

TS11

TS11 CTRL LGC
CZTSICO

COPYRIGHT (c) 1979-84
AH-E458C-MC
FICHE 01 OF 01

JUL 1984
digital
Made In USA

Table with multiple columns and rows of data, including various numerical values and text labels, likely representing control logic or system parameters. The content is extremely faint and difficult to read.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

.REM \

IDENTIFICATION

PRODUCT CODE: AC-E457C-MC
PRODUCT NAME: CZTSICO TS11 CTRL LGC
PRODUCT DATE: 15 MARCH 1984
MAINTAINER: / DIAGNOSTIC ENGINEERING
AUTHOR: J. MITT

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 DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1979, 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL PDP UNIBUS MASSBUS
DEC DECUS DECTAPE

42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93

USER DOCUMENTATION

USER DOCUMENTATION TABLE OF CONTENTS

GLOSSARY

- 1.0 GENERAL INFORMATION
 - 1.1 PROGRAM ABSTRACT
 - 1.1.1 FUNCTIONAL DESCRIPTION
 - 1.1.2 STRUCTURE OF PROGRAM
 - 1.1.3 MEMORY MAP
 - 1.1.4 DIAGNOSTIC INFORMATION
 - 1.2 SYSTEM REQUIREMENTS
 - 1.2.1 HARDWARE REQUIREMENTS
 - 1.2.2 SOFTWARE REQUIREMENTS
 - 1.3 RELATED DOCUMENTS AND STANDARDS
 - 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
 - 1.5 ASSUMPTIONS
 - 1.6 DIAGNOSTIC HISTORY
- 2.0 OPERATING INSTRUCTIONS
 - 2.1 HARDWARE PARAMETERS
 - 2.2 SOFTWARE PARAMETERS
 - 2.4 EXECUTION TIMES
 - 2.4.1 SYSTEM CONFIGURATION
 - 2.4.2 TEST EXECUTION TIMES
- 3.0 ERROR INFORMATION
 - 3.1 ERROR REPORTING
 - 3.2 ERROR HALTS

```

94          4.0  DEVICE INFORMATION TABLES
95
96          4.1  DIAG REGISTERS
97          4.2  GENERAL
98          4.3  UNIBUS INTERFACE SPECIFICATIONS
99          4.4  BIT DEFINITIONS FOR TS11/TS04 REGISTERS
100
101          5.0  TEST SUMMARIES
102
103          5.1  TEST 1 - PDP-11/TS11 WRAP TEST
104          5.2  TEST 2 - PDP-11/TS04 WRAP TEST
105          5.3  TEST 3 - SET TS04 CHARACTERISTIC
106          5.4  FORMATTER BOARD DATA WRAP TESTS (TESTS 4-7).
107
108          5.4.1 TRACK ACTIVE/TRACK INACTIVE TESTS (TEST 4)
109
110          5.4.1.1 SUBTEST 1 - TRACK INACTIVE TEST
111          5.4.1.2 SUBTEST 2 - TRACK ACTIVE TEST
112          5.4.1.3 SUBTEST 3 - TRACK INACTIVE TEST
113          5.4.1.4 SUBTEST 4 - TRACK ACTIVE TEST
114
115          5.4.2 P.E. DATA TESTS (TEST 5)
116
117          5.4.2.1 SUBTEST 1 - P.E. DATA TEST/0 PATTERN
118          5.4.2.2 SUBTEST 2 - P.E. DATA TEST/1 PATTERN
119          5.4.2.3 SUBTEST 3 - P.E. DATA TEST/SHIFTING 1 PATTERN
120          5.4.2.4 SUBTEST 4 - P.E. DATA TEST/SHIFTING 0 PATTERN
121
122          5.4.3 PE SKEW TEST (TEST 6)
123
124          5.4.3.1 SUBTEST 1 - P.E. SKEW TEST
125          5.4.3.2 SUBTEST 2 - P.E. SKEW TEST
126
127          5.4.4 DEAD TRACK LOGIC TEST (TEST 7)
128
129          5.4.4.1 SUBTEST 1 - P.E. DEAD TRACK TEST
130          5.4.4.2 SUBTEST 2 - P.E. DEAD TRACK TEST
131
132          5.5  TEST 8 - ROM LOOKUP TABLE TEST.
133          5.6  TEST 9 - INLINE MICRO DIAG TEST
134          5.7  TEST 10 - INIT MICRO DIAG TEST

```

135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169

GLOSSARY

ACT	AUTOMATED COMPUTER TEST
APT	AUTOMATED PRODUCT TEST SYSTEM USED IN MANUFACTURING.
BYTE/RECORD/FILE COUNT BRF	IS STORED IN THE 4TH WORD OF THE COMMAND PACKET AND IT'S USE BY THE TS04 DEPENDS ON THE TYPE OF COMMAND.
CMD	TS04 COMMAND
COMMAND PACKET CMDPKT	FOUR WORD PACKET IN THE CPU MEMORY WHICH CONTAINS ALL INFORMATION NEEDED BY THE TS04 TO EXECUTE A COMMAND.
EXTENDED STATUS	FOUR WORDS OF TS04 STATUS WHICH ARE TRANSFERRED AS PART OF THE MESSAGE PACKET AT THE COMPLETION OF A COMMAND.
MESSAGE PACKET	SEVEN WORD PACKET IN THE CPU MEMORY INTO WHICH THE TS04 STORES STATUS AT THE COMPLETION OF A COMMAND.
PC	PROGRAM COUNTER
PSW	PROCESSOR STATUS WORD
RESIDUAL FRAME COUNT RFC	THIS COUNT IS PART OF THE MESSAGE PACKET AND CONTAINS THE NUMBER OF BYTES/RECORDS /FILES REMAINING TO BE PROCESSED AT THE COMPLETION OF A COMMAND.

170		
171	TERMINATION CLASS CODE	THREE BIT CODE IN THE TSSR WHICH INDI-
172	TCC	CATES THE TYPE OF COMMAND TERMINATION.
173		
174	TSBA	TAPE SYSTEM BUS ADDRESS REGISTER.
175		
176	TSDB	TAPE SYSTEM DATA BUFFER REGISTER.
177		
178	TSSR	TAPE SYSTEM STATUS REGISTER.
179		
180	XST0	EXTENDED STATUS REGISTER 0
181		
182	XST1	EXTENDED STATUS REGISTER 1
183		
184	XST2	EXTENDED STATUS REGISTER 2
185		
186	XST3	EXTENDED STATUS REGISTER 3
187		
188	XXDP.	XXDP. IS A "CATCH-ALL" NAME FOR A GROUP OF PDP-11
189		DIAGNOSTIC PACKAGES AVAILABLE ON MULTIMEDIA.

190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
2131.0 GENERAL INFORMATION
-----1.1 PROGRAM ABSTRACT
-----1.1.1 FUNCTIONAL DESCRIPTION

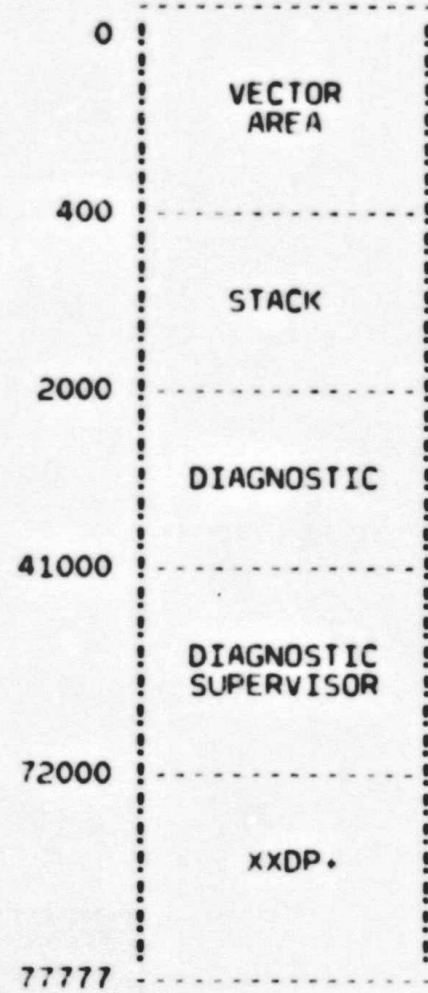
THIS PROGRAM PERFORMS CORRECTIVE MAINTENANCE BY EXECUTING TS11
AND TSO% WRAPAROUNDS FOR THE PURPOSE OF IDENTIFYING FAILED
MODULES.

1.1.2 STRUCTURE OF PROGRAM

THIS DIAG IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE USER, BUT
IT CONTAINS A CONTROL MODULE RELEASED INDEPENDENTLY AS A DIAG SUPERVISOR

214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264

1.1.3 MEMORY MAP



FREE MEMO SPACE RESIDES BETWEEN THE DIAG AND THE SUPERVISOR.

1.1.4 DIAGNOSTIC INFORMATION

THIS DIAGNOSTIC TESTS ONE UNIT AT A TIME, BUT WILL SEQUENTIALLY TEST UP TO 4 UNITS. THE 4 UNITS ARE ASSIGNED LOGICAL UNIT NUMBERS 0 - 3 BY THE DIAGNOSTIC. THE UNITS DO NOT HAVE TO BE ON LINE AND A TAPE DOES NOT HAVE TO BE LOADED TO RUN THIS DIAGNOSTIC.

265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306RECOMMANDATION:

IT IS RECOMMENDED TO RUN THIS DIAG WITH NO TAPE LOADED BECAUSE OF MUCH FASTER EXECUTION TIME (SEE 2.4).

1.2 SYSTEM REQUIREMENTS
-----1.2.1 HARDWARE REQUIREMENTS

PDP-11 PROCESSOR WITH 16K OR MORE OF MEMORY
CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
PROGRAM LOAD DEVICE

1.2.2 SOFTWARE REQUIREMENTS

DIAGNOSTIC SUPERVISOR

1.3 RELATED DOCUMENTS AND STANDARDS

XXDP+ USERS MANUAL MD-11-CHQUS
DIAGNOSTIC SUPERVISOR PROGRAM LISTING
PDP-11 DIAGNOSTIC SUPERVISOR INTERFACE SPECIFICATION.
PDP-11 DIAGNOSTIC SUPERVISOR PROGRAMMER'S GUIDE
TS11/TS04 PROGRAMMING SPECIFICATION.
TS04/TS11 ENGINEERING SPECIFICATION.
TS11/TS04 COMMAND PACKET SPECIFICATION.

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THIS PROGRAM SHOULD BE RUN BEFORE ANY OTHER HOST CPU DIAGNOSTIC PROGRAM.

307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE SUBSYSTEM BEING TESTED IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, MEMORY, ETC., DO NOT FUNCTION PROPERLY.

1.6 DIAGNOSTIC HISTORY

- REV A - FEB 79
- ORIGINAL RELEASE
- REV B - AUG 79
- ADDED AN INTERRUPT TEST
- REVISED ALL TESTS BECAUSE OF AN HARDWARE ECO:
WATCH FOR TAPE MOTION DURING A LOAD SEQUENCE RESULTING FROM AN INIT SENT TO THE TS11.
- CONVERTED DIAG TO SUPERVISOR REV: C.
- ADDED SEVERAL SECTIONS: PROTECT TABLE, AUTO-DROP CODE, HARD CODED PARAMETER TABLE...
- REV C - MAR 84
- CORRECTED MODULE CALLOUT FOR ERROR 17. TEST 7 (USUALLY IS ONE OF G157 BOARDS - IDENTIFIED AS M8922)

332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
3672.0 OPERATING INSTRUCTIONS
-----2.1 HARDWARE PARAMETERS

ON A START COMMAND, THE DIAG ASKS: "CHANGE HW?".
ON A "N" ANSWER, THERE SHALL BE NO HARDWARE DIALOGUE AND THE
DIAG SHALL RUN ASSUMING A UNIT AT TSSR = 177522 WITH VECTOR OF
224.

ON A "Y" ANSWER TO "CHANGE HW?" QUESTION, THEN AND ONLY THEN,
THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE
VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT
VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

TSSR ADDRESS (172522) ?

VECTOR (224) ?

THE VALIDITY OF THESE PARAMETERS CAN BE CHECKED BEFORE RUNNING
THE TESTS BY SETTING THE FLAG "ADR" ON A STA, RES, OR CON COMMAND.
THE SO CALLED AUTO-DROP CODE SHALL THEN BE EXECUTED AFTER THE INIT
CODE AND BEFORE THE HARDWARE TESTS ARE RUN.
THAT CODE FIRST TESTS THE ADDRESS OF THE TSSR. IF NO RESPONSE FROM
THE INTERFACE, THE UNIT IS DROPPED IMMEDIATELY WITH THE FOLLOWING
MESSAGE:

BUS TRAP AT XXXXXX
INTERFACE BAD OR NOT SETTO ABOVE AD

(XXXXXX = TSSR AD)

ON A RESPONSE FROM THE INTERFACE, THE READY STATUS OF THE
UNIT IS CHECKED. IF NOT READY, THE UNIT IS DROPPED IMMEDIATELY.

368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397

2.2 SOFTWARE PARAMETERS

THE FOLLOWING QUESTION WILL BE ASKED ON REQUEST - I.E., ANSWERING
"Y" TO "CHANGE SW (L)?" QUESTION - ON A START, RESTART OR CONTINUE COMMA

ENABLE DATA COMPARE ERROR PRINTS FOR TESTS 4-7(L) N ?

IF "Y" IS THE RESPONSE TO THIS SOFTWARE QUESTION, THEN WILL DATA
COMPARE ERRORS BE PRINTED, PROVIDED IER FLAG IS RESET.

2.4 EXECUTION TIMES

2.4.1 SYSTEM CONFIGURATION

PDP11/70
MOS MEMORY
LA36
TS11/TS04

2.4.2 TESTS EXECUTION TIMES

TAPE NOT LOADED:	40	SEC
TAPE LOADED, UNIT OFF-LINE:	90	SEC
TAPE LOADED, UNIT ON-LINE:	145	SEC

398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
4463.0 ERROR INFORMATION
-----3.1 ERROR REPORTING

- 1 - COMMAND PACKET ADR NOT ON MODULO 4 BOUNDARY
- 2 - TSO4 NOT READY-SSR NOT SET
- * 3 - PDP11-TS11 WRAP FAILURE
- * 4 - PDP11-TS11 WRAP FAILURE ON TSSR EXT ADDR BITS
- * 5 - PDP11-TSO4 WRAP FAILURE-TSBA INCORRECT
- * 6 - PDP11-TSO4 WRAP FAILURE-TSSR INCORRECT
- 7 - TRACK ACTIVE NOT 0 FOR 1 OR MORE TRACKS IN TEST4 SUB1
- 8 - TRACK ACTIVE NOT 0 FOR 1 OR MORE TRACKS IN TEST4 SUB3
- 9 - TRACK ACTIVE NOT 1 FOR 1 OR MORE TRACKS IN TEST4 SUB2
- 10 - TRACK ACTIVE NOT 1 FOR 1 OR MORE TRACKS IN TEST4 SUB4
- 11 - TRACK ACTIVE ERROR
- 12 - PE WRAP DATA ON 0'S PATTERN
- 13 - PE WRAP DATA ERROR ON 1'S PATTERN
- 14 - PE WRAP DATA ERROR ON SHIFTING 1 PATTERN
- 15 - PE WRAP DATA ERROR ON SHIFTING 0 PATTERN
- 16 - PE SKEW ERROR
- 17 - PE DEAD TRACK ERROR
- 18 - ROM LOOKUP TABLE ERROR
- 19 - MICRO DIAGNOSTIC ERROR
- * 20 - SET CHAR ERROR-TSSR. NBA NOT SET ON COMMANDS BEFORE SET CHAR ISSUED.
- * 21 - SET CHAR ERROR-TSSR. NBA NOT CLEARED WHEN SET CHAR ISSUED
- * 22 - SET CHAR ERROR-TSBA NOT POINTING PAST MSG PACKET
- * 23 - SET CHAR ERROR-MSG PACKET CONTENTS IN DOUBT
- * 24 - SET CHAR ERROR-I/O SEQ CROM PARITY ERR (FC=1)
- * 25 - SET CHAR ERROR-SERIAL BUS PARITY ERR (FC=2)
- * 26 - SET CHAR ERROR-SILO PARITY ERR (FC=2)
- * 27 - SET CHAR ERROR-UP CROM PARITY ERROR OR FATAL MICRO ERROR HALTS (FC=2)
- * 28 - SET CHAR ERROR-AC DETECTED LOW AT TSO4 (FC=3)
- * 29 - TS11 DID NOT DROP READY UPON COMMAND ISSUANCE
- 30 - PE DATA ERROR
- 31 - NO INTERRUPT

* = DEVICE FATAL ERRORS - CALL BAD TS11 BOARD OR IO BOARDS

OTHER ERRORS ARE HARD ERRORS

NOTE: EXPECTED AND ACTUAL DATA AND/OR PRINTOUTS WILL OCCUR WITH SOME OF THE ABOVE MESSAGES WHEN APPLICABLE.

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

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION
 WITH /FLAG:MOE. THERE ARE NO OTHER HALTS.

4.0 DEVICE INFORMATION TABLES

4.1 "DIAG" REGISTERS

;THE FOLLOWING ARE REGISTER AND BIT DEFINITIONS FOR THE TS04
 ;REGISTERS OF INTEREST IN THIS DIAGNOSTIC.

;FMCTLO - FORMATTER MAJOR STATE CONTROL REG
 FMCTLO=4
 ;WRITE REGISTER 4

;THIS REGISTER IS SET UP FOR THE PORTION OF THE
 ;RECORD WE ARE PRESENTLY READING

FC.RD= 200 ;WE ARE DOING NORMAL READ. IF 0, WE DISABLE SOME ERROR
 ;CORRECTION LOGIC SO WE'RE MORE DISCRIMINATING
 ;FOR READ AFTER WRITE.
 FC.FLO=100 ;SETTING THIS BIT CAUSES .FMFLO ON BBUS TO BE TRUE
 ;(NEEDED BY THE RD PE ROUTINE)
 FC.DAT=10 ;DATA MODE
 FC.PRE=4 ;PREAMBLE MODE
 FC.VCO=2 ;VCO SYNC MODE
 FC.NRZ=1 ;NRZI MODE (FORCE SKEW WINDOW TO STAY OPEN)
 ;CAUSES FMTAUI TO BE CLRED WHEN YOU WRITE
 ;TO THE FMCLDO REGISTER.

;RDCTLO - READ CONTROL REGISTER
 RDCTLO=20
 ;WRITE REGISTER 20

;THIS REGISTER CONTROLS THE FORMATTER MODE AND THRESHOLD.

RD.REV=200 ;1 FOR REV MOTION, 0=FWD
 RD.MAI=100 ;I/O FORMATTER DATA WRAPAROUND
 RD.SPC=40 ;WE ARE SPACING RECORDS. (THIS BIT IS UNUSED IN
 ;THE HARDWARE BUT IS A SOFTWARE FLAG IN THE READ CODE)
 RD.SKP=20 ;WE ARE SKIPPING FILES. (THIS BIT IS UNUSED IN
 ;THE HARDWARE BUT IS A SOFTWARE FLAG IN THE READ CODE)

498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547

```

;THE FOLLOWING THRESHOLDS ARE AVAILABLE:
;NORM USE          DIAG USE
RD.110-7          ;
RD.90-6           ;
RD.75-5           ;
RD.68-4           ;
RD.40-3           ;NRZ WRT
RD.20-2           ;NRZ RD, PE WRT
RD.12-1           ;PE READ, NRZ ERR RECOV RD/WRT CROSSTALK
RD.07-0           ;PE ERR RECOV          ERASE FUNCTION
;DATA PORTION ONLY

```

.....

```

;IOSCO - I/O SEQUENCER SILO CONTROL BUFFER OUT
IOSCO=14
;WRITE          REGISTER 14

```

```

;THIS REGISTER CONTAINS THE SILO CONTROL BITS FOR DATA WRITTEN
;BY THE MAIN OR I/O MICRO TO THE SILO. THE DATA IN THIS REG
;IS PAIRED WITH THE IOSDO REG AND PUT IN THE SILO WHENEVER
;THE IOSICO REG IS CLOCKED. THIS REGISTER NEED ONLY BE WRITTEN
;ONCE IF THE SAME OLD DATA IS OK TO BE WRITTEN IN THE SILO.
;NOTE THAT IF THE I/O IS WRITING THE SILO, THE MAIN MUST PUT EVEN
;PARITY IN THE BITS 357 OR THE I/O WILL WRITE PAR ERRS IN THE SILO.
;(THE IS.DAP BIT IS DON'T CARE HERE FOR I/O WRITING THE SILO)

```

```

IS.PAR= 200      ;ODD PAR BIT FOR ALL 16 BITS (CNRTL AND DATA)
;NOTE THAT THE BITS MASKED BY 357 MUST BE
;EVEN PARITY BECAUSE THE 9 DATA BITS ARE ODD

IS.IVP= 100      ;INVERT CNTRL SILO PAR BIT BEFORE MOVING
;TO WRITE BOARD (WRITE EVEN PARITY ON TAPE)

IS.NRZ= 40       ;INVERT WRITE BUFFER BIT IF ASSOCIATED SILO
;DATA BIT IS A 1. (IF IS.NRZ=0, WE'RE IN PE MODE)

IS.DAP= 20       ;ODD PARITY FOR THE 8 DATA BITS IN IOSDO

IS.LRC= 10       ;CAUSES SYNCHRONOUS CLR ON WRT BOARD TO WRITE THE LRC CHA
IS.WRF= 4        ;THIS FLG BIT SHOWS UP AT THE WRITE BOARD
;WITH THE CORRESPONDING DATA. THE FUNCTION OF
;THE BIT WILL BE DEFINED BY THE WRITE BOARD.
;NOTE HOWEVER THAT IF THE WRITE BOARD SEES
;THE BIT 1, THE PA.WRF BIT IN THE
;PRATNI REG WILL ALSO BE 1 (IF ENABLED
;TO AFFECT THE ATTN REG).
;THE 2 LOW BITS ARE WRITABLE AND AFFECT THE PARITY TREES BUT
;OTHERWISE ARE UNIMPLEMENTED

```

.....

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

4.2 GENERAL

THE TS04 TAPE SUBSYSTEM CONSISTS OF A TS11 UNIBUS TO SERIAL BUS CONTROLLER CONNECTED TO A TS04 DRIVE. FROM A SOFTWARE VIEWPOINT THIS CONFIGURATION IS UNIQUE (FOR A UNIBUS DEVICE) IN A NUMBER OF WAYS:

- A. ONLY ONE REGISTER MAY BE WRITTEN - TSDB (TAPE SYSTEM DATA BUFFER).
- B. TWO REGISTERS MAY BE READ - TSSR AND TSBA (TAPE SYSTEM STATUS REGISTER AND TAPE SYSTEM BUS ADDRESS REGISTER).
- C. COMMANDS ARE NOT WRITTEN TO THE DRIVE; RATHER, COMMAND POINTERS ARE WRITTEN WHICH POINT TO COMMAND PACKETS SOMEWHERE IN CPU MEMORY. THE COMMAND POINTER IS USED BY THE TS04 SUBSYSTEM TO FETCH THE WORD(S) WITHIN THE COMMAND PACKET. THE WORDS WITHIN THE COMMAND PACKET ARE:
 - 1. COMMAND WORD
 - 2. LOW ORDER BUFFER ADDRESS
 - 3. HIGH ORDER BUFFER ADDRESS
 - 4. BYTE COUNT
- D. THE TSSR CONTAINS ALL THE INFORMATION WHICH WILL BE NECESSARY TO DETERMINE WHETHER:
 - 1. THE DRIVE IS READY TO ACCEPT ANOTHER COMMAND.
 - 2. THE PREVIOUS COMMAND WAS EXECUTED WITHOUT ERROR.IF EITHER OF THE ABOVE CONDITIONS IS UNTRUE AT "JOB DONE" OR "COMMAND INITIATION" TIME, IT MAY BE NECESSARY TO GET THE EXTENDED STATUS REGISTERS TO DETERMINE WHAT ACTION IS TO BE TAKEN AND/OR LOG THE ERROR INFORMATION.
- E. EXTENDED STATUS REGISTERS ARE NOT READ DIRECTLY FROM DRIVE REGISTERS; RATHER, A "GET STATUS" COMMAND IS ISSUED WHICH WILL CAUSE THE TS04 TO TRANSFER EXTENDED STATUS INFORMATION TO THE MEMORY AREA POINTED TO BY THE BUFFER ADDRESS OF THE "GET STATUS" COMMAND. THERE ARE FOUR EXTENDED STATUS REGISTERS. SEE 6.3.
- F. THE TSDB MUST BE WRITTEN WITH A DATO INSTRUCTION TO PROPERLY WRITE THE COMMAND POINTER. A DATOB WILL CAUSE A MAINTENANCE FUNCTION. A DATO TO THE TSSR WILL CAUSE SUBSYSTEM INIT.
- G. COMMAND PACKETS MUST RESIDE ON DIVIDE BY FOUR MEMORY BOUNDARIES (AS OPPOSED TO DIVIDE BY 2 OR WORD BOUNDARIES).

600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620

4.3 UNIBUS INTERFACE SPECIFICATIONS

TS11/ TS04 -----	INT. VECTOR -----	UNIBUS ADDRESS -----	REGISTER -----
FIRST	224	772520 772522	TSBA/TSDB TSSR
SECOND	154	772524 772526	TSBA/TSDB TSSR
THIRD	160	772530 772532	TSBA/TSDB TSSR
FOURTH	164	772534 772536	TSBA/TSDB TSSR

621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676

4.4 BIT DEFINITIONS FOR TS11/TS04 REGISTERS

4.4.1 TS11/TS04 REGISTER SUMMARY

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	
TSBA	!A15!	!A14!	!A13!	!A12!	!A11!	!A10!	!A09!	!A08!	!A07!	!A06!	!A05!	!A04!	!A03!	!A02!	!A01!	
TSDB	!P15!	!P14!	!P13!	!P12!	!P11!	!P10!	!P09!	!P08!	!P07!	!P06!	!P05!	!P04!	!P03!	!P02!	!P01!	
TSSR	!SC!	!UPE!	!SPE!	!RMR!	!NXM!	!NBA!	!A17!	!A16!	!SSR!	!OFL!	!FC1!	!FC0!	!TC2!	!TC1!	!TC0!	
XST0	!TMK!	!RLS!	!LET!	!RLL!	!WLE!	!NEF!	!ILC!	!ILA!	!MOT!	!ONL!	!IE!	!VCK!	!PED!	!WLK!	!BO!	
XST1	!DLT!		!COR!	!CRS!	!TIG!	!DBF!	!SCK!		!IPR!	!SYN!	!IPO!	!IED!	!POS!	!POL!	!UN!	
XST2	!OPM!	!SIP!	!BPE!	!CAF!		!WCF!		!DTP!	!DT7!	!DT6!	!DT5!	!DT4!	!DT3!	!DT2!	!DT1!	
XST3	MICRO DIAGNOSTIC ERROR CODE							!LMX!	!OPI!	!REV!	!CRF!	!DCK!	!NOI!	!LX!		

TERMINATION CLASS CODES (TSSR TC0-TC2):

- 0 = NORMAL TERMINATION
- 1 = ATTENTION CONDITION
- 2 = TAPE STATUS ALERT
- 3 = FUNCTION REJECT
- 4 = RECOVERABLE ERROR - TAPE POSITION = ONE RECORD
DOWN TAPE FROM START OF FUNCTION
- 5 = RECOVERABLE ERROR - TAPE NOT MOVED
- 6 = UNRECOVERABLE ERROR - TAPE POSITION LOST
- 7 = FATAL CONTROLLER ERROR

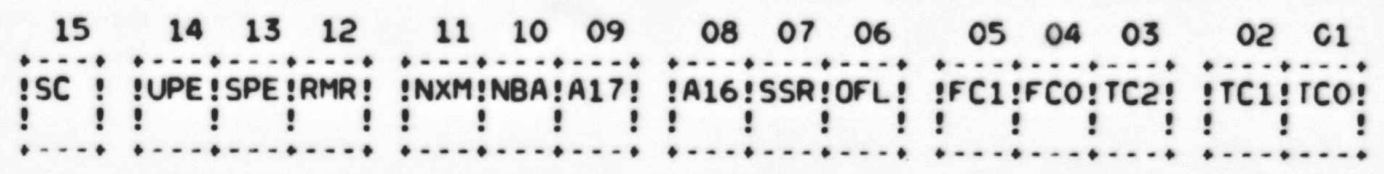
FATAL CLASS CODES (TSSR FC0-FC1):

- 0 = MICRO DIAGNOSTIC FAILURE (DISPLAYED IN TS04 OPERATOR PANEL AND
- 1 = I/O SEQUENCER CROM PARITY ERROR.
- 2 = MICROPROCESSOR CROM PARITY ERROR.
SILO PARITY ERROR.
SERIAL BUS PARITY ERROR DETECTED AT TS11 (SPE).
SERIAL BUS PARITY ERROR DETECTED AT TS04 (BPE).
FATAL ERROR HALTS 1750-1777 IN TS04 PROGRAM COUNTER DISPLAY.
- 3 = LOSS OF AC POWER HAS BEEN DETECTED.

677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728

4.4.2 TS11 STATUS REGISTER (TSSR)

UNIBUS ADDRESS + 2 - READ ONLY



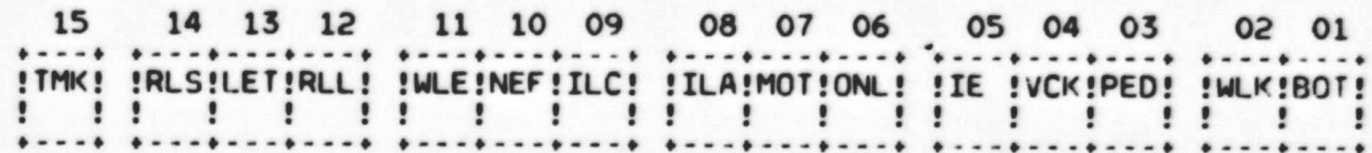
BIT	NAME	TCC	DEFINITION
15	SC	S	SPECIAL CONDITION. WHEN SET, INDICATES THAT THE LAST COMMAND DID NOT COMPLETE WITHOUT INCIDENT. SPECIFICALLY, EITHER AN ERROR WAS DETECTED OR AN EXCEPTION CONDITION OCCURRED. EXCEPTION CONDITIONS CAN BE TAPE MARKS ON READ COMMANDS, REVERSE MOTION AND AT BOT, EOT WHILE WRITING, ETC. MAY ALSO BE SET BY THE ERROR BITS CONTAINED IN THE TSSR REGISTER: UPE, SPE, RMR, AND NXM. THE TERMINATION CLASS BITS ARE SOMET OTHER THAN 0 (UNLESS RMR IS THE ONLY ERROR - SEE RM
14	UPE	4/5	UNIBUS PARITY ERROR. SET BY THE TS11 WHEN IT DETECTS A PARITY ERROR ON THE UNIBUS DATA WHEN TRANSFERRING TO OR FROM THE CPU'S MEMORY.
13	SPE	7	SERIAL BUS PARITY ERROR. THIS BIT IS SET BY THE TS11 WHEN IT DETECTS A SERIAL BUS PARITY ERROR ON DATA RECEIVED FROM THE TSO4.
12	RMR	S	REGISTER MODIFICATION REFUSED. SET BY THE TS11 WHEN A COMMAND POINTER IS LOADED INTO T508 AND SUB-SYSTEM READY (SSR) IS NOT SET. NOTE THAT THIS BIT CAUSES SPECIAL CONDITION BUT NO TERMINATIO CLASS (IN FACT, THE TSO4 NEVER SEES THIS ERROR) BECAUSE ON A SYSTEM WITH NO BUGS, THIS BIT MAY COME UP ON AN ATTENTION MESSAGE. IF ATTN5 ARE NOT ENABLED, THIS BIT COMING UP IS AN INDICATION OF EITHER A FATAL CONTROLLER ERROR OR A SOFTWARE BUG.
11	NXM	4/5	NON-EXISTENT MEMORY. SET BY THE TS11 WHEN TRYING TO TRANSFER TO OR FROM A MEMORY LOCATION WHICH DOES NOT EXIST. MAY OCCUR WHEN FETCHING THE COMMAND PACKET, FETCHING OR STORING DATA, OR STORING THE MESSAGE PACKET.

729	10	NBA	S	NEED BUFFER ADDRESS. WHEN SET, INDICATES THAT THE TS04 NEEDS A MESSAGE BUFFER ADDRESS. THIS BIT IS CLEARED DURING THE SET CHARACTERISTICS COMMAND (IF A GOOD ADDRESS WAS GIVEN).
730				
731				
732				
733				
734	09	A17	S	BUS ADDRESS BIT 17. A17 AND A16 (BIT 08) TRACK THE VALUES OF BITS 17 AND 16 OF THE TSBA REGISTER.
735				
736				
737				
738	08	A16	S	BUS ADDRESS BIT 16. SEE A17 (BIT 09).
739				
740	07	SSR	S	SUB-SYSTEM READY. WHEN SET, INDICATES THAT THE TS11/TS04 SUBSYSTEM IS NOT BUSY AND IS READY TO ACCEPT A NEW COMMAND POINTER.
741				
742				
743				
744	06	OFL	S,1,3	OFF-LINE. WHEN SET, INDICATES THAT THE TS04 IS OFF-LINE AND UNAVAILABLE FOR ANY TAPE MOTION COMMANDS. THIS BIT CAN CAUSE A TERMINATION CLASS OF 1 (ON ATTN INTERRUPT) OR 3 (RESULTS IN NEF).
745				
746				
747				
748				
749	05	FC1	7	FATAL TERMINATION CLASS 01. FC1 AND FC0 (BIT 04) ARE USED TO INDICATE THE TYPE OF FATAL ERROR WHICH HAS OCCURRED ON THE TS04. THESE BITS ARE VALID ONLY WHEN SC IS SET AND THE TERMINATION CLASS CODE BITS ARE ALL SET (111).
750				
751				
752				
753				
754				
755	04	FC0	7	FATAL TERMINATION CLASS 00. SEE FC1 (BIT 05).
756				
757	03	TC2	S	TERMINATION CLASS BIT 02. THIS BIT, ALONG WITH THE TC1 AND TC0 BITS, ACT AS AN OFFSET VALUE WHENEVER AN ERROR OR EXCEPTION CONDITION OCCURS ON A COMMAND. EACH OF THE EIGHT POSSIBLE VALUES OF THIS FIELD REPRESENT A PARTICULAR CLASS OF ERRORS OR EXCEPTIONS. THE CONDITIONS IN EACH CLASS HAVE SIMILAR SIGNIFICANCE AND, AS APPLICABLE, RECOVERY PROCEDURES. THE CODE PROVIDED IN THIS FIELD IS EXPECTED TO BE UTILIZED AS AN OFFSET INTO A DISPATCH TABLE FOR HANDLING OF THE CONDITION.
758				
759				
760				
761				
762				
763				
764				
765				
766				
767				
768				
769	02	TC1	S	TERMINATION CLASS BIT 01. SEE TC2 (BIT 03).
770				
771	01	TC0	S	TERMINATION CLASS BIT 00. SEE TC2 (BIT 03).
772				
773	00	-	-	NOT USED.
774				
775				
776				

UNIBUS ADDRESS * 2 - WRITE ONLY
 SUBSYSTEM INITIALIZE

777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821

4.4.3 EXTENDED STATUS REGISTER 0 (XSTAT0)



BIT	NAME	TCC	DEFINITION
15	TMK	5,2	TAPE MARK DETECTED. SET WHENEVER A TAPE MARK WAS DETECTED DURING A 'READ, SPACE, OR SKIP COMMAND AND AS A RESULT OF THE WRITE TAPE MARK OR WITE TAPE MARK RETRY COMMANDS.
14	RLS	2	RECORD LENGTH SHORT. THIS BIT INDICATES THAT EITHER THE RECORD'S LENGTH WAS SHORTER THAN THE BYTE COUNT ON READ OPERATIONS, A SPACE RECORD OPERATION ENCOUNTERED A TAPE MARK OR BOT BEFORE THE POSITION COUNT WAS EXHAUSTED, OR A SKIP TAPE MARKS COMMAND WAS TERMINATED BY ENCOUNTERING BOT OR A DOUBLE TAPE MARK (IF THAT OPERATIONAL MODE IS ENABLED, SEE LET) PRIOR TO EXHAUSTING THE POSITION COUNTER.
13	LET	2	LOGICAL END OF TAPE. SET ONLY ON THE SKIP TAPE MARKS COMMAND WHEN EITHER TWO CONTIGUOUS TAPE MARKS ARE DETECTED OR WHEN MOVING OFF OF BOT AND THE FIRST RECORD ENCOUNTERED IS A TAPE MARK. THE SETTING OF THIS BIT WILL NOT OCCUR UNLESS THIS MODE OF TERMINATION IS ENABLED THROUGH USE OF THE SET CHARACTERISTICS COMMAND.
12	RLL	2	RECORD LENGTH LONG. WHEN SET, THIS BIT INDICATES THAT THE RECORD READ WAS LONGER THAN THE BYTE COUNT SPECIFIED.
11	WLE	3,6	WRITE LOCK ERROR. WHEN SET, INDICATES THAT A WRITE OPERATION WAS ISSUED BUT THE MOUNTED TAPE DID NOT CONTAIN A WRITE ENABLE RING OR THE WRT LOCK SWITCH ACTIVATED DURING THE OPERATION.

