ;MOVE POINTER TO NEXT SLAVE
1007 005712* 000756 BR Z6 ;GO LOOK FOR IT
1008 005714* 005067 172454 38: CLR SLAVNO ;CLEAR COUNTER
1009 005720* 012767 000001 172450 MOV #1,POINT ;RESET POINTER
1010 005726* 000261 SEC ;SET CARRY: NO SLAVES
1011 005730* 000205 RTS R5 ;RETURN, NO SLAVES FOUND
1012 ;
1013
1014 ;THIS SUBROUTINE IS USED FOR CONTROLLING SLAVE REWINDS. IT FIRST ISSUES
1015 ;REWIND COMMANDS TO ALL SLAVES, IN REWIND MODE (FLAG BIT11 = 0). WHEN ALL
1016 ;SLAVES ARE STARTED, IT CHANGES TO CHECK-REWIND MODE (FLAG BIT1 = 1) AND
1017 ;REPEATS THE SAME BASIC CODE, THIS TIME WAITING FOR EACH SLAVE TO COMPLETE
1018 ;REWIND (REACH POT).
1019 005732* 142767 000002 174143 REWIND: RIBC #BIT1,FLAG ;REWIND MODE
1020 005740* 012767 177777 172414 105: MOV #1,DRIVE ;RESET DRIVE #
1021 005746* 016767 172404 172404 MOV DVICE,DRIVE ; RESET DRIVE POINTER
1022 005754* 012767 000001 172414 18: MOV #1,POINT ;SETUP SLAVE POINTER
1023 005762* 005067 172406 CLR SLAVNO ;CLEAR SLAVE COUNTER
1024 005766* 004567 176622 JSR R5,SELDRV ; GET A VALID DRIVE NUMBER
1025 005772* 132767 000010 174103 RTRB #BIT3,FLAG ; ALL DRIVES DONE ?
1026 006000* 001071 PHE Z6 ; YES, GET OUT
1027 006002* 016700 172354 MOV DRYVE,R0 ; COPY DRIVE # INTO REG. 0
1028 006006* 006300 ASL R0 ; MAKE IT A WORD ADDRESS
1029
1030 006010* 136760 172362 000316* 29: BITB POINT,SEL(R0) ;DOES THIS SLAVE EXIST?
1031 006016* 001451 REQ Z6 ;NO GET NEXT SLAVE
1032 006020* 016777 172336 172234 MOV DRYVE,RRHCS2 ;LOAD UNIT #
1033 006026* 016777 172342 172250 MOV SLAVNO,@MTTC ;SELECT SLAVE
1034
1035 006034* 132767 000002 174041 RTRB #BIT1,FLAG ;ARE WE SETTING UP REWINDS OR CHECKING?
1036 006042* 001417 BEQ Z6 ;SETTING UP
1037 006044*
1038 006044* 104407 000000* 88: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
1039 006050* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1040 006054* 032777 020000 172202 BIT #BIT13,@MTDS ;HAS PIP RESET YET?
1041 006062* 001370 BNE Z6 ;NO
1042 006064* 032777 000002 172172 BIT #BIT1,@MTDS ;MAKE SURE WE'RE AT BOT
1043 006072* 001023 BNE Z6 ;WE ARE
1044 006074* 004567 177054 JSR R5,DRSV ;WE'RE NOT: DROP THE SLAVE
```