822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877

10 NEF 3

NON-EXECUTABLE FUNCTION. WHEN SET, INDICATES THAT THE COMMAND COULD NOT BE EXECUTED DUE TO ONE OF THE FOLLOWING CONDITIONS:

- THE COMMAND SPECIFIED REVERSE TAPE DIRECTION BUT THE TAPE WAS ALREADY POSITIONED AT BOT.
- THE ISSUING OF ANY COMMAND, EXCEPT REWIND, UNLOAD, OR A COMMAND WITH THE CLEAR VOLUME CHECK (CVC) BIT SET, WHEN THE VOLUME CHECK BIT IS SET.
- ANY COMMAND, EXCEPT GET STATUS OR DRIVE INITIALIZE, WHEN THE TS04 IS OFF-LINE.
- ANY WRITE COMMAND WHEN THE TAPE DOES NOT CONTAIN A WRITE ENABLE RING (WRITE LOCK STATUS - WLS).

09 ILC 3

ILLEGAL COMMAND. SET WHEN A COMMAND IS ISSUED AND EITHER ITS COMMAND FIELD OR ITS COMMAND MODE FIELD CONTAINS CODES WHICH ARE NOT SUPPORTED BY THE TS04.

08 ILA 3

ILLEGAL ADDRESS. (MORE THAN 18 BITS OR ODD WHEN AN EVEN ADDRESS IS REQUIRED.)

07 MOT S

TAPE IS MOVING.

06 ONL S

ON LINE. WHEN SET, INDICATES THAT THE TS04 IS ON-LINE AND OPERABLE.

05 IE S

INTERRUPT ENABLE. REFLECTS THE STATE OF THE INTERRUPT ENABLE BIT SUPPLIED ON THE LAST COMMAND.

04 VCK S

VOLUME CHECK. WHEN SET, INDICATES THAT THE DRIVE HAS BEEN EITHER POWERED DOWN OR TURNED OFF-LINE. CLEARED BY THE CLEAR VOLUME CHECK (CVC) BIT IN THE COMMAND HEADER WORD. THIS BIT CAN CAUSE A TERMINATION CLASS OF 3.

03 PED S

PHASE ENCODED DRIVE. WHEN SET, INDICATES THAT THE TS04 IS CAPABLE OF READING AND WRITING ONLY 1600 BPI PHASE ENCODED DATA. WHEN RESET, INDICATES THAT THE TS04 HAS ONLY 800 BPI NRZI DATA CAPABILITIES.

02 WLK S,3

WRITE LOCKED. WHEN SET, INDICATES THAT THE MOUNTED REEL OF TAPE DOES NOT HAVE A WRITE-ENABLE RING INSTALLED. THE TAPE IS, THEREFORE, WRITE PROTECTED.

01 BOT S,3

BEGINNING OF TAPE. WHEN SET, INDICATES THAT THE TAPE IS POSITIONED AT THE LOAD POINT AS DENOTED BY THE BOT REFLECTIVE STRIP ON THE TAPE.

J2

SVC.MLB SOURCE FILE MACY11 30(1046) 09-APR-84 14:40 PAGE 24
CZTSIC.P11 09-APR-84 14:37 MISCNTOP: GPRM COUNT OPTION

SEQ 0022

878
879
880
881
882

00 EOT S.2

END OF TAPE. THIS BIT IS SET WHENEVER THE TAPE IS POSITIONED AT OR BEYOND THE END OF TAPE REFLECTIVE STRIP. DOES NOT RESET UNTIL THE TAPE PASSES OVER THE REFLECTIVE STRIP IN THE REVERSE DIRECTION UNDER PROGRAM CONTROL.

883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938

4.4.4 EXTENDED STATUS REGISTER 1 (XSTAT1)

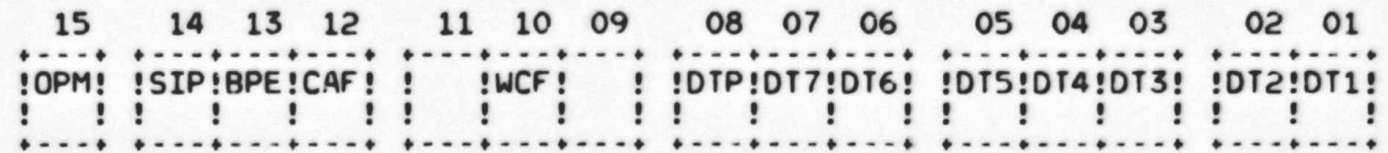


BIT	NAME	TCC	DEFINITION
15	DLT	4	DATA LATE. SET WHEN THE I/O SILO IS FULL ON A READ OR EMPTY ON A WRITE. THESE CONDITIONS OCCUR WHENEVER THE UNIBUS LATENCY EXCEEDS THE DATA TRANSFER RATE OF THE TSO4.
14	-	-	NOT USED.
13	COR	5	CORRECTABLE DATA. IN PHASE ENCODED MODE, A CORRECTABLE DATA ERROR HAS BEEN ENCOUNTERED.
12	CRS	4	CREASE DETECTED. FOR NRZI, ALL DATA TRACKS DROPPED OUT FOR MORE THAN THREE CHARACTER TIMES BUT FOR LESS THAN .1 INCHES OF TAPE. FOR PE, EIGHT OUT OF NINE DATA TRACKS WENT DEAD FOR LESS THAN .1 INCHES BEFORE A VALID POSTAMBLE WAS DETECTED.
11	TIG	4	TRASH IN THE GAP. NON-ERASED DATA WAS DETECTED IN A GAP DURING A READ, WRITE, WRITE TAPE MARK, OR ERASE COMMAND.
10	DBF	4	EXCESSIVE SKEW. FOR NRZI, DATA OCCURRED BETWEEN THE 50% MARK AND THE 100% MARK OF THE NRZI DATA WINDOW. FOR PE, IT TOOK MORE THAN FIVE CHARACTERS IN READ-AFTER-WRITE OR TEN CHARACTERS IN READ TO PROPERLY CENTER THE WINDOWS OF THE FORMAT CHANNEL LOGIC.
	NZO	4	NRZ FIFO OVERRUN.
09	SCK	4	SPEED CHECK. TAPE SPEED WAS OFF BY MORE THAN 5% DURING A WRITE DATA OPERATION. NOTE THAT SPEED AVERAGED OVER 8 TICKS AND THE AVERAGE MUST BE OFF 5% TO CAUSE THIS ERROR.
08	-	-	NOT USED.
07	IPR	5,4	INVALID PREAMBLE. SET ON A PE DRIVE IF THE PREAMBLE APPEARS TO BE SHORTER THAN 36 CHARACTERS OR LONGER THAN 44 CHARACTERS. ALSO SET IF THE PREAMBLE IS INCORRECTLY ENCODED BEYOND THE FIFTEENTH CHARACTER IN READ OR THE

939				TENTH CHARACTER IN READ-AFTER-WRITE.
940				
941		06	SYN 4	SYNCH FAILURE. SET ON A PE DRIVE IF THE FORMATTER WAS UNABLE TO ACHIEVE SYNCHRONIZATION IN THE PREAMBLE.
942				
943				
944				
945			DRP 4	NRZ RECORD DROPPED A CHARACTER (THE NEXT CHARACTER WAS TO BE CONSIDERED CRC).
946				
947				
948		05	IPO S,4	INVALID POSTAMBLE. SET ON A PE DRIVE DURING READ OR WRITE IF ANY OF THE FIRST 39 CHARACTERS OF THE POSTAMBLE ARE NOT READ CORRECTLY.
949				
950				
951				
952			ITM S,4	ILLEGAL TAPE MARK FOR NRZ.
953				
954		04	IED 4	INVALID END DATA. FOR PE, EIGHT OUT OF NINE TRACKS WENT DEAD BEFORE THE POSTAMBLE WAS DETECTED.
955				
956				
957			LRO 4	FOR NRZI, DATA WAS NOT DETECTED IN EITHER THE LRCC OR CRCC WINDOWS. (LRC WAS ZERO)
958				
959				
960		03	POS S,4	POSTAMBLE SHORT. SET ON PE DRIVES DURING A READ OR WRITE WHEN LESS THAN 38 ALL-ZEROES CHARACTERS ARE READ FOLLOWING THE ALL-ONES CHARACTER.
961				
962				
963				
964				
965			NZN S,4	NRZ NOISE RECORD (FEWER THAN 13(10) FRAMES).
966				
967		02	POL 4	POSTAMBLE LONG. SET ON PE DRIVES DURING READ OR WRITE OPERATIONS WHEN THE POSTAMBLE EXCEEDS 42 CHARACTERS.
968				
969				
970				
971			LRC 4	LRC ERROR. SET ON NRZI DRIVES WHEN THE LRCC CHARACTER WAS FOUND IN ERROR.
972				
973				
974		01	UNC 4	UNCORRECTABLE DATA. SET ON PE DRIVES WHEN A PARITY ERROR OCCURRED WITHOUT A CORRESPONDING DEAD TRACK INDICATION.
975				
976				
977				
978			CRC 4	CRC ERROR. SET ON NRZI DRIVES WHEN THE CRC CHARACTER WAS FOUND TO BE IN ERROR.
979				
980				
981		00	MTE 4	MULTI-TRACK ERROR. SET ON PE DRIVES WHEN MORE THAN ONE DEAD TRACK OCCURRED IN THE PREAMBLE OR IN THE DATA FIELD.
982				
983				
984				
985			VPE 4	VERTICAL PARITY ERROR. SET ON NRZI DRIVES WHEN A CHARACTER DID NOT CONTAIN AN ODD NUMBER OF ONE BITS.
986				
987				

988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028

4.4.5 EXTENDED STATUS REGISTER 2 (XSTAT2)



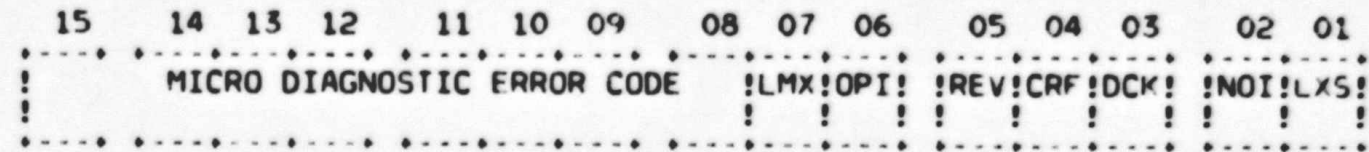
BIT	NAME	TCC	DEFINITION
15	OPM	5	OPERATION IN PROGRESS. (TAPE MOVING)
14	SIP	7	SILO PARITY ERROR. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
13	BPE	7	SERIAL BUS PARITY ERROR AT DRIVE. SET BY THE TS04 WHEN A PARITY ERROR IS DETECTED ON DATA TRANSMITTED FROM THE TS11 TO THE TS04. CAUSES FATAL CLASS 2 BECAUSE THE ERROR MIGHT HAVE OCCURRED DURING THE TRANSMISSION OF THE MESSAGE PACKET.
12	CAF	7	CAPSTAN ACCELERATION FAIL. AFTER ACCELERATING TAPE FOR .2 INCHES, THE TAPE SPEED WAS CHECKED AND FOUND TO BE OUT OF TOLERANCE BY MORE THAN 10%.
11	-	-	NOT USED.
10	WCF	7	DESKEW BUFFER FAIL. ONE OF THE DESKEW BUFFERS FAILED TO ASSERT "OUTPUT READY" WITHIN 20 MICROSECONDS AFTER BEING ENABLED. THE DEAD TRACK BITS WILL INDICATE ON WHICH TRACKS THIS FAILURE OCCURRED.
09	-	-	NOT USED.

1029				
1030				
1031	08	DTP	S	DEAD TRACK PARITY. THE BITS DTP THROUGH DTO
1032				INDICATE WHICH TRACK(S) WENT DEAD, IF ANY,
1033				DURING THE LAST DATA TRANSFER OPERATION. IF
1034				DESKEW BUFFER FAIL (DBF) IS SET, THESE BITS
1035				INDICATE WHICH CHANNEL FAILED.
1036	07	DT7	S	DEAD TRACK 7. SEE DTP.
1037				
1038	06	DT6	S	DEAD TRACK 6. SEE DTP.
1039				
1040	05	DT5	S	DEAD TRACK 5. SEE DTP.
1041				
1042	04	DT4	S	DEAD TRACK 4. SEE DTP.
1043				
1044	03	DT3	S	DEAD TRACK 3. SEE DTP.
1045				
1046	02	DT2	S	DEAD TRACK 2. SEE DTP.
1047				
1048	01	DT1	S	DEAD TRACK 1. SEE DTP.
1049				
1050	00	DT0	S	DEAD TRACK 0. SEE DTP.
1051				
1052				
1053				
1054				

NOTE: ON A SET CHARACTERISTICS COMMAND, THE UCODE LEVEL IS RETURNED IN DT7 THRU DTO. ON A GET STATUS COMMAND, THE RESIDUAL CAPSTAN TICK COUNT (INTERNALLY R7) IS RETURNED IN DT7 THRU DTO.

1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095

4.4.6 EXTENDED STATUS REGISTER 3 (XSTAT3)



BIT	NAME	TCC	DEFINITION
15 TO 08			MICRO DIAGNOSTIC ERROR CODE. (SEE LIST OF CODES BELOW). ALL ERROR CODES IN THE TABLE WILL BE DISPLAYED ON THE TSO4 CONTROL PANEL BUT ONLY CODES HIGHER THAN 110 WILL BE AVAILABLE TO CPU DIAGNOSTICS FOR PRINTOUT IN THE MICRO DIAGNOSTIC ERROR CODE FIELD OF XSTAT3. THIS ERROR CODE FIELD IS VALID ONLY WHEN THE TERMINATION CLASS CODE IN THE TSSR EQUALS 7 AND THE FATAL CLASS CODE IN THE TSSR EQUALS 0, INDICATING AN INTERNAL DIAGNOSTIC FAILURE.
07	NTL	6	LIMIT EXCEEDED. SET WHEN THE TAPE TENSION ARMS HAVE EXCEEDED THEIR ALLOWABLE TRAVEL AND HAVE CAUSED THE ACTIVATION OF THE LIMIT SWITCHES. NO TENSION EXISTS ON THE MOUNTED TAPE.
06	OPI	6	OPERATION INCOMPLETE. SET WHEN A READ, SPACE, OR SKIP OPERATION HAS MOVED 25 FEET OF TAPE WITHOUT DETECTING ANY DATA ON THE TAPE.
05	REV	5	DIRECTION OF CURRENT OPERATION WAS REVERSE (BUT IS 0 IF REWIND OR FORWARD)
04	CRF	7	CAPSTAN RESPONSE FAILURE. A MOTION COMMAND WAS GIVEN TO THE CAPSTAN BUT WE DID NOT GET A TICK BACK WITHIN A REASONABLE AMOUNT OF TIME.

1096				
1097				
1098				
1099				
1100				
1101				
1102				
1103				
1104				
1105				
1106				
1107				
1108				
1109				
1110				
1111				
1112				
1113				
1114				
1115				
1116				
1117				
1118				
1119				
1120				
1121				
1122				
1123				
1124				

	03	DCK	5,6	DENSITY CHECK. SET ON PE DRIVES WHEN A PE IDENTIFICATION BURST WAS NOT DETECTED WHEN MOVING OFF OF BOT. SET ON NRZI DRIVES WHEN A NON-NRZI IDENTIFICATION BURST WAS FOUND WHEN MOVING OFF OF BOT.
	02	NOI	6	NOISE RECORD. SET DURING A READ OR SPACE OPERATION WHEN A BURST OF FLUX CHANGES, WHICH DO NOT QUALIFY AS A RECORD (BUT TOO MANY TO IGNORE), ARE DETECTED: NRZI: AT LEAST TWO CHARACTERS IN A ROW BUT LESS THAN TWELVE, FOLLOWED BY A CHARACTER IN EITHER THE CRCC OR LRCC WINDOWS. PE: AT LEAST 24 CHARACTERS IN A ROW THAT DO NOT QUALIFY AS A TAPE MARK OR A DATA PREAMBLE.
	01	LXS	5	LIMIT EXCEEDED STATICALLY. THIS BIT IS SET ANY TIME THE LIMIT SWITCHES ARE EXCEEDED. THIS BIT CAN ONLY BE CLEARED BY MANUALLY LOADING THE TAPE.
	00	RIB	2	REVERSE INTO BOT. A READ, SPACE, OR SKIP COMMAND ALREADY IN PROGRESS HAS ENCOUNTERED THE BOT MARKER WHEN MOVING TAPE IN THE REVERSE DIRECTION. TAPE MOTION WILL BE HALTED AT BOT.

1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175

4.4.7 MICRO DIAGNOSTIC ERROR CODES

FOLLOWING IS A LIST OF THE ERRORS WHICH ARE DISPLAYED IN THE MICRO DIAGNOSTIC ERROR CODE (XSTAT3 BITS 15 - 08) AND ALSO IN THE LIGHTS ON THE TS04 CONTROL PANEL, DUE TO FAILURES ON THE CAPSTAN BOARD, I/O BOARDS, WRITE BOARD, READ BOARD, OR FORMATTER BOARD. THE MICRO WILL BE IN A TIGHT LOOP IN THE DISPM PROGRAM, WAITING FOR OPERATOR OR CPU INTERVENTION WHILE THE ERROR IS BEING DISPLAYED IN THE CONSOLE LIGHTS. IT IS APPARENT THAT AN ERROR IS BEING DISPLAYED IF THE "UOK" LIGHT IS NOT LIGHTED, THE PROCESSOR IS NOT STOPPED, AND AN OCTAL NUMBER (100-377) IS BEING DISPLAYED IN THE LIGHTS. TO SCOPE LOOP THESE TESTS, ENTER MAINTENANCE MODE (ON-LINE SWITCH TO "OFF" POSITION, MAINTENANCE SWITCH UP, PRESS RESET), ENTER THE OFF-LINE TEST NUMBER (SEE SCOPE LOOP COLUMN BELOW) IN THE OPERATOR CONSOLE LIGHTS (ENTER ONES WITH LEFT-MOST SWITCH, ENTER ZEROES WITH RIGHT-MOST SWITCH), AND PRESS ON-LINE BUTTON. TEST WILL LOOP UNTIL ON-LINE SWITCH IS RETURNED TO OFF-LINE POSITION, ERRORS WILL BE DISPLAYED CONTINUOUSLY.

ERROR PROGRAM (DISPLAY)	PROGRAM	ERROR DESCRIPTION	LIKELY MODULE	SCOPE LOOP
337	OPERATIONAL CODE	CAPSTAN RUNAWAY ERROR (M3.RNY). CAPSTAN DIDN'T STOP WITHIN ACCEPTABLE WINDOW AFTER LAST COMMAND.		
100	IOTSM	BASIC I/O MICRO FAILURE (PARITY ERROR, IOATN, HANDSHAKING, AND DATA WINDOW TEST BETWEEN THE I/O AND MAIN MICROS. NOTE: CAN ALSO BE CAUSED BY THE SERIAL BUS .SHIN (SHIFT IN) STUCK ASSERTED.	M8967	14
101	IOTSM	ERROR IN I/O CONTROL REGISTER TEST	M8966 M8967	15
102	IOTSM	FAILURE OF FRAME COUNTER TEST	M8966	15
103	IOTSM	FAILURE OF I/O SILO NON-PARITY ERROR DATA TEST OR THE WRITE FLAG.	M8966 M8963	16
104	IOTSM	FAILURE OF I/O SILO PARITY ERROR TEST OR DATA LATE TEST.	M8966	17
105	IOTSM	FAILURE OF SHIFT LOOP WITH ZEROES.	M8965	20
106	IOTSM	FAILURE OF SHIFT LOOP WITH ONES.	M8965	21

1176						
1177						
1178						
1179						
1180						
1181						
1182						
1183						
1184						
1185						
1186						
1187						
1188						
1189						
1190						
1191						
1192						
1193						
1194						
1195						
1196						
1197						
1198						
1199						
1200						
1201						
1202						
1203						
1204						
1205						
1206						
1207						
1208						
1209						
	107	IOTSM	FAILURE OF SHIFT LENGTH MUX.	M8965	22	
	110	IOTSM	FAILURE TO RECEIVE CORRECT OP-CODE FROM TS11 WHEN WE SENT DATA OVER THE SERIAL BUS.	M8965	47	TS11 MOTHER BD SBUS CABLE
	111	CATSM	FAILURE OF 1 KHZ CLOCK TEST. TSTS TAC SYNC FLOP AND ATTN, TOO.	G159	2	CBUS CABLE M8963
	112	CATSM	LIGHT REGISTER CHANGED WHEN MOTION REGISTER WAS CLEARED.	G159	3.4	
	113	CATSM	FWD OR MVG BITS WRONG AFTER 1 TICK OF SIMULATED COMMAND AND TACH PULSES.	G159	3.4	
	114	CATSM	FAILURE OF SIMULATED CAPSTAN SPEED TEST. THE CAPSTAN SPEED COUNTER WAS OUT OF RANGE WHEN TAPE MOTION AT SPEED WAS SIMULATED.	G159	3.4	
	115	CATSM	FAILURE OF SIMULATED SLOW CAPSTAN TEST. SPEED COUNTER DID NOT LATCH UP WITH MAX COUNT WHEN SLOW TACH TICKS WERE SIMULATED.	G159	3.4	
	116	CATSM	FAILURE OF SIMULATED CAPSTAN DECEL TEST. COUNTER NOT ZERO FOR FORWARD OR 377 FOR REVERSE WHILE DECELERATING. OR MVG BIT NOT 1.	G159	3.4	

1210						
1211	117	CATSM	FAILURE OF MOVING FLOP TO GO TO ZERO AFTER STOPPING (DIRECTION REVERSAL FOR ONE TACH TICK).	G159	3.4	
1212						
1213						
1214						
1215	120	PETSM	FAILURE OF WRITE BOARD TO TURN ON AND EMPTY THE SILO, OR DATA LATE BIT DOESN'T WORK.	M8929 M8966	23	
1216						
1217						
1218						
1219	121	PETSM	FAILURE OF WRITE BOARD TO EMPTY SILO AT CORRECT SPEED.	M8929	23	
1220						
1221						
1222	124	PETSM	FORMATTER FLAG DOESN'T WORK ON THE M8922.	M8922	24	
1223						
1224						
1225	125	PETSM	FORMATTER SILO FILLING AND DATA ERROR	M8922 M8923 M8924	24	
1226						
1227						
1228						
1229	126	PETSM	PEAK SHIFT TEST ERROR	M8922 M8923 M8924	25	
1230						
1231						
1232						
1233	127	PETSM	FORMATTER TABLE LOOKUP ROM CHECKSUM TEST ERROR	M8922 M8923 M8924	26	
1234						
1235						

1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
12755.0 TEST SUMMARIES

5.1 TEST 1 - PDP-11/TS11 WRAP TEST

TEST TO INSURE PROPER COMMUNICATION BETWEEN THE PDP11 AND THE TS11 BY WRAPPING THE FOLLOWING PATTERNS:
A 1 IN A FIELD OF 0'S A 0 IN A FIELD OF 1'S.

WHEN DATA IS WRITTEN TO THE TSDB HI BYTE, THE DATA IS WRAPPED AROUND WITHIN THE TS11 AND APPEARS IN THE TSBA LO AND HI BYTES. THE 2 LOW ORDER BITS OF THE DATA WILL BE REFLECTED IN THE TSSR EXTENDED ADDRESS BITS.
R4 CONTAINS A COPY OF THE DATA SENT.
R3 CONTAINS THE EXPECTED TSBA RESULTS
R2 CONTAINS THE EXPECTED STATE OF THE TWO EXTENDED ADDRESS BITS IN THE TSSR.

5.2 TEST 2 - PDP-11/TS04 WRAP TEST

TEST TO INSURE PROPER COMMUNICATION BETWEEN THE PDP11 AND THE TS04 BY WRAPPING THE FOLLOWING PATTERN:
A 1 IN A FIELD OF 0'S; A 0 IN A FIELD OF 1'S

WHEN THE DATA IS WRITTEN TO THE TSDB LO BYTE, THE DATA IS SENT TO THE TS04, VIA THE SERIAL LINE, WHERE IT IS WRAPPED AROUND BACK OVER THE SERIAL LINE TO THE TS11. THE DATA THEN APPEARS IN THE TSBA LO AND TSSR LO BYTES.
R4 CONTAINS THE EXPECTED TSBA RESULTS AND THE EXPECTED TSSR RESULTS.

5.3 TEST 3 - SET TS04 CHARACTERISTIC

THE FUNCTION OF THIS TEST IS TO ISSUE A "SET CHARACTERISTIC" COMMAND TO TELL THE TS04 WHERE IN CORE THE MESSAGE PACKET RESIDES AND TO VERIFY THAT A MESSAGE PACKET WAS STORED.

1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306

5.4

TESTS 4 - 7 PERFORM DATA WRAPS ON THE P.E. READ FORMATTER BOARDS. COMMUNICATION BETWEEN THE PDP11 AND TS04 OCCURS BY USING THE DIA (DIAGNOSTIC) COMMAND WHICH SENDS A COPY OF THE DIABLK TABLE, RESIDING IN CORE, TO THE TS04 CONTROLLER. THE FORMAT OF THE DIABLK IS SHOWN IN THE FOLLOWING TABLE. NOTE THAT THE TABLE IS FILLED IN REVERSE ORDER, THAT IS, THE LAST LOGICAL ENTRY OF THE TABLE IS LABELED DIABLK, WHILE THE FIRST LOGICAL ENTRY OF THE TABLE IS LABELED DIABLK+DIAEXT, WHE DIAEXT IS THE LENGTH (EXTENT) OF THE TABLE IN BYTES.

WHEN THE DIA COMMAND IS EXECUTED, THE DIABLK IS LOADED ONTO THE TS04 STA WITH THE FIRST LOGICAL ENTRY AT THE TOP OF THE STACK, AS SHOWN BELOW. THE TS04 THEN JUMPS TO THE P.E. WRAP TASK, IN ROM, WHERE THE FUNCTION IS EXECUTED USING THE REMAINING STACK ENTRIES AS ARGUMENTS.

DIABLK+DIAEXT:	TS04 PE WRAP TASK ADDR LO	
	TS04 PE WRAP TASK ADDR HI	
	READ CONTROL	(RDCTLO)
	FORMAT CONTROL	(FMCTLO)
	DATA	
	CONTROL	(IOSCO)
	DATA	
	CONTROL	(IOSCO)
	DATA	
	CONTROL	(IOSCO)
	DATA	
	CONTROL	(IOSCO)
	DATA	
	CONTROL	(IOSCO)
	DATA	
DIABLK:	CONTROL	(IOSCO)

1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350

5.4.1 TRACK ACTIVE/TRACK INACTIVE TESTS (TEST 4)

TEST 4 CHECKS THAT THE TRACK ACTIVE FLOP CAN SET AND CLEAR IN NRZI MODE FOR ALL CHANNELS. IF THE DATA DOES NOT MAKE A TRANSITION WHEN THE WRITE FLAG IS UP, THE TRACK ACTIVE FLOP WILL CLEAR. HOWEVER, IT WILL SET IF THERE IS A DATA TRANSITION WHILE THE WRITE FLAG IS UP.

5.4.1.1 SUBTEST 1 - TRACK INACTIVE TEST

THIS TEST FORCES THE TRACK ACTIVE TO CLEAR BY WRITING ALL 0'S DATA. THE PATTERN IS AS FOLLOWS FOR EACH CHANNEL:

DATA: 000000
WRTFLG: 011100
TRACK ACTIVE: SHOULD BE 0 FOR ALL TRACKS

5.4.1.2 SUBTEST 2 - TRACK ACTIVE TEST

THIS TEST FORCES TRACK ACTIVE TO SET BY WRITING THE FOLLOWING PATTERN ON EACH CHANNEL:

DATA: 110000
WRTFLG: 011100
TRACK ACTIVE: SHOULD BE 1 FOR ALL TRACKS.

5.4.1.3 SUBTEST 3 - TRACK INACTIVE TEST

THIS TEST FORCES THE TRACK ACTIVE TO CLEAR BY WRITING THE FOLLOWING PATTERN ON EACH CHANNEL:

DATA: 111111
WRTFLG: 011100
TRACK ACTIVE: SHOULD BE 0 FOR ALL TRACKS

5.4.1.4 SUBTEST 4 - TRACK ACTIVE TEST

THIS TEST FORCES THE TRACK ACTIVE FLOP TO SET BY WRITING THE FOLLOWING PATTERN ON EACH CHANNEL:

DATA: 001111
WRTFLG: 011100
TRACK ACTIVE: SHOULD BE 1 FOR ALL TRACKS.

1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396

5.4.2 P.E. DATA TEST (TEST 5)

TEST 5 WRAPS A DATA PATTERN TO CHECK EACH TRACK FOR BIT PICKUPS AND DROPS.

REGISTER USAGE IS AS FOLLOWS:

- R2 = PREAMBLE DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.
- R3 = 1ST BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8. THIS IS THE DATA INTEREST AFTER EXECUTING THE TS04 DIA COMMAND.
- R4 = 2ND BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.

5.4.2.1 SUBTEST 1 - P.E. DATA TEST/0 PATTERN.

THIS TEST WRAPS AN ALL 0'S PATTERN.

5.4.2.2 SUBTEST 2 - P.E. DATA TEST/1 PATTERN.

THIS TEST WRAPS AN ALL 1'S PATTERN.

5.4.2.3 SUBTEST 3 - P.E. DATA TEST/SHIFTING 1 PATTERN.

THIS TEST RIPPLES A 1 IN A FIELD OF 0'S.

5.4.2.4 SUBTEST 4 - P.E. DATA TEST/SHIFTING 0 PATTERN.

THIS TEST RIPPLES A 0 IN A FIELD OF 1'S.

5.4.3 TEST 6 SKEWS THE DATA ON A TRACK BY ONE BYTE WITH RESPECT TO ALL THE OTHER TRACKS. THAT IS, THE DATA IS ONE BYTE LATE ON THE ONE TRACK. EACH TRACK IS TESTED FOR SKEW IN THIS MANNER. REGISTER ASSIGNMENTS ARE AS FOLLOWS:

- R2 = PREAMBLE DATA
- R3 = BYTE 1 DATA (WITH THE EXCEPTION OF THE SKEWED TRACK. THAT TRACK CONTAINS PREAMBLE DATA)
- R4 = BYTE 2 DATA (WITH THE EXCEPTION OF THE SKEWED TRACK. THAT TRACK CONTAINS BYTE 1 DATA)

5.4.3.1 SUBTEST 1 - P.E. SKEW TEST

THIS TEST WRITE AN ALL 1'S PREAMBLE (SKEWED), AN ALL 0'S BYTE 1 (SKEWED), AND AN ALL 1'S BYTE 2 DATA (SKEWED).

1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446

5.4.3.2 SUBTEST 2 - P.E. SKEW TEST

THIS TEST SENDS AN ALL 1'S PREAMBLE (SKEWED), AN ALL 1'S BYTE 1 DATA (SKEWED) AND AN ALL 0'S BYTE 2 DATA (SKEWED).

5.4.4 TEST 7 CHECKS THE DEAD TRACK LOGIC BY RIPPLING A DEAD TRACK THRU A FIELD OF LIVE TRACKS AND ONE LIVE TRACK THRU A FIELD OF DEAD TRACKS. ADDITIONALLY, EACH SUBTEST WILL SEND 1'S DATA AND 0'S DATA IN ORDER TO TEST THE 1'S OR DEAD REGISTER AND THE 0'S OR DEAD REGISTER.

REGISTER USAGE:

R2 = PREAMBLE ALL 1'S CHARACTER
R3 = 1ST DATA BYTE (BITS 0-8)
R4 = 2ND DATA BYTE (BITS 0-8)

DTKIDN = DEAD TRACK DEFINED IN BITS 0-8 (0=LIVE TRK; 1=DEAD TRK)

5.4.4.1 SUBTEST 1 - P.E. DEAD TRACK TEST

THIS TEST RIPPLES A DEAD TRACK IN A FIELD OF LIVE TRACKS.

5.4.4.2 SUBTEST 2 - P.E. DEAD TRACK TEST

THIS TEST RIPPLES A LIVE TRACK IN A FIELD OF DEAD TRACKS.

5.5 TEST 8 - LOOKUP TABLE TEST

THIS TEST VERIFIES THAT THE CONTENTS OF THE ROM LOOKUP TABLE ARE CORRECT. THE ROM CONTENTS IN ADDRESSES 1777-0 ARE CHECKED.

DATA AND REGISTER USAGE:

ROMLKI = ROM LOOKUP TABLE ADDRESS.
ERRLFG = ERROR FLAG.
R5 = DIABLK INDEX.

5.6 TEST 9 - INLINE MICRO DIAG TEST

ALLOWS INLINE MICRO DIAGS TO RUN, THEN CHECKS THE STATUS THEY RETURN.

5.7 TEST 10 - INIT MICRO DIAG TEST

ALLOWS INIT MICRO DIAGS TO RUN, THEN CHECKS THE STATUS THEY RETURN.

```

1447 .TITLE PROGRAM HEADER AND TABLES
1448 .SBTTL PROGRAM HEADER
1449
1450 .ENABL ABS,AMA
1451 = 2000
1452
1453 002000 BGNMOD
1454
1455
1456 ;**
1457 ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1458 ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1459 ;--
1460 002000 POINTER BGNAU,BGN DU,BGNSW,BGNSFT,BGNSETUP
1461
1462
1463 002000 HEADER CZTSI.C.0,25,0
1464 002000 L$NAME:: ;DIAGNOSTIC NAME
1465 002000 103 .ASCII /C/
1466 002001 132 .ASCII /Z/
1467 002002 124 .ASCII /T/
1468 002003 123 .ASCII /S/
1469 002004 111 .ASCII /I/
1470 002005 000 .BYTE 0
1471 002006 000 .BYTE 0
1472 002007 000 .BYTE 0
1473 002010 L$REV:: ;REVISION LEVEL
1474 002010 103 .ASCII /C/
1475 002011 L$DEPO:: ;0
1476 002011 060 .ASCII /O/
1477 002012 L$UNIT:: ;NUMBER OF UNITS
1478 002012 000001 .WORD T$PTHV
1479 002014 L$TIML:: ;LONGEST TEST TIME
1480 002014 000025 .WORD 25
1481 002016 L$HPCP:: ;POINTER TO H.W. QUES.
1482 002016 034216 .WORD L$HARD
1483 002020 L$SPCP:: ;POINTER TO S.W. QUES.
1484 002020 034270 .WORD L$SOFT
1485 002022 L$HPTP:: ;PTR. TO DEF. H.W. PTABLE
1486 002022 002204 .WORD L$HW
1487 002024 L$SPTP:: ;PTR. TO S.W. PTABLE
1488 002024 002212 .WORD L$SW
1489 002026 L$LADP:: ;DIAG. END ADDRESS
1490 002026 034664 .WORD L$LAST
1491 002030 L$STA:: ;RESERVED FOR APT STATS
1492 002030 000000 .WORD 0
1493 002032 L$CO:: .WORD 0
1494 002032 000000 .WORD 0
1495 002034 L$DTYP:: ;DIAGNOSTIC TYPE
1496 002034 000000 .WORD 0
1497 002036 L$APT:: ;APT EXPANSION
1498 002036 000000 .WORD 0
1499 002040 L$DTP:: ;PTR. TO DISPATCH TABLE
1500 002040 002124 .WORD L$DISPAT
1501 002042 L$PRIO:: ;DIAGNOSTIC RUN PRIORITY
1502 002042 000000 .WORD 0

```

1503	002044		L\$ENVI::	;FLAGS DESCRIBE HOW IT WAS SETUP		
1504	002044	000000			.WORD	0
1505	002046		L\$EXP1::	;EXPANSION WORD		
1506	002046	000000			.WORD	0
1507	002050		L\$MREV::	;SVC REV AND EDIT #		
1508	002050	003			.BYTE	C\$REVISI
1509	002051	003			.BYTE	C\$EDIT
1510	002052		L\$EF::	;DIAG. EVENT FLAGS		
1511	002052	000000			.WORD	0
1512	002054	000000			.WORD	0
1513	002056		L\$SPC::			
1514	002056	000000			.WORD	0
1515	002060		L\$DEVP::	; POINTER TO DEVICE TYPE LIST		
1516	002060	002174			.WORD	L\$DVTYP
1517	002062		L\$REPP::	;PTR. TO REPORT CODE		
1518	002062	000000			.WORD	0
1519	002064		L\$EXP4::			
1520	002064	000000			.WORD	0
1521	002066		L\$EXP5::			
1522	002066	000000			.WORD	0
1523	002070		L\$AUT::	;PTR. TO ADD UNIT CODE		
1524	002070	030230			.WORD	L\$AU
1525	002072		L\$DUT::	;PTR. TO DROP UNIT CODE		
1526	002072	030212			.WORD	L\$DU
1527	002074		L\$LUN::	;LUN FOR EXERCISERS TO FILL		
1528	002074	000000			.WORD	0
1529	002076		L\$DESP::	;POINTER TO DIAG. DESCRIPTION		
1530	002076	002150			.WORD	L\$DESC
1531	002100		L\$LOAD::	;GENERATE SPECIAL AUTOLOAD EMT		
1532	002100	104035			EMT	E\$LOAD
1533	002102		L\$ETP::	;POINTER TO ERRtbl		
1534	002102	000000			.WORD	0
1535	002104		L\$ICP::	;PTR. TO INIT CODE		
1536	002104	027434			.WORD	L\$INIT
1537	002106		L\$CCP::	;PTR. TO CLEAN-UP CODE		
1538	002106	030170			.WORD	L\$CLEAN
1539	002110		L\$ACP::	;PTR. TO AUTO CODE		
1540	002110	027716			.WORD	L\$AUTO
1541	002112		L\$PRT::	;PTR. TO PROTECT TABLE		
1542	002112	027426			.WORD	L\$PROT
1543	002114		L\$TEST::	;TEST NUMBER		
1544	002114	000000			.WORD	0
1545	002116		L\$DLY::	;DELAY COUNT		
1546	002116	000000			.WORD	0
1547	002120		L\$HIME::	;PTR. TO HIGH MEM		
1548	002120	000000			.WORD	0
1549						

1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587

002122
002122 000012
002124
002124 030246
002126 030602
002130 030762
002132 031454
002134 032362
002136 033024
002140 033406
002142 033770
002144 034124
002146 034172

002150
002150
002150 047503 052116 047522
002156 020114 047514 044507
002164 020103 042524 052123
002172 000
002174
002174
002174
002174 051524 030461 000
002202

.SBTTL DISPATCH TABLE

; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
;--

DISPATCH 10

L\$DISPATCH::

.WORD 10
.WORD T1
.WORD T2
.WORD T3
.WORD T4
.WORD T5
.WORD T6
.WORD T7
.WORD T8
.WORD T9
.WORD T10

.SBTTL DESCRIPTIVE TEXT

; 2 LINES OF TEXT PRINTED TO THE OPERATOR TO IDENTIFY THE DIAG AND THE DEVICE
;--

DESCRIPT <CONTROL LOGIC TEST>

L\$DESC::

.ASCIZ /CONTROL

DEVTYP <TS11>

L\$DVTYP::

.EVEN

.ASCIZ /TS11/
.EVEN

1588
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596 002202
 1597 002202 000002
 1598 002204
 1599 002204
 1600
 1601 002204 172522
 1602 002206 000224
 1603
 1604 002210
 1605 002210

.SBTTL DEFAULT HARDWARE P-TABLE

 ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
 ; THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
 ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.