```

1045 006100* 000420 BP 48
1046 006102* 004567 177130 75: JSR R5,READY ;MAKE SURE IT IS READY
1047 006106* 103403 RCS 36 ;READY, CONTINUE
1048 006110* 004567 177160 JSR R5,NOTPDY ;NOT READY, GO WAIT
1049 006111* 103412 RCS 46 ;STILL NOT READY, SLAVE DROPPED
1050 006110* 004567 175650 36: JSR F5,REWIND ;GO REWIND A SLAVE
1051 006122* 012703 000177 MOV #177,R3 ;SET UP COUNTER FOR WAIT LOOP
1052 006126*
1053 006126* 104407 000000* 118: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR....
1054 006132* 104407 000000* BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
1055 006136* 005303 DEC F3
1056 006140* 001372 FNE 116 ;WAIT A WHILE LONGER
1057 006142* 022767 000007 172224 46: CMP #7,SLAVNO ;ALL SLAVES TESTED?
1058 006150* 001701 BEQ 18 ;YES, GO CHECK NEXT DRIVE
1059 006157* 005267 172216 INC SLAVNO ;NO, COUNT A SLAVE
1060 006156* 006367 172214 ASL POINT ;POINT TO NEXT SLAVE
1061 006162* 000712 BPL 26 ;GO LOOK FOR NEXT SLAVE
1062
1063 006161* 132767 000002 173711 58: BITH #BIT1,FLAG ;CHECK WHICH MODE WE'RE IN NOW
1064 006172* 001004 BNE 66 ;DONE BOTH MODES...EXIT ROUTINE
1065 006174* 152767 000002 173701 BISH #BIT1,FLAG ;SET CHECK-REWIND MODE
1066 006202* 000656 BR 106 ;WAIT FOR DRIVES TO REWIND
1067 006204* 000205 RTS #5 ;RETURN
1068
1069
1070
1071 ;SWAPS DENSITY IF ALLOWED AND IF HARDWARE ALLOWS.
1072 006206* 032767 000040 171602 SWAP.DENSITY: BIT #BIT5,SW1 ;+TM03/TE16
1073 006214* 001004 RNE 18 ;+1M02/TU16
1074 006216* 032777 000002 172040 BIT #BIT1,#MINS ;+SLAVES AT BOT?
1075 006224* 001423 BEQ 36 ;+CANT SWAP,RETURN
1076 006226* 022767 002300 172114 18: CMP #2300,DENSITY ;+PRESENTLY AT 1600?
1077 006234* 001010 BNE 28 ;+NO-600BPI SET.
1078 006236* 032767 000200 171552 BIT #BIT7,SW1 ;+800BPI ALLOWED?
1079 006244* 001013 BNE 36 ;+CANT SWAP,RETURN
1080 006246* 012767 001300 172074 MOV #1300,DENSITY ;+SET DENSITY TO 800BPI
1081 006254* 000407 BR 36 ;+RETURN
1082 006256* 032767 000100 171532 26: BIT #BIT6,SR1 ;+1600BPI ALLOWED?
1083 006264* 001003 BNE 36 ;+CANT SWAP,RETURN
1084 006266* 012767 002300 172054 MOV #2300,DENSITY ;+SET DENSITY TO 1600BPI
1085 006274* 000207 RTS PC ;+RETURN
1086
1087
1088 ;CLEAR SOFT ERROR TABLE AND SLAVE TABLE IF OPERATOR DID
1089 ;NOT SELECT THE SLAVES. IF OPERATOR SELECTED SLAVES,
1090 ;COUNT THEM.
1091 006276* 012700 000416* CLTAB: MOV #SOFTERR,RO ;GET ADDRESS OF TABLE
1092 006302* 012701 000200* MOV #128,,R1 ;SETUP COUNTER
1093 006306* 005020 18: CLR (R0)+ ;CLEAR TABLE
1094 006310* 005301 DEC R1 ;DONE YET?
1095 006312* 001375 RNE 18 ;NO
1096 006314* 032767 000002 171474 BIT #BIT1,SR1 ;DID OPERATOR SEL SLAVES?
1097 006322* 001425 BEQ 56 ;NO
1098 006324* 005000 CLR R0 ;CLEAR R0 WHICH =DRIVE INDEX
1099 006326* 005067 172004 CLR SLACT ;CLEAR SLAVE COUNTER
1100 006332* 012767 000001 172036 26: MOV #1,POINT ;SETUP POINTER
    
```

```

1101 006340* 036760 172032 000316* 38: BIT POINT,SEL(R0) ;DOES THIS SLAVE EXIST?
1102 006346* 001402 BEQ 48 ;NO
1103 006350* 005267 171762 INC SLACT ;YES, INCREMENT COUNTER
1104 006354* 006367 172116 48: ASL POINT ;MOVE POINTER LEFT
1105 006360* 103367 RCC 36 ;IST NEXT BIT IF WD IS NOT FINISHED
1106 006362* 062700 000002 ADD #2,R0 ;UPDATE DRIVE INDEX FOR SLAVE TABLE
1107 006366* 022700 000020 CMP #16,,R0 ;DONE ALL DRIVES YET?
1108 006372* 001357 BNE 26 ;NO TEST FOR MORE SLAVES
1109 006374* 000411 BR 76 ;DONE COUNTING SLAVES GET OUT
1110 006376* 005067 171734 58: CLR SLACT ;CLEAR SLAVE COUNTER
1111 006402* 012700 000316* MOV #SEL,R0 ;GET ADDRESS OF SLAVE TABLE
1112 006406* 012701 000010 MOV #8,,R1 ;SETUP COUNTER FOR 8. WORDS
1113 006412* 005020 66: CLR (R0)+ ;CLEAR OUT A WORD
1114 006414* 005301 DEC R1 ;ALL DONE?
1115 006416* 001375 BNE 68 ;NO, KEEP CLEARING
1116 006420* 000205 78: RTS R5 ;DONE, RETURN
1117
1118
1119
1120 ;SOFT ERROR 'MATRIX' CALCULATION: THERE ARE 16 WORDS FOR EACH OF
1121 ;THE POSSIBLE 8 DRIVES (ROW) AND A COLUMN FOR EACH SLAVE NUMBER; EACH
1122 ;SLAVE HAS TWO WORDS; ERRORS IN THIS PASS THROUGH TAPE, AND TOTAL
1123 ;ERROR COUNT. ON EXIT, R0 WILL CONTAIN TABLE OFFSET.
1124
1125 006422* 006300 CALTAB: ASL R0 ;MULTIPLY DRIVE NUMBER BY 16 ('ROW #')
1126 006424* 006300 ASL R0
1127 006426* 006300 ASL R0
1128 006430* 006300 ASL R0
1129 006432* 016701 171736 MOV SLAVNO,R1 ;GET SLAVE NUMBER
1130 006436* 006301 ASL R1 ;MULTIPLY IT
1131 006440* 006301 ASL R1 ;BY 4 TO GET 'MATHIX COLUMN'
1132 006442* 060100 ADD R1,R0 ;CREATE THE TABLE OFFSET
1133 006444* 000205 RTS R5 ;RETURN
1134
1135 006446* 012704 000003 SWITCH: MOV #3,R4 ;LOAD REG. 4 WITH DEFAULT RETRY LIMIT
1136 006452* 110467 173422 MOVB R4,LIM1 ;SET WRITE RETRY LIMIT
1137 006456* 110467 173417 MOVB R4,LIM2 ;SET READ REVERSE LIMIT
1138 006462* 110467 173414 MOVB R4,LIM3 ;SET READ FORWARD LIMIT
1139 006466* 032767 000010 171322 BIT #BIT3,SR1 ;OVERRIDE DEFAULT RETRY LIMITS?
1140 006474* 001411 BEQ 16 ;NO, CONTINUE
1141 006476* 116767 173373 173374 MOVB #LIM,LIM1 ;YES, SET WRITE LIMIT
1142 006504* 116767 173366 173367 MOVB #RRLIM,LIM2 ;READ REVERSE LIMIT
1143 006512* 116767 173361 173362 MOVB #RFLIM,LIM3 ;READ FORWARD LIMIT
1144 006520* 000205 18: RTS R5 ;RETURN
1145
1146
1147
1148 ;ADD SOFT ERROR TO SOFT ERROR TABLE - IF 20 SOFT ERRORS PER PASS
1149 ;ON A SLAVE, DROP IT.
1150
1151 006522* 016700 171634 SFTCNT: MOV DRYVE,R0 ;GET DRIVE NUMBER
1152 006526* 006300 ASL R0 ;MAKE IT A WORD ADDRESS
1153 006530* 004567 177666 JSR R5,CALTAB ;CALCULATE OFFSET FOR ERROR TABLE
1154 006534* 005260 000416* INC SOFTERR(R0) ;ADD 1 TO ERROR FOR THIS PASS
1155 006540* 022760 000024 000416* CMP #20,,SOFTERR(R0) ;20 SOFT ERRORS?
1156 006546* 000222 BNE 18 ;NO
    