BGNHW DFPTBL

.WORD L10000-L

L\$HW::
DFPTBL::

172522 ;TSSR ADDRESS
 224 ;VECTOR ADDRESS

ENDHW
L10000:

C4

PROGRAM HEADER AND TABLES
CZTSIC.P11 09-APR-84 14:37

MACY11 30(1046) 09-APR-84 14:40 PAGE 43
SOFTWARE P-TABLE

SEQ 0041

1606		
1607		
1608		
1609		
1610		
1611		
1612		
1613		
1614	002210	
1615	002210	000001
1616	002212	
1617	002212	
1618		
1619	002212	000
1620		
1621	002213	000
1622		
1623	002214	
1624	002214	
1625		
1626	002214	

.SBTTL SOFTWARE P-TABLE

```

***
; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
---
```

BGNSW SFPTBL

L\$SW::
SFPTBL::

CMPFLG:: .BYTE 0
 .BYTE 0

```

;ENABLE DATA COMPARE ERROR PRINT FLAG
;0=DO NOT ENABLE IS DEFAULT
;SPARE FLAG
```

.WORD L10001-L

ENDSW
L10001:
ENDMOD

```

1627
1628
1629
1630 002214
1631
1632
1633
1634
1635
1636
1637 002214
1638
1639
1640
1641 100000
1642 040000
1643 020000
1644 010000
1645 004000
1646 002000
1647 001000
1648 000400
1649 000200
1650 000100
1651 000040
1652 000020
1653 000010
1654 000004
1655 000002
1656 000001
1657
1658 001000
1659 000400
1660 000200
1661 000100
1662 000040
1663 000020
1664 000010
1665 000004
1666 000002
1667 000001
1668
1669
1670
1671
1672 000040
1673 000037
1674 000036
1675 000035
1676 000034
1677
1678
1679
1680
1681 000340
1682 000300

```

```

.TITLE GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION
      BGNMOD

***
; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
; ARE USED IN MORE THAN ONE TEST.
;--

      EQUALS

;
; BIT DIFINITIONS
;
BIT15== 100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000
BIT08== 400
BIT07== 200
BIT06== 100
BIT05== 40
BIT04== 20
BIT03== 10
BIT02== 4
BIT01== 2
BIT00== 1

;
BIT9== BIT09
BIT8== BIT08
BIT7== BIT07
BIT6== BIT06
BIT5== BIT05
BIT4== BIT04
BIT3== BIT03
BIT2== BIT02
BIT1== BIT01
BIT0== BIT00

;
; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
;
EF.START== 32. ; START COMMAND WAS ISSUED
EF.RESTART== 31. ; RESTART COMMAND WAS ISSUED
EF.CONTINUE== 30. ; CONTINUE COMMAND WAS ISSUED
EF.NEW== 29. ; A NEW PASS HAS BEEN STARTED
EF.PWR== 28. ; A POWER-FAIL/POWER-UP OCCURRED

;
; PRIORITY LEVEL DEFINITIONS
;
PRI07== 340
PRI06== 300

```

1683 000240
 1684 000200
 1685 000140
 1686 000100
 1687 000040
 1688 000000
 1689
 1690
 1691
 1692 000004
 1693 000010
 1694 000020
 1695 000040
 1696 000100
 1697 000200
 1698 000400
 1699 001000
 1700 002000
 1701 004000
 1702 010000
 1703 020000
 1704 040000
 1705 100000
 1706
 1707
 1708
 1709
 1710 001000
 1711 000400
 1712 000200
 1713 000100
 1714 000040
 1715 000020
 1716 000010
 1717 000004
 1718 000002
 1719 000001
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727 000005
 1728 000200
 1729 000000
 1730 000033
 1731
 1732
 1733
 1734

PRI05== 240
 PRI04== 200
 PRI03== 140
 PRI02== 100
 PRI01== 40
 PRI00== 0

;
 ;OPERATOR FLAG BITS

;
 EVL== 4
 LOT== 10
 ADR== 20
 IDU== 40
 ISR== 100
 UAM== 200
 BOE== 400
 PNT== 1000
 PRI== 2000
 IXE== 4000
 IBE== 10000
 IER== 20000
 LOE== 40000
 MOE== 100000

;
 ;BIT DEFINITIONS USED TO SPECIFY THE ROM ADDRESS LINES IN THE ROMLOK
 ;SUBROUTINE

A9== 1000
 A8== 400
 A7== 200
 A6== 100
 A5== 40
 A4== 20
 A3== 10
 A2== 4
 A1== 2
 A0== 1

;
 ;*****
 ;*****

;
 ; THE FOLLOWING DEFINITIONS MAY CHANGE ON MICRO-CODE REASSEMBLY:

;
 WRPLO==5 ;WRITE WRAP TASK ADR LO.
 WRPHI==200 ;WRITE WRAP TASK ADR HI.
 POPJHI==0 ;TS04 POPJ ADDRESS HI (RTS)
 POPJLO==33 ;TS04 POPJ ADDRESS LO (RTS)

;
 ;*****
 ;*****

```

1735 ; TSSR REGISTER BIT DEFINITIONS.
1736
1737 001000 TS.XA1==1000 ;EXTENDED ADDRESS BIT 1
1738 000400 TS.XA0==400 ;EXTENDED ADDRESS BIT 0
1739 000200 TS.SSR==200 ;SUBSYSTEM READY BIT.
1740 002000 TS.NBA==2000 ;NEED BUFFER ADDRESS-CLEARED BY SET CHAR
1741 ; -SET BY CMD WITHOUT SET CHAR
1742 020000 TS.SPE==20000 ;SERIAL BUS PARITY ERROR AT TS11
1743 177717 FCMASK==177717 ;FATAL CLASS CODE MASK
1744 177701 TCFMK==177701 ;TERMINATION AND FATAL CLASSES CODE MASK
1745
1746 ;THE FOLLOWING ARE BIT DEFINITIONS FOR THE COMMAND WORD
1747
1748 100000 ACK.C==100000 ;ACKNOWLEDGE BIT
1749 040000 CVC.C==40000 ;CLEAR VOLUME CHECK.
1750 020000 OPP.C==20000 ;OPPOSITE BIT
1751 010000 SWB.C==10000 ;SWAP BYTE BIT
1752 004000 MOD.C3==4000 ;MODE BIT 3
1753 002000 MOD.C2==2000 ;MODE BIT 2
1754 001000 MOD.C1==1000 ;MODE BIT 1
1755 000400 MOD.C0==400 ;MODE BIT 0
1756 000200 IE.C==200 ;INTERRUPT ENABLE
1757 000100 FMT.C1==100 ;FORMAT BIT 1
1758 000040 FMT.C0==40 ;FORMAT BIT 0.
1759 000020 CMD.C4==20 ;COMMAND BIT 4
1760 000010 CMD.C3==10 ;COMMAND BIT 3
1761 000004 CMD.C2==4 ;COMMAND BIT 2
1762 000002 CMD.C1==2 ;COMMAND BIT 1
1763 000001 CMD.C0==1 ;COMMAND BIT 0
1764
1765 ; BIT DEFINITIONS FOR DEVICE CHARACTERISTICS.
1766
1767 000200 CH.ESS==200
1768 000040 CH.EAI==40
1769 000020 CH.ERI==20
1770
1771 ;ROM LOOKUP TABLE BIT DEFINITIONS
1772
1773 000200 .MULT==200 ;MULT TRACKS
1774 000100 .RDFMK==100 ;READ FILE MARK PATTERN
1775 000040 .PREAM==40 ;PREAMBLE
1776 000020 .9OF9==20 ;9 OF 9 TRACKS
1777 000010 .0OF9==10 ;NONE OF 9 TRACKS
1778 000004 .CORD==4 ;CORRECTABLE DATA
1779 000002 .INCOR==2 ;INCORRECTABLE DATA
1780 000001 .8OF9==1 ;8 OF 9 TRACKS
1781
1782 ; MISCELLANEOUS DEFINITIONS.
1783
1784 000340 INTPRI==PRI07 ;PRIORITY TO BE USED IN THE INTERRUPT STATE.
1785 000010 SCHEXT==10 ;BUFFER LENGTH. (EVEN #)
1786 000016 MSGEXT==16 ;MESSAGE BUFFER LENGTH IN BYTES. (EVEN #)
1787 000020 DIAEXT==20 ;DIABLK EXTENT IN OCTAL.
1788 100006 DIA==ACK.C!CMD.C2!CMD.C1 ;DIA CMD WORD.
1789 140004 SCH==ACK.C!CVC.C!CMD.C2 ;SCH (SET CHAR) CMD WORD.
1790 100017 GES==ACK.C!CMD.C0!CMD.C1!CMD.C2!CMD.C3 ;GET STATUS COMMAND

```

GLOBAL AREAS MACY11 30(1046) 09-APR-84 14:40 PAGE 47
 CZTSIC.P11 09-APR-84 14:37 GLOBAL EQUATES SECTION

SEQ 0045

```

1791      020000      BPE==20000      ;XSTAT2, SERIAL BUS PARITY ERROR AT TS04
1792      040000      SIP==40000      ;XSTAT2, SILO PARITY ERROR
1793      000200      MOT==200      ;XSTAT0, TAPE MOVING
1794      177777      ENDTBL==177777 ;END OF A TABLE FLAG
1795
1796
1797      ;THE FOLLOWING INDICATES THE RELATIVE POSITIONS OF THE STATUS WORDS
1798      ;IN THE MESSAGE BUFFER.
1799      000004      MS$RFC==4      ;RESIDUAL FRAME COUNT.
1800      000006      MS$XS0==6      ;EXT STATUS REG 0
1801      000010      MS$XS1==10     ;EXT STATUS REG 1
1802      000012      MS$XS2==12     ;EXT STATUS REG 2
1803      000014      MS$XS3==14     ;EXT STATUS REG 3
1804      000004      C18$OR==4      ;INDEX FOR OUTPUT READY INFO (CHAN 1-8).
1805      000005      C18$1D==5      ;INDEX FOR 1 OR DEAD INFO (CHAN 1-8)
1806      000007      C18$TA==7      ;INDEX FOR TRACK ACTIVE INFO. (CHAN 1-8)
1807      000010      C18$DA==10     ;INDEX FOR DATA INFO. (CHAN 1-8)
1808      000011      C18$TD==11     ;INDEX FOR TRACK DEAD INFO. (CHAN 1-8)
1809      000012      C18$OD==12     ;INDEX FOR 0 OR DEAD INFO. (CHAN 1-8)
1810      000013      ROM$LK==13     ;INDEX FOR LOOKUP TABLE
1811      000014      PRCHST==14     ;STATUS OF THE PARITY CHANNEL (CHANNEL 9)
1812      000001      CH9.OR==1      ;BIT POSITION FOR OUTPUT RDY FOR CHAN 9
1813      000002      CH9.1D==2      ;BIT POS FOR 1 OR DEAD INFO FOR CHAN 9.
1814      000004      CH9.TA==4      ;BIT POSITION OF THE TRACK ACTIVE INFO CHAN 9.
1815      000010      CH9.DA==10     ;BIT POS OF DATA INFO FOR CHAN 9.
1816      000020      CH9.TD==20     ;BIT POS OF TRK DEAD INFO FOR CHAN 9.
1817      000040      CH9.OD==40     ;BIT POS OF 0 OR DEAD INFO FOR CHAN 9.
1818
1819      ;THE FOLLOWING DEFINITIONS SHOW THE RELATIVE POSITIONS OF THE COMMAND
1820      ;PACKET ENTRIES.
1821
1822      000000      CP$CMD==0      ;CMDPKT.0==TS04 COMMAND.
1823      000002      CP$ADL==2      ;CMDPKT.2==BUFFER ADDRESS LOW.
1824      000004      CP$ADH==4      ;CMDPKT.4==BUFFER ADDRESS HIGH.
1825      000006      CP$CNT==6      ;CMDPKT.6==BYTE/FILE/RECORD COUNT.
1826

```

```

1827 ;THE FOLLOWING ARE REGISTER AND BIT DEFINITIONS FOR THE T504
1828 ;REGISTERS OF INTEREST IN THIS DIAGNOSTIC.
1829
1830 ;*****
1831 ;FMCTLO - FORMATTER MAJOR STATE CONTROL REG
1832 000004 FMCTLO==4
1833 ;WRITE REGISTER 4
1834
1835 ;THIS REGISTER IS SET UP FOR THE PORTION OF THE
1836 ;RECORD WE ARE PRESENTLY READING
1837
1838
1839 000200 FC.RD== 200 ;WE ARE DOING NORMAL READ. IF 0, WE DISABLE SOME ERROR
1840 ;CORRECTION LOGIC SO WE'RE MORE DISCRIMINATING
1841 ;FOR READ AFTER WRITE.
1842
1843 000100 FC.FLO==100 ;SETTING THIS BIT CAUSES .FMFLO ON BBUS TO BE TRUE
1844 ;(NEEDED BY THE RD PE ROUTINE)
1845 000010 FC.DAT==10 ;DATA MODE
1846 000004 FC.PRE==4 ;PREAMBLE MODE
1847 000002 FC.VCO==2 ;VCO SYNC MODE
1848 000001 FC.NRZ==1 ;NRZI MODE (FORCE SKEW WINDOW TO STAY OPEN)
1849 ;CAUSES FMTAUI TO BE CLRED WHEN YOU WRITE
1850 ;TO THE FMCLDO REGISTER.
1851 ;*****
1852
1853 ;RDCTLO - READ CONTROL REGISTER
1854 000020 RDCTLO==20
1855 ;WRITE REGISTER 20
1856
1857 ;THIS REGISTER CONTROLS THE FORMATTER MODE AND THRESHOLD.
1858
1859 000200 RD.REV==200 ;1 FOR REV MOTION, 0==FWD
1860 000100 RD.MAI==100 ;I/O FORMATTER DATA WRAPAROUND
1861
1862 000040 RD.SPC==40 ;WE ARE SPACING RECORDS. (THIS BIT IS UNUSED IN
1863 ;THE HARDWARE BUT IS A SOFTWARE FLAG IN THE READ CODE)
1864
1865 000020 RD.SKP==20 ;WE ARE SKIPPING FILES. (THIS BIT IS UNUSED IN
1866 ;THE HARDWARE BUT IS A SOFTWARE FLAG IN THE READ CODE)
1867
1868 ;THE FOLLOWING THRESHOLDS ARE AVAILABLE:
1869 ;NORM USE DIAG USE
1870 000007 RD.110==7 ; HI PREAMP GAIN
1871 000006 RD.90==6 ; LO PREAMP, BAD TAPE CLEA
1872 000005 RD.75==5 ; RESIDUAL ERASE CHK
1873 000004 RD.68==4 ; FWD/REV AMP BALANCE
1874 000003 RD.40==3 ;NRZ WRT
1875 000002 RD.20==2 ;NRZ RD, PE WRT
1876 000001 RD.12==1 ;PE READ, NRZ ERR RECOV RD/WRT CROSSTALK
1877 000000 RD.07==0 ;PE ERR RECOV ERASE FUNCTION
1878 ;DATA PORTION ONLY
1879

```

```

1880
1881
1882           000014
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894           000200
1895
1896
1897
1898           000100
1899
1900
1901           000040
1902
1903
1904           000020
1905
1906           000010
1907           000004
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917

```

```

;*****
;IOSCO - I/O SEQUENCER SILO CONTROL BUFFER OUT
IOSCO==14
;WRITE REGISTER 14

;THIS REGISTER CONTAINS THE SILO CONTROL BITS FOR DATA WRITTEN
;BY THE MAIN OR I/O MICRO TO THE SILO. THE DATA IN THIS REG
;IS PAIRED WITH THE IOSDO REG AND PUT IN THE SILO WHENEVER
;THE IOSICO REG IS CLOCKED. THIS REGISTER NEED ONLY BE WRITTEN
;ONCE IF THE SAME OLD DATA IS OK TO BE WRITTEN IN THE SILO.
;NOTE THAT IF THE I/O IS WRITING THE SILO, THE MAIN MUST PUT EVEN
;PARITY IN THE BITS 357 OR THE I/O WILL WRITE PAR ERRS IN THE SILO.
;(THE IS.DAP BIT IS DON'T CARE HERE FOR I/O WRITING THE SILO)

IS.PAR== 200 ;ODD PAR BIT FOR ALL 16 BITS (CNRTL AND DATA)
;NOTE THAT THE BITS MASKED BY 357 MUST BE
;EVEN PARITY BECAUSE THE 9 DATA BITS ARE ODD

IS.IVP== 100 ;INVERT CNTRL SILO PAR BIT BEFORE MOVING
;TO WRITE BOARD (WRITE EVEN PARITY ON TAPE)

IS.NRZ== 40 ;INVERT WRITE BUFFER BIT IF ASSOCIATED SILO
;DATA BIT IS A 1. (IF IS.NRZ==0, WE'RE IN PE MODE)

IS.DAP== 20 ;ODD PARITY FOR THE 8 DATA BITS IN IOSDO

IS.LRC== 10 ;CAUSES SYNCHRONOUS CLR ON WRT BOARD TO WRITE THE
IS.WRF== 4 ;THIS FLG BIT SHOWS UP AT THE WRITE BOARD
;WITH THE CORRESPONDING DATA. THE FUNCTION OF
;THE BIT WILL BE DEFINED BY THE WRITE BOARD.
;NOTE HOWEVER THAT IF THE WRITE BOARD SEES
;THE BIT 1, THE PA.WRF BIT IN THE
;PRATNI REG WILL ALSO BE 1 (IF ENABLED
;TO AFFECT THE ATTN REG).
;THE 2 LOW BITS ARE WRITABLE AND AFFECT THE PARITY TREES BUT
;OTHERWISE ARE UNIMPLEMENTED
;*****

```



```

1918 .SBTTL GLOBAL DATA SECTION
1919
1920
1921 ;**
1922 ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1923 ; IN MORE THAN ONE TEST.
1924 ;--
1925
1926 ; TS04 REGISTER ADDRESSES.
1927 002214 000000 TSDB:: 0 ;TS04 DATA BUFFER ADDRESS.
1928 002216 000000 TSDBHI:: 0 ;TSDB HI BYTE ADDRESS.
1929 002220 000000 TSBA:: 0 ;TS04 BUFFER ADDRESS REG ADDRESS.
1930 002222 000000 TSBAHI:: 0 ;TSBA HI BYTE ADDRESS.
1931 002224 000000 TSSR:: 0 ;TS04 STATUS REGISTER ADDRESS.
1932 002226 000000 TSVCT:: 0 ;TS04 VECTOR ADDRESS.
1933
1934 ; THE FOLLOWING IS THE TS04 COMMAND PACKET.
1935 ; (MUST BE ON A MODULO 4 BOUNDARY)
1936
1937 002230 .-.+3&177774 ;FORCES CMD PKT ON MODULO 4 BOUNDARY.
1938
1939 002230 000000 CMDPKT::0 ;1ST WORD:: TS04 COMMAND
1940 002232 000000 0 ;2ND WORD:: BUFFER LOW ADDR.
1941 002234 000000 0 ;3RD WORD:: BUFFER HIGH ADDR.
1942 002236 000000 0 ;4TH WORD:: BYTE/RECORD/FILE COUNT.
1943
1944 ;SET CHAR PACKET
1945
1946 002240 .-.+3&177774
1947
1948 002240 140004 SCHPKT::SCH ;SET CHAR COMD HEADER
1949 002242 002276 SCHBLK ;SET CHAR BLK AD LO
1950 002244 000000 000000 ;SET CHAR BLK AD HI
1951 002246 000010 SCHEXT ;BPCR=LENGTH OF SCHAR BLOCK
1952
1953 ;DIAG PACKET
1954
1955 002250 .-.+3&177774
1956
1957 002250 100006 DIAPKT::DIA ;DIAG COMD HEADER
1958 002252 002316 DIABLK ;DIAG BLOCK AD LO
1959 002254 000000 000000 ;DIAG BLOCK AD HI
1960 002256 000020 DIAEXT ;BPCR=DIABLK LENGTH
1961
1962 ; THIS IS THE MESSAGE PACKET.
1963
1964 002260 000000 MSGPKT:: 0 ;1ST WORD:: MESSAGE WORD.
1965 002262 000000 MSGDFL:: 0 ;2ND WORD:: DATA FIELD LENGTH
1966 002264 000000 RFC:: 0 ;3RD WORD:: RESIDUAL FRAME COUNT.
1967 002266 000000 XSTAT0:: 0 ;4TH WORD:: EXTENDED STATUS REG 0.
1968 002270 000000 XSTAT1:: 0 ;5TH WORD:: EXTENDED STATUS REG 1.
1969 002272 000000 XSTAT2:: 0 ;6TH WORD:: EXTENDED STATUS REG 2.
1970 002274 000000 XSTAT3:: 0 ;7TH WORD:: EXTENDED STATUS REG 3.
1971 002276 MSGEND= . ;PAST MSG PKT POINTER
1972
1973 ; THE SET CHARACTERISTIC BLOCK.
    
```

1974				
1975	002276	002260	SCHBLK:: MSGPKT	:1ST WORD:: MSGPKT ADDR LO.
1976	002300	000000	0	:2ND WORD MSGPKT ADDR HI.
1977	002302	000016	MSGEXT	:3RD WORD:: MSG BUFFER LENGTH(BYTES).
1978	002304	000000	0	:4TH WORD:: CHARACTERISTIC WORD.
1979			:	ADDRESSES OF INTERRUPT HANDLING ROUTINES.
1980				
1981	002306	014454	TS4INT:: TS4IN0	:DEVICE 0.
1982	002310	014464	TS4IN1	:DEVICE 1.
1983	002312	014474	TS4IN2	:DEVICE 2.
1984	002314	014504	TS4IN3	:DEVICE 3.
1985				
1986				
1987			:	THE DIAGNOSTIC COMMAND BUFFER. WHEN A DIAG COMMAND IS
1988			:	EXECUTED, THE TS04 PLACES THE CONTENTS OF THIS BUFFER ONTO THE TS04
1989			:	STACK IN REVERSE ORDER, THEN EXECUTES WHAT IS ON THE STACK BY
1990			:	PERFORMING A COMMAND SIMILAR TO AN RTS.
1991			:	THE BUFFER IS INITIALLY LOADED WITH THE TS04 CODE FOR
1992			:	AN RTS COMMAND.
1993				
1994				
1995	002316	000	DIABLK:: .BYTE POPJHI	
1996	002317	033	.BYTE POPJLO	
1997	002320	000	.BYTE POPJHI	
1998	002321	033	.BYTE POPJLO	
1999	002322	000	.BYTE POPJHI	
2000	002323	033	.BYTE POPJLO	
2001	002324	000	.BYTE POPJHI	
2002	002325	033	.BYTE POPJLO	
2003	002326	000	.BYTE POPJHI	
2004	002327	033	.BYTE POPJLO	
2005	002330	000	.BYTE POPJHI	
2006	002331	033	.BYTE POPJLO	
2007	002332	000	.BYTE POPJHI	
2008	002333	033	.BYTE POPJLO	
2009	002334	000	.BYTE POPJHI	
2010	002335	033	.BYTE POPJLO	

```

2011                                     ; TABLES OF FORMATTER AND WRITE CONTROL REG ACTUAL & EXPECTED
2012
2013 002336                               EXPTBL::
2014 002336 000000                       EXORDY:: 0
2015 002340 000000                       EX1DTR:: 0
2016 002342 000000                       EXTRAC:: 0
2017 002344 000000                       EXDATA:: 0
2018 002346 000000                       EXTRDD:: 0
2019 002350 000000                       EXODTR:: 0
2020 002352 000000                       EXROML:: 0
2021
2022 002354                               ACTTBL::
2023 002354 000000                       ACORDY:: 0
2024 002356 000000                       AC1DTR:: 0
2025 002360 000000                       ACTRAC:: 0
2026 002362 000000                       ACDATA:: 0
2027 002364 000000                       ACTRDD:: 0
2028 002366 000000                       ACODTR:: 0
2029 002370 000000                       ACROML:: 0
2030
2031 002372                               ORDTBL::
2032 002372 000000                       ORORDY:: 0
2033 002374 000000                       OR1DTR:: 0
2034 002376 000000                       ORTRAC:: 0
2035 002400 000000                       ORDATA:: 0
2036 002402 000000                       ORTRDD:: 0
2037 002404 000000                       ORODTR:: 0
2038
2039 002406 000000                       ACTRK1:: 0
2040 002410 000000                       ACTRK2:: 0
2041 002412 000000                       ACTRK3:: 0
2042 002414 000000                       ACTRK4:: 0
2043
2044 002416 000000                       DTKIDN:: 0
2045
2046 002420 000000                       ROMLKI:: 0
2047 002422 000000                       UNIT:: 0
2048 002424 000000                       TS4CL:: 0
2049 002426 000000                       TIME:: 0
2050 002430 000000                       TEMPO:: 0
2051 002432 000000                       TEMP1:: 0
2052 002434 000000                       TEMP2:: 0
2053
2054
2055
2056                                     ; PROGRAM CONTROL FLAGS.
2057
2058 002436 000                           ERRFLG:: .BYTE 0
2059 002437 000                           CTLFLG:: .BYTE 0
2060 002440 000000                       T4S4:: .WORD 0
2061                                     .EVEN

```

```

; TABLE 1 EXPECTED REGS
; EXPECTED OUTPUT READY INFO
; EXPECTED 1 OR DEAD TRACK INFO
; EXPECTED TRACK ACTIVE INFO
; EXPECTED DATA INFO
; EXPECTED TRACK DEAD INFO
; EXPECTED 0 OR DEAD TRACK INFO.
; EXPECTED ROM LOOKUP TABLE INFO.

; TABLE 2 ACTUAL REGS
; ACTUAL OUTPUT READY INFO
; ACTUAL 1'S OR DEAD TRACK INFO
; ACTUAL TRACK ACTIVE INFO
; ACTUAL DATA INFO
; ACTUAL TRACK DEAD INFO
; ACTUAL 0 OR DEAD TRACK INFO
; ACTUAL ROM LOOKUP TABLE INFO

; TABLE 3 SUBTESTS' ORED ACTUAL REGS
; OUTPUT READY
; 1 OR DEAD TRACK
; TRACK ACTIVE
; DATA
; TRACK DEAD
; 0 OR DEAD TRACK

; ACTUAL TRACK ACTIVE TEST 4, SUB 1
; ACTUAL TRACK ACTIVE TEST 4, SUB 2
; ACTUAL TRACK ACTIVE TEST 4, SUB 3
; ACTUAL TRACK ACTIVE TEST 4, SUB 4

; DEAD TRACK IDENTIFICATION REG. (BITS 0 THRU 8)
; 1=DEAD; 0=LIVE.
; ROM LOOKUP TABLE ADDRESS
; CURRENT UNIT # FOR PRINTS.
; TS04 MICRO-CODE LEVEL - STORED IN SCH TEST.
; TIMEOUT COUNTER.
; GENERAL PURPOSE LOCATION 0.
; GENERAL PURPOSE LOCATION 1.
; GENERAL PURPOSE LOCATION 2.

```

```

; 0 = NO ERROR
; ALL PURPOSE PROGRAM CONTROL FLAG
; BTL TEST 4 SUBTEST 4 FLAG

```

2062
2063
2064
2065
2066
2067
2068
2069
2070
2071

.SBTTL GLOBAL TEXT SECTION

;THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
;MESSAGES, AND ASCII INFORMATION THAT ARE USED IN MORE THAN
;ONE TEST.

; ASCII MESSAGES USED IN ERROR REPORT HEADERS.

.NLIST BEX

002442	051524	032117	041440	DESCM:: .ASCIZ /TS04 CONTROL LOGIC TEST PROGRAM/
002502	047503	046515	047101	MODUER:: .ASCIZ /COMMAND PACKET ADDR NOT ON MODULO 4 BOUNDARY: RELOAD!/
002570	051524	032060	047040	SSROFF:: .ASCIZ /TS04 NOT READY - SSR NOT SET/
002625	124	030123	020064	SSRON:: .ASCIZ /TS04 DID NOT DROP READY ON COMD ISSUANCE-SSR NOT 0/
002710	042120	030520	020061	WRPER1:: .ASCIZ /PDP11 - TS11 WRAP FAILURE/
002742	042120	030520	020061	WRPER2:: .ASCIZ /PDP11 - TS11 WRAP FAIL ON TSSR EXT ADDR BITS/
003017	120	050104	030461	WRPER3:: .ASCIZ /PDP11 - TS04 WRAP FAILURE/
003051	123	052105	041440	SCHERR:: .ASCIZ /SET CHARACTERISTIC ERROR/
003102	051124	041501	020113	TAER1:: .ASCIZ /TRACK ACTIVE NOT 0 FOR 1 OR MORE TRACKS/
003152	051124	041501	020113	TAER2:: .ASCIZ /TRACK ACTIVE NOT 1 FOR 1 OR MORE TRACKS/
003222	051124	041501	020113	TAERR:: .ASCIZ /TRACK ACTIVE ERROR/
003245	120	042456	020056	DATER:: .ASCIZ /P.E. DATA WRAP ERROR/
003272	027120	027105	051440	SKEWER:: .ASCIZ /P.E. SKEW ERROR/
003312	027120	027105	042040	DDER:: .ASCIZ /P.E. DEAD TRACK ERROR/
003340	040504	040524	047440	DASKDD:: .ASCIZ /DATA OR SKEW OR DEAD TRK ERR/
003375	122	046517	046040	ROMER:: .ASCIZ /ROM LOOKUP TABLE ERROR/
003424	046511	051120	050117	SPECON:: .ASCIZ /IMPROPER TERMINATION - SPECIAL CONDITION BIT SET/
003505	116	020117	047111	NINTM:: .ASCIZ /NO INTERRUPT/
003522	047111	042524	051122	UINTM:: .ASCIZ /INTERRUPT OCCURRED WHEN DISABLED/
003563	115	041511	047522	MICROE:: .ASCIZ /MICRO DIAGNOSTIC ERROR TSSR:FC=0,TC=7/

.LIST BEX

.EVEN

2072

2073
2074
2075

:ASCII BASIC MESSAGES USED IN ERROR REPORTS

003632	040445	042523	020105
003653	045	044501	020106
003716	051445	022462	042501
003777	045	041501	042117
004060	040445	030061	020060
004157	045	030501	030460
004233	045	030501	031060
004310	040445	030061	020063
004413	045	030501	032060
004500	040445	030061	020065
004551	045	030501	033060
004622	040445	030061	020067
004707	045	030501	030061
005000	051445	034463	040445
005041	045	041101	042101
005064	040445	040502	020104
005140	040445	044103	041505
005176	040445	040502	020104
005253	045	041101	042101
005330	040445	040502	020104
005405	045	041101	042101
005454	040445	040502	020104
005523	045	041101	042101
005572	040445	044103	041505
005650	040445	051105	047522
005722	040445	034115	033071
005765	045	047501	020513
005775	045	041501	042510

```

.NLIST BEX
TERM01:: .ASCIZ /#ASEE OP PANEL#N/
TERM02:: .ASCIZ /#AIF UOK LIT THEN BAD TS11 BOARD#N/
TERM03:: .ASCIZ /#S2#AELSE OP PANEL=MICRO IO ERROR CODE 100-110#N/
TERM05:: .ASCIZ /#ACODE LOOP#S10#ADESCRIPTION#S9#AMODULE(SLOT)#N2/
TERM06:: .ASCIZ /#A100 14 IO MICRO SSTEP,IOATN,...#S5#AM8967(12),M8963(11)#N/
TERM07:: .ASCIZ /#A101 15 IOCNO REG TEST#S15#AM8967(12)#N/
TERM10:: .ASCIZ /#A102 15 FRAME CNTR TEST#S14#AM8966(14)#N/
TERM11:: .ASCIZ /#A103 16 SILO GOOD PAR DATA - WRT FLG#S1#AM8966(14),M8963(11)
TERM12:: .ASCIZ /#A104 17 SILO BAD PAR - DATA LATE#S5#AM8966(14)#N/
TERM13:: .ASCIZ /#A105 20 IO LOOP-0'S#S18#AM8965(15)#N/
TERM14:: .ASCIZ /#A106 21 IO LOOP-1'S#S18#AM8965(15)#N/
TERM15:: .ASCIZ /#A107 22 IO LOOP-SHIFT LENGTH MUX#S5#AM8965(15)#N/
TERM16:: .ASCIZ /#A110 47 SERIAL BUS - TS11 ALIVE#S6#AM8965(15),TS11#N/
TERM17:: .ASCIZ /#S39#AMOTHER BOARD, SBUS CABLE#N/

TS11BD:: .ASCIZ /#ABAD TS11 BOARD#N/
FMTCTR:: .ASCII /#ABAD FORMATTER CONTROL BOARD M8922 SLOT 7#N/
          .ASCIZ /#ACHECK VCO ADJUST ON M8922#N/
FMTCH6:: .ASCIZ /#ABAD FORMATTER CHANNEL BOARD M8924 SLOT 6#N/
FMTCH5:: .ASCIZ /#ABAD FORMATTER CHANNEL BOARD M8924 SLOT 5#N/
FMTCH4:: .ASCIZ /#ABAD FORMATTER CHANNEL BOARD M8924 SLOT 4#N/
RDCH2:: .ASCIZ /#ABAD READ CHANNEL BOARD G157 SLOT 2#N/
RDCH1:: .ASCIZ /#ABAD READ CHANNEL BOARD G157 SLOT 1#N/
RDCHP3:: .ASCII /#ABAD READ CHANNEL BOARD M8923 SLOT 3#N/
          .ASCIZ /#ACHECK SKEW AND THRESHOLD ADJUSTS ON M8923#N/
IOBRD:: .ASCII /#AERROR IN IO BOARDS(SLOT) OR SBUS CABLE#N/
          .ASCIZ /#AM8965(15),M8966(14),M8967(12)#N2/
CRMSG:: .ASCIZ /#AOK!#N/
T4S4MG:: .ASCIZ /#ACHECK G157 BOARDS FOR CORRECT PLACEMENT OF C6#N/

```

:BTL

2076
2077

006060

.LIST BEX
.EVEN

2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094 006060
2095 006060
2096 006060
2097 006060
2098 006060 021127 177777
2099 006064 001415
2100 006066
2101 006066 011137 006124
2102 006072
2103 006072 013746 006124
2104 006076 012746 000001
2105 006102 010600
2106 006104 104414
2107 006106 062706 000004
2108 006112
2109 006112 062701 000002
2110 006116
2111 006116 000760
2112 006120
2113 006120
2114 006120 000167
2115 006122 000002
2116 006124 000000
2117 006126
2118 006126
2119 006126 104423
2120
2121 006130
2122 006130
2123 006130
2124 006130 012746 005041
2125 006134 012746 000001
2126 006140 010600
2127 006142 104414
2128 006144 062706 000004
2129 006150
2130 006150 013746 002422
2131 006154 013746 002432
2132 006160 010346
2133 006162 012746 006206

.SBTTL GLOBAL ERROR REPORT SECTION

```

; **
; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
; THAT ARE USED IN MORE THAN ONE TEST. IT ALSO INCLUDES THE ASCII MESSAGES
; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
; **

```

```

;LONG MESSAGE PRINT SUBR
;ENTER WITH R1=AD OF MSG AD TABLE
;MSG AD TABLE CONTAINS AD OF MSGS TO PRINT, ONE .WORD AD
;PER MSG. THE TABLE MUST END WITH .WORD ENDTBL=177777
;AS END OF TABLE FLAG

```

```

BGNMSG LONMSG
LONMSG::
  WHILE (R1) NE #177777 DO

```

```

    LET MSGADR := (R1)
    PRINTB     MSGADR

```

LET R1 := R1 + #2

ENDDO

EXIT MSG

```

MSGADR: .WORD 0
L10002: ENDMMSG

```

```

BGNMSG WRAPR1
WRAPR1::
  PRINTB #TS11BD

```

PRINTX #EEM1A,R3,TEMP1,UNIT

```

50000$:
  CMP     (R1),#17
  BEQ    50001$
  MOV     (R1),MSG
  MOV     MSGADR,
  MOV     #1,-(SP)
  MOV     SP,R0
  TRAP   C$PNTB
  ADD     #4,SP
  ADD     #2,R1
  BR     50000$
50001$:
  .WORD  J$JMP
  .WORD  L10002-2
  TRAP   C$MSG
  MOV     #TS11BD,
  MOV     #1,-(SP)
  MOV     SP,R0
  TRAP   C$PNTB
  ADD     #4,SP
  MOV     UNIT,-(S
  MOV     TEMP1,-(
  MOV     R3,-(SP)
  MOV     #EEM1A,

```

2134 006166 012746 000004
2135 006172 010600
2136 006174 104415
2137 006176 062706 000012
2138 006202
2139 006202 000167
2140 006204 000074
2141

EXIT MSG
EEM1A: .NLIST BEX
.ASCII /#ATSBA EXPECTED=#B16#N/
.ASCIZ /#ATSBA ACTUAL=#S2#B16#S3#AUNIT:#D1#N2/
.LIST BEX
.EVEN
ENDMSG

MOV #4,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #12,SP
.WORD J\$JMP
.WORD L10003-2

006206 040445 051524 040502
006234 040445 051524 040502

2142
2143 006302
2144 006302
2145 006302 104423
2146

L10003:
BGNMSG WRAPR2
WRAPR2:: PRINTB #IOBRD

TRAP C\$MSG

2147 006304
2148 006304
2149 006304
2150 006304 012746 005650
2151 006310 012746 000001
2152 006314 010600
2153 006316 104414
2154 006320 062706 000004
2155 006324
2156 006324 013746 002422
2157 006330 013746 002432
2158 006334 010446
2159 006336 012746 006362
2160 006342 012746 000004
2161 006346 010600
2162 006350 104415
2163 006352 062706 000012
2164 006356
2165 006356 000167
2166 006360 000072
2167

PRINTX #EEM2A,R4,TEMP1,UNIT

MOV #IOBRD,-
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP

006362 040445 051524 040502
006407 045 052101 041123

2168
2169 006454
2170 006454
2171 006454 104423
2172

EXIT MSG
EEM2A: .NLIST BEX
.ASCII /#ATSBA EXPECTED=#B8#N/
.ASCIZ /#ATSBA ACTUAL=#S2#B8#S3#AUNIT:#D1#N2/
.LIST BEX
.EVEN
ENDMSG

MOV UNIT,-(S
MOV TEMP1,-(
MOV R4,-(SP)
MOV #EEM2A,-
MOV #4,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #12,SP

.WORD J\$JMP
.WORD L10004-2

2173 006456
2174 006456
2175 006456
2176 006456 012746 005650
2177 006462 012746 000001
2178 006466 010600
2179 006470 104414
2180 006472 062706 000004
2181 006476
2182 006476 013746 002422
2183 006502 013746 002434

L10004:
BGNMSG WRAPR3
WRAPR3:: PRINTB #IOBRD

TRAP C\$MSG

PRINTX #EEM3A,R4,TEMP2,UNIT

MOV #IOBRD,-
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #4,SP

MOV UNIT,-(S
MOV TEMP2,-(

Line No.	Module	Offset	Inst 1	Inst 2	Inst 3	Inst 4	Inst 5	Inst 6	Inst 7
2184	006506	010446							
2185	006510	012746	006534						MOV R4, -(SP)
2186	006514	012746	000004						MOV @EEM3A, -
2187	006520	010600							MOV @4, -(SP)
2188	006522	104415							MOV SP, R0
2189	006524	062706	000012						TRAP C\$PNTX
2190	006530					EXIT	MSG		ADD @12, SP
2191	006530	000167							
2192	006532	000072							.WORD J\$JMP
2193									.WORD L10005-2
	006534	040445	051524	051123		EEM3A:	.NLIST	BEX	
	006561	045	052101	051523			.ASCII	/@ATSSR EXPECTED=@B8@N/	
							.ASCIZ	/@ATSSR ACTUAL=@S2@B8@S3@AUNIT:@D1@N2/	
							.LIST	BEX	
							.EVEN		
							ENDMSG		
2194						L10005:			
2195	006626								
2196	006626								
2197	006626	104423							TRAP C\$MSG
2198									
2199	006630					WRAPR4::	BGNMSG	WRAPR4	
2200	006630						PRINTB	@TS11BD	
2201	006630								
2202	006630	012746	005041						
2203	006634	012746	000001						MOV @TS11BD,
2204	006640	010600							MOV @1, -(SP)
2205	006642	104414							MOV SP, R0
2206	006644	062706	000004						TRAP C\$PNTB
2207	006650						PRINTX	@EEM4A, R2, TEMP2, UNIT	ADD @4, SP
2208	006650	013746	002422						
2209	006654	013746	002434						MOV UNIT, -(S
2210	006660	010246							MOV TEMP2, -(
2211	006662	012746	006706						MOV R2, -(SP)
2212	006666	012746	000004						MOV @EEM4A, -
2213	006672	010600							MOV @4, -(SP)
2214	006674	104415							MOV SP, R0
2215	006676	062706	000012						TRAP C\$PNTX
2216	006702						EXIT	MSG	ADD @12, SP
2217	006702	000167							
2218	006704	000072							.WORD J\$JMP
2219									.WORD L10006-2
	006706	040445	051524	051123		EEM4A:	.NLIST	BEX	
	006733	045	052101	051523			.ASCII	/@ATSSR EXPECTED=@B8@N/	
							.ASCIZ	/@ATSSR ACTUAL=@S2@B8@S3@AUNIT:@D1@N2/	
							.LIST	BEX	
							.EVEN		
							ENDMSG		
2220						L10006:			
2221	007000								
2222	007000								
2223	007000	104423							TRAP C\$MSG
2224									
2225	007002					TAEM::	BGNMSG	TAEM	
2226	007002						PRINTX	@TAEMA, UNIT	
2227	007002								
2228	007002	013746	002422						MOV UNIT, -(S
2229	007006	012746	007102						MOV @TAEMA, -
2230	007012	012746	000002						MOV @2, -(SP)
2231	007016	010600							MOV SP, R0
2232	007020	104415							TRAP C\$PNTX
2233	007022	062706	000006						ADD @6, SP

2234	007026					PRINTX	#TAEMB,EXTRAC		
2235	007026	013746	002342					MOV	EXTRAC,-
2236	007032	012746	007153					MOV	#TAEMB,-
2237	007036	012746	000002					MOV	#2,-(SP)
2238	007042	010600						MOV	SP,RO
2239	007044	104415						TRAP	C\$PNTX
2240	007046	062706	000006					ADD	#6,SP
2241	007052					PRINTX	#TAEMC,ACTRAC		
2242	007052	013746	002360					MOV	ACTRAC,-
2243	007056	012746	007207					MOV	#TAEMC,-
2244	007062	012746	000002					MOV	#2,-(SP)
2245	007066	010600						MOV	SP,RO
2246	007070	104415						TRAP	C\$PNTX
2247	007072	062706	000006					ADD	#6,SP
2248	007076					EXIT	MSG		
2249	007076	000167							
2250	007100	000156						.WORD	J\$JMP
2251								.WORD	L10007-2
	007102	040445	051124	041501	TAEMA:	.NLIST	BEX		
	007153	045	052101	045522	TAEMB:	.ASCIZ	/#ATRACK:#S9#S6#AP76543210#S3#AUNIT:#D1#N/		
	007207	045	052101	045522	TAEMC:	.ASCIZ	/#ATRK ACT EXP DATA:#S4#B9#N/		
						.LIST	BEX		
						.EVEN			
						ENDMSG			
2252		007260				L10007:			
2253	007260								
2254	007260								
2255	007260	104423						TRAP	C\$MSG
2256									
2257	007262				SKDAEM:	BGNMSG	SKDAEM		
2258	007262								
2259	007262					PRINTX	#DAEM,UNIT		
2260	007262	013746	002422						
2261	007266	012746	007472					MOV	UNIT,-(S
2262	007272	012746	000002					MOV	#DAEM,-(
2263	007276	010600						MOV	#2,-(SP)
2264	007300	104415						MOV	SP,RO
2265	007302	062706	000006					TRAP	C\$PNTX
2266	007306					PRINTX	#DAEMA		
2267	007306	012746	007545					ADD	#6,SP
2268	007312	012746	000001					MOV	#DAEMA,-
2269	007316	010600						MOV	#1,-(SP)
2270	007320	104415						MOV	SP,RO
2271	007322	062706	000004					TRAP	C\$PNTX
2272	007326					PRINTX	#DAEMB,EXORDY,ACORDY,EX1DTR,AC1DTR		
2273	007326	013746	002356					ADD	#4,SP
2274	007332	013746	002340					MOV	AC1DTR,-
2275	007336	013746	002354					MOV	EX1DTR,-
2276	007342	013746	002336					MOV	ACORDY,-
2277	007346	012746	007614					MOV	EXORDY,-
2278	007352	012746	000005					MOV	#DAEMB,-
2279	007356	010600						MOV	#5,-(SP)
2280	007360	104415						MOV	SP,RO
2281	007362	062706	000014					TRAP	C\$PNTX
2282	007366					PRINTX	#DAEMC,EXTRAC,ACTRAC,EXDATA,ACDATA		
2283	007366	013746	002362					ADD	#14,SP
2284	007372	013746	002344					MOV	ACDATA,-
2285	007376	013746	002360					MOV	EXDATA,-
								MOV	ACTRAC,-

2286 007402 013746 002342
 2287 007406 012746 007703
 2288 007412 012746 000005
 2289 007416 010600
 2290 007420 104415
 2291 007422 062706 000014
 2292 007426
 2293 007426 013746 002366
 2294 007432 013746 002350
 2295 007436 013746 002364
 2296 007442 013746 002346
 2297 007446 012746 007765
 2298 007452 012746 000005
 2299 007456 010600
 2300 007460 104415
 2301 007462 062706 000014
 2302 007466
 2303 007466 000167
 2304 007470 000362
 2305

PRINTX #DAEMD,EXTRDD,ACTRDD,EXODTR,ACODTR

MOV EXTRAC,-
 MOV #DAEMC,-
 MOV #5,-(SP)
 MOV SP,RO
 TRAP C#PNTX
 ADD #14,SP
 MOV ACODTR,-
 MOV EXODTR,-
 MOV ACTRDD,-
 MOV EXTRDD,-
 MOV #DAEMD,-
 MOV #5,-(SP)
 MOV SP,RO
 TRAP C#PNTX
 ADD #14,SP

EXIT MSG

.WORD J\$JMP
 .WORD L10010-2

007472 051445 022471 032523
 007545 045 052101 040522
 007614 040445 052517 050124
 007650 040445 020061 051117
 007703 045 052101 040522
 007737 045 042101 052101
 007765 045 052101 040522
 010020 040445 020060 051117

.NLIST BEX
 DAEM: .ASCIZ /#S9#S5#AEXPECTED#S7#AACTUAL#S3#AUNIT:#D1#N/
 DAEMA: .ASCIZ /#ATRACK:#S8#AP76543210#S5#AP76543210#N/
 DAEMB: .ASCII /#AOUTPUT READY:#S#B9#S5#B9#N/
 .ASCIZ /#A1 OR DEAD:#S4#B9#S5#B9#N/
 DAEMC: .ASCII /#ATRACK ACTIVE:#S#B9#S5#B9#N/
 .ASCIZ /#ADATA:#S9#B9#S5#B9#N/
 DAEMD: .ASCII /#ATRACK DEAD:#S3#B9#S5#B9#N/
 .ASCIZ /#A0 OR DEAD:#S4#B9#S5#B9#N2/
 .LIST BEX
 .EVEN
 ENDMSG

2306
 2307 010054
 2308 010054
 2309 010054 104423
 2310
 2311 010056
 2312 010056
 2313 010056
 2314 010056 013746 002422
 2315 010062 013746 002370
 2316 010066 013746 002352
 2317 010072 013746 002420
 2318 010076 012746 010122
 2319 010102 012746 000005
 2320 010106 010600
 2321 010110 104415
 2322 010112 062706 000014
 2323 010116
 2324 010116 000167
 2325 010120 000102
 2326

L10010:

TRAP C#MSG

BGNMSG ROMEM
 ROMEM::
 PRINTX #ROMEMA,ROMLKI,EXROML,ACROML,UNIT

MOV UNIT,-(S
 MOV ACROML,-
 MOV EXROML,-
 MOV ROMLKI,-
 MOV #ROMEMA,
 MOV #5,-(SP)
 MOV SP,RO
 TRAP C#PNTX
 ADD #14,SP

EXIT MSG

.WORD J\$JMP
 .WORD L10011-2

010122 040445 042101 051104
 010143 045 042501 050130
 010163 045 040501 052103

.NLIST BEX
 ROMEMA: .ASCII /#ADDRESS=#S#04#N/
 .ASCII /#AEXPECTED=#03#N/
 .ASCIZ /#AACTUAL=#S2#03#S3#AUNIT:#D1#N2/
 .LIST BEX
 .EVEN
 ENDMSG

2327 010224
 2328 010224

Line	Area	Code	Msg	Time	Text	Op	Target
2329	010224				L10011:		
2330	010224	104423					TRAP C\$MSG
2331							
2332	010226	040445	051524	051123	TERMA: .NLIST BEX .ASCIZ /#ATSSR =#S#06#S3#AUNIT:#D1#N/ .LIST BEX .EVEN		
2333		010264					
2334							
2335							
2336							
2337	010264				BGNMSG SCHER1		
2338	010264				SCHER1::		
2339	010264				PRINTB #TS11BD		
2340	010264	012746	005041				MOV #TS11BD,
2341	010270	012746	000001				MOV #1,-(SP)
2342	010274	010600					MOV SP,RO
2343	010276	104414					TRAP C\$PNTB
2344	010300	062706	000004				ADD #4,SP
2345	010304				PRINTX #SCHERA		
2346	010304	012746	010330				MOV #SCHERA,
2347	010310	012746	000001				MOV #1,-(SP)
2348	010314	010600					MOV SP,RO
2349	010316	104415					TRAP C\$PNTX
2350	010320	062706	000004				ADD #4,SP
2351	010324				EXIT MSG		
2352	010324	000167					.WORD J\$JMP
2353	010326	000064					.WORD L10012-2
2354	010330	040445	051524	051123	SCHERA: .NLIST BEX .ASCIZ /#ATSSR NBA NOT SET ON COMD BEFORE SET CHAR ISSUED#N/ .LIST BEX .EVEN		
2355					ENDMSG		
2356	010414				L10012:		
2357	010414						
2358	010414	104423					TRAP C\$MSG
2359							
2360	010416				BGNMSG SCHER2		
2361	010416				SCHER2::		
2362	010416				PRINTB #TS11BD		
2363	010416	012746	005041				MOV #TS11BD,
2364	010422	012746	000001				MOV #1,-(SP)
2365	010426	010600					MOV SP,RO
2366	010430	104414					TRAP C\$PNTB
2367	010432	062706	000004				ADD #4,SP
2368	010436				PRINTX #SCHERB		
2369	010436	012746	010462				MOV #SCHERB,
2370	010442	012746	000001				MOV #1,-(SP)
2371	010446	010600					MOV SP,RO
2372	010450	104415					TRAP C\$PNTX
2373	010452	062706	000004				ADD #4,SP
2374	010456				EXIT MSG		
2375	010456	000167					.WORD J\$JMP
2376	010460	000046					.WORD L10013-2
2377	010462	040445	051524	051123	SCHERB: .NLIST BEX .ASCIZ /#ATSSR NBA NOT CLEARED BY SET CHAR#N/ .LIST BEX .EVEN		
2378		010530					


```

2430
2431 010772          BGNMSG  SCHER5
2432 010772          SCHER5::
2433 010772          PRINTB  #SCHERE
2434 010772 012746 011036          MOV  #SCHERE,
2435 010776 012746 000001          MOV  #1,-(SP)
2436 011002 010600          MOV  SP,RO
2437 011004 104414          TRAP C:PNTB
2438 011006 062706 000004          ADD  #4,SP
2439 011012          PRINTX  #SCHERF
2440 011012 012746 011122          MOV  #SCHERF,
2441 011016 012746 000001          MOV  #1,-(SP)
2442 011022 010600          MOV  SP,RO
2443 011024 104415          TRAP C:PNTX
2444 011026 062706 000004          ADD  #4,SP
2445 011032          EXIT  MSG
2446 011032 000167          .WORD J$JMP
2447 011034 000146          .WORD L10016-2
2448
      011036 040445 051524 030461  SCHERE: .NLIST  BEX
      011122 040445 042523 044522  SCHERF: .ASCIZ  /#ATS11 BOARD OR SERIAL BUS CABLE OR M8965 SLOT 15#N/
      .ASCIZ  /#ASERIAL BUS PARITY ERROR SPE OR BPE TSSR FC=2#N/
      .LIST  BEX
      .EVEN
      ENDMSG
2449
2450 011204          L10016:
2451 011204
2452 011204 104423          TRAP  C$MSG
2453
2454 011206          BGNMSG  SCHER6
2455 011206          SCHER6::
2456 011206          PRINTB  #SCHERG
2457 011206 012746 011252          MOV  #SCHERG,
2458 011212 012746 000001          MOV  #1,-(SP)
2459 011216 010600          MOV  SP,RO
2460 011220 104414          TRAP C:PNTB
2461 011222 062706 000004          ADD  #4,SP
2462 011226          PRINTX  #SCHERH
2463 011226 012746 011274          MOV  #SCHERH,
2464 011232 012746 000001          MOV  #1,-(SP)
2465 011236 010600          MOV  SP,RO
2466 011240 104415          TRAP C:PNTX
2467 011242 062706 000004          ADD  #4,SP
2468 011246          EXIT  MSG
2469 011246 000167          .WORD J$JMP
2470 011250 000070          .WORD L10017-2
2471
      011252 040445 034115 033071  SCHERG: .NLIST  BEX
      011274 040445 044523 047514  SCHERH: .ASCIZ  /#AM8966 SLOT 14#N/
      .ASCIZ  /#ASILO PARITY ERROR SIP TSSR FC=2#N/
      .LIST  BEX
      .EVEN
      ENDMSG
2472
2473 011342          L10017:
2474 011342
2475 011342 104423          TRAP  C$MSG
2476
2477
2478 011344          BGNMSG  SCHER7
2479 011344          SCHER7::

```

```

2480 011344 PRINTB #SCHERI
2481 011344 012746 011434 MOV #SCHERI,
2482 011350 012746 000001 MOV #1,-(SP)
2483 011354 010600 MOV SP,R0
2484 011356 104414 TRAP C$PNTB
2485 011360 062706 000004 ADD #4,SP
2486 011364
2487 011364 012707 011522 LET R2 := #PCHLTB
2488 011370 WHILE (R2) NE #177777 DO MOV #PCHLTB,
2489 011370
2490 011370 021227 177777 50002$:
2491 011374 001415 CMP (R2),#17
2492 011376 LET PCHTAD := (R2) BEQ 50003$
2493 011376 011237 011516 MOV (R2),PCH
2494 011402 PRINTB PCHTAD
2495 011402 013746 011516 MOV PCHTAD,-
2496 011406 012746 000001 MOV #1,-(SP)
2497 011412 010600 MOV SP,R0
2498 011414 104414 TRAP C$PNTB
2499 011416 062706 000004 ADD #4,SP
2500 011422 LET R2 := R2 * #2
2501 011422 062702 000002 ADD #2,R2
2502 011426 ENDDO
2503 011426 000760
2504 011430
2505 011430 EXIT MSG 50003$:
2506 011430 000167 .WORD J$JMP
2507 011432 000064 .WORD L10020-2
2508 011434 040445 051524 051123
2509
2510 011516 000000 SCHERI: .NLIST BEX
2511 011520 .ASCIZ /#ATSSR FC=2 FATAL MICRO ERR HALTS PC=1750-1777#N2/
2512 011520 .LIST BEX
2513 011520 104423 PCHTAD: .EVEN
2514 .WORD 0
2515 011522 011614 011710 011760 L10020: ENDMMSG
2516 011530 012022 012101 012143 TRAP C$MSG
2517 011536 012217 012301 012363
2518 011544 012426 012467 012530
2519 011552 012576 012640 012710
2520 011560 012766 013044 013111
2521 011566 013157 013225 013277
2522 011574 013361 013437 013521
2523 011602 013575 013664 013751
2524 011610 014033
2525 011612 177777
2526
2527
011614 040445 041520 042440 PCHDR: .ASCIZ /#APC ERROR HALTS BAD MODULES: M8962,M8963,M8964,OR M8967#N/
011710 051445 031462 040445 PCHDRA: .ASCIZ /#S23#ASLOT: 8 11 9 12#N/
011760 040445 050040 022503 PCHDRB: .ASCIZ /#A PC#S17#ACOMMENT#S13#AMODULE#N2/
012022 040445 033461 030065 PCHT50: .ASCIZ /#A1750 OPERATION IO PROBLEM#S14#AM8967,TS11#N/
012101 045 030501 032467 PCHT51: .ASCIZ /#A1751 SPURIOUS ATTN#S22#AM8963#N/

```

012143	045	030501	032467	PCHT52:	.ASCIZ	/#A1752 STK PAR ERR ON OVERFLOW#S12#AM8963#N/	
012217	045	030501	032467	PCHT53:	.ASCIZ	/#A1753 STK NOT EMPTY AT END OF TASK	M8963#N/
012301	045	030501	032467	PCHT54:	.ASCIZ	/#A1754 STK PTR REG WON'T HOLD DATA	M8963#N/
012363	045	030501	032467	PCHT55:	.ASCIZ	/#A1755 UTSTM - BRANCH#S21#AM8962#N/	
012426	040445	033461	033065	PCHT56:	.ASCIZ	/#A1756 UTSTM - ZBIT#S23#AM8962#N/	
012467	045	030501	032467	PCHT57:	.ASCIZ	/#A1757 UTSTM - NBIT#S23#AM8962#N/	
012530	040445	033461	030066	PCHT60:	.ASCIZ	/#A1760 UTSTM - NOT Z BIT#S18#AM8962#N/	
012576	040445	033461	030466	PCHT61:	.ASCIZ	/#A1761 UTSTM - C BIT#S22#AM8962#N/	
012640	040445	033461	031066	PCHT62:	.ASCIZ	/#A1762 UTSTM - PCB LO TEST#S16#AM8962#N/	
012710	040445	033461	031466	PCHT63:	.ASCIZ	/#A1763 UTSTM - INTERNAL REG ERROR#S9#AM8962#N/	
012766	040445	033461	032066	PCHT64:	.ASCIZ	/#A1764 UTSTM - INTERNAL REG ERROR#S9#AM8962#N/	
013044	040445	033461	032466	PCHT65:	.ASCIZ	/#A1765 UTSTM - ADD TEST#S19#AM8962#N/	
013111	045	030501	033067	PCHT66:	.ASCIZ	/#A1766 UTSTM - A SUBTEST#S18#AM8962#N/	
013157	045	030501	033067	PCHT67:	.ASCIZ	/#A1767 UTSTM - BSUB TEST#S18#AM8962#N/	
013225	045	030501	033467	PCHT70:	.ASCIZ	/#A1770 UTSTM - SHIFT LR TEST#S14#AM8962#N/	
013277	045	030501	033467	PCHT71:	.ASCIZ	/#A1771 UTSTM - LOGICAL:AND,OR,XOR,NAND	M8962#N/
013361	045	030501	033467	PCHT72:	.ASCIZ	/#A1772 STAKM - STACK PARITY TEST#S10#AM8963#N/	
013437	045	030501	033467	PCHT73:	.ASCIZ	/#A1773 STAKM - STACK UNDER-OVERFLOW	M8963#N/
013521	045	030501	033467	PCHT74:	.ASCIZ	/#A1774 STAKM - STACK DATA TEST#S12#AM8963#N/	
013575	045	030501	033467	PCHT75:	.ASCIZ	/#A1775 STAKM - CAPSTAN BUS DATA TEST	M8963,G159#N/
013664	040445	044103	041505	PCH75A:	.ASCIZ	/#ACHECK TACH PHASE,DECEL AND SPEED ADJUSTS ON G159#N/	
013751	045	030501	033467	PCHT76:	.ASCIZ	/#A1776 STAKM - CBUS BRANCH TEST	M8963#N/
014033	045	030501	033467	PCHT77:	.ASCIZ	/#A1777 STAKM - SIMULATED LIMIT ATTN TEST	M8963#N/

2528		014116		.LIST	BEX		
2529	014116			.EVEN			
2530	014116			BGNMSG	SCHERO		
2531	014116			SCHERO::			
2532	014116	012746	014162	PRINTB	#SCHERL		MOV #SCHERL,
2533	014122	012746	000001				MOV #1,-(SP)
2534	014126	010600					MOV SP,RO
2535	014130	104414					TRAP C\$PNTB
2536	014132	062706	000004				ADD #4,SP
2537	014136			PRINTX	#SCHERM		
2538	014136	012746	014205				MOV #SCHERM,
2539	014142	012746	000001				MOV #1,-(SP)
2540	014146	010600					MOV SP,RO
2541	014150	104415					TRAP C\$PNTX
2542	014152	062706	000004				ADD #4,SP
2543	014156			EXIT	MSG		
2544	014156	000167					.WORD J\$JMP
2545	014160	000042					.WORD L10021-2
2546				.NLIST	BEX		
	014162	040445	041501	SCHERL:	.ASCIZ	/#AAC LOW AT TS04#N/	
	014205	045	052101	SCHERM:	.ASCIZ	/#ATSSR FC=3#N/	
2547		014224		.LIST	BEX		
2548	014224			.EVEN			
2549	014224			ENDMSG			
2550	014224	104423		L10021:			TRAP C\$MSG
2551							
2552							
2553	014226			BGNMSG	ROMLER		
2554	014226			ROMLER::			


```

2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600 014454
2601 014454
2602 014454
2603 014454 112737 000001 002437
2604 014462
2605 014462
2606 014462 000002
2607
2608 014464
2609 014464
2610 014464
2611 014464 112737 000001 002437
2612 014472
2613 014472
2614 014472 000002
2615
2616 014474
2617 014474
2618 014474
2619 014474 112737 000001 002437
2620 014502
2621 014502
2622 014502 000002
2623
2624 014504
2625 014504
2626 014504
2627 014504 112737 000001 002437
2628 014512
2629 014512
2630 014512 000002

.SBTTL GLOBAL SUBROUTINES SECTION

;+
; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
; THAT ARE USED IN MORE THAN ONE TEST.
;--

; MODULES TO HANDLE TS04 INTERRUPTS.

TS4IN0: BGNSRV TS4IN0 ;DEVICE 0.
        LET CTLFLG :B= #1 ;SET INTERRUPT OCCURRED FLAG.
        ENDSRV ;MOV#1.CTLFL
L10024: RTI

TS4IN1: BGNSRV TS4IN1 ;DEVICE 1.
        LET CTLFLG :B= #1 ;SET INTERRUPT OCCURRED FLAG.
        ENDSRV ;MOV#1.CTLFL
L10025: RTI

TS4IN2: BGNSRV TS4IN2 ;DEVICE 2.
        LET CTLFLG :B= #1 ;SET INTERRUPT OCCURRED FLAG.
        ENDSRV ;MOV#1.CTLFL
L10026: RTI

TS4IN3: BGNSRV TS4IN3 ;DEVICE 3.
        LET CTLFLG :B= #1 ;SET INTERRUPT OCCURRED FLAG.
        ENDSRV ;MOV#1.CTLFL
L10027: RTI

```

```

2631 ; SUBROUTINE TO WAIT UP TO 1 MINUTE FOR THE SSR BIT
2632 ; TO SET - REPORT AN ERROR IF IT DOESN'T SET AND ATTEMPT TO DROP UNIT
2633 ;
2634 014514 005037 002426 WT4SSR:: CLR TIME ;INIT TIMEOUT COUNTER.
2635 014520 1$: BREAK ;GO TO SUPERVISOR.
2636 014520 104422 TRAP C$BRK
2637 014522 105777 165476 TSTB @TSSR ;READY?
2638 014526 100432 BMI 2$ ;BR IF SO
2639 014530 005337 002426 DEC TIME ;WAIT...
2640 014534 001371 BNE 1$ ;DONE WAITING? BR IF NOT.
2641 014536 017737 165462 002434 MOV @TSSR,TEMP2 ;GET TSSR
2642 014544 012701 014616 MOV @TERMTB,R1 ;
2643 014550 ERRDF 2,SSROFF,LONMSG ;IF SO, REPORT AN ERR IF SSR NOT SET YET.
2644 014550 104455 TRAP C$ERDF
2645 014552 000002 .WORD 2
2646 014554 002570 .WORD SSROFF
2647 014556 006060 .WORD LONMSG
2648 014560 PRINTX @TERMA,TEMP2,UNIT ;PRINT TSSR CONTENTS
2649 014560 013746 002422 MOV UNIT, -(S
2650 014564 013746 002434 MOV TEMP2, -(
2651 014570 012746 010226 MOV @TERMA, -
2652 014574 012746 000003 MOV #3, -(SP)
2653 014600 010600 MOV SP,RO
2654 014602 104415 TRAP C$PNTX
2655 014604 062706 000010 ADD #10,SP
2656 014610 CALL DROPU ;ATTEMPT TO DROP UNIT
2657 014610 004737 014654 JSR PC,DROPU
2658 ;IF DROPPED PROG ABORTS CURRENT PASS
2659 ;IF INHIBIT DROP FLAG SET BY OPERATOR (IDU)
2660 ;THEN PROGRAM CONTINUES
2661 014614 000207 2$: RTS PC ;RETURN.
2662
2663 014616 003632 003653 003716 TERMTB: .WORD TERM01,TERM02,TERM03
2664 014624 003777 004060 .WORD TERM05,TERM06
2665 014630 004157 004233 004310 .WORD TERM07,TERM10,TERM11
2666 014636 004413 004500 004551 .WORD TERM12,TERM13,TERM14
2667 014644 004622 004707 005000 .WORD TERM15,TERM16,TERM17
2668 014652 177777 .WORD ENDTBL
2669
2670 ; SUBR TO ATTEMPT TO DROP A UNIT FLAGGED WITH A DEVICE FATAL ERROR
2671
2672 014654 DROPU:: DODU LUNIT ;EXECUTE DROP MACRO
2673 014654 013700 027712 MOV LUNIT,RO
2674 014660 104451 TRAP C$DODU
2675 014662 GPHARD LUNIT,TIME ;IS UNIT DROPPED
2676 014662 013700 027712 MOV LUNIT,RO
2677 014666 104442 TRAP C$GPHRD
2678 014670 010037 002426 MOV RO,TIME
2679 014674 BCOMPLETE 1$ ;NO RETURN TO CALLER TO PROCEED
2680 014674 103401 DOCLN ;YES EXEC CLEAN-UP CODE TO START NEW PASS
2681 014676 BCS 1$
2682 014676 104444 TRAP C$DCLN
2683 014700 000207 1$: RTS PC
2684
2685 ; SUBR TO WAIT AN EXTRA LONG TIME FOR THE SSR BIT TO SET
2686
    
```

```

2687
2688
2689
2690
2691 014702
2692 014702
2693 014702 012737 000001 002426
2694 014710 000402
2695 014712
2696 014712 005237 002426
2697 014716
2698 014716 023727 002426 000027
2699 014724 003023
2700 014726
2701 014726 032777 000200 165270
2702 014734 001401
2703 014736
2704 014736 000416
2705 014740
2706 014740
2707 014740
2708 014740 012727 000372
2709 014744 000000
2710 014746 013727 002116
2711 014752 000000
2712 014754 005367 177772
2713 014760 001375
2714 014762 005367 177756
2715 014766 001367
2716 014770
2717 014770 104422
2718 014772
2719 014772
2720 014772
2721 014772 000747
2722 014774
2723 014774
2724 014774
2725 014774
2726 014774 023727 002426 000027
2727 015002 003511
2728 015004
2729 015004 012746 015230
2730 015010 012746 000001
2731 015014 010600
2732 015016 104417
2733 015020 062706 000004
2734 015024
2735 015024
2736 015024 012737 000001 002426
2737 015032 000402
2738 015034
2739 015034 005237 002426
2740 015040
2741 015040 023727 002426 000160
2742 015046 003023

```

```

:   NEEDED AFTER AN INIT IS SENT TO TS11 TO WAIT FOR LOAD SEQ
:   TO COMPLETE.  REPORT ERROR AND DROP UNIT IF EXTRA
:   TIME IS EXCEEDED.
WAITJR::BEGIN COUNTER1
      INCR TIME FROM #1 TO #23. BY #1
      IF #TS.SSR SETIN #TSSR THEN
          LEAVE COUNTER1
      ELSE
          DELAY 250.
      BREAK
      ENDIF
      ENDINC
      END COUNTER1
      IF TIME GT #23. THEN
          PRINTF #WAITMR
          BEGIN COUNTER2
            INCR TIME FROM #1 TO #160 BY #1

```

```

MOV #1, TIME
BR 50005$
50006$: INC TIME
50005$: CMP TIME, #23
BGT 50007$
BIT #TS.SSR,
BEQ 50010$
BR 50004$
50010$: MOV #250., (P
        .WORD 0
        MOV L$DLY., (P
        .WORD 0
        DEC -6(PC)
        BNE .-4
        DEC -22(PC)
        BNE .-20
        TRAP C$BRK
50011$: BR 50006$
50007$: BR 50004$
50004$: CMP TIME, #23
BLE 50012$
MOV #WAITMR,
MOV #1., (SP)
MOV SP, RO
TRAP C$PNTF
ADD #4, SP
MOV #1, TIME
BR 50014$
50015$: INC TIME
50014$: CMP TIME, #160
BGT 50016$

```

2743	015050				IF @TS.SSR SETIN @TSSR THEN		
2744	015050	032777	000200	165146		BIT	@TS.SSR,
2745	015056	001401				BEQ	50017\$
2746	015060				LEAVE COUNTER2		
2747	015060	000416				BR	50013\$
2748	015062				ELSE		
2749	015062				DELAY 200.	50017\$:	
2750	015062						
2751	015062	012727	000310			MOV	@200.,(P
2752	015066	000000				.WORD	0
2753	015070	013727	002116			MOV	L\$DLY,(P
2754	015074	000000				.WORD	0
2755	015076	005367	177772			DEC	-6(PC)
2756	015102	001375				BNE	.-4
2757	015104	005367	177756			DEC	-22(PC)
2758	015110	001367				BNE	.-20
2759	015112				BREAK		
2760	015112	104422				TRAP	C\$BRK
2761	015114				ENDIF		
2762	015114				ENDINC	50020\$:	
2763	015114						
2764	015114	000747				BR	50015\$
2765	015116				END COUNTER2	50016\$:	
2766	015116						
2767	015116				IF TIME GT @160 THEN	50013\$:	
2768	015116						
2769	015116	023727	002426	000160		CMP	TIME,@16
2770	015124	003430			LET TEMP2 := @TSSR	BLE	50021\$
2771	015126				LET R1 := @TERMTB	MOV	@TSSR,TE
2772	015126	017737	165072	002434	ERRDF 2,SSROFF,LONMSG	MOV	@TERMTB,
2773	015134					TRAP	C\$ERDF
2774	015134	012701	014616			.WORD	2
2775	015140	104455				.WORD	SSROFF
2776	015140	000002			PRINTX @TERMA,TEMP2,UNIT	.WORD	LONMSG
2777	015142	000002				MOV	UNIT,-(S
2778	015144	002570				MOV	TEMP2,-(
2779	015146	006060				MOV	@TERMA,-
2780	015150					MOV	@3,-(SP)
2781	015150	013746	002422			MOV	SP,R0
2782	015154	013746	002434			TRAP	C\$PNTX
2783	015160	012746	010226		JSR PC,DROPU	ADD	@10,SP
2784	015164	012746	000003		ELSE		
2785	015170	010600					
2786	015172	104415			PRINTF @CRMSG	50021\$:	
2787	015174	062706	000010				
2788	015200	004737	014654			MOV	@CRMSG,
2789	015204					MOV	@1,-(SP)
2790	015204	000410				MOV	SP,R0
2791	015206					TRAP	C\$PNTF
2792	015206					ADD	@4,SP
2793	015206	012746	005765				
2794	015212	012746	000001				
2795	015216	010600					
2796	015220	104417					
2797	015222	062706	000004				
2798	015226				ENDIF		

2799	015226					500221:
2800	015226				ENDIF	500121:
2801	015226					
2802	015226	000207			RTS PC	
2803						
2804	015230	047045	040445	040527	WAITMR: .ASCIZ /N#AWAITING FOR RDY.../	
2805	015236	052111	047111	020107		
2806	015244	047506	020122	042122		
2807	015252	027131	027056	000		
2808		015260			.EVEN	
2809						
2810						
2811						

```

2812
2813
2814      :      SUBR TO WAIT FOR TAPE MOTION TO CEASE WHEN AN INIT TO THE
2815      :      ON-LINE TS11 INITIATED A LOAD SEQUENCE.
2816 015260 000240      WAITMT:: NOP
2817 015262 000240      NOP
2818 015264 000240      NOP
2819 015266 000240      NOP
2820 015270      IF #MOT SETIN XSTATO THEN
2821 015270 032737 000200 002266
2822 015276 001501
2823 015300      PRINTF #WAITMS
2824 015300 012746 015504
2825 015304 012746 000001
2826 015310 010600
2827 015312 104417
2828 015314 062706 000004
2829 015320      BEGIN COUNTER
2830 015320      INCR TIME FROM #1 TO #160 BY #1
2831 015320 012737 000001 002426
2832 015326 000402
2833 015330
2834 015330 005237 002426
2835 015334
2836 015334 023727 002426 000160
2837 015342 003031
2838 015344      LET CMDPKT := #GES
2839 015344 012737 100017 002230
2840 015352      LET #TSDB := #CMDPKT
2841 015352 012777 002230 164634
2842 015360      DELAY 200.
2843 015360 012727 000310
2844 015364 000000
2845 015366 013727 002116
2846 015372 000000
2847 015374 005367 177772
2848 015400 001375
2849 015402 005367 177756
2850 015406 001367
2851 015410      BREAK
2852 015410 104422
2853 015412
2854 015412 032737 000200 002266
2855 015420 001001
2856 015422      LEAVE COUNTER
2857 015422 000401
2858 015424      ENDIF
2859 015424
2860 015424      ENDINC
2861 015424 000741
2862 015426
2863 015426
2864 015426
2865 015426
2866 015426 023727 002426 000160
2867 015434 003412

```

```

      BIT #MOT,XST
      BEQ 50023$
      MOV #WAITMS,
      MOV #1,-(SP)
      MOV SP,R0
      TRAP C$PNTF
      ADD #4,SP
      MOV #1,TIME
      BR 50025$
50026$:
      INC TIME
50025$:
      CMP TIME,#16
      BGT 50027$
      MOV #GES,CMD
      MOV #CMDPKT,
      MOV #200..(P
      .WORD 0
      MOV L$DLY,(P
      .WORD 0
      DEC -6(PC)
      BNE -.4
      DEC -22(PC)
      BNE .-20
      TRAP C$BRK
      BIT #MOT,XST
      BNE 50030$
      BR 50024$
50030$:
      BR 50026$
50027$:
      BR 50024$:
      CMP TIME,#16
      BLE 50031$

```

2868	015436				PRINTB #WAITEM		
2869	015436	012746	015542			MOV	#WAITEM,
2870	015442	012746	000001			MOV	#1,-(SP)
2871	015446	010600				MOV	SP,RO
2872	015450	104414				TRAP	C\$PNTB
2873	015452	062706	000004			ADD	#4,SP
2874	015456				DOCLN		
2875	015456	104444				TRAP	C\$DCLN
2876	015460				ELSE		
2877	015460	000410				BR	50032\$
2878	015462						
2879	015462				PRINTF #CRMSG	50031\$:	
2880	015462	012746	005765			MOV	#CRMSG,-
2881	015466	012746	000001			MOV	#1,-(SP)
2882	015472	010600				MOV	SP,RO
2883	015474	104417				TRAP	C\$PNTF
2884	015476	062706	000004			ADD	#4,SP
2885	015502				ENDIF		
2886	015502					50032\$:	
2887	015502				ENDIF		
2888	015502					50023\$:	
2889	015502	000207			RTS PC		
2890							
2891							
2892	015504	040445	040527	052111	WAITMS: .ASCIZ /#AWAITING FOR TAPE TO STOP.../		
2893	015512	047111	020107	047506			
2894	015520	020122	040524	042520			
2895	015526	052040	020117	052123			
2896	015534	050117	027056	000056			
2897	015542	040445	040527	052111	WAITEM: .ASCII /#AWAIT TOO LONG#N/		
2898	015550	052040	047517	046040			
2899	015556	047117	022507	116			
2900	015563	045	051501	044527	.ASCIZ /#ASWITCH UNIT OFF-LINE, RESTART DIAG#N/		
2901	015570	041524	020110	047125			
2902	015576	052111	047440	043106			
2903	015604	046055	047111	026105			
2904	015612	051040	051505	040524			
2905	015620	052122	042040	040511			
2906	015626	022507	000116				
2907							
2908					.EVEN		

```

2909 ;S/R TO GENERATE AN EVEN OR ODD PAR BIT. ENTER WITH R1 CONTAINING THE #
2910 ;FOR WHICH YOU WISH TO HAVE PAR GEN. EXIT WITH THE PARITY BIT IN THE
2911 ;LOW ORDER BIT OF TEMPO. CLOBBERREDS TEMPO
2912
2913 015632 012737 000001 002430 ODDPAR:: MOV #1,TEMPO ;INIT THIS REG FOR ADD PAR.
2914 015640 000137 015650 ;CONTINUE
2915 015644 005037 002430 EVNPAR: JMP PARGEN ;INIT THIS REG FOR EVEN PAR.
2916 015650 012737 000020 002434 PARGEN: MOV #20,TEMP2 ;INIT COUNTER
2917 015656 005337 002434 1$: DEC TEMP2 ;DONE ALL BITS?
2918 015662 100405 BMI 2$ ;BR IF SO
2919 015664 006001 ROR R1 ;ROTATE THE DATA.
2920 015666 103773 BCS 1$
2921 015670 005237 002430 INC TEMPO
2922 015674 000770 BR 1$ ;DO IT AGAIN
2923
2924 015676 006001 2$: ROR R1 ;ROTATE ONCE MORE TO GET ORIG CONTENTS
2925 015700 042737 177776 002430 BIC #177776,TEMPO ;CLR ALL BUT LO ORDER BIT IN TEMPO.
2926 ;LO ORDER BIT IS THE PAR BIT
2927 ;RETURNED TO THE CALLING ROUTINE.
2928 015706 000241 CLC ;CLR CARRY BIT
2929 015710 000207 RTS PC ;RETURN
2930
2931
2932
2933
2934 ;S/R TO LOAD THE DIABLK WITH R3 (DATA) BITS AND R4 (CONTROL) BITS WITH
2935 ;CORRECT PARITY.
2936 ; INPUTS: R3 DATA, R4 CONTROL, R5 DIABLK INDEX.
2937 ; OUTPUTS: R3 DATA (UNCHANGED), R4 CONTROL (WITH GOOD PARITY), R5
2938 ; UPDATED DIABLK INDEX.
2939 ; CALLS: ODDPAR SUBROUTINE
2940
2941 015712 110345 TALOAD:: MOVB R3,-(R5) ;LOAD DATA
2942 015714 005703 TST R3 ;ALL 0'S DATA?
2943 015716 001403 BEQ 1$ ;BR IF SO
2944 015720 052704 000020 BIS #IS.DAP,R4 ;IF NOT, SET THE PARITY DATA ALSO IN R4
2945 015724 000402 BR 3$ ;CONTINUE.
2946
2947 015726 042704 000020 1$: BIC #IS.DAP,R4 ;CLR THE PAR DATA BIT IN CNTRL WORD.
2948 015732 010401 3$: MOV R4,R1 ;SAVE IT.
2949 015734 004737 015632 JSR PC,ODDPAR ;CALC CONTROL WORD PARITY.
2950 015740 005737 002430 TST TEMPO ;IS PARITY BIT SET?
2951 015744 001402 BEQ 2$ ;BR IF NOT
2952 015746 052704 000200 BIS #IS.PAR,R4 ;IF SO, SET THE PAR BIT.
2953 015752 110445 MOVB R4,-(R5) ;LOAD THE CONTROL WORD
2954 015754 000207 RTS PC ;RETURN.

```



```

2955                                     ;S/R TO COMPLETE FILLING THE DIABLK WITH ENTRIES WHOSE FORMAT THE TS04
2956                                     ;RECOGNIZES.  UNUSED DIABLK LOCATIONS TO CONTAIN A TS04 RTS COMMAND.
2957
2958 015756 020527 002317             FIXDIA:: CMP      R5,#DIABLK+1           ;ARE WE ON DIABLK ODD BOUNDARY?
2959 015762 001406                   BEQ        1$                      ;BR IF SO
2960 015764 100407                   BMI        2$                      ;ARE WE ON DIABLK EVEN BOUNDARY?
2961                                     ;BR IF SO-WE'RE DONE.
2962 015766 112745 000033             MOVB      #POPJLO,-(R5)          ;LOAD TS04
2963 015772 112745 000000             MOVB      #POPJHI,-(R5)          ;RTS COMMAND.
2964 015776 000767                   BR         FIXDIA                ;REPEAT.
2965
2966
2967 016000 112745 177777             1$:      MOVB      #-1,-(R5)      ;MUST END ON AN EVEN BOUNDARY.
2968 016004 000207                   2$:      RTS        PC              ;RETURN.
2969
2970
2971
2972                                     ;S/R TO SETUP THE TS04 COMMAND PACKET AND INITIATE EXECUTION OF THE DIA
2973                                     ;COMMAND.
2974                                     ;
2975                                     ;   CALLS:: WT4SSR SUBROUTINE.
2976 016006 004737 014514             DIAEXE:: JSR      PC,WT4SSR        ;SEE IF SSR IS SET.  SHOULD BE
2977 016012 004737 016056             JSR      PC,INITM                ;INITIALIZE THE MESSAGE PACKET.
2978 016016 012777 002250 164170     MOV       #DIAPKT,@TSDB          ;INITIATE EXECUTION
2979 016024 004737 014514             JSR      PC,WT4SSR                ;WAIT FOR SSR TO SET
2980 016030 000207                   RTS        PC

```

```

2981 ;S/R TO SETUP THE COMMAND PACKET TO DO AN SCH COMMAND AND ISSUE THE
2982 ;COMMAND.
2983
2984 ; CALLS:: WT4SSR SUBROUTINE.
2985
2986 016032 004737 014702 SCHEXE:: JSR PC,WAITSR ;WAIT FOR SSR BIT TO SET.
2987 016036 004737 016056 JSR PC,INITM ;INITIALIZE THE MESSAGE PACKET.
2988 016042 012777 002240 164144 MOV #SCHPKT,@TSDB ;INITIATE EXECUTION.
2989 016050 004737 014702 JSR PC,WAITSR ;WAIT FOR SSR.
2990 016054 000207 RTS PC ;RETURN.
2991
2992
2993
2994 ;S/R TO INITIALIZE THE MESSAGE PACKET WITH ALL ONES.
2995 ; REGISTERS USED: R1
2996
2997 016056 INITM: LET R1 := #0 ;CLEAR LOCATION COUNTER.
2998 016056 005001 WHILE R1 NE #MSGEXT DO ;WHILE THERE ARE MORE LOCATIONS:
2999 016060 ; 50033$:
3000 016060 ; CMP R1,#MSGE
3001 016060 020127 000016 ; BEQ 50034$
3002 016064 001406 ; LET MSGPKT(R1) := #-1 ;INIT ONE LOCATION IN MSG PACKET.
3003 016066 ; LET R1 := R1 + #2 ;UPDATE COUNTER
3004 016066 012761 177777 002260 ; MOV #-1,MSGP
3005 016074 ; ADD #2,R1
3006 016074 062701 000002 ;END OF INIT LOOP.
3007 016100 ENDDO ;BR 50033$
3008 016100 000767
3009 016102
3010 016102 000207 RTS PC ;RETURN.

```

```

3011 ;S/R TO SHIFT BYTE A 1 IN A FIELD OF 0'S. WHEN WE'RE THROUGH
3012 ;DOING THAT, SHIFT BYTE A 0 IN A FIELD OF 1'S.
3013
3014 ; INPUTS: R4=DATA TO SHIFT.
3015 ; OUTPUTS: R4=SHIFTED DATA.
3016
3017 016104 006204 SHWRAP:: ASR R4 ;SHIFT THE DATA.
3018 016106 105704 TSTB R4 ;SHIFTING A 1 OR A 0?
3019 016110 100404 BMI 1$ ;BR IF SHIFTING A 0.
3020 016112 001005 BNE 2$ ;DONE SHIFTING A 1? BR IF NOT.
3021 016114 012704 177577 MOV #177577,R4 ;IF SO, SETUP TO SHIFT 0 IN FIELD OF 1'S
3022 016120 000402 BR 2$ ;CONTINUE.
3023
3024 016122 052704 000200 1$: BIS #200,R4 ;SET THE MSB.
3025 016126 000207 2$: RTS PC ;RETURN.
3026
3027
3028
3029 ;S/R TO CHECK FOR TRACK ACTIVE ERRORS.
3030 ;
3031 ;
3032 ; OUTPUTS:ACTRAC CONTAINS THE ACTUAL TRACK ACTIVE DATA IN BITS 0 THRU 8
3033
3034 016130 113737 002267 002360 TKACER:: MOVB MSGPKT-C18$TA,ACTRAC ;GET ACTUAL STATE OF ACTIVE BITS FOR CHA
3035 016136 LET R2 :B= MSGPKT-PRCHST ;GET THE TRK ACT DATA FOR CHAN 9
3036 016136 113702 002274 ;MOV B MSGPKT-P
3037 016142 032702 000004 BIT #CH9.TA,R2 ;IS IT SET?
3038 016146 001404 BEQ 1$ ;BR IF NOT
3039 016150 052737 000400 002360 BIS #BIT8,ACTRAC ;IF SO, SET THE APPROPRIATE BIT IN "ACTU
3040 016156 000403 BR 3$ ;CONTINUE.
3041
3042 016160 042737 000400 002360 1$: BIC #BIT8,ACTRAC ;CLEAR THE BIT IN "ACTUAL" REG.
3043 016166 042737 177000 002360 3$: BIC #177000,ACTRAC ;CLR GARBAGE.
3044 016174 023737 002342 002360 CMP EXTRAC,ACTRAC ;EXPECTED & ACTUAL DATA IDENTICAL?
3045 016202 001402 BEQ 2$ ;BR IF SO.
3046 016204 105237 002436 INCF ERRFLG ;SET THE ERR FLAG.
3047 016210 000207 2$: RTS PC ;RETURN.
3048

```

```

3049                                     ;COMMON S/R USED BY THE PE DATA TESTS TO BUILD THE DIABLK
3050                                     ;IN CORE.
3051                                     ;
3052                                     ;   INPUTS:
3053                                     ;           R2      =   PREAMBLE DATA.
3054                                     ;           R3      =   BYTE 1 DATA.
3055
3056 016212 004737 016240                 DATBLD:: JSR   PC,PEINIT                   ;GO SET THE WRAP TASK ADDR IN DIABLD
3057                                     ;AND INIT DIABLK INDEX.
3058 016216 112745 000012                 MOVB   #FC.DAT!FC.VCO,-(R5)       ;SET DATA & VCO MODE IN FMT CNTRL.
3059 016222 010304                       MOV    R3,R4                      ;SETUP SECOND BYTE OF DATA.
3060 016224 005104                       COM    R4                          ;COMPLEMENT THE BYTE 2 DATA.
3061 016226 042704 177000                 BIC   #177000,R4                 ;CLR GARBAGE BITS.
3062 016232 004737 016262                 JSR   PC,PEDATA                  ;LOAD THE PREAMBLE AND THE 2 DATA BYTES
3063                                     ;IN DIABLK
3064 016236 000207                       RTS    PC                          ;RETURN.
3065
3066
3067
3068                                     ;S/R TO LOAD THE DIABLK WITH THE TSO4 WRAP TASK ADDRESS.
3069                                     ;AND SETUP THE READ CONTROL WORD (RDCTLO) FOR MAINTENANCE MODE.
3070                                     ;
3071                                     ;IMPLICIT INPUTS:   DIABLK, DIAEXT, WRPLO, WRPHI
3072                                     ;OUTPUTS:           R5=DIABLK INDEX.
3073
3074 016240 012705 002336                 PEINIT:: MOV  #DIABLK-DIAEXT,R5    ;SAVE ADR OF LAST WORD IN DIABLK
3075                                     ;SO WE CAN FILL DIABLK IN REVERSE ORDER.
3076 016244                                     LET  -(R5) :B= #WRPLO              ;LOAD THE NEW WRAP TASK ADDR LO
3077 016244 112745 000005                                     MOVB  #WRPLO,-
3078 016250 112745 000200                 MOVB  #WRPHI,-(R5)              ;LOAD WRAP TASK ADDR HI.
3079 016254 112745 000100                 MOVB  #RD.MAI,-(R5)            ;SET MAINT MODE IN RDCTLO.
3080 016260 000207                       RTS    PC
3081
3082

```

```

3083 ;S/R TO LOAD THE DIABLK WITH THE ALL 1'S PREAMBLE CHARACTER AND 2 BYTES
3084 ;OF DATA AND CONTROL. IT ALSO LOADS THE EXPECTED UNDESKEWED DATA IN THE
3085 ;FORMATTER AND WRITE CONTROL REGISTERS.
3086 ;
3087 ; INPUTS: R2=PREAMBLE ALL 1'S CHARACTER.
3088 ; R3=1ST BYTE OF DATA
3089 ; R4=2ND BYTE OF DATA
3090 ; CALLS: PELOAD SUBROUTINE
3091
3092 016262 010237 002432 PEDATA:: MOV R2,TEMP1 ;SETUP TO LOAD DIABLK PREAMBLE
3093 016266 004737 016420 JSR PC,PELOAD ;DO IT
3094 016272 010337 002432 MOV R3,TEMP1 ;SETUP TO LOAD DIABLK & 1ST DATA BYTE
3095 016276 004737 016420 JSR PC,PELOAD ;DO IT
3096 016302 010437 002432 MOV R4,TEMP1 ;SETUP TO LOAD DIABLK WITH 2ND DATA BYTE
3097 016306 004737 016420 JSR PC,PELOAD ;DO IT
3098
3099 ;LOAD THE EXPECTED RESULTS.
3100
3101 016312 013737 002416 002342 4$: MOV DTKIDN,EXTRAC ;IDENTIFY DEAD TRACKS.
3102 016320 005137 002342 COM EXTRAC ;ALL TRACKS NOT DEAD ARE ACTIVE.
3103 016324 042737 177000 002342 BIC #177000,EXTRAC ;CLR GARBAGE.
3104 016332 013737 002416 002346 MOV DTKIDN,EXTRDD ;TRACKS DEAD DEFINED IN DTKIDN REG.
3105 016340 012737 000777 002336 MOV #777,EXORDY ;OUTPUT READY HI ON ALL TRACKS.
3106 016346 010337 002344 MOV R3,EXDATA ;EXPECTED DATA INFO (NOT SO FOR SKEW TEST)
3107 016352 043737 002416 002344 BIC DTKIDN,EXDATA ;ZERO THE DATA CORRESPONDING TO THE DEAD TRACKS.
3108 016360 013737 002416 002340 MOV DTKIDN,EX1DTR ;EXPECTED ONES OR DEAD TRK INFO.
3109 016366 050337 002340 BIS R3,EX1DTR
3110 016372 005103 COM R3 ;COMPLEMENT DATA.
3111 016374 013737 002416 002350 MOV DTKIDN,EXODTR ;EXPECTED 0 OR DEAD TRK INFO.
3112 016402 050337 002350 BIS R3,EXODTR
3113 016406 005103 COM R3 ;RESTORE R3.
3114 016410 042737 177000 002350 BIC #177000,EXODTR ;CLR GARBAGE.
3115 016416 000207 RTS PC ;RETURN
3116

```

```

3117 ;S/R TO LOAD CONTENTS OF TEMP1 INTO THE DIABLK BY GENERATING THE TWO
3118 ;BIT CELLS OF DATA AND CONTROL. THE ROUTINE ALSO LOADS DATA TO SIMULATE
3119 ;DEAD TRACKS.
3120
3121 ;
3122 ; INPUT: TEMP1=DATA TO LOAD.
3123 ; R5=DIABLK INDEX.
3124 ; DTKIDN=DEAD TRACK IDEN REG. 1=DEAD; 0=LIVE.
3125 ;
3126 ;CLOBBERS TEMPO,R1,TEMP1
3127 ;
3128 ; OUTPUT: UPDATED DIABLK DATA
3129 ; UPDATED DIABLK INDEX
3130 ;
3131 ;NOTE THAT A LOGICAL "1" IS SENT TO THE TSO4 AS A 1 TO 0 TRANSITION.
3132 ;LIKewise, THE TSO4 INTERPRETS A 0 TO 1 TRANSITION AS A LOGICAL 0.
3133 ;THEREFORE, A LOGICAL 1 HAS A 1 BIT CELL FOLLOWED BY A 0 BIT CELL, AND
3134 ;LOGICAL 0 CONSISTS OF A 0 BIT CELL FOLLOWED BY A 1 BIT CELL. THE FIRST
3135 ;PASS OF THIS SUBROUTINE LOADS THE FIRST BIT CELL, WHILE THE SECOND PASS
3136 ;LOADS BIT CELL 2.
3137 ;DEAD TRACK BIT CELLS ARE EITHER BOTH 0 OR BOTH 1. (NO TRANSITION.)
3138
3139 016420 012701 000044 PELOAD:: MOV #IS.NRZ!IS.WRF,R1 ;INIT CONTROL WORD STORAGE REG
3140 016424 113745 002432 1$: MOVB TEMP1,-(R5) ;LOAD DATA WORD (BIT CELL)
3141 016430 000301 SWAB R1 ;LOAD THE
3142 016432 153701 002432 BISB TEMP1,R1 ;DATA WORD IN R1 ALSO TO
3143 016436 000301 SWAB R1 ;GENERATE PARITY.
3144 016440 032737 000400 002432 BIT #BIT8,TEMP1 ;WHAT ABOUT PARITY TRACK. SET?
3145 016446 001402 BEQ 2$ ;BR IF NOT
3146 016450 052701 000020 BIS #IS.DAP,R1 ;IF SO, SET THE BIT IN THE CNTRL WORD.
3147 016454 004737 015632 2$: JSR PC,ODDPAR ;CALCULATE ODD PARITY. RESULT IN TEMPO BIT 1
3148 016460 005737 002430 TST TEMPO ;PARITY BIT SET?
3149 016464 001402 BEQ 3$ ;BR IF NOT
3150 016466 052701 000200 BIS #IS.PAR,R1 ;IF SO SET PARITY BIT FOR BOTH THE
3151 ;DATA AND CONTROL WORD.
3152 016472 110145 3$: MOVB R1,-(R5) ;LOAD THE CNTRL WORD (BIT CELL)
3153 016474 005737 002432 TST TEMP1 ;DONE WITH PASS 2?
3154 016500 100432 BMI 4$ ;BR IF SO
3155 016502 005737 002416 TST DTKIDN ;LOOKING FOR ANY DEAD TRACKS?
3156 016506 001424 BEQ 5$ ;BR IF NOT.
    
```

```

3157                                     ;XOR THE DATA WITH THE DEAD TRACKS SO NO TRANSITIONS OCCUR ON
3158                                     ;THE DEAD TRACK BIT CELLS.
3159                                     ;XOR = (A)(NOT B) + (NOT A)(B) = NOT(A + NOT B) + NOT(NOT A + B)
3160
3161 016510 013737 002432 002430      MOV     TEMP1,TEMPO      ;SAVE ARG A.
3162 016516 013701 002416             MOV     DTKIDN,R1      ;SAVE ARG B.
3163 016522 005137 002430             COM     TEMPO          ;INVERT ARG A.
3164 016526 005101                   COM     R1             ;INVERT ARG B.
3165 016530 053701 002432             BIS     TEMP1,R1      ;GENERATE (A) + (NOT B)
3166 016534 053737 002416 002430     BIS     DTKIDN,TEMPO  ;GENERATE (NOT A) + (B)
3167 016542 005137 002430             COM     TEMPO          ;INVERT
3168 016546 005101                   COM     R1             ;INVERT
3169 016550 053701 002430             BIS     TEMPO,R1      ;OR THE TWO ARGUMENTS TO GIVE THE XOR.
3170 016554 010137 002432             MOV     R1,TEMP1      ;TEMP1 = TEMP1 XOR DTKIDN
3171 016560 005137 002432             5$:   COM     TEMP1    ;SET UP TO LOAD BIT CELL 2.
3172 016564 000715                   BR      PELOAD        ;DO IT
3173 016566 000207             4$:   RTS     PC      ;RETURN
3174
    
```

```

3175 ;S/R TO RETRIEVE THE TS04 FORMATTER AND WRITE CONTROL REGISTERS AND
3176 ;CHECK TO SEE IF THEY CONTAIN EXPECTED DATA.
3177 ;
3178 ; INPUTS: EXPECTED DATA PREVIOUSLY SET UP IN PEDATA SUBROUTINE
3179 ; MSGPKT - TS04 MESSAGE PACKET.
3180 ;
3181 ; OUTPUT: ERROR FLAG, ACTUAL DATA
3182 ;
3183 ;CLOBBERS R5,R1
3184
3185 PEERCK:: INCR R1 FROM #0 TO #12 BY #2
3186 016570 005001
3187 016572 000402
3188 016574
3189 016574 062701 000002
3190 016600
3191 016600 020127 000012
3192 016604 003003
3193 016606
3194 016606 005061 002354
3195 016612
3196 016612 000770
3197 016614
3198
3199
3200 016614 113737 002264 002354 MOVB MSGPKT.C18#OR,ACORDY ;ACTUAL OUTPUT READY INFO
3201 016622 113737 002265 002356 MOVB MSGPKT.C18#ID,AC1DTR ;ACTUAL 1'S OR DEAD TRACK INFO.
3202 016630 113737 002267 002360 MOVB MSGPKT.C18#TA,ACTRAC ;ACTUAL TRACK ACTIVE INFO
3203 016636 113737 002270 002362 MOVB MSGPKT.C18#DA,ACDATA ;ACTUAL DATA INFO
3204 016644 113737 002271 002364 MOVB MSGPKT.C18#TD,ACTRDD ;ACTUAL TRACK DEAD INFO
3205 016652 113737 002272 002366 MOVB MSGPKT.C18#OD,ACODTR ;ACTUAL 0 OR DEAD TRACK INFO.
3206 016660 LET TEMP2 :B= MSGPKT.PRCMST ;GET THE TRK ACT DATA FOR CHAN 9
3207 016660 113737 002274 002434 MOVB MSGPKT.P
3208 016666 012737 000006 002432
3209 016674 012701 002354
3210
3211 016700 006237 002434 1$: ASR TEMP2 ;INIT COUNTER
3212 016704 103003 BCC 2$ ;SET INDEX OF THE FORMATTER AND
3213 016706 052721 000400 BIS #BIT8,(R1). ;WRITE CONTROL REG., TABLE.
3214 ;SHIFT CHAN 9 DATA INTO CARRY BIT
3215 016712 000402 BR 3$ ;BR IF DATA WAS CLR.
3216 016714 042721 000400 2$: BIC #BIT8,(R1). ;IF DATA WAS SET THEN SET THE CHAN 9
3217 ;BIT IN THE APPROPRIATE "ACTUAL" REG.
3218 016720 005337 002432 3$: DEC TEMP1 ;IF CLR, THEN CLR THE CHAN 9 BIT IN
3219 016724 001365 BNE 1$ ;THE APPROPRIATE "ACTUAL" REG.
3220 ;DONE CHECKING ALL THE NEEDED BITS?
3221 ;BR IF NOT
3222 ;NOW COMPARE THE ACTUAL & EXPECTED DATA AND SET THE ERROR FLAG
3223 ;IF WE HAD AN INVALID COMPARISON.
3224 016726 012701 002336 MOV #EXPTBL,R1 ;SAVE "ACTUAL" TABLE INDEX
3225 016732 012737 000006 002432 MOV #6,TEMP1 ;INIT COUNTER
3226 016740 4$: LET R5 := (R1) XOR 16(R1)
3227 016740 011105 MOV (R1),R5
3228 016742 016146 000016 MOV 16(R1),
3229 016746 040516 BIC R5,(SP)
3230 016750 046105 000016 BIC 16(R1),R

```


3231	016754	052605							
3232	016756			IF R5 NE #0 THEN			BIS	(SP),R5	
3233	016756	005705					TST	R5	
3234	016760	001402					BEQ	50040\$	
3235	016762	005237	002436	INC ERRFLG					
3236	016766			ENDIF					
3237	016766					50040\$:			
3238	016766			LET 34(R1) := 34(R1) SET BY R5					
3239	016766	050561	000034				BIS	R5,34(R1)	
3240	016772			LET R1 := R1 * #2					
3241	016772	062701	000002				ADD	#2,R1	
3242									
3243	016776	005337	002432	DEC	TEMP1				
3244	017002	001356		BNE	#4				
3245	017004	000207		RTS	PC				

:DONE CHECKNING ALL DATA?
:BR IF NOT

```

3246 ;S/R TO SETUP THE DIABLK IN SUCH A WAY THAT WILL ENABLE THE ROM
3247 ;LOOKUP TABLE LOCATION, SPECIFIED IN ROMLKI, TO BE EXAMINED.
3248
3249 ;10.0 ADDRESS BITS ARE NEEDED TO ADDRESS THE ROM FROM 0-1777. THE 9 LOW
3250 ;ORDER ROM ADDRESS BITS ARE GATED FROM THE TRACK ACTIVE DATA WHILE THE
3251 ;MSB OF THE ROM ADDRESS IS GATED FROM THE VERTICAL PARITY ERROR (VPE).
3252 ;
3253 ;THIS S/R LOADS THE DIABLK WITH AN ALL 1'S PREAMBLE; CAUSES A VPE TO SET
3254 ;OR CLEAR IN BYTE 1 DEPENDING ON THE STATE OF ROMLKI (BIT 9); AND CAUSES
3255 ;DEAD TRACKS IN BYTE 2 CORRESPONDING TO BITS 0-8 OF ROMLKI.
3256 ;
3257
3258 017006 005037 002416 ROMLOK:: CLR DTKIDN ;CLR DEAD TRACK REG.
3259 017012 012737 000777 002432 MOV #777,TEMP1 ;SET UP TO LOAD ALL 1'S PREAMBLE
3260 017020 004737 016420 JSR PC,PELOAD ;LOAD PREAMBLE & CONTROL.
3261 017024 032737 001000 002420 BIT #BIT9,ROMLKI ;MSB OF ROM ADDR SET?
3262 017032 001403 BEQ 1$ ;BR IF NOT
3263 017034 005037 002432 CLR TEMP1 ;CAUSE VPE=1.
3264 017040 000403 BR 2$ ;CONTINUE
3265
3266 017042 012737 000777 002432 1$: MOV #777,TEMP1 ;CAUSE VPE=0.
3267 017050 004737 016420 2$: JSR PC,PELOAD ;LOAD BYTE 1 * CNTRL.
3268 017054 005037 002432 CLR TEMP1 ;INIT TEMP1.
3269 017060 013737 002420 002416 MOV ROMLKI,DTKIDN ;LOAD ROM ADDR IN DEAD TRACK REG.
3270 017066 005137 002416 COM DTKIDN ;INVERT IT.
3271 017072 042737 177000 002416 BIC #177000,DTKIDN ;CLR GARBAGE. MSB ADDRESS
3272 ;WAS TAKEN CARE OF IN BYTE 1.
3273 017100 004737 016420 JSR PC,PELOAD ;LOAD BYTE 2 DATA AND CONTROL.
3274 017104 000207 RTS PC ;RETURN.

```

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

```

;S/R TO GENERATE THE EXPECTED ROM LOOKUP TABLE OUTPUT FROM THE ADDRESS
;IN ROMLKI.
;
;      INPUTS:      ROMLKI = ROM ADDRESS
;      OUTPUTS:     EXROML = EXPECTED ROM LOOKUP TABLE OUTPUT.
;      CLOBBERS:    TEMPO, R1, R2
;
;THE ROM CONTENTS ARE SHOWN IN THE FOLLOWING TABLE:
;
;OUTPUT      ADDRESSES WHICH ASSERT*      COMMENTS
;-----
;-----
;MULT      (MSB)      X----- X = DON'T CARE. MULT = 1 IF
;                                     ANY 2 ADDRESS LINES (OTHER
;                                     THAN A9) ARE 1.
;RDTMK      XXX0100X11      X = DON'T CARE.
;           X110X001XX
;PREAM      XAACBCCABB      X = DON'T CARE. MUST HAVE AT
;                                     LEAST ONE OF A & B & C. MUST
;                                     HAVE ALL OF A OR B OR C.
;9 OF 9      X111111111      X = DON'T CARE.
;0 OF 9      X000000000      X = DON'T CARE.
;COR DATA      1000000001
;           1000000010
;           1000000100
;           1000001000
;           1000010000
;           1000100000
;           1001000000
;           1010000000
;INCOR DATA      1000000000
;8 OF 9      (LSB)      X111111111      X = DON'T CARE.
;           X111111110
;           X111111101
;           X111111011
;           X111110111
;           X111101111
;           X111011111
;           X110111111
;           X101111111
;           X011111111
;
; * = A0-A7 = MUX 2(0)-2(7)
;      A8 = MUX PARITY
;      A9 = LATCHED PARITY ERROR

```

```

3330 017106 005037 002352      ROMEX:: CLR      EXROML      ;INIT. EXPECTED ROM REG.
3331
3332                          ;CHECK FOR 8 OF 9.
3333
3334 017112 012702 000001      X8OF9: MOV      #1,R2      ;INIT 0'S CNTR.
3335 017116 012737 000011 002430  MOV      #11,TEMPO      ;LOAD BIT CNTR. NOT INTERESTED IN BIT10.
3336 017124 013701 002420      MOV      ROMLKI,R1      ;SAVE ROM TABLE ADDRESS
3337 017130 000241              1$: CLC              ;CLR CARRY TO PREPARE FOR SHIFT
3338 017132 006201              ASR      R1              ;SHIFT IT. WAS IS A 1?
3339 017134 103401              BCS     2$              ;BR IF SO
3340 017136 005302              DEC      R2              ;DECR 0 CNTR.
3341 017140 005337 002430      2$: DEC      TEMPO      ;DECR BIT CNTR DONE?
3342 017144 001371              BNE     1$              ;BR IF NOT
3343 017146 005702              TST     R2              ;8 OF 9?
3344 017150 100403              BMI     XCORDA          ;BR IF NOT
3345 017152 052737 000001 002352  BIS     #.80F9,EXROML  ;IF SO, SET THE BIT
3346
3347                          ;CHECK FOR CORRECTABLE DATA
3348
3349 017160 032737 001000 002420  XCORDA: BIT     #A9,ROMLKI ;COULD THIS BE CORRECTABLE DATA
3350 017166 001427              BEQ     XINCOR          ;BR IF NOT
3351 017170 032737 000400 002420  BIT     #A8,ROMLKI      ;COULD THIS BE CORRECTABLE DATA?
3352 017176 001023              BNE     XINCOR          ;BR IF NOT.
3353 017200 012702 000002              MOV     #2,R2          ;INIT 1'S CNTR.
3354 017204 012737 000012 002430  MOV     #12,TEMPO      ;LOAD BIT CNTR. NOT INTERESTED IN MSB.
3355 017212 013701 002420      MOV     ROMLKI,R1      ;SAVE ROM TABLE ADDR.
3356 017216 000241              1$: CLC              ;INIT CARRY TO PREPARE FOR SHIFT.
3357 017220 006201              ASR     R1              ;SHIFT IT. WAS IT A 1?
3358 017222 103001              BCC     2$              ;BR IF NOT
3359 017224 005302              DEC     R2              ;DECR. 1'S CNTR.
3360 017226 005337 002430      2$: DEC     TEMPO      ;DECR BIT CNTR. DONE?
3361 017232 001371              BNE     1$              ;BR IF NOT
3362 017234 005702              TST     R2              ;CORRECTABLE DATA?
3363 017236 001003              BNE     XINCOR          ;BR IF NOT
3364 017240 052737 000004 002352  BIS     #.CORD,EXROML  ;IF SO, SET THE BIT.
3365
3366                          ;CHECK FOR INCORRECTABLE DATA
3367
3368 017246 022737 001000 002420  XINCOR: CMP     #1000,ROMLKI ;INCORRECTABLE DATA?
3369 017254 001003              BNE     X00F9          ;BR IF NOT
3370 017256 052737 000002 002352  BIS     #.INCOR,EXROML ;IF SO, SET THE "INCOR" BIT.
3371
3372                          ;CHECK FOR 0 OF 9
3373
3374 017264 032737 000777 002420  X00F9: BIT     #777,ROMLKI ;0 OF 9?
3375 017272 001003              BNE     X90F9          ;BR IF NOT
3376 017274 052737 000010 002352  BIS     #.00F9,EXROML  ;IF SO, SET THE BIT.
3377
3378                          ;CHECK FOR 9 OF 9
3379
3380 017302 022737 000777 002420  X90F9: CMP     #777,ROMLKI ;9 OF 9?
3381 017310 001404              BEQ     1$              ;BR IF SO
3382 017312 022737 001777 002420  CMP     #1777,ROMLKI   ;9 OF 9?
3383 017320 001003              BNE     XMULT          ;BR IF NOT
3384 017322 052737 000020 002352  1$: BIS     #.90F9,EXROML ;SET THE BIT.
3385

```

```

3386                                     ;CHECK FOR MULTIPLE TRACKS
3387
3388 017330 012702 000001                XMULT: MOV    #1,R2          ;INIT 1'S CNTR.
3389 017334 012737 000011 002430        MOV    #11,TEMPO      ;LOAD BIT CNTR.
3390 017342 013701 002420                MOV    ROMLKI,R1     ;SAVE ROM TABLE ADDR
3391 017346 000241                1$:   CLC              ;INIT CARRY TO PREPARE FOR SHIFT.
3392 017350 006201                ASR    R1             ;SHIFT IT WAS IT A 1?
3393 017352 103001                BCC   2$             ;BR IF NOT
3394 017354 005302                DEC   R2             ;DECR 1'S CNTR.
3395 017356 005337 002430                2$:   DEC   TEMPO     ;DECR BIT CNTR. DONE?
3396 017362 001371                BNE   1$             ;BR IF NOT
3397 017364 005702                TST   R2             ;MULTIPLE TRACKS?
3398 017366 100003                BPL   XPREAM        ;BR IF NOT
3399 017370 052737 000200 002352        BIS   #.MULT,EXROML ;IF SO, SET BIT BIT
3400
3401                                     ;CHECK FOR PREAMBLE
3402                                     ;ADDR LINES 9876543210
3403                                     ; XAACBCCABB
3404                                     ;PREAM=1 IF HAVE ONE OF AEBEC AND
3405                                     ; ALL OF A OR B OR C.                X = 0 OR 1.
3406
3407 017376 005002                XPREAM: CLR   R2
3408 017400 013701 002420                MOV    ROMLKI,R1     ;SAVE ROM TABLE ADDRESS.
3409 017404 010137 002430                MOV    R1,TEMPO      ;GENERATE THE
3410 017410 005137 002430                COM    TEMPO         ;COMPLIMENT OF THE ADDR.
3411 017414 032737 000604 002430        BIT    #A8!A7!A2,TEMPO ;ALL OF GROUP A?
3412 017422 001003                BNE   1$             ;BR IF NOT
3413 017424 052702 000011                BIS   #BIT0!BIT3,R2 ;SET "ALL OF A" AND "1 OF A" INDICATION
3414 017430 000405                BR    2$
3415
3416 017432 032701 000604                1$:   BIT    #A8!A7!A2,R1 ;1 OF GROUP A?
3417 017436 001440                BEQ   7$             ;BR IF NOT - NOT A PREAM
3418 017440 052702 000001                BIS   #BIT0,R2       ;SET "1 OF A" INDICATION.
3419 017444 032737 000043 002430        2$:   BIT    #A5!A1!A0,TEMPO ;ALL OF GROUP B ACTIVE?
3420 017452 001003                BNE   3$             ;BR IF NOT
3421 017454 052702 000012                BIS   #BIT1!BIT3,R2 ;SET "ALL OF B" + "1 OF B".
3422 017460 000405                BR    4$             ;CONTINUE
3423 017462 032701 000043                3$:   BIT    #A5!A1!A0,R1 ;"1 OF B" ACTIVE.
3424 017466 001424                BEQ   7$             ;BR IF NOT - NOT A PREAM.
3425 017470 052702 000002                BIS   #BIT1,R2       ;SET "1 OF B" INDICATION.
3426 017474 032737 000130 002430        4$:   BIT    #A6!A4!A3,TEMPO ;ALL OF GROUP C ACTIVE?
3427 017502 001003                BNE   5$             ;BR IF NOT
3428 017504 052702 000014                BIS   #BIT2!BIT3,R2 ;SET ALL OF C AND 1 OF C
3429 017510 000405                BR    6$             ;INDICATION.
3430 017512 032701 000130                5$:   BIT    #A6!A4!A3,R1 ;1 OF C?
3431 017516 001410                BEQ   7$             ;BR IF NOT-NO PREAMBLE
3432 017520 052702 000004                BIS   #BIT2,R2       ;SET 1 OF C INDICATION.
3433
3434                                     ;IN ORDER FOR THE PREAMBLE CONDITION TO BE TRUE, R2 MUST CONTAIN
3435                                     ;17 AT THIS POINT.
3436
3437 017524 022702 000017                6$:   CMP    #17,R2     ;PREAMBLE?
3438 017530 001003                BNE   7$             ;BR IF NOT
3439 017532 052737 000040 002352        BIS   #.PREAM,EXROML ;IF SO, SET THE PREAMBLE BIT IN
3440                                     ;EXPECTED LOOKUP TABLE REG.
3441 017540 000240                7$:   NOP

```

```

3442
3443
3444
3445
3446
3447
3448
3449
3450 017542 013701 002420
3451 017546 010137 002430
3452 017552 005137 002430
3453 017556 032701 000130
3454 017562 001013
3455 017564 032737 000043 002430
3456 017572 001404
3457 017574 032737 000604 002430
3458 017602 001003
3459 017604 052737 000100 002352
3460
3461 017612 000207

;CHECK FOR READ FILE MARK PATTERN
;ADDR LINES 9876543210
;RDFMK-1 IF XXX0100X11 - CONDITION 1
; OR X110X001XX - CONDITION 2
; IS TRUE.
; X = 0 OR 1

XRDFMK: MOV ROMLKI,R1 ;SAVE ROM TABLE ADDR.
MOV R1,TEMPO ;GENERATE THE
COM TEMPO ;COMPLIMENT OF THE ADDR.
BIT #A6!A4!A3,R1 ;CHECK THE 0'S IN COND 1 AND COND 2.
BNE 3$ ;BR IF COND 1 AND COND 2 NOT TRUE
BIT #A5!A1!A0,TEMPO ;CHECK 1'S IN COND 1.
BEQ 2$ ;BR IF COND 1 IS TRUE
BIT #A8!A7!A2,TEMPO ;CHECK 1'S IN CONDITION 2
BNE 3$ ;BR IF COND 2 IS FALSE
2$: BIS #.RDFMK,EXROML ;WE ARE READING A FILE MARK.
;SET APPROPRIATE BIT.
3$: RTS PC ;RETURN

```

```

3462 ;S/R TO RETRIEVE THE ACTUAL ROM LOOKUP TABLE OUTPUT AND COMPARE IT
3463 ;WITH THE EXPECTED OUTPUT.
3464 ;
3465 ; INPUT: EXROML - EXPECTED ROM OUTPUT.
3466 ; MSGPKT - TS04 MESSAGE PACKET.
3467 ;
3468 ; OUTPUT: ERRFLG - ERROR FLAG
3469 ; ACROML - ACTUAL ROM OUTPUT.
3470 ;
3471 ;
3472 ;
3473 017614 ROMCK:: LET ACROML :B= MSGPKT*ROM$LK ;GET THE ACTUAL ROM OUTPUT
3474 J17614 113737 002273 002370 ;MOV B,MSGPKT
3475 017622 123737 002370 002352 ;MOV B,ROM$LK
3476 017630 001402 ;MOV B,ACROML
3477 017632 005237 002436 ;MOV B,EXROML
3478 017636 000207 ;MOV B,ERRFLG

CMPB ACROML,EXROML ;ARE THEY THE SAME.
BEQ 1$ ;BR IF SO
INC ERRFLG ;IF NOT SET THE ERROR FLAG
1$: RTS PC

```

3479
3480
3481
3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534

017640 005037 002432
017644
017644 010102
017646 010246
017650 042716 000007
017654 042602
017656
017656 010103
017660 010346
017662 042716 000070
017666 042603
017670
017670 010104
017672 010446
017674 042716 000700
017700 042604
017702
017702 005005
017704 004737 020510
017710
017710 005705
017712 001026
017714
017714 023727 002440 177777
017722 001011
017724
017724 012746 005775
017730 012746 000001
017734 010600
017736 104414
017740 062706 000004

: SUBROUTINES MSORT 1 AND MSORT2 SELECT BAD BOARDS FOR TESTS 4 - 7 PER
: TABLE BELOW WHERE:

22 = M8922 57 = G157
24 = M8924 23 = M8923

BAD BOARD SELECTION TABLE			
R5	BD/SLOT/BITS	MASK	PRINTB MSG
0	22/7/0-8		FMTCTR
1	24/6/0,1,2	R2=7	FMTCH6
2	24/5/3,4,5	R3=70	FMTCH5
3	24/4/6,7,8	R4=700	FMTCH4
400	57/2/0,1,5,6	R2=143	RDCH2
1000	57/1/2,3,4,7	R3=234	RDCH1
1400	23/3/8	R4=400	RDCHP3

: TEST 4 -- MODULE SORT

```

MSORT1:: CLR      TEMP1          ;TEMP1=BOARDS 24 SELECTION FLAG
          LET R2 := R1 AND #7    ;R2=BOARD 24/6 MASK
                                          MOV      R1,R2
                                          MOV      R2,-(SP)
                                          BIC      #7,(SP)
                                          BIC      (SP),R2
          LET R3 := R1 AND #70    ;R3=BOARD 24/5 MASK
                                          MOV      R1,R3
                                          MOV      R3,-(SP)
                                          BIC      #70,(SP)
                                          BIC      (SP),R3
          LET R4 := R1 AND #700   ;R4=BOARD 24/4 MASK
                                          MOV      R1,R4
                                          MOV      R4,-(SP)
                                          BIC      #700,(SP)
                                          BIC      (SP),R4
          LET R5 := #0           ;R5=BOARD 22/7 FLAG
                                          CLR      R5
          JSR      PC,PAIRST      ;SORT 22 AND 24 BOARDS, RESULT IN R5 L0
          IF R5 EQ #0 THEN
          IF T4S4 EQ #-1 THEN    ;BTL
                                          TST      R5
                                          BNE      50041$
          PRINTB #T4S4MG        ;BTL CALL BAD G157
                                          CMP      T4S4,-1
                                          BNE      50042$
          MOV      #T4S4MG,
          MOV      #1,-(SP)
          MOV      SP,R0
          TRAP   C$PNTB
          ADD      #4,SP

```


3535	017744			ELSE	;BTL		
3536	017744	000410				BR	50043\$
3537	017746					50042\$:	
3538	017746			PRINTB #FMTCTR	;22 BOARD SELECTED, CALL BAD 22/7	MOV	#FMTCTR,
3539	017746	012746	005064			MOV	#1,-(SP)
3540	017752	012746	000001			MOV	SP,R0
3541	017756	010600				TRAP	C\$PNTB
3542	017760	104414				ADD	#4,SP
3543	017762	062706	000004				
3544	017766			ENDIF	;BTL		
3545	017766					50043\$:	
3546	017766			ELSE	;24 BOARD SELECTED, CONTINUE SORTING	BR	50044\$
3547	017766	000561				50041\$:	
3548	017770				;R2=BOARD 57/2 MASK	MOV	R1,R2
3549	017770			LET R2 := R1 AND #143		MOV	R2,-(SP)
3550	017770	010102				BIC	#143,(SP)
3551	017772	010246				BIC	(SP)+,R2
3552	017774	042716	000143				
3553	020000	042602					
3554	020002			LET R3 := R1 AND #234	;R3=BOARD 57/1 MASK	MOV	R1,R3
3555	020002	010103				MOV	R3,-(SP)
3556	020004	010346				BIC	#234,(SP)
3557	020006	042716	000234			BIC	(SP)+,R3
3558	020012	042603					
3559	020014			LET R4 := R1 AND #400	;R4=BOARD 23/3 MASK	MOV	R1,R4
3560	020014	010104				MOV	R4,-(SP)
3561	020016	010446				BIC	#400,(SP)
3562	020020	042716	000400			BIC	(SP)+,R4
3563	020024	042604					
3564	020026			LET R5 := SWAP R5		SWAB	R5
3565	020026	000305				RESULT	IN R5 HI
3566	020030	004737	020510	JSR PC,PAIRST	;SORT 57 AND 23 BOARDS,		
3567	020034			LET R5 := SWAP R5	;ANALYZE R5 LO BYTE	SWAB	R5
3568	020034	000305					
3569	020036			IF R5 EQ #1 THEN		CMP	R5,#1
3570	020036	020527	000001			BNE	50045\$
3571	020042	001011					
3572	020044			PRINTB #FMTCH6	;CALL BAD 24/6	MOV	#FMTCH6,
3573	020044	012746	005176			MOV	#1,-(SP)
3574	020050	012746	000001			MOV	SP,R0
3575	020054	010600				TRAP	C\$PNTB
3576	020056	104414				ADD	#4,SP
3577	020060	062706	000004				
3578	020064			ELSE		BR	50046\$
3579	020064	000432				50045\$:	
3580	020066						
3581	020066			IF R5 EQ #2 THEN		CMP	R5,#2
3582	020066	020527	000002			BNE	50047\$
3583	020072	001011					
3584	020074			PRINTB #FMTCH5	;CALL BAD 24/5	MOV	#FMTCH5,
3585	020074	012746	005253			MOV	#1,-(SP)
3586	020100	012746	000001			MOV	SP,R0
3587	020104	010600				TRAP	C\$PNTB
3588	020106	104414				ADD	#4,SP
3589	020110	062706	000004				
3590	020114			ELSE			

```

3591 020114 000416
3592 020116
3593 020116
3594 020116 020527 000003
3595 020122 001011
3596 020124
3597 020124 012746 005330
3598 020130 012746 000001
3599 020134 010600
3600 020136 104414
3601 020140 062706 000004
3602 020144
3603 020144 000402
3604 020146
3605 020146 005237 002432
3606 020152
3607 020152
3608 020152
3609 020152
3610 020152
3611 020152
3612 020152
3613 020152 005737 002432
3614 020156 001465
3615 020160
3616 020160 013701 002406
3617 020164 053701 002412
3618 020170
3619 020170 005701
3620 020172 001430
3621 020174
3622 020174 010501
3623 020176
3624 020176 010546
3625 020200 042716 000003
3626 020204 042605
3627 020206 004737 020556
3628 020212
3629 020212 012746 005064
3630 020216 012746 000001
3631 020222 010600
3632 020224 104414
3633 020226 062706 000004
3634 020232
3635 020232 010105
3636 020234 000305
3637 020236
3638 020236 010546
3639 020240 042716 000003
3640 020244 042605
3641 020246 004737 020700
3642 020252
3643 020252 000427
3644 020254
3645 020254
3646 020254 010501

IF R5 EQ #3 THEN
    PRINTB #FMTCH4 ;CALL BAD 24/4
ELSE
    INC TEMP1 ;TEPM1 NOT=0 WHEN NO BAD 24 WERE CALLED
    ENDIF
ENDIF
ENDIF
IF TEMP1 NE #0 THEN ;IF NO BAD 24 BOARDS CALLED, CONTINUE SO
    LET R1 := ACTRK1 OR ACTRK3 ;ANALYZE TRACK INACTIVE SUBTSTS RESULTS
    IF R1 NE #0 THEN ;R1 NOT=0, TRACK INACTIVE FAILED
        LET R1 := R5 ;SAVE R5
        LET R5 := R5 AND #3 ;KEEP 2 LSB'S
        JSR PC,FMTSEL ;CALL APPROPRIATE BAD 24
        PRINTB #FMTCTR ;CALL BAD 22
        LET R5 := SWAP R1 ;FETCH R5 HI BYTE
        LET R5 := R5 AND #3 ;KEEP 2 LSB'S
        JSR PC,RDCSEL ;CALL APPROPRIATE BAD 57/23
    ELSE ;R1=0, TRACK INACTIVE DID NOT FAIL
        LET R1 := R5 ;SAVE R5
    ENDIF
ENDIF

```

```

50047$: BR 50050$
CMP R5,#3
BNE 50051$
MOV #FMTCH4,
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
BR 50052$
50051$:
50052$:
50050$:
50046$:
TST TEMP1
BEQ 50053$
MOV ACTRK1,R
BIS ACTRK3,R
TST R1
BEQ 50054$
MOV R5,R1
MOV R5,-(SP)
BIC #3,(SP)
BIC (SP)+,R5
MOV #FMTCTR,
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
MOV R1,R5
SWAB R5
MOV R5,-(SP)
BIC #3,(SP)
BIC (SP)+,R5
BR 50055$
50054$:
MOV R5,R1

```

3647	020256			LET R5 := SWAP R5			
3648	020256	000305				SWAB	R5
3649	020260			LET R5 := R5 AND #3	;KEEP 2 LSB'S OF HI BYTE		
3650	020260	010546				MOV	R5, -(SP)
3651	020262	042716	000003			BIC	#3, (SP)
3652	020266	042605				BIC	(SP)+, R5
3653	020270	004737	020700	JSR PC, RDCSEL	;CALL APPROPRIATE BAD 57/23		
3654	020274			LET R5 := R1 AND #3	;FETCH 2 LSB'S OF R5 LO BYTE		
3655	020274	010105				MOV	R1, R5
3656	020276	010546				MOV	R5, -(SP)
3657	020300	042716	000003			BIC	#3, (SP)
3658	020304	042605				BIC	(SP)+, R5
3659	020306	004737	020556	JSR PC, FMTSEL	;CALL APPROPRIATE BAD 24		
3660	020312			PRINTB #FMTCTR	;CALL BAD 22		
3661	020312	012746	005064			MOV	#FMTCTR,
3662	020316	012746	000001			MOV	#1, -(SP)
3663	020322	010600				MOV	SP, R0
3664	020324	104414				TRAP	C#PNTB
3665	020326	062706	000004			ADD	#4, SP
3666	020332			ENDIF			
3667	020332						
3668	020332			ENDIF		50055\$:	
3669	020332						
3670	020332			ENDIF		50053\$:	
3671	020332						
3672	020332	000207		RTS PC		50044\$:	

```

3673
3674
3675
3676 020334
3677 020334 005701
3678 020336 001442
3679 020340
3680 020340 010102
3681 020342 010246
3682 020344 042716 000007
3683 020350 042602
3684 020352
3685 020352 010103
3686 020354 010346
3687 020356 042716 000070
3688 020362 042603
3689 020364
3690 020364 010105
3691 020366 010546
3692 020370 042716 000700
3693 020374 042605
3694 020376
3695 020376 005005
3696 020400 004737 020510
3697 020404
3698 020404 005705
3699 020406 001011
3700 020410
3701 020410 012746 005064
3702 020414 012746 000001
3703 020420 010600
3704 020422 104414
3705 020424 062706 000004
3706 020430
3707 020430 000404
3708 020432
3709 020432 004737 020556
3710 020436 004737 020452
3711 020442
3712 020442
3713 020442
3714 020442 000402
3715 020444
3716 020444 004737 020452
3717 020450
3718 020450
3719 020450 000207
3720
3721
3722
3723 020452
3724 020452 013701 002374
3725 020456 053701 002404
3726 020462
3727 020462 005701
3728 020464 001410

```

```

: TEST 5 - 7 -- MODULE SORT
MSORT2:: IF R1 NE #0 THEN
:ERROR FROM 24 BOARDS, WHICH ONE?
TST R1
BEQ 50056$
LET R2 := R1 AND #7
:R2=24/6 MASK
MOV R1,R2
MOV R2,-(SP)
BIC #7,(SP)
BIC (SP),R2
LET R3 := R1 AND #70
:R3=24/5 MASK
MOV R1,R3
MOV R3,-(SP)
BIC #70,(SP)
BIC (SP),R3
LET R5 := R1 AND #700
:R4=24/4 MASK
MOV R1,R5
MOV R5,-(SP)
BIC #700,(SP)
BIC (SP),R5
LET R5 := #0
:R5=22/7 FLAG
CLR R5
JSR PC,PAIRST
IF R5 EQ #0 THEN
:SORT 22 AND 24 BOARDS
:
TST R5
BNE 50057$
PRINTB #FMTCTR ;CALL 22/7
MOV #FMTCTR,
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
ELSE
:24 SELECTED, WHICH ONE?
BR 50060$
50057$:
:CALL APPROPRIATE 24
:CHECK OR FUNCTION OF 22
50060$:
BR 50061$
50056$:
:NO ERROR FROM 24, CHECK OR FUNCTION OF
50061$:
RTS PC
:CHECK DEAD TRACK ORING ON 22 BOARD
DTRCHK: LET R1 := OR1DTR OR OR0DTR
:GET OR DATA
MOV OR1DTR,R
BIS OR0DTR,R
IF R1 NE #0 THEN
TST R1
BEQ 50062$

```

3729	020466		
3730	020466	012746	005064
3731	020472	012746	000001
3732	020476	010600	
3733	020500	104414	
3734	020502	062706	000004
3735	020506		
3736	020506		
3737	020506	000207	

PRINTB @FMTCTR

;OR DATA BAD, CALL 22 BOARD

MOV	@FMTCTR,
MOV	#1, -(SP)
MOV	SP, R0
TRAP	C\$PNTB
ADD	#4, SP

ENDIF

RTS PC

50062\$:

```

3738 ;PAIR OF BOARDS SORTING SUBR
3739
3740 ;(R2)(R3) * (R2)(R4) * (R3)(R4) NEQ 0 ----> R5=0
3741 ;
3742 ; R2 NEQ 0 ----> R5=1
3743 ; R3 NEQ 0 ----> R5=2
3744 ; R4 NEQ 0 ----> R5=3
3745
3746 020510 PAIRST: IF R2 NE #0 THEN
3747 020510 005702 TST R2
3748 020512 001407 BEQ 50063$
3749 020514 IF R3 EQ #0 AND R4 EQ #0 THEN
3750 020514 005703 TST R3
3751 020516 001004 BNE 50064$
3752 020520 005704 TST R4
3753 020522 001002 BNE 50064$
3754 020524 LET R5 := R5 SET.BY #1
3755 020524 052705 000001 BIS #1,R5
3756 020530 ENDIF
3757 020530 ELSE 50064$:
3758 020530 BR 50065$
3759 020530 000411 IF R3 NE #0 THEN 50063$:
3760 020532 TST R3
3761 020532 005703 BEQ 50066$
3762 020532 001405 IF R4 EQ #0 THEN
3763 020534 005704 TST R4
3764 020536 001002 BNE 50067$
3765 020536 005704 LET R5 := R5 SET.BY #2
3766 020540 001002 ENDIF
3767 020542 052705 000002 BIS #2,R5
3768 020542 020546 ENDIF
3769 020546 ELSE 50067$:
3770 020546 BR 50070$
3771 020546 000402 LET R5 := R5 SET.BY #3 50066$:
3772 020546 000402 LET R5 := R5 SET.BY #3
3773 020550 000402 ENDIF
3774 020550 052705 000003 BIS #3,R5
3775 020550 020554 ENDIF
3776 020554 000207 ENDIF
3777 020554 HTS PC 50070$:
3778 020554 50065$:
3779 020554
3780 020554
3781
3782 ;FORMATTERS 24 SELECT
3783
3784 020556 FMTSEL: SELECT R5 OF 3 VERIFY
3785 020556 010546 MOV R5, -(SP)
3786 020560 003403 BLE 50071$
3787 020562 021627 000003 CMP (SP), #3
3788 020566 003402 BLE 50072$
3789 020570 50071$:
3790 020570 012716 000004 MOV #4, (SP)
3791 020574 50072$:
3792 020574 006316 ASL (SP)
3793 020576 060716 ADD PC, (SP)
    
```

```

3794 020600 063607
3795 020602
3796 020602 000010
3797 020604 000032
3798 020606 000054
3799 020610 000074
3800 020612
3801 020612
3802 020612
3803 020612 012746 005176
3804 020616 012746 000001
3805 020622 010600
3806 020624 104414
3807 020626 062706 000004
3808 020632
3809 020632 000421
3810 020634
3811 020634
3812 020634 012746 005253
3813 020640 012746 000001
3814 020644 010600
3815 020646 104414
3816 020650 062706 000004
3817 020654
3818 020654 000410
3819 020656
3820 020656
3821 020656 012746 005330
3822 020662 012746 000001
3823 020666 010600
3824 020670 104414
3825 020672 062706 000004
3826 020676
3827 020676
3828 020676 000207
3829
3830
3831
3832 020700
3833 020700 010546
3834 020702 003403
3835 020704 021627 000003
3836 020710 003402
3837 020712
3838 020712 012716 000004
3839 020716
3840 020716 006316
3841 020720 060716
3842 020722 063607
3843 020724
3844 020724 000010
3845 020726 000032
3846 020730 000054
3847 020732 000074
3848 020734
3849 020734

```

```

CASE 1
PRINTB @FMTCH6

CASE 2
PRINTB @FMTCH5

CASE 3
PRINTB @FMTCH4

ENDSELECT
RTS PC

;READ CHANNEL 57/23 SELECT
RDCSEL: SELECT R5 OF 3 VERIFY

CASE 1

```

```

50073$: ADD @ (SP),P
.WORD 50077$-5
.WORD 50076$-5
.WORD 50075$-5
.WORD 50074$-5

50077$:
MOV @FMTCH6,
MOV @1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD @4,SP

50076$: BR 50074$

MOV @FMTCH5,
MOV @1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD @4,SP

50075$: BR 50074$

MOV @FMTCH4,
MOV @1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD @4,SP

50074$:
MOV R5,-(SP)
BLE 50100$
CMP (SP),@3
BLE 50101$

50100$: MOV @4,(SP)
50101$: ASL (SP)
ADD PC,(SP)
ADD @ (SP),P

50102$: .WORD 50106$-5
.WORD 50105$-5
.WORD 50104$-5
.WORD 50103$-5

50106$:

```

3850	020734			PRINTB	#RDCH2		
3851	020734	012746	005405				MOV #RDCH2,-
3852	020740	012746	000001				MOV #1,-(SP)
3853	020744	010600					MOV SP,R0
3854	020746	104414					TRAP C\$PNTB
3855	020750	062706	000004				ADD #4,SP
3856	020754			CASE 2			
3857	020754	000421					BR 50103\$
3858	020756					50105\$:	
3859	020756			PRINTB	#RDCH1		
3860	020756	012746	005454				MOV #RDCH1,-
3861	020762	012746	000001				MOV #1,-(SP)
3862	020766	010600					MOV SP,R0
3863	020770	104414					TRAP C\$PNTB
3864	020772	062706	000004				ADD #4,SP
3865	020776			CASE 3			
3866	020776	000410					BR 50103\$
3867	021000					50104\$:	
3868	021000			PRINTB	#RDCHP3		
3869	021000	012746	005527				MOV #RDCHP3,-
3870	021004	012746	000001				MOV #1,-(SP)
3871	021010	010600					MOV SP,R0
3872	021012	104414					TRAP C\$PNTB
3873	021014	062706	000004				ADD #4,SP
3874	021020			ENDSELECT			
3875	021020					50103\$:	
3876	021020	000207		RTS PC			


```

3877
3878
3879
3880 021022
3881 021022 017701 161176
3882 021026 042701 177701
3883 021032
3884 021032 020127 000016
3885 021036 001075
3886 021040
3887 021040 104456
3888 021042 000023
3889 021044 003563
3890 021046 014250
3891 021050
3892 021050 013701 002274
3893 021054 000301
3894 021056
3895 021056 042701 177400
3896 021062
3897 021062 020127 000337
3898 021066 001011
3899 021070
3900 021070 012746 027374
3901 021074 012746 000001
3902 021100 010600
3903 021102 104414
3904 021104 062706 000004
3905 021110
3906 021110 000450
3907 021112
3908 021112
3909 021112 020127 000127
3910 021116 101034
3911 021120
3912 021120 042701 177740
3913 021124
3914 021124 006301
3915 021126
3916 021126 062701 021234
3917 021132
3918 021132 111102
3919 021134 106302
3920 021136
3921 021136 062702 021314
3922 021142
3923 021142 011103
3924 021144 000303
3925 021146
3926 021146 010346
3927 021150 042716 000377
3928 021154 042603
3929 021156
3930 021156
3931 021156
3932 021156 011246

```

```

: TEST 9, 10 -- MODULE SORT
MSORT3:: LET R1 := @TSSR CLR.BY #TCFCMK ;GET TERMINATION AND FATAL CLASS CODES
IF R1 EQ #16 THEN ;FC=0, TC=7
ERRHRD 19,MICROE,MICERR
LET R1 := SWAP XSTAT3 ;INPUT MICRO CODE ERROR
LET R1 := R1 CLR.BY #177400 ;KEEP 8 LSB'S
IF R1 EQ #337 THEN
PRINTB #COD337
ELSE
IF R1 LOS #127 THEN
LET R1 := R1 CLR.BY #177740
LET R1 := R1 SHIFT 1 ;R1X2 FOR WORD ADDRESSING
LET R1 := R1 * #LOOKTB ;INDEX INTO LOOK-UP TBL
LET R2 := R1 * #LOOKTB ;GET WORD OFFSET INTO CODTBL FROM LOOKTB
LET R2 := R2 * #CODTBL ;INDEX INTO CODTBL
LET R3 := SWAP (R1) ;GET NO OF LINE TO PRINT
LET R3 := R3 AND #377 ;FROM LOOKTB
REPEAT
PRINTB (R2)

```

```

MOV @TSSR,R1
BIC #TCFCMK,
CMP R1,#16
BNE 50107$
TRAP C$ERHRD
.WORD 19
.WORD MICROE
.WORD MICERR
MOV XSTAT3,R
SWAB R1
BIC #177400,
CMP R1,#337
BNE 50110$
MOV #COD337,
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
BR 50111$
50110$:
CMP R1,#127
BHI 50112$
BIC #177740,
ASL R1
ADD #LOOKTB,
MOVB (R1),R2
ASLB R2
ADD #CODTBL,
MOV (R1),R3
SWAB R3
MOV R3,-(SP)
BIC #377,(SP)
BIC (SP),R3
50113$:
PRINT CURRENT LINE
MOV (R2),-(S

```

3933 021160 012746 000001
 3934 021164 010600
 3935 021166 104414
 3936 021170 062706 000004
 3937 021174
 3938 021174 062702 000002
 3939 021200
 3940 021200 005303
 3941 021202
 3942 021202 005703
 3943 021204 001364
 3944 021206
 3945 021206 000411
 3946 021210
 3947 021210
 3948 021210 010146
 3949 021212 012746 027336
 3950 021216 012746 000002
 3951 021222 010600
 3952 021224 104414
 3953 021226 062706 000006
 3954 021232
 3955 021232
 3956 021232
 3957 021232
 3958 021232
 3959 021232
 3960 021232 000207
 3961

LET R2 := R2 + #2
 LET R3 := R3 - #1
 UNTIL R3 EQ #0
 ELSE
 PRINTB #CODXXX,R1
 ENDIF
 ENDIF
 ENDIF
 RTS PC

; THEN THE NEXT,

; IF ANY

50112\$:

50114\$:

50111\$:

50107\$:

MOV #1,-(SP)
 MOV SP,R0
 TRAP C\$PNTB
 ADD #4,SP
 ADD #2,R2
 DEC R3
 TST R3
 BNE 50113\$
 BR 50114\$
 MOV R1,-(SP)
 MOV #CODXXX,
 MOV #2,-(SP)
 MOV SP,R0
 TRAP C\$PNTB
 ADD #6,SP

```
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
```

```

: MICRO DIAG ERROCR CODE LOOKUP TABLE
: COD00 THRU COD27
: COD22, COD23, COD30 AND ABOVE, ARE UNDEFINED

```

LOOKTB:	.BYTE	
	0,3	
	3,2	
	5,1	
	6,2	
	10,2	
	12,1	
	13,1	
	14,1	
	15,3	
	20,3	
	23,2	
	25,3	
	30,4	
	34,4	
	40,4	
	44,3	
	47,2	
	51,2	
	53,1	
	54,1	
	55,2	
	57,3	
	62,3	
	65,3	
	↑↑↑	
:	!	
:	!	-----> NO OF LINES TO PRINT, IN HI BYTE (ODD)
:	!	-----> OFFSET IN CODTBL, IN LO BYTE (EVEN)
:		

```

3994
3995
3996 021314 021474 021602 021667
3997 021322 021751 022050
3998 021326 022072
3999 021330 022165 022273
4000 021334 022351 022457
4001 021340 022506
4002 021342 022605
4003 021344 022702
4004 021346 022773 023101 023203
4005 021354 023234 023317 023403
4006 021362 023503 023606
4007 021366 023666 023765 024037
4008 021374 024117 024221 024301
4009 021402 024354
4010 021404 024434 024535 024621
4011 021412 024675
4012 021414 024755 025057 025137
4013 021422 025221
4014 021424 025301 025405 025464
4015 021432 025544 025652
4016 021436 025752 026060
4017 021442 026106
4018 021444 026143
4019 021446 026200 026306
4020 021452 026344 026452 026526
4021 021460 026623 026706 026762
4022 021466 027057 027165 027241
4023
4024
4025
4026

```

;MICRO CODE ERROR AD TABLE

```

CODTBL: .WORD COD00,COD00A,COD00B      :0      3
         .WORD COD01,COD01A           :3      2
         .WORD COD02                   :5      1
         .WORD COD03,COD03A           :6      2
         .WORD COD04,COD04A           :10     2
         .WORD COD05                   :12     1
         .WORD COD06                   :13     1
         .WORD COD07                   :14     1
         .WORD COD10,COD10A,COD10B    :15     3
         .WORD COD11,COD11A,COD11B    :20     3
         .WORD COD12,COD12A           :23     2
         .WORD COD13,COD13A,COD13B    :25     3
         .WORD COD14,COD14A,COD14B,COD14C :30     4
         .WORD COD15,COD15A,COD15B,COD15C :34     4
         .WORD COD16,COD16A,COD16B,COD16C :40     4
         .WORD COD17,COD17A,COD17B    :44     3
         .WORD COD20,COD20A           :47     2
         .WORD COD21,COD21A           :51     2
         .WORD COD22                   :53     1
         .WORD COD23                   :54     1
         .WORD COD24,COD24A           :55     2
         .WORD COD25,COD25A,COD25B    :57     3
         .WORD COD26,COD26A,COD26B    :62     3
         .WORD COD27,COD27A,COD27B    :65     3
         ;                               :↑↑    ↑
         ;                               :!     !
;OFFSET INTO THIS TABLE <-----
;NO OF .WORDS=NO OF LINES TO PRINT <-----

```

4027
 4028
 4029

:MICRO CODE ERROR MSGS

				.NLIST	BEX
021474	040445	030061	020060	COD00:	.ASCIZ /#A100 BASIC IO MICRO FAIL:PAR ERR, IOATN,HANDSHAKE, M8967(12)
021602	040445	020040	020040	COD00A:	.ASCIZ /#A DATA WINDOW TEST BETWEEN IO AND MAIN MICROS,#N/
021667	045	020101	020040	COD00B:	.ASCIZ /#A SERIAL BUS.SHIN (SHIFT IN) STUCK TRUE,...#N/
021751	045	030501	030460	COD01:	.ASCIZ /#A101 ERROR IN IO CONTROL REGISTER TEST#S13#AM8966(14) 15#N/
022050	051445	030065	040445	COD01A:	.ASCIZ /#S50#AM8967(12)#N/
022072	040445	030061	020062	COD02:	.ASCIZ /#A102 FAILURE OF FRAME COUNTER TEST#S17#AM8966(14) 15#N/
022165	045	030501	031460	COD03:	.ASCIZ /#A103 FAILURE OF IO SILO NON-PARITY ERROR DATA M8966(14)
022273	045	020101	020040	COD03A:	.ASCIZ /#A TEST OR THE WRITE FLAG#S24#AM8963(11)#N/
022351	045	030501	032060	COD04:	.ASCIZ /#A104 FAILURE OF IO SILO PARITY ERROR TEST OR M8966(14)
022457	045	020101	020040	COD04A:	.ASCIZ /#A DATA LATE TEST#N/
022506	040445	030061	020065	COD05:	.ASCIZ /#A105 FAILURE OF SHIFT LOOP WITH ZEROES#S13#AM8965(15) 20#N/
022605	045	030501	033060	COD06:	.ASCIZ /#A106 FAILURE OF SHIFT LOOP WITH ONES#S15#AM8965(15) 21#N/
022702	040445	030061	020067	COD07:	.ASCIZ /#A107 FAILURE OF SHIFT LENGTH MUX#S19#AM8965(15) 22#N/
022773	045	030501	030061	COD10:	.ASCIZ /#A110 FAILURE TO RECEIVE CORRECT OP-CODE FROM TS11 M8965(15)
023101	045	020101	020040	COD10A:	.ASCIZ /#A WHEN DATA SENT OVER THE SERIAL BUS#S12#ATS11,MOTHER BOARD
023203	045	032523	022460	COD10B:	.ASCIZ /#S50#ASERIAL BUS CABLE#N/
023234	040445	030461	020061	COD11:	.ASCIZ /#A111 FAILURE OF 1KHZ CLOCK TEST#S20#AG159#S9#A2#N/
023317	045	020101	020040	COD11A:	.ASCIZ /#A TSTS TAC SYNC FLOP AND ATTN#S19#ACBUS CABLE#N/
023403	045	041501	042510	COD11B:	.ASCIZ /#ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159 M8963(11)#N
023503	045	030501	031061	COD12:	.ASCIZ /#A112 LIGHT REG CHANGED WHEN MOTION REG WAS CLEARED G159#S9#A3,
023606	040445	044103	041505	COD12A:	.ASCIZ /#ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159#N/
023666	040445	030461	020063	COD13:	.ASCIZ /#A113 FWD OR MVG BITS WRONG AFTER 1 TICK OF#S9#AG159#S9#A3,4#N/
023765	045	020101	020040	COD13A:	.ASCIZ /#A SIMULATED COMMAND AND TACH PULSES#N/
024037	045	041501	042510	COD13B:	.ASCIZ /#ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159#N/
024117	045	030501	032061	COD14:	.ASCIZ /#A114 FAILURE OF SIMULATED CAPSTAN SPEED TEST:#S6#AG159#S9#A3,4
024221	045	020101	020040	COD14A:	.ASCIZ /#A CAPSTAN SPEED COUNTER OUT OF RANGE WHEN#N/
024301	045	020101	020040	COD14B:	.ASCIZ /#A TAPE MOTION AT SPEED WAS SIMULATED#N/
024354	040445	044103	041505	COD14C:	.ASCIZ /#ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159#N/
024434	040445	030461	020065	COD15:	.ASCIZ /#A115 FAILURE OF SIMULATED SLOW CAPSTAN TEST:#S7#AG159#S9#A3,4#
024535	045	020101	020040	COD15A:	.ASCIZ /#A SPEED COUNTER NOT LATCHED UP WITH MAX COUNT#N/
024621	045	020101	020040	COD15B:	.ASCIZ /#A WHEN SLOW TACK TICKS WERE SIMULATED#N/
024675	045	041501	042510	COD15C:	.ASCIZ /#ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159#N/
024755	045	030501	033061	COD16:	.ASCIZ /#A116 FAILURE OF SIMULATED CAPSTAN DECEL TEST:#S6#AG159#S9#A3,4
025057	045	020101	020040	COD16A:	.ASCIZ /#A COUNTER NOT ZERO FOR FORWARD OR 377 FOR#N/
025137	045	020101	020040	COD16B:	.ASCIZ /#A REVERSE WHILE DECELERATING OR MVG BIT=0#N/
025221	045	041501	042510	COD16C:	.ASCIZ /#ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159#N/
025301	045	030501	033461	COD17:	.ASCIZ /#A117 FAILURE OF MOVING FLOP TO RESET AFTER STOP#S4#AG159#S9#A3
025405	045	020101	020040	COD17A:	.ASCIZ /#A (DIRECTION REVERSAL FOR ONE TACH TICK)#N/
025464	040445	044103	041505	COD17B:	.ASCIZ /#ACHECK TACH PHASE,SKEW,SPEED ADJUSTS ON G159#N/
025544	040445	031061	020060	COD20:	.ASCIZ /#A120 FAILURE OF WRITE BOARD TO TURN ON AND EMPTY M8929(13)
025652	040445	020040	020040	COD20A:	.ASCIZ /#A THE SILO OR DATA LATE BIT NOT WORKING M8966(14)#N
025752	040445	031061	020061	COD21:	.ASCIZ /#A121 FAILURE OF WRITE BOARD TO EMPTY SILO AT M8929(13)
026060	040445	020040	020040	COD21A:	.ASCIZ /#A CORRECT SPEED#N/
026106	040445	031061	020062	COD22:	.ASCIZ /#A122 UNDEFINED ERROR CODE#N/
026143	045	030501	031462	COD23:	.ASCIZ /#A123 UNDEFINED ERROR CODE#N/
026200	040445	031061	020064	COD24:	.ASCIZ /#A124 FORMATTER CTRL BOARD: FORMATTER FLAG FAILED M8922(7)
026306	040445	044103	041505	COD24A:	.ASCIZ /#ACHECK VCO ADJUST ON M8922#N/
026344	040445	031061	020065	COD25:	.ASCIZ /#A125 FORMATTER SILO FILLING AND DATA ERROR M8922(7)
026452	040445	044103	041505	COD25A:	.ASCIZ /#ACHECK VCO ADJUST ON M8922#S25#AM8923(3)#N/
026526	040445	044103	041505	COD25B:	.ASCIZ /#ACHECK SKEW, THRESHOLD ADJUSTS ON M8923#S12#AM8924(4,5,6)#N/
026623	045	030501	033062	COD26:	.ASCIZ /#A126 PEAK SHIFT TEST ERROR#S25#AM8922(7)#S5#A25#N/
026706	040445	044103	041505	COD26A:	.ASCIZ /#ACHECK VCO ADJUST ON M8922#S25#AM8923(3)#N/
026762	040445	044103	041505	COD26B:	.ASCIZ /#ACHECK SKEW, THRESHOLD ADJUSTS ON M8923#S12#AM8924(4,5,6)#N/

027057 045 030501 033462
027165 045 041501 042510
027241 045 041501 042510

027336 047445 022463 020101
027374 040445 031463 020067

4030 027426
4031
4032 027426
4033
4034
4035
4036
4037
4038
4039
4040
4041 027426
4042 027426
4043 027426 000000
4044 027430 177777
4045 027432 177777
4046 027434

COD27: .ASCIZ /#A127 FORMATTER TABLE LOOKUP ROM CHECKSUM TEST ERR M8922(7)
COD27A: .ASCIZ /#ACHECK VCO ADJUST ON M8922#S25#AM8923(3)#N/
COD27B: .ASCIZ /#ACHECK SKEW, THRESHOLD ADJUSTS ON M8923#S12#AM8924(4,5,6)#N/

CODXXX: .ASCIZ /#03#A UNDEFINED ERROR CODE#N/

COD337: .ASCIZ /#A337 CAPSTAN RUNAWAY#N/
.LIST BEX
.EVEN

ENDMOD

.SBTTL LOAD DEV PROTECTION TABLE

; **
; TABLE FOR SUPERVISOR TO IDENTIFY P-TBL FOR LOAD DEV
; AND TO WARN OPERATOR WHEN HE TRIES TO TEST THE LOAD DEVICE
; --

BGNPROT
L\$PROT::
.WORD 0 ;P-TBL OFFSET OF TSSR
.WORD -1 ;P-TBL OFFSET OF MASS BUS UNIT#: -1 = NOT A MASS
.WORD -1 ;P-TBL OFFSET OF DRIVE #: -1 = NONE, ONE DRIVE P
ENDPROT

```

4047
4048
4049
4050 027434
4051
4052
4053
4054
4055
4056 027434
4057 027434
4058
4059
4060 027434 032727 000003 002230
4061 027442 001421
4062 027444
4063 027444 104454
4064 027446 000001
4065 027450 002502
4066 027452 000000
4067 027454
4068 027454 012727 000310
4069 027460 000000
4070 027462 013727 002116
4071 027466 000000
4072 027470 005367 177772
4073 027474 001375
4074 027476 005367 177756
4075 027502 001367
4076 027504 000757
4077
4078 027506
4079 027506 012700 000035
4080 027512 104447
4081 027514
4082 027514 103003
4083 027516
4084 027516 012737 177777 027712
4085 027524
4086 027524 012700 000034
4087 027530 104447
4088 027532
4089 027532 103402
4090 027534
4091 027534 005237 027712
4092 027540
4093 027540 013700 027712
4094 027544
4095 027544 020027 000003
4096 027550 003401
4097 027552
4098 027552 104444
4099 027554
4100 027554
4101
4102 027554

```

```

.TITLE MISCELLANEOUS SECTIONS
.SBTTL INITIALIZE SECTION
      BGNMOD
; **
; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
; AT THE BEGINNING OF EACH PASS.
; --
      BGNINIT
L$INIT::
      BIT      #BIT0!BIT1,#CMDPKT      ;; IS CMD PKT ON A MODULO 4 BOUNDARY?
      BEQ      2$                       ;; BR IF SO, OK.
1$:     ERRSF   1,MODUER                 ;; IF NOT, TELL HIM THE PROGRAM IS SCREWE
                                          TRAP      C$ERSF
                                          .WORD    1
                                          .WORD    MODUER
                                          .WORD    0
      DELAY 200.                        ;GO TO THE SUPERVISOR, WAIT 1 SECOND.
                                          MOV      #200.,(P
                                          .WORD    0
                                          MOV      L$DLY,(P
                                          .WORD    0
                                          DEC      -6(PC)
                                          BNE     .-4
                                          DEC      -22(PC)
                                          BNE     .-20
      BR       1$                       ;; TELL HIM AGAIN IF HE INSISTS ON CONTIN
2$:     READEF #EF.NEW                   ; IS THIS A NEW PASS?
                                          MOV      #EF.NEW,
                                          TRAP    C$REFG
      BNCOMPLETE 3$                       ; BR IF NOT.
                                          BCC     3$
      LET LUNIT := #-1                   ; INIT THE LOGICAL UNIT #.
3$:     READEF #EF.PWR                   ; HAS THERE BEEN A POWER FAILURE?
                                          MOV      #EF.PWR,
                                          TRAP    C$REFG
      BCOMPLETE 4$                       ; BRANCH IF SO - KEEP CURRENT UNIT #.
                                          BCS    4$
      LET LUNIT := LUNIT + #1           ; UPDATE UNIT #.
4$:     LET RO := LUNIT                  ; PREPARE TO PASS # TO SUPER.
      IF RO GT #3 THEN                   ; IF # IS PASS THE LIMIT THEN:
                                          MOV      LUNIT,RO
                                          CMP      RO,#3
                                          BLE     50115$
      DOCLN                               ; DO CLEANUP AND TERMINATE PASS.
                                          TRAP    C$DCLN
                                          50115$:
      ENDIF
      GPHARD RO,RO                       ; SETUP TO RETRIEVE HRD P-TABLE DATA.

```

```

4103 027554 104442                                TRAP   C$GPHRD
4104 027556                                         ;BR IF THIS UNIT HAS BEEN DROPPED.
4105 027536 103362                                BCC   3$
4106 027560 011037 002224                        MOV   (R0),TSSR                        ;GET THE TSSR ADDR.
4107 027564 013737 002224 002214                MOV   TSSR,TSDB                       ;CALCULATE THE
4108 027572 162737 000002 002214                SUB   #2,TSDB                          ;TSDB ADDRESS.
4109 027600 013737 002214 002220                MOV   TSDB,TSBA                        ;LOAD TSBA ADDRESS.
4110 027606 013737 002220 002222                MOV   TSBA,TSBAHI                      ;CALCULATE THE TSBA
4111 027614 062737 000001 002222                ADD   #1,TSBAHI                        ;HI BYTE ADDRESS.
4112 027622 013737 002214 002216                MOV   TSDB,TSDBHI                      ;CALCULATE THE
4113 027630 062737 000001 002216                ADD   #1,TSDBHI                        ;TSDB HI BYTE ADDR.
4114 027636 016037 000002 002226                MOV   2(R0),TSVCT                      ;GET THE VECTOR ADDRESS.
4115 027644                                         LET R5 := LUNIT SHIFT 1                ;GET LOGICAL UNIT # X 2.
4116 027644 013705 027712                        MOV   LUNIT,R5
4117 027650 006305                                ASL   R5
4118 027652                                         ;SET UP INTERRUPT PROCESSING CON
4119 027652 012746 000340                        SETVEC TSVCT,TS4INT(R5),#INTPRI
4120 027656 016546 002306                        MOV   #INTPRI,
4121 027662 013746 002226                        MOV   TS4INT(R
4122 027666 012746 000003                        MOV   TSVCT,-(
4123 027672 104437                                MOV   #3,-(SP)
4124 027674 062706 000010                        TRAP  C$SVEC
4125 027700                                         ADD   #10,SP
4126 027700 013737 027712 002422                MOV   LUNIT,UN
4127 027706                                         TRAP  C$EXIT
4128 027706 104432                                         .WORD L10031-.
4129 027710 000004
4130
4131 ; LOCAL STORAGE THAT IS USED ONLY DURING THE INITIALIZE SECTION.
4132
4133 027712 000000                                LUNIT: .WORD 0                        ;CURRENT LOGICAL UNIT #.
4134
4135 .EVEN
4136
4137 027714                                ENDINIT
4138 027714                                L10031:
4139 027714 104411                                TRAP  C$INIT
4140 .SBTTL AUTO DROP SECTION
4141
4142
4143 ;++
4144 ;SECTION EXECUTED AFTER THE INIT CODE WHEN "ADR" FLAG IS SET BY OPERATOR
4145 ;SECTION CHECKS FOR A VALID INTERFACE LOCATION. DROPS UNIT IF NO RESPONSE
4146 ;FROM INTERFACE
4147 ;--
4148 027716                                BGNAUTO
4149 027716                                L$AUTO::
4150
4151 027716                                LET TRAPD4 :B= #0
4152 027716 105037 030166                        CLR   TRAPD4
4153 027722                                SETVEC #4,#TRAP4,#PRI07
4154 027722 012746 000340                        MOV   #PRI07,-
4155 027726 012746 030160                        MOV   #TRAP4,-
4156 027732 012746 000004                        MOV   #4,-(SP)
4157 027736 012746 000003                        MOV   #3,-(SP)
4158 027742 104437                                TRAP  C$SVEC

```


4159 027744 062706 000010
 4160 027750
 4161 027750 017701 152250
 4162 027754
 4163 027754 012700 000004
 4164 027760 104436
 4165 027762
 4166 027762 105737 030166
 4167 027766 001415
 4168 027770
 4169 027770 013746 002224
 4170 027774 012746 030064
 4171 030000 012746 000002
 4172 030004 010600
 4173 030006 104417
 4174 030010 062706 000006
 4175 030014 004737 014654
 4176 030020
 4177 030020 000420
 4178 030022
 4179 030022
 4180 030022 032777 000200 152174
 4181 030030 001014
 4182 030032
 4183 030032 013746 002422
 4184 030036 012746 002570
 4185 030042 012746 000002
 4186 030046 010600
 4187 030050 104417
 4188 030052 062706 000006
 4189 030056 004737 014654
 4190 030062
 4191 030062
 4192 030062
 4193 030062
 4194
 4195 030062
 4196 030062
 4197 030062 104461
 4198
 4199 030064 040445 052502 020123
 4200 030072 051124 050101 040440
 4201 030100 020124 047445 022466
 4202 030106 116
 4203 030107 045 044501 052116
 4204 030114 051105 040506 042503
 4205 030122 041040 042101 047440
 4206 030130 020122 047516 020124
 4207 030136 042523 020124 047524
 4208 030144 040440 047502 042526
 4209 030152 040440 022504 000116
 4210
 4211
 4212
 4213
 4214

LET R1 := @TSSR
 CLRVEC #4
 IFB TRAPD4 NE #0 THEN
 PRINTF #AUTODM,TSSR
 JSR PC,DROPU
 ELSE
 IF #TS.SSR NOTSETIN @TSSR THEN
 PRINTF #SSROFF,UNIT
 JSR PC,DROPU
 ENDIF
 ENDIF
 ENDAUTO
 L10032:
 AUTODM: .ASCII /#ABUS TRAP AT #06#N/
 .ASCIZ /#AINTERFACE BAD OR NOT SET TO ABOVE AD#N/
 .EVEN
 :
 : DEVICE BUS TRAP HANDLER
 : OUTPUT: TRAPD4 BYTE 1: TRAPED AT 4
 : 0: NO TRAP

ADD #10,SP
 MOV @TSSR,R1
 MOV #4,R0
 TRAP C\$CVEC
 TSTB TRAPD4
 BEQ 50116\$
 MOV TSSR,-(S
 MOV #AUTODM,
 MOV #2,-(SP)
 MOV SP,R0
 TRAP C\$PNTF
 ADD #6,SP
 BR 50117\$
 50116\$:
 BIT #TS.SSR,
 BNE 50120\$
 MOV UNIT,-(S
 MOV #SSROFF,
 MOV #2,-(SP)
 MOV SP,R0
 TRAP C\$PNTF
 ADD #6,SP
 50120\$:
 50117\$:
 TRAP C\$AUTO

4215
4216 030160
4217 030160 105237 030166
4218 030164 000002
4219
4220 030166 000
4221 030170
4222

TRAP4: LET TRAPD4 :B= TRAPD4 * #1
RTI
TRAPD4: .BYTE 0 ;TRAPED AT 4 FLAG
.EVEN

INCB TRAPD4

```

4223                                     .SBTTL  CLEANUP CODING SECTION
4224
4225                                     ;**
4226                                     ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4227                                     ; AT THE END OF EACH PASS.
4228                                     ;--
4229
4230 030170                               BGNCLN
4231 030170                               L$CLEAN::
4232
4233 030170                               CLRVEC  TSVCT                               ;RELEASE THE INTERRUPT VECTOR.
4234 030170 013700 002226                               MOV      TSVCT,RO
4235 030174 104436                               TRAP    C$CVEC
4236 030176                               SETPRI  PRI07
4237 030176 013700 000340                               MOV      PRI07,RO
4238 030202 104441                               TRAP    C$SPRI
4239
4240                               EXIT      CLN
4241 030204 104432
4242 030206 000002                               TRAP    C$EXIT
4243                               .WORD    L10033..
4244
4245                               .EVEN
4246
4247 030210                               ENDCLN
4248 030210                               L10033:
4249 030210 104412                               TRAP    C$CLEAN

```

```

4250 .SBTTL DROP UNIT SECTION
4251
4252
4253 ;**
4254 ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4255 ; TO NO LONGER BE TESTED.
4256 ;--
4257 030212 BGNDU
4258 030212 L$DU::
4259
4260
4261 030212 000240 NOP
4262 030214 000240 NOP
4263 030216 000240 NOP
4264 030220 000240 NOP
4265
4266 030222 EXIT DU
4267 030222 000167
4268 030224 000000 .WORD JSJMP
4269 .WORD L10034-2
4270
4271 .EVEN
4272
4273 030226 ENDDU
4274 030226 L10034:
4275 030226 104453 TRAP C$DU

```

4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305

030230
030230

030230 000240
030232 000240
030234 000240
030236 000240

030240
030240 000167
030242 000000

030244
030244
030244 104452

030246

.SBTTL ADD UNIT SECTION

; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
; TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
; "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.

L\$AU:: BGNU

NOP
NOP
NOP
NOP

EXIT AU

.WORD J\$JMP
.WORD L10035-2

.EVEN

ENDAU
L10035:

TRAP C\$AU

ENDMOD

```

4306 .TITLE HARDWARE TESTS
4307
4308 .SBTTL TEST 1: PDP11/TS11 WRAP TEST.
4309 030246 BGNMOD
4310
4311 ;**
4312 ; TEST TO INSURE PROPER COMMUNICATION BETWEEN THE PDP11 AND THE TS11 BY
4313 ; WRAPPING THE FOLLOWING PATTERNS:
4314 ; A 1 IN A FIELD OF 0'S; A 0 IN A FIELD OF 1'S.
4315
4316 ; WHEN DATA IS WRITTEN TO THE TSDB HI BYTE, THE DATA IS WRAPPED
4317 ; AROUND WITHIN THE TS11 AND APPEARS IN THE TSBA LO
4318 ; AND HI BYTES. THE 2 LOW ORDER BITS OF THE DATA WILL BE
4319 ; REFLECTED IN THE TSSR EXTENDED ADDRESS BITS.
4320 ; R4 CONTAINS A COPY OF THE DATA SENT.
4321 ; R3 CONTAINS THE EXPECTED TSBA RESULTS
4322 ; R2 CONTAINS THE EXPECTED STATE OF THE TWO EXTENDED ADDRESS
4323 ; BITS IN THE TSSR.
4324 ;--
4325
4326
4327
4328 030246 T1:: BGNST
4329 030246
4330
4331 030246 T1.1: BGNSUB
4332 030246
4333 030246 104402 BGNSEG TRAP C$BSUB
4334 030250 104404 BRESET ;RESET THE BUS. TRAP C$BSEG
4335 030252 104433 IF #TS.SSR SETIN @TSSR THEN TRAP C$RESET
4336 030254 032777 000200 151742 ERRDF 29,SSRON BIT #TS.SSR,
4337 030262 001416 ERRDF 29,SSRON BEQ 50121$
4338 030264 104455 PRINTB #TS11BD TRAP C$ERDF
4339 030266 000035 PRINTB #TS11BD .WORD 29
4340 030270 002625 PRINTB #TS11BD .WORD SSRON
4341 030272 000000 PRINTB #TS11BD .WORD 0
4342 030274 012746 005041 MOV #TS11BD,
4343 030300 012746 000001 MOV #1,-(SP)
4344 030304 010600 MOV SP,R0
4345 030306 104414 TRAP C$PNTB
4346 030310 062706 000004 ADD #4,SP
4347 030314 004737 014654 CALL DROPU JSR PC,DROPU
4348 030320 ENDIF
4349 030320 ENDSEG 50121$:
4350 030320 104405 CALL WAITSR TRAP C$ESEG
4351 030322 004737 014702 CALL WAITSR JSR PC,WAIT$
4352 030326 ENDSUB
    
```



```

4461 .SBTTL TEST 2: PDP11/TS04 WRAP TEST.
4462 ;
4463 ; TEST TO INSURE PROPER COMMUNICATION BETWEEN THE PDP11 AND
4464 ; THE TS04 BY WRAPPING THE FOLLOWING PATTERN:
4465 ; A 1 IN A FIELD OF 0'S, A 0 IN A FIELD OF 1'S
4466 ;
4467 ; WHEN THE DATA IS WRITTEN TO THE TSDB LO BYTE, THE DATA IS
4468 ; SENT TO THE TS04, VIA THE SERIAL LINE, WHERE IT IS WRAPPED
4469 ; AROUND BACK OVER THE SERIAL LINE TO THE TS11. THE DATA THEN
4470 ; APPEARS IN THE TSBA LO AND TSSR LO BYTES.
4471 ; R4 CONTAINS THE EXPECTED TSBA RESULTS AND THE
4472 ; EXPECTED TSSR RESULTS.
4473 ;
4474 030602 BGNTST
4475 030602 T2::
4476
4477 030602 012704 000200 MOV @200,R4 ;INIT THE DATA.
4478 ;REPEAT FOLLOWING UNTIL DATA ALL SHIFTED
4479 030606 005077 151412 T2WRAP: CLR @TSSR
4480 030612 CALL WAITSR
4481 030612 004737 014702
4482 030616 032777 000200 151400 BIT @TS.SSR,@TSSR JSR PC,WAIT5
4483 030624 001452 BEQ T2SHFT
4484 030626 BGNSUB
4485 030626 T2.1:
4486 030626 104402
4487 030630 110477 151360 MOVB R4,@TSDB ;SENf DATA. TRAP C$BSUB
4488 030634 DELAY 10. ;LJ TO SUPERVISOR, WAIT 1 MSEC.
4489 030634 012727 000012 MOV #10.,(PC
4490 030640 000000 .WORD 0
4491 030642 013727 002116 MOV L$DLY,(P
4492 030646 000000 .WORD 0
4493 030650 005367 177772 DEC -6(PC)
4494 030654 001375 BNE -.4
4495 030656 005367 177756 DEC -22(PC)
4496 030662 001367 BNE .-20
4497 030664 017737 151330 002432 MOV @TSBA,TEMP1 ;SAVE TSBA.
4498 030672 IFB TEMP1 NE R4 THEN
4499 030672 123704 002432
4500 030676 001406 CMPB TEMP1,R4
4501 030700 ERRDF 5,WRPER3,WRAPR2 BEQ 50125$
4502 030700 104455
4503 030702 000005 TRAP C$ERDF
4504 030704 003017 .WORD 5
4505 030706 006304 .WORD WRPER3
4506 030710 .WORD WRAPR2
4507 030710 004737 014654 CALL DROPU JSR PC,DROPU
4508 030714 ENDIF
4509 030714
4510 030714 017737 151304 002434 MOV @TSSR,TEMP2 ;SAVE TSSR. 50125$:
4511 030722 IFB TEMP2 NE R4 THEN
4512 030722 123704 002434
4513 030726 001406 CMPB TEMP2,R4
4514 030730 ERRDF 6,WRPER3,WRAPR3 BEQ 50126$
4515 030730 104455 TRAP C$ERDF
4516 030732 000006 .WORD 6
    
```

4517 030734 003017
 4518 030736 006456
 4519 030740
 4520 030740 004737 014654
 4521 030744
 4522 030744
 4523 030744
 4524 030744
 4525 030744 104403
 4526 030746
 4527 030746 004737 016104
 4528 030752 120427 000377
 4529 030756 001313
 4530
 4531
 4532
 4533
 4534 030760
 4535 030760
 4536 030760 104401

CALL DROPU
 ENDF
 ENDSUB
 L10042:
 CALL SHWRAP
 J2SHFT: CMPB R4,#377
 BNE T2WRAP
 .EVEN
 ENDTST
 L10041:

.WORD WRPER3
 .WORD WRAPR3
 JSR PC,DROPU
 501264:
 TRAP C\$ESUB
 JSR PC,SHWRA
 TRAP C\$ETST

```

4537 .SBTTL TEST 3: SET TS04 CHARACTERISTIC VERIFICATION.
4538 ;
4539 ; THE FUNCTION OF THIS TEST IS TO ISSUE A "SET CHARACTERISTIC" COMMAND
4540 ; TO TELL THE TS04 WHERE IN CORE THE MESSAGE PACKET RESIDES.
4541
4542 030762 BGNTST
4543 030762 T3.:
4544
4545 030762 BGNSUB
4546 030762 T3.1:
4547 030762 104402 CLR      CTLFLG          TRAP      C$BSUB
4548 030764 105037 002437 CLR      @TSSR          ;INIT TS11-TS04
4549 030770 005077 151230 JSR PC,WAITSR
4550 030774 004737 014702 IF #TS.NBA NOTSETIN @TSSR THEN ;NBA SHOULD BE SET SINCE A COMD
4551 031000
4552 031000 032777 002000 151216 BIT      #TS.NBA,
4553 031006 001006 ERRDF 20,SCHERR,SCHER1 ;WAS ISSUED WITHOUT SET CHAR ISSUED FIRS
4554 031010
4555 031010 104455 TRAP      C$ERDF
4556 031012 000024 .WORD    20
4557 031014 003051 .WORD    SCHERR
4558 031016 010264 .WORD    SCHER1
4559 031020 CALL DROPU ;IF NBA NOTSET, THEN CALL ERROR, DROP UN
4560 031020 004737 014654 JSR      PC,DROPU
4561 031024
4562 031024 ENDIF
4563 031024 ENDSUB
4564 031024 L10044:
4565 031024 104403 TRAP      C$ESUB
4566
4567 031026 BGNSUB
4568 031026 T3.2:
4569 031026 104402 TRAP      C$BSUB
4570 031030 CALL SCHEXE JSR      PC,SCHEX
4571 031030 004737 016032 IF #TS.NBA SETIN @TSSR THEN
4572 031034
4573 031034 032777 002000 151162 BIT      #TS.NBA,
4574 031042 001406 BEQ      50130$
4575 031044
4576 031044 104455 TRAP      C$ERDF
4577 031046 000025 .WORD    21
4578 031050 003051 .WORD    SCHERR
4579 031052 010416 .WORD    SCHER2
4580 031054 CALL DROPU JSR      PC,DROPU
4581 031054 004737 014654
4582 031060 ENDIF
4583 031060 IF @TSBA NE #MSGEND THEN 50130$:
4584 031060
4585 031060 027727 151134 002276 CMP      @TSBA,#M
4586 031066 001406 BEQ      50131$
4587 031070
4588 031070 104455 TRAP      C$ERDF
4589 031072 000026 .WORD    22
4590 031074 003051 .WORD    SCHERR
4591 031076 010532 .WORD    SCHER3
4592 031100 CALL DROPU

```



```

4782 031454
4783 031454
4784
4785 031454
4786 031454 004737 016032
4787 031460
4788 031460 004737 015260
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799 031464
4800 031464
4801 031464 104402
4802
4803 031466 004737 016240
4804 031472 112745 000001
4805
4806
4807
4808 031476 005003
4809 031500 012704 000040
4810 031504 004737 015712
4811 031510 012704 000044
4812 031514 004737 015712
4813 031520 004737 015712
4814 031524 004737 015712
4815 031530 012704 000040
4816 031534 004737 015712
4817 031540 004737 015712
4818 031544 004737 016006
4819
4820
4821
4822 031550 005037 002436
4823 031554 005037 002342
4824 031560 004737 016130
4825 031564
4826 031564 013737 002360 002406
4827 031572
4828 031572 105737 002436
4829 031576 001407
4830 031600 105737 002212
4831 031604 001404
4832 031606
4833 031606 104456
4834 031610 000007
4835 031612 003102
4836 031614 007002
4837 031616

T4::
BGNTST
CALL SCHEXE ;SET CHAR
CALL WAITMT JSR PC,SCHEX
JSR PC,WAITM

; THIS SUBTEST FORCES THE TRACK ACTIVE TO CLEAR BY WRITING
; ALL 0'S DATA. THE PATTERN IS AS FOLLOWS FOR EACH CHANNEL:
;
; DATA: 000000
; WRTFLG: 011100
; TRACK ACTIVE: SHOULD BE 0 FOR ALL TRACKS

;INITIALIZAITON
T4.1:
BGNSUB
TRAP C$BSUB
JSR PC,PEINIT ;GO LOAD DIABLK INDEX AND TSO4 TASK ADDR
MOV #FC.NRZ,-(R5) ;SET NRZ1 MODE IN FMT CNTRL.

;SETUP DATA & CONTROL WORDS IN DIABLK.
CLR R3 ;SETUP DATA FOR WORD 1
MOV #IS.NRZ,R4 ;SETUP CONTROL WORD 1
JSR PC,TALOAD ;LOAD IT.
MOV #IS.NRZ!IS.WRF,R4 ;SET WRITE FLAG IN CONTROL WORD 2
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 2
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 3
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 4.
MOV #IS.NRZ,R4 ;SETUP CNTRL WORD 5
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 5
JSR PC,TALOAD ;LOAD DATA & CNTRL WORD 6.
JSR PC,DIAEXE ;EXECUTE THE CMD.

;NOW CHECK FOR ERRORS.
CLR ERRFLG ;CLR THE ERROR FLAG.
CLR EXTRAC ;TRACK ACTIVE 0 FOR ALL CHAN
JSR PC,TKACER ;CHECK FOR ERRORS.
LET ACTRK1 := ACTRAC
MOV ACTRAC,A
IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
TSTB ERRFLG
BEQ 50151$
TSTB CMPFLG
BEQ 50151$

ERRHRD 7,TAER1,TAEM
TRAP C$ERHRD
.WORD 7
.WORD TAER1
.WORD TAEM

ENDIF

```


4838 031616
 4839
 4840 031616
 4841 031616
 4842 031616 104403
 4843
 4844
 4845
 4846
 4847
 4848
 4849
 4850
 4851 031620
 4852 031620
 4853 031620 104402
 4854
 4855
 4856
 4857 031622 004737 016240
 4858 031626 112745 000001
 4859
 4860
 4861
 4862 031632 012703 177777
 4863 031636 012704 000040
 4864 031642 004737 015712
 4865 031646 012704 000044
 4866 031652 004737 015712
 4867 031656 005003
 4868 031660 004737 015712
 4869 031664 004737 015712
 4870 031670 012704 000040
 4871 031674 004737 015712
 4872 031700 004737 015712
 4873 031704 004737 016006
 4874
 4875
 4876
 4877 031710 005037 002436
 4878 031714 012737 000777 002342
 4879 031722 004737 016130
 4880 031726
 4881 031726 013737 002360 002410
 4882 031734 005137 002410
 4883 031740
 4884 031740 042737 177000 002410
 4885 031746
 4886 031746 105737 002436
 4887 031752 001407
 4888 031754 105737 002212
 4889 031760 001404
 4890 031762
 4891 031762 104456
 4892 031764 000011
 4893 031766 003152

50151\$:

ENDSUB
L10047:

TRAP C\$ESUB

; THIS SUBTEST FORCES TRACK ACTIVE TO SET BY WRITING THE FOLLOWING
; PATTERN ON EACH CHANNEL:

; DATA: 110000
; WRTFLG: 011100
; TRACK ACTIVE: SHOULD BE 1 FOR ALL TRACKS.

BGNSUB
T4.2:

TRAP C\$BSUB

; INITIALIZATION

JSR PC, PEINIT ;
MOVB #FC.NRZ, -(R5) ; NRZI MODE

; SET UP DATA & CNTRL WORDS IN DIABLK

MOV #-1, R3 ; WORD 1 DATA
MOV #IS.NRZ, R4 ; WORD 1 CONTROL
JSR PC, TALOAD ; LOAD IT
MOV #IS.NRZ!IS.WRF, R4 ; WORD 2 CONTROL -- WRT FLAG SET
JSR PC, TALOAD ; LOAD IT WORD 2 (DATA & CONTROL)
CLR R3 ; WORD 3 DATA
JSR PC, TALOAD ; LOAD DATA & CNTRL - WORD 3
JSR PC, TALOAD ; LOAD DATA & CNTRL - WORD 4
MOV #IS.NRZ, R4 ; CLR WRITE FLAG IN WORD 5
JSR PC, TALOAD ; LOAD DATA & CNTRL WORD 5
JSR PC, TALOAD ; LOAD DATA & CNTRL WORD 6
JSR PC, DIAEXE ; DO THE TS04 WRAP.

; CHECK FOR ERRORS

CLR ERRFLG ; INIT ERROR FLAG.
MOV #777, EXTRAC ; SHOULD BE DATA.
JSR PC, TKACER ; CHECK FOR ERRORS
LET ACTRK2 := COMP ACTRAC

MOV ACTRAC, A
COM ACTRK2

LET ACTRK2 := ACTRK2 CLR.BY #177000

BIC #177000.

IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN

TSTB ERRFLG
BEQ 50152\$
TSTB CMPFLG
BEQ 50152\$

ERRHRD 9, TAER2, TAEM

TRAP C\$ERRRD
.WORD 9
.WORD TAER2

```

4894 031770 007002                                     .WORD  TAEM
4895 031772                                     ENDIF
4896 031772                                     50152$:
4897
4898 031772                                     ENDSUB
4899 031772 L10050:
4900 031772 104403                                     TRAP  C$ESUB
4901
4902                                     ;THIS SUBTEST FORCES THE TRACK ACTIVE TO CLEAR BY WRITING THE FOLLOWING
4903                                     ;PATTERN ON EACH CHANNEL:
4904                                     ;
4905                                     ;   DATA:           111111
4906                                     ;   WRTFLG:         011100
4907                                     ;   TRACK ACTIVE:   SHOULD BE 0 FOR ALL TRACKS
4908
4909 031774                                     BGNSUB
4910 031774
4911 031774 104402                                     T4.3:                                     TRAP  C$BSUB
4912
4913                                     ;
4914                                     ;   INITIALIZATION
4915 031776 004737 016240                               JSR   PC,PEINIT
4916 032002 112745 000001                               MOVB  #FC.NRZ,-(R5)           ;NRZI MODE.
4917
4918                                     ;SET UP DATA AND CONTROL WORDS IN D.ABLK.
4919
4920 032006 012703 177777                               MOV   #-1,R3                ;SET UP DATA.
4921 032012 012704 000040                               MOV   #IS.NRZ,R4            ;SET UP CNTRL WORD.
4922 032016 004737 015712                               JSR   PC,TALOAD              ;LOAD DATA & CNTRL WORD 1
4923 032022 012704 000044                               MOV   #IS.NRZ!IS.WRF,R4     ;WORD 2 CNTRL
4924 032026 004737 015712                               JSR   PC,TALOAD              ;LOAD DATA - CNTRL WORD 2
4925 032032 004737 015712                               JSR   PC,TALOAD              ;LOAD DATA - CNTRL WORD 3
4926 032036 004737 015712                               JSR   PC,TALOAD              ;LOAD DATA - CNTRL WORD 4
4927 032042 012704 000040                               MOV   #IS.NRZ,R4            ;CNTRL WORD 5
4928 032046 004737 015712                               JSR   PC,TALOAD              ;LOAD DATA - CNTRL WORD 5
4929 032052 004737 015712                               JSR   PC,TALOAD              ;LOAD DATA - CNTRL WORD 6
4930 032056 004737 016006                               JSR   PC,DIAEXE              ;DO THE WRAP.
4931
4932                                     ;CHECK FOR ERRORS
4933
4934 032062 005037 002436                               CLR   ERRFLG                 ;INIT ERROR FLAG.
4935 032066 005037 002342                               CLR   EXTRAC                 ;SHOULD BE DATA=0.
4936 032072 004737 016130                               JSR   PC,TKACER              ;CHECK FOR ERRORS
4937 032076
4938 032076 013737 002360 002412                               LET  ACTRK3 := ACTRAC
4939 032104
4940 032104 105737 002436                               IFB  ERRFLG NE #0 ANDB  CMPFLG NE #0 THEN
4941 032110 001407
4942 032112 105737 002212
4943 032116 001404
4944 032120
4945 032120 104456
4946 032122 000010
4947 032124 003102
4948 032126 007002
4949 032130
ERRHRD  8,TAER1,TAEM
MOV     ACTRAC,A
TSTB   ERRFLG
BEQ    50153$
TSTB   CMPFLG
BEQ    50153$
TRAP   C$ERHRD
      .WORD  8
      .WORD  TAER1
      .WORD  TAEM
ENDIF

```

```

4950 032130
4951
4952 032130
4953 032130
4954 032130 104403
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964 032132
4965 032132
4966 032132 104402
4967
4968
4969 032134 012737 177777 002440
4970 032142 004737 016240
4971 032146 112745 000001
4972
4973
4974
4975 032152 005003
4976 032154 012704 000040
4977 032160 004737 015712
4978 032164 012704 000044
4979 032170 004737 015712
4980 032174 012703 177777
4981 032200 004737 015712
4982 032204 004737 015712
4983 032210 012704 000040
4984 032214 004737 015712
4985 032220 004737 015712
4986 032224 004737 016006
4987
4988
4989
4990 032230 005037 002436
4991 032234 012737 000777 002342
4992 032242 004737 016130
4993 032246
4994 032246 013737 002360 002414
4995 032254 005137 002414
4996 032260
4997 032260 042737 177000 002414
4998 032266
4999 032266 105737 002436
5000 032272 001407
5001 032274 105737 002212
5002 032300 001404
5003 032302
5004 032302 104456
5005 032304 000012

                                50153$:

                                ENDSUB
L10051:
                                TRAP C$ESUB

;THIS SUBTEST FORCES THE TRACK ACTIVE FLOP TO SET BY WRITING THE FOLLOWING
;PATTERN ON EACH CHANNEL:
;
;   DATA:           001111
;   WRNFLG:         011100
;   TRACK ACTIVE:   SHOULD BE 1 FOR ALL TRACKS.

                                BGNSUB
T4.4:
                                TRAP C$BSUB

;   INITIALIZATION.
MOV   #-1,T4S4           ;BTL SET THE TEST 4 SUBTEST 4 FLAG
JSR   PC,PEINIT
MOV   #FC.NRZ,-(R5)      ;NRZI MODE

;SET UP DATA AND CNTRL WORD IN DIABLK.
CLR   R3
MOV   #IS.NRZ,R4         ;WORD 1 DATA
JSR   PC,TALOAD          ;WORD 1 CNTRL - WRITE FLAG CLR
MOV   #IS.NRZ!IS.WRF,R4 ;LOAD DATA & CNTRL WORD 1
JSR   PC,TALOAD          ;WORD 2 CNTRL-SET WRT FLAG
MOV   #-1,R3            ;LOAD DATA & CNTRL WORD 2
JSR   PC,TALOAD          ;WORD 3 DATA-DO A DATA TRANSITION
JSR   PC,TALOAD          ;LOAD DATA & CNTRL WORD 3
JSR   PC,TALOAD          ;LOAD DATA & CNTRL WORD 4
MOV   #IS.NRZ,R4         ;LOAD DATA & CNTRL WORD 4
JSR   PC,TALOAD          ;CNTRL WORD 5 - CLR WRITE FLAG.
JSR   PC,TALOAD          ;LOAD DATA & CNTRL WORD 5
JSR   PC,TALOAD          ;LOAD DATA & CNTRL WORD 6
JSR   PC,DIAEXE         ;GO WRAP THE DATA.

;CHECK FOR ERRORS.
CLR   ERRFLG
MOV   #777,EXTRAC       ;INIT ERROR FLAG.
JSR   PC,TKACER         ;TRACK ACTIVE SHOULD BE SET FOR ALL TRAC
LET   ACTRK4 := COMP ACTRAC ;CHECK FOR ERRORS.

                                MOV   ACTRAC,A
                                COM   ACTRK4
LET   ACTRK4 := ACTRK4 CLR.BY #177000
                                BIC   #177000.
IFB  ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
                                TSTB  ERRFLG
                                BEQ   50154$
                                TSTB  CMPFLG
                                BEQ   50154$

ERRHRD  10,TAER2,TAEM
                                TRAP  C$ERHRD
                                .WORD 10

```

G10

HARDWARE TESTS MACY11 30(1046) 09-APR-84 14:40 PAGE 125
 CZTSIC.P11 09-APR-84 14:37 TEST 4: TRACK INACTIVE/ACTIVE TEST.

SEQ 0123

5006	032306	003152					
5007	032310	007002				.WORD	TAER2
5008	032312			ENDIF		.WORD	TAEM
5009	032312						
5010						50154\$:	
5011	032312			ENDSUB			
5012	032312		L10052:				
5013	032312	104403				TRAP	C\$ESUB
5014							
5015							
5016	032314			LET R1 := ACTRK1 OR ACTRK2 OR ACTRK3			
5017	032314	013701	002406			MOV	ACTRK1,R
5018	032320	053701	002410			BIS	ACTRK2,R
5019	032324	053701	002412			BIS	ACTRK3,R
5020	032330			LET R1 := R1 OR ACTRK4			
5021	032330	053701	002414			BIS	ACTRK4,R
5022	032334			IF R1 NE #0 THEN			
5023	032334	005701				TST	R1
5024	032336	001406				BEQ	50155\$
5025	032340			ERRHRD	11.TAERR		
5026	032340	104456				TRAP	C\$ERHRD
5027	032342	000013				.WORD	11
5028	032344	003222				.WORD	TAERR
5029	032346	000000				.WORD	0
5030	032350			CALL MSORT1			
5031	032350	004737	017640			JSR	PC,MSORT
5032	032354			ENDIF			
5033	032354					50155\$:	
5034							
5035				.EVEN			
5036							
5037	032354	005037	002440	CLR	T4S4		:BTL CLEAR THE TEST 4 SUBTEST 4 FLAG
5038							
5039	032360			ENDTST			
5040	032360						
5041	032360	104401	L10046:			TRAP	C\$ETST
5042							

```

5043 .SBTTL TEST 5: P.E. DATA TEST.
5044
5045 ;TEST 5 WRAPS A DATA PATTERN TO CHECK EACH
5046 ;TRACK FOR BIT PICKUPS AND DROPS.
5047
5048 ;
5049 ;REGISTER USAGE IS AS FOLLOWS:
5050 ; R2=PREAMBLE DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.
5051 ; R3=1ST BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8. THIS
5052 ; IS THE DATA OF INTEREST AFTER EXECUTING THE TSO4 DIA COMMAND.
5053 ; R4=2ND BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.
5054 032362
5055 032362
5056
5057 032362
5058 032362 004737 016032
5059 032366
5060 032366 004737 015260
5061 032372
5062 032372 005001
5063 032374 000402
5064 032376
5065 032376 062701 000002
5066 032402
5067 032402 020127 000012
5068 032406 003003
5069 032410
5070 032410 005061 002372
5071 032414
5072 032414 000770
5073 032416
5074
5075 032416
5076 032416
5077 032416 104402
5078
5079
5080
5081 032420 005037 002416
5082 032424 012702 000777
5083 032430 005003
5084 032432 004737 016212
5085 032436 004737 016006
5086
5087
5088
5089 032442 005037 002436
5090 032446 004737 016570
5091 032452
5092 032452 105737 002436
5093 032456 001407
5094 032460 105737 002212
5095 032464 001404
5096 032466
5097 032466 104456
5098 032470 000014

;TEST 5 WRAPS A DATA PATTERN TO CHECK EACH
;TRACK FOR BIT PICKUPS AND DROPS.
;REGISTER USAGE IS AS FOLLOWS:
; R2=PREAMBLE DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.
; R3=1ST BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8. THIS
; IS THE DATA OF INTEREST AFTER EXECUTING THE TSO4 DIA COMMAND.
; R4=2ND BYTE OF DATA FOR TRACKS 1-9 IN BIT POSITION 0-8.

T5::
    BGNTST
    CALL SCHEXE ;SET CHAR
    CALL WAITMT JSR PC,SCHEX
    INCR R1 FROM #0 TO #12 BY #2 JSR PC,WAITM
    CLR R1
    BR 50157$
    ADD #2,R1 50156$:
    CMP R1,#12
    BGT 50160$
    CLR ORDTBL(R)
    BR 50157$
    BR 50160$:

    LET ORDTBL(R1) := #0 ;CLEAR 6 LOC OF ORED WRAP REG TABLE
    ENDINC
    BGNSUB
T5.1:
    TRAP C$BSUB

;THIS SUBTEST WRAPS AN ALL 0'S PATTERN.
    CLR DTKIDN ;CLR DEAD TRK REG.
    MOV #777,R2 ;SETUP THE PREAMBLE DATA.
    CLR R3 ;BYTE 2 DATA. ALL 0'S.
    JSR PC,DATBLD ;BUILD THE DIAG BLK.
    JSR PC,DIAEXE ;EXECUTE IT.

;CHECK FOR ERRORS.
    CLR ERRFLG ;INIT ERROR FLAG.
    JSR PC,PEERCK ;CHECK FOR ERRORS.
    IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
        TSTB ERRFLG
        BEQ 50161$
        TSTB CMPFLG
        BEQ 50161$
        TRAP C$ERRRD
        .WORD 12

ERRHRD 12,DATER,SKDAEM

```

```

5099 032472 003245
5100 032474 007262
5101 032476
5102 032476
5103
5104 032476
5105 032476
5106 032476 104403
5107
5108 032500
5109 032500
5110 032500 104402
5111
5112
5113
5114 032502 005037 002416
5115 032506 012702 000777
5116 032512 012703 000777
5117 032516 004737 016212
5118 032522 004737 016006
5119
5120
5121
5122 032526 005037 002436
5123 032532 004737 016570
5124 032536
5125 032536 105737 002436
5126 032542 001407
5127 032544 105737 002212
5128 032550 001404
5129 032552
5130 032552 104456
5131 032554 000015
5132 032556 003245
5133 032560 007262
5134 032562
5135 032562
5136
5137 032562
5138 032562
5139 032562 104403
5140
5141 032564
5142 032564
5143 032564 104402
5144
5145
5146
5147
5148 032566 005037 002416
5149 032572 012702 000777
5150 032576 012703 000400
5151 032602 005037 002416
5152 032606 004737 016212
5153 032612 004737 016006
5154

```

```

                    ENDIF
                    50161$:
                    ENDSUB
L10054:
                    BGNSUB
T5.2:
                    TRAP C$ESUB
                    TRAP C$BSUB
;THIS SUBTEST WRAPS AN ALL 1'S PATTERN.
                    CLR DTKIDN ;CLR DEAD TRK IDN REG.
                    MOV #777,R2 ;LOAD PREAMBLE REG.
                    MOV #777,R3 ;BYTE 1 DATA.
                    JSR PC,DATBLD ;BUILD THE DIA BLK.
                    JSR PC,DIAEXE ;EXECUTE IT.
;CHECK FOR ERRORS.
                    CLR ERRFLG ;INIT ERROR FLAG.
                    JSR PC,PEERCK ;CHECK FOR ERRORS.
                    IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
                    TSTB ERRFLG
                    BEQ 50162$
                    TSTB CMPFLG
                    BEQ 50162$
                    TRAP C$ERHRD
                    .WORD 13
                    .WORD DATER
                    .WORD SKDAEM
                    ERRHRD 13,DATER,SKDAEM
                    50162$:
                    ENDSUB
L10055:
                    BGNSUB
T5.3:
                    TRAP C$ESUB
                    TRAP C$BSUB
;THIS SUBTEST RIPPLES A 1 IN A FIELD OF 0'S.
                    CLR DTKIDN ;CLR DEAD TRK IDN REG.
                    MOV #777,R2 ;SETUP THE PREAMBLE DATA IN R2 BITS 0-9.
                    MOV #400,R3 ;SETUP 1ST BYTE OF DATA.
                    CLR DTKIDN ;NO DEAD TRACKS DESIRED.
                    JSR PC,DATBLD ;BUILD THE DIAG BLOCK.
                    JSR PC,DIAEXE ;EXECUTE THE DATA WRAP
                    1$:

```


5211	032744				50164\$:	
5212						
5213						
5214						
5215				;IF NOT DONE, SHIFT THE 0 AND DO IT AGAIN.		
5216	032744	052703	001000	BIS #BIT9,R3		;PREPARE FOR SHIFT.
5217	032750	006203		ASR R3		;SHIFT THE DATA. HAS THE SHIFITNG 0
5218						;REACHED THE CARRY BIT YET?
5219	032752	103752		BCS 1\$;IF NOT, CONTINUE.
5220						
5221	032754			ENDSUB		
5222	032754			L10057:		
5223	032754	104403				TRAP C\$ESUB
5224						
5225						
5226	032756			LET R1 := ORORDY OR ORTRAC OR ORDATA		
5227	032756	013701	002372			MOV ORORDY,R
5228	032762	053701	002376			BIS ORTRAC,R
5229	032766	053701	002400			BIS ORDATA,R
5230	032772			LET R1 := R1 OR ORTRDD		
5231	032772	053701	002402			BIS ORTRDD,R
5232	032776			IF R1 NE #0 THEN		
5233	032776	005701				TST R1
5234	033000	001406				BEQ 50165\$
5235	033002			ERRHRD 30,DATER		
5236	033002	104456				TRAP C\$ERHRD
5237	033004	000036				.WORD 30
5238	033006	003245				.WORD DATER
5239	033010	000000				.WORD 0
5240	033012			CALL MSORT2		
5241	033012	004737	020334			JSR PC,MSORT
5242	033016			ENDIF		
5243	033016					50165\$:
5244						
5245	033016			EXIT TST		
5246	033016	104432				TRAP C\$EXIT
5247	033020	000002				.WORD L10053-.
5248				.EVEN		
5249						
5250	033022			ENDTST		
5251	033022			L10053:		
5252	033022	104401				TRAP C\$ETST


```

5253 .SBTTL TEST 6: P.E. SKEW TEST
5254
5255 ;THE NEXT TWO SUBTESTS SKEW THE DATA ON A TRACK BY ONE BYTE WITH RESPECT TO ALL
5256 ;THE OTHER TRACKS. THAT IS, THE DATA IS ONE BYTE LATE ON THE ONE TRACK.
5257 ;EACH TRACK IS TESTED FOR SKEW IN THIS MANNER. REGISTER ASSIGNMENTS
5258 ;ARE AS FOLLOWS:
5259 ;
5260 ; R2=PREAMBLE DATA
5261 ; R3=BYTE 1 DATA (WITH THE EXCEPTION OF THE SKEWED TRACK. THAT
5262 ; TRACK CONTAINS PREAMBLE DATA)
5263 ; R4=BYTE 2 DATA (WITH THE EXCEPTION OF THE SKEWED TRACK. THAT
5264 ; TRACK CONTAINS BYTE 1 DATA)
5265
5266 033024 BGNTST
5267 033024 T6::
5268
5269 033024 CALL SCHEXE JSR PC,SCHEX
5270 033024 004737 016032 CALL WAITMT JSR PC,WAITM
5271 033030 INCR R1 FROM #0 TO #12 BY #2 CLR R1
5272 033030 004737 015260 CLR BR 50166$
5273 033034 ADD #2,R1
5274 033034 005001 CLR R1,#12
5275 033036 000402 BGT 50170$
5276 033040 062701 000002 LET ORDTBL(R1) := #0 ;CLEAR 6 LOC OF ORED WRAP REG TABLES
5277 033040 020127 000012 ENDINC CLR ORDTBL(R
5278 033044 BR 50167$
5279 033044 003003 BR 50170$
5280 033050 LET ORDTBL(R1) := #0 ;CLEAR 6 LOC OF ORED WRAP REG TABLES
5281 033052 ENDINC CLR ORDTBL(R
5282 033052 005061 002372 BR 50167$
5283 033056 BR 50170$
5284 033056 000770 BR 50167$
5285 033060 BR 50170$
5286
5287 033060 BGNSUB
5288 033060 T6.1:
5289 033060 104402 TRAP C$BSUB
5290
5291 ;THIS SUBTEST WRITES AN ALL 1'S PREAMBLE (SKEWED), AN ALL 0'S
5292 ;BYTE 1 DATA (SKEWED), AND AN ALL 1'S BYTE 2 DATA (SKEWED).
5293
5294 033062 005037 002416 CLR DTKIDN ;CLR DEAD TRK IDN REG.
5295 033066 012702 000377 MOV #377,R2 ;SET UP PREAMBLE DATA. START BY SKEWING TRK 9.
5296 033072 004737 016240 JSR PC,PEINIT ;SET WRAP TASK ADDR IN DIABLK AND INIT INDEX.
5297 033076 112745 000012 MOV #FC.DAT!FC.VCO,-(R5) ;SET DATA & VCO MODE IN FORMAT CNTRL.
5298 033102 010203 MOV R2,R3 ;SETUP BYTE 1 DATA
5299 033104 005103 COM R3 ;=0
5300 033106 042703 177000 BIC #177000,R3 ;CLR GARBAGE BITS
5301 033112 010204 MOV R2,R4 ;SETUP BYTE 2 DATA=1.
5302 033114 004737 016262 JSR PC,PEDATA ;LOAD DATA IN DIABLK
5303 033120 005037 002344 CLR EXDATA ;THE EXPECTED DESKEWED DATA IN EXDATA IS 0
5304 033124 012737 000777 002350 MOV #777,EXODTR ;THE EXPECTED DESKEWED DATA IN EXODTR IS
5305 ;1 FOR EACH TRACK.
5306 033132 005037 002340 CLR EX1DTR ;THE EXPECTED DESKEWED DATA IN EX1DTR IS 0
5307 033136 004737 016006 JSR PC,DIAEXE ;EXECUTE THE DATA WRAP.
5308

```

```

5309                                     ;CHECK FOR ERRORS AND REPORT IF ANY
5310
5311 033142 005037 002436                CLR     ERRFLG           ;INIT ERROR FLAG.
5312 033146 004737 016570                JSR     PC,PEERCK       ;CHECK FOR ERRORS
5313 033152                                     IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
5314 033152 105737 002436                                     TSTB   ERRFLG
5315 033156 001407                                     BEQ    50171$
5316 033160 105737 002212                                     TSTB   CMPFLG
5317 033164 001404                                     BEQ    50171$
5318 033166                                     ERRHRD  16,SKEWER,SKDAEM
5319 033166 104456                                     TRAP   C$ERHRD
5320 033170 000020                                     .WORD  16
5321 033172 003272                                     .WORD  SKEWER
5322 033174 007262                                     .WORD  SKDAEM
5323 033176                                     ENDIF
5324 033176                                     50171$:
5325
5326                                     ;SHIFT THE SKEW TO THE NEXT TRACK AND REPEAT.
5327
5328 033176 052702 001000                BIS     #BIT9,R2        ;PREPARE FOR SHIFT OF SKEWED TRK.
5329 033202 006202                                     ASR     R2              ;SHIFT THE PREAMBLE DATA. HAVE ALL THE
5330                                     ;TRACKS BEEN SKEWED?
5331 033204 103732                                     BCS    1$              ;BR IF NOT. CONTINUE.
5332
5333 033206                                     ENDSUB
5334 033206                                     L10061:
5335 033206 104403                                     TRAP   C$ESUB
5336
5337 033210                                     BGNSUB
5338 033210                                     T6.2:
5339 033210 104402                                     TRAP   C$BSUB
5340
5341                                     ;THIS SUBTEST SENDS AN ALL 1'S PREAMBLE (SKEWED), AN ALL 1'S
5342                                     ;BYTE 1 DATA (SKEWED) AND AN ALL 0'S BYTE 2 DATA (SKEWED).
5343
5344 033212 005037 002416                CLR     DTKIDN          ;CLR DEAD TRK IDEN REG.
5345 033216 012702 000377                MOV     #377,R2        ;SET UP PREAMBLE DATA. START WITH CHAN 9 SKEWED
5346 033222 004737 016240                JSR     PC,PEINIT      ;SET WRAP TASK ADR IN DIABLK & INIT INDEX.
5347 033226 112745 000012                MOV     #FC.DAT!FC.VCO, -(R5) ;SET DATA & VCO MODE IN FMT CNTRL.
5348 033232 012703 000777                MOV     #777,R3        ;SETUP BYTE 1 DATA=1
5349 033236 010204                MOV     R2,R4          ;SETUP BYTE 2 DATA
5350 033240 005104                COM     R4              ;=0
5351 033242 042704 177000                BIC     #177000,R4     ;CLR GARBAGE BITS
5352 033246 004737 016262                JSR     PC,PEDATA      ;LOAD THE DATA IN DIABLK
5353 033252 012737 000777 002344                MOV     #777,EXDATA    ;THE EXPECTED DESKEWED DATA IS 1
5354 033260 005037 002350                CLR     EXODTR         ;THE EXPECTED "0 OR DEAD" DATA=1 FOR EACH TRACK.
5355 033264 012737 000777 002340                MOV     #777,EX1DTR    ;THE EXPECTED DESKEWED "1 OR DEAD" DATA=1 FOR AL
5356 033272 004737 016006                JSR     PC,DIAEXE      ;EXECUTE THE WRAP.
5357
5358                                     ;CHECK FOR ERRORS
5359
5360 033276 005037 002436                CLR     ERRFLG           ;INIT ERROR FLAG.
5361 033302 004737 016570                JSR     PC,PEERCK       ;CHECK FOR ERRORS.
5362 033306                                     IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN
5363 033306 105737 002436                                     TSTB   ERRFLG
5364 033312 001407                                     BEQ    50172$

```

5365	033314	105737	002212						
5366	033320	001404						TSTB	CMPFLG
5367	033322							BEQ	50172\$
5368	033322	104456		ERRHRD	16,SKEWER,SKDAEM				
5369	033324	000020						TRAP	C\$ERHRD
5370	033326	003272						.WORD	16
5371	033330	007262						.WORD	SKEWER
5372	033332			ENDIF				.WORD	SKDAEM
5373	033332								
5374									50172\$:
5375									
5376									;SHIFT THE SKEW TO THE NEXT TRACK AND REPEAT.
5377	033332	052702	001000	BIS	#BIT9,R2				;PREPARE TO SHIFT SKEWED TRK.
5378	033336	006202		ASR	R2				;SKEW THE NEXT TRACK. ARE WE DONE YET?
5379	033340	103730		BCS	1\$;BR IF NOT.
5380									
5381	033342			ENDSUB					
5382	033342			L10062:					
5383	033342	104403							
5384								TRAP	C\$ESUB
5385	033344			LET R1 :=	ORORDY OR ORTRAC OR ORDATA				
5386	033344	013701	002372					MOV	ORORDY,R
5387	033350	053701	002376					BIS	ORTRAC,R
5388	033354	053701	002400					BIS	ORDATA,R
5389	033360			LET R1 :=	R1 OR ORTRDD				
5390	033360	053701	002402					BIS	ORTRDD,R
5391	033364			IF R1 NE	#0 THEN				
5392	033364	005701						TST	R1
5393	033366	001406						BEQ	50173\$
5394	033370			ERRHRD	16,SKEWER				
5395	033370	104456						TRAP	C\$ERHRD
5396	033372	000020						.WORD	16
5397	033374	003272						.WORD	SKEWER
5398	033376	000000						.WORD	0
5399	033400			CALL	MSORT2				
5400	033400	004737	020334					JSR	PC,MSORT
5401	033404			ENDIF					
5402	033404								50173\$:
5403									
5404									.EVEN
5405									
5406									
5407	033404								ENDTST
5408	033404			L10060:					
5409	033404	104401						TRAP	C\$ETST

5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425 033406
5426 033406
5427
5428 033406
5429 033406 004737 016032
5430 033412
5431 033412 004737 015260
5432 033416
5433 033416 005001
5434 033420 000402
5435 033422
5436 033422 062701 000002
5437 033426
5438 033426 020127 000012
5439 033432 003003
5440 033434
5441 033434 005061 002372
5442 033440
5443 033440 000770
5444 033442
5445
5446 033442
5447 033442
5448 033442 104402
5449
5450
5451
5452 033444 012702 000777
5453 033450 012703 000777
5454 033454 005004
5455 033456 012737 000400 002416
5456 033464 004737 016240
5457 033470 112745 000012
5458 033474 004737 016262
5459 033500 004737 016006
5460
5461
5462
5463 033504 005037 002436
5464 033510 004737 016570
5465 033514

.SBTTL TEST 7: P.E. DEAD TRACK TEST.

;TEST 7 CHECKS THE DEAD TRACK LOGIC BY RIPPLING A DEAD TRACK THRU
;A FIELD OF LIVE TRACKS AND ONE LIVE TRACK THRU A FIELD OF DEAD
;TRACKS. ADDITIONALLY, EACH SUBTEST WILL SEND 1'S DATA AND 0'S
;DATA IN ORDER TO TEST THE 1'S OR DEAD REGISTER AND THE
;0'S OR DEAD REGISTER.

;REGISTER USAGE:

; R2=PREAMBLE ALL 1'S CHARACTER
; R3=1ST DATA BYTE (BITS 0-8)
; R4=2ND DATA BYTE (BITS 0-8)

; DTKIDN=DEAD TRACK DEFINED IN BITS 0-8 (0=LIVE TRK; 1=DEAD TRK.)

BGNTST

T7::

CALL SCHEXE

JSR PC,SCHEX

CALL WAITMT

JSR PC,WAITM

INCR R1 FROM #0 TO #12 BY #2

CLR R1

BR 50174#

50175#:

ADU #2,R1

50174#:

CMP R1,#12

BGT 50176#

LET ORDTBL(R1) := #0

;CLEAR 6 LOC OF ORED WRAP REG TABLES

CLR ORDTBL(R

ENDINC

BR 50175#

50176#:

BGNSUB

T7.1:

TRAP C#BSUB

;THIS SUBTEST RIPPLES A DEAD TRACK IN A FIELD OF LIVE TRACKS.

MOV #777,R2

;PREAMBLE

MOV #777,R3

;BYTE 1 DATA.

CLR R4

;BYTE 2 DATA.

MOV #400,DTKIDN

;SETUP TO SHIFT 1 DEAD TRACK.

JSR PC,PEINIT

;SET DATA AND VCO MODE IN FMT CONTROL.

MOVB #FC,DAT!FC.VCO, (R5)

JSR PC,PEDATA

;LOAD DATA IN DIABLK AND CALCULATE EXPECT

JSR PC,DIAEXE

;EXECUTE IT.

;CHECK FOR ERRORS AND REPORT IF ANY.

CLR ERRFLG

;INIT ERROR FLAG.

JSR PC,PEERCK

;CHECK FOR ERRORS.

IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN

```

5466 033514 105737 002436
5467 033520 001407
5468 033522 105737 002212
5469 033526 001404
5470 033530
5471 033530 104456
5472 033532 000021
5473 033534 003312
5474 033536 007262
5475 033540
5476 033540
5477
5478
5479
5480 033540 000241
5481 033542 006237 002416
5482 033546 103346
5483
5484
5485
5486 033550 105703
5487 033552 001404
5488 033554 005003
5489 033556 012704 000777
5490 033562 000735
5491 033564
5492 033564 104432
5493 033566 000002
5494
5495 033570
5496 033570
5497 033570 104403
5498
5499 033572
5500 033572
5501 033572 104402
5502
5503
5504
5505 033574 012702 000777
5506 033600 012703 000777
5507 033604 005004
5508 033606 012737 000377 002416
5509 033614 004737 016240
5510 033620 112745 000012
5511 033624 004737 016262
5512 033630 004737 016006
5513
5514
5515
5516 033634 005037 002436
5517 033640 004737 016570
5518 033644
5519 033644 105737 002436
5520 033650 001407
5521 033652 105737 002212

ERRHRD 17,DDER,SKDAEM

ENDIF

;SHIFT THE DEAD TRACK AND REPEAT.

CLC
ASR DTKIDN
BCC 2$

;CLR CARRY TO PREPARE FOR SHIFT.
;SHIFT DEAD TRACK
;BR IF ALL TRACKS NOT KILLED YET.

;CHANGE THE DATA AND REPEAT.

TSTB R3
BEQ 4$
CLR R3
MOV #777,R4
BR 1$
EXIT SUB

;DONE?
;BR IF SO.
;BYTE 1 DATA.
;BYTE 2 DATA.
;REPEAT TEST WITH COMPLEMENTED DATA.

TRAP C$EXIT
.WORD L10064-.

ENDSUB
L10064:

BGNSUB
T7.2:

;THIS SUBTEST RIPPLES A LIVE TRACK IN A FIELD OF DEAD TRACKS.

MOV #777,R2
MOV #777,R3
CLR R4
MOV #377,DTKIDN
JSR PC,PEINIT
MOV #FC.DAT!FC.VCO,-(R5)
JSR PC,PEDATA
JSR PC,DIAEXE

;PREAMBLE.
;BYTE 1 DATA.
;BYTE 2 DATA.
;SETUP TO SHIFT 1 LIVE TRACK.
;SET WRAP TSK ADR AND DIABLK INDEX.
;SET DATA AND VCO MODE IN FMT CNTRL.
;LOAD DIABLD AND GENERATE EXPECTED DATA.
;DO IT.

;CHECK FOR ERRORS.

CLR ERRFLG
JSR PC,PEERCK
IFB ERRFLG NE #0 ANDB CMPFLG NE #0 THEN

TRAP C$ERRHRD
.WORD 17
.WORD DDER
.WORD SKDAEM

50177$:

TSTB ERRFLG
BEQ 50177$
TSTB CMPFLG

```



```

5578 .SBTTL TEST 8: LOOKUP TABLE TEST
5579
5580 ;THIS TEST VEFIFIES THAT THE CONTENTS OF THE ROM LOOKUP TABLE ARE CORRECT.
5581 ;THE ROM CONTENTS IN ADDRESS 0-1777 ARE CHECKED.
5582
5583 ;
5584 ; DATA & REGISTER USAGE:
5585 ;
5586 ; ROMLKI = ROM LOOKUP TABLE ADDRESS.
5587 ; ERRFLG = ERROR FLAG.
5588 ; R5 = DIABLK INDEX.
5589 ;
5590 033770 BGNTST
5591 033770 T8::
5592
5593 ;OPEN A ROM LOOKUP TABLE LOCATION.
5594
5595 033770 004737 016032 JSR PC,SCHEXE ;EXECUTE AN SCH COMMAND.
5596 033774 CALL WAITMT
5597 033774 004737 015260 JSR PC,WAITM
5598 034000 012737 001777 002420 MOV #1777,ROMLKI ;INIT THE ROM TABLE ADDRESS
5599 034006 004737 016240 1$: JSR PC,PEINIT ;SETUP TASK ADR + READ CNTRL.
5600 034012 112745 000003 MOVB #FC.VCO!FC.NRZ,-(R5) ;SET VCO + NRZ1 MODE IN FORMAT CNTRL.
5601 034016 004737 017006 JSR PC,ROMLOK ;LOAD THE DIABLK.
5602 034022 004737 016006 JSR PC,DIAEXE ;EXECUTE IT.
5603
5604 ;CHECK FOR VALID DATA IN THAT LOCATION.
5605
5606 034026 005037 002436 CLR ERRFLG ;INIT ERROR FLAG
5607 034032 004737 017106 JSR PC,ROMEX ;GET EXPECTED DATA
5608 034036 004737 017614 JSR PC,ROMCK ;GET ACTUAL DATA AND COMPARE WITH
5609 ;EXPECTED.
5610 034042 IFB ERRFLG NE #0 THEN
5611 034042 105737 002436 TSTB ERRFLG
5612 034046 001411 BEQ 50202$
5613 034050 105237 002437 INCB CTLFLG
5614 034054 IFB CMPFLG NE #0 THEN
5615 034054 105737 002212 TSTB CMPFLG
5616 034060 001404 BEQ 50203$
5617 034062 ERRHRD 18,ROMER,ROMEM
5618 034062 104456 TRAP C$ERHRD
5619 034064 000022 .WORD 18
5620 034066 003375 .WORD ROMER
5621 034070 010056 .WORD ROMEM
5622 034072 ENDF
5623 034072 50203$:
5624 034072 ENDF
5625 034072 50202$:
5626
5627 ;UPDATE THE ADDRESS AND REPEAT THE TEST UNTIL DONE.
5628
5629 034072 005337 002420 DEC ROMLKI ;DECR ROM ADDRESS - DONE?
5630 034076 100343 BPL 1$ ;BR IF NOT.
5631 034100 IFB CTLFLG NE #0 THEN
5632 034100 105737 002437 TSTB CTLFLG
5633 034104 001404 BEQ 50204$

```

5634 034106
 5635 034106 104456
 5636 034110 000022
 5637 034112 003375
 5638 034114 014226
 5639 034116
 5640 034116
 5641
 5642
 5643
 5644 034116
 5645 034116 104432
 5646 034120 000002
 5647
 5648
 5649 034122
 5650 034122
 5651 034122 104401

ERRMRD 18,ROMER,ROMLER

TRAP C\$ERMRD
 .WORD 18
 .WORD ROMER
 .WORD ROMLER

ENDIF

50204\$:

EXIT TST

TRAP C\$EXIT
 .WORD L10066-

.EVEN

ENDTST

L10066:

TRAP C\$ETST

5652								
5653								
5654								
5655	034124							
5656	034124							
5657								
5658	034124							
5659	034124	004737	016032					
5660	034130							
5661	034130	004737	015260					
5662	034134							
5663	034134	012727	000310					
5664	034140	000000						
5665	034142	013727	002116					
5666	034146	000000						
5667	034150	005367	177772					
5668	034154	001375						
5669	034156	005367	177756					
5670	034162	001367						
5671	034164							
5672	034164	004737	021022					
5673								
5674	034170							
5675	034170							
5676	034170	104401						

.SBTTL TEST 9: IN-LINE MICRO DIAGNOSTIC TEST

T9::

BGNTST

CALL SCHEXE

JSR PC.SCHEX

CALL WAITMT

JSR PC.WAITM

DELAY 200.

MOV #200.,(P

.WORD 0

MOV L\$DLY,(P

.WORD 0

DEC -6(PC)

BNE -.4

DEC -22(PC)

BNE -.20

CALL MSORT3

JSR PC.MSORT

ENDTST

L10067:

TRAP C\$ETST

H11

```
5677
5678
5679
5680 034172
5681 034172
5682
5683 034172 005077 146026
5684 034176
5685 034176 004737 014702
5686 034202
5687 034202 004737 016032
5688 034206
5689 034206 004737 021022
5690
5691
5692 034212
5693 034212
5694 034212 104401
5695
5696 034214
5697

          .SBTTL TEST 10: INIT MICRO DIAGNOSTIC TEST
          BGNTST
          T10::
          CLR      @TSSR
          CALL     WAITSR
          CALL     SCHEXE
          CALL     MSORT3
          JSR      PC.WAITS
          JSR      PC.SCHEX
          JSR      PC.MSORT
          .EVEN
          ENDTST
          L10070:
          TRAP     C$ETST
          ENDMOD
```

```

5698 .TITLE PARAMETER CODING
5699
5700 .SBTTL HARDWARE PARAMETER CODING SECTION
5701
5702 034214 BGNMOD
5703
5704
5705 : **
5706 : THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
5707 : THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
5708 : MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5709 : INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
5710 : MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
5711 : WITH THE OPERATOR.
5712 : **
5713 034214 BGNHRD
5714 034214 000024 .WORD L10071-L$H
5715 034216 L$HRD::
5716
5717 034216 GPRMA TS4ADR,0,0,160002,177564,YES
5718 034216 000031 .WORD T$CODE
5719 034220 034242 .WORD TS4ADR
5720 034222 160002 .WORD T$LLOLIM
5721 034224 177564 .WORD T$HILIM
5722 034226 GPRMD TS4VCT,2,0,777,60,776,YES
5723 034226 001032 .WORD T$CODE
5724 034230 034257 .WORD TS4VCT
5725 034232 000777 .WORD 777
5726 034234 000060 .WORD T$LLOLIM
5727 034236 000776 .WORD T$HILIM
5728
5729 034240 EXIT HRD
5730 034240 013004 .WORD T$CODE
5731
5732 034242 051524 051123 040440 .NLIST BEX
034257 126 041505 047524 TS4ADR: .ASCIZ /TSSR ADDRESS/
TS4VCT: .ASCIZ /VECTOR/
.LIST BEX
.EVEN
5733
5734
5735 034266 ENDHRD
5736
5737 034266 L10071: .EVEN

```

```

5738          .SBTTL  SOFTWARE PARAMETER CODING SECTION
5739
5740          ;**
5741          ;IT CONTAINS MACRO THAT ARE USED BY THE SUPERVISOR
5742          ;TO BUILD SOFT P-YABLES.  THE MACROS ARE INTERPRETED
5743          ;BY THE SUPERVISOR AS DATA STRUCTURES.  THE MACROS
5744          ;ALLOW THE SUPERVISOR TO COMMUNICATE WITH THE OPERATOR.
5745          ;--
5746
5747 034266          BGNSFT
5748 034266 000034          .WORD  L10072-L$S
5749 034270          L$SOFT::
5750
5751 034270          GPRML  ENBMSG,0,1,YES
5752 034270 000130          .WORD  T$CODE
5753 034272 034300          .WORD  ENBMSG
5754 034274 000001          .WORD  1
5755
5756 034276          EXIT SFT
5757 034276 031004          .WORD  T$CODE
5758
5759 034300 047105 041101 042514  ENBMSG: .NLIST BEX
          .ASCIZ  /ENABLE DATA COMPARE ERROR PRINTS FOR