```

```

1157 ;*****
1158 ;CONVERT DRYVE TO ASCII AND
1159 ;STORE AT TABLSF
1160 006550 104420 000000 000362 OTOAS,BEGIN,DRYVE,TABLSF
1161 006556 002016
1162 ;*****
1163 ;CONVERT SLAVNO TO ASCII AND
1164 ;STORE AT TABLS1
1165 006560 104420 000000 000374 OTOAS,BEGIN,SLAVNO,TABLS1
1166 006566 002026
1167 ;*****
1168 ;ASCII MESSAGE CALL WITH COMMON HEADER
1169 006570 104403 000000 007124 MSGNS,BEGIN,SFTLIM
1170 ;+ 20+ SFT ERRS DRPNG SLAVE
1171 006576 005720 TST (R0)+ ; POINT TO TOTAL ERRORS COUNTER
1172 006600 005260 000416 INC SOFTER(R0) ;ADD ONE TO TOTAL ERROR COUNTER
1173 006604 005725 TST (R5)+ ;CLEAR RETURN TO "ERRORS"
1174 006606 005725 TST (R5)+ ;CLEAR RETURN TO "INTRPT"
1175 006610 000167 174540 JMP NEXTB ;GO DROP SLAVE AND CONTINUE
1176 006614 005720 18: TST (R0)+ ; POINT TO TOTAL ERRORS COUNTER
1177 006616 005260 000416 INC SOFTER(R0) ;ADD 1 TO TOTAL ERRORS
1178 006622 000205 RTS R5 ;RETURN
1179 ;
1180 -----
1181
1182
1183 006624 016700 171156 SETUP: MOV ADPP,R0 ;GET DEVICE ADDRESS
1184 006630 010067 171416 MOV R0,RHCS1 ;GENERATE REGISTER ADDRESS
1185 006634 005720 TST (R0)+
1186 006636 010067 171412 MOV R0,MTNC
1187 006642 005720 TST (R0)+
1188 006644 010067 171406 MOV R0,PHBA
1189 006650 005720 TST (R0)+
1190 006652 010067 171402 MOV R0,MTFC
1191 006656 005720 TST (R0)+
1192 006660 010067 171376 MOV R0,RHCS2
1193 006664 005720 TST (R0)+
1194 006666 010067 171372 MOV R0,MIDS
1195 006672 005720 TST (R0)+
1196 006674 010067 171366 MOV R0,MTFR
1197 006700 005720 TST (R0)+
1198 006702 010067 171362 MOV R0,MTAS
1199 006706 005720 TST (R0)+
1200 006710 010067 171356 MOV R0,MTCK
1201 006714 005720 TST (R0)+
1202 006716 010067 171352 MOV R0,MTFB
1203 006722 005720 TST (R0)+
1204 006724 010067 171346 MOV R0,MTMH
1205 006730 005720 TST (R0)+
1206 006732 010067 171342 MOV R0,MTDT
1207 006736 005720 TST (R0)+
1208 006740 010067 171336 MOV R0,MTSN
1209 006744 005720 TST (R0)+
1210 006746 010067 171332 MOV R0,MTIC
1211 006752 032767 001000 171076 BIT #ADDR22,RES1 ;11/70?
1212 006760 001006 BEQ R5 ;NO, SKIP IT ALL
    
```

```

1213 006762 005720 TST (R0)+ ;YES IT IS
1214 006764 010067 171316 MOV R0,RHBAF ;A 11/70
1215 006770 005720 TST (R0)+ ;SETUP THESE
1216 006772 010067 171312 MOV R0,RHCS3 ;TWO REGISTERS
1217 006776 016700 171006 18: MOV VECTOR,R0 ;GET VECTOR ADDR
1218 007002 012720 004032 MOV #INTRPT,(R0)+ ;LOAD IT
1219 007006 116710 171000 MOVB BPL,(R0) ;SET BR LEVEL
1220 007012 000207 RTS PC
1221 ;
1222 -----
1223
1224
1225
1226
1227 007014 007656 LDRIVE: MSL
1228 007016 177777 177777
1229 007020 007200 TREN: MS1
1230 007022 177777 177777
1231 007024 007256 MCFERR: MS2
1232 007026 177777 177777
1233 007030 007305 MDPERR: MS3
1234 007032 177777 177777
1235 007034 007433 HARD: MS8
1236 007036 177777 177777
1237 007040 007221 NODENS: MSGNDS
1238 007042 007341 MS4
1239 007044 177777 177777
1240 007046 007354 SOFT: MS5
1241 007050 177777 177777
1242 007052 007176 DRP: CPLF
1243 007054 007625 MSF ; SLAVE
1244 007056 007602 MS9 ;NOT READY = DRIVE
1245 007060 002016 TABLSF
1246 007062 007625 MSE
1247 007064 002026 TABLS1
1248 007066 007341 MS4
1249 007070 177777 177777
1250
1251 007072 007370 WRIT: MS6 ;+WRITE RETRY
1252 007074 007635 MSLIM ;+LIMIT EXCEEDED
1253 007076 177777 177777
1254 007100 007406 RDR: MS7 ;+READ REVERSE RETRY
1255 007102 007635 MSLIM ;+LIMIT EXCEEDED
1256 007104 177777 177777
1257 007106 007447 RDF: MS8 ;+READ FOWARD RETRY
1258 007110 007635 MSLIM ;+LIMIT EXCEEDED
1259 007112 177777 177777
1260 007114 007473 DD: MSA
1261 007116 002016 TABLSF
1262 007120 007176 CRLF
1263 007122 177777 177777
1264
1265 007124 007473 SFTLIM: MSA ;+DROPPED DRIVE
1266 007126 002016 TABLSF
1267 007130 007625 MSE ;+ SLAVE
1268 007132 002026 TABLS1
    
```


Table listing symbols (MSG, MSGS, MSL, MSLIM, MS0, MS1, MS2, MS3, MS4, MS5, MS6, MS7, MS8, MS9, MTAS, MTCK, MTDB, MTDS, MTDI, MTER, MTEC, MTHR, MTSN, MTTC, MTWC, NEXI, NFXTA, NEXTB, NODEKS, NOTRDY, NTRUPT, NULL, OPEN, OTOA, PASCNT, PASS, PA18, PA22, PIRQ, POINT, POPSP, POPSP2, PRY, PRY0, PRY1) and their corresponding values across multiple columns.

Table listing symbols (PRY2, PRY3, PRY4, PRY5, PRY6, PRY7, PS, PSM, PUSH, PUSH2, RANDS, RANUM, RB, RBSAV, RBUEA, RBUFFA, RBUFFSZ, RBUFFVA, RDEA, RDF, RDFWD, RDR, RDREV, READY, REST, RESTRT, RES1, RES2, RETRY1, RETRY2, RETRY3, REWIND, REWIND, REZLT, RFLIM, RHBA, RHBAE, RHCS1, RHCS2, RHCS3, RRLIM, RSTR, SBADR, SEL, SELDFV, SETUP, SFICNT, SFTLIM, SLACT, SLAV, SLAVNO, SOFCNT, SOFER, SOFPA) and their corresponding values across multiple columns.

SOFT	007046R	691	1240#																
SOFTER	000416R	336#	775	782*	784	1091	1154*	1155	1172*	1177*									
SOFTLC	000404R	333#	775#	779															
SOFTTO	004406R	334#	784#	768															
SPOINT	000032P	234#																	
SPSIZ	000040P	1#	278																
SR1	000016R	227#	381	386	435	540	556	578	594	689	749	1072	1078	1082					
		1096	1139																
SR2	000020R	228#																	
SR3	000022R	229#																	
SP4	000024R	230#																	
SSL	002614R	474	479#	484	519														
START	002104R	233	373#	448															
STAT	000026R	232#																	
SVR0	000062R	247#																	
SVR1	000064R	248#																	
SVR2	000066R	249#																	
SVR3	000070R	250#																	
SVR4	000072P	251#																	
SVP5	000074R	252#																	
SVR6	000076R	253#																	
SWAP,D	006206R	477	495	1072#															
SWITCH	006446R	407	1135#																
SYSCNT	000052R	242#																	
TABLE	000252H	285#	544	560	582	708	728	925											
TABLSP	002016P	340#	766	821	866	884	1160	1245	1261	1266	1275	1279							
TABL51	002026R	342#	772	890	1166	1247	1268	1281											
TABL52	002036P	344#	779	1283															
TABL53	002046R	346#	788	1285															
TOTALE	007152P	791	1278#																
TPMK	003426R	521	609#																
TRER	007020R	712	1229#																
TRPDFD	000022	285#																	
TRY1	002072P	358#	485#	534*	537														
TRY2	002073P	359#	486#	550*	553														
TRY3	002074R	360#	487*	572*	575														
VECTOR	000010R	223#	1217																
WASADR	000104R	257#	937*																
WB	000364P	325#	460*	461	462*	464*	615												
WBSAV	000366R	326#	461*	463*	614														
WBUFEA	000136R	272#	465																
WBUFPA	000134R	271#	616																
WBUFPQ	000140R	273#																	
WBUFSZ	000142P	274#	460																
WDFR	000116R	264#	373*																
WDJO	000114R	263#	374*																
WLIM	002075P	361#	1141																
WREA	000342R	316#	465*	466*	467*	468*	469*	617											
WRIT	007072R	539	1251#																
WRITE	003444R	489	612#																
XFLAG	000005R	221#																	
XMEM	002060R	351#	617*	624*	631*	640*	641*	642*	643*	649*	650								
ZERO	000354R	321#	724																
.	007722R	338#	339#	340#	342#	344#	346#	504											

. APS, 000000 000
 007722 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

PACK:XTMBK0,PACK:XTMBK0/SOL/CRF;SYM=DDXCOM,PACK:XTMBK0
 RUN-TIME: 6 11 1 SECONDS
 RUN-TIME RATIO: 50/19=2.5
 CORE USED: 7K (13 PAGES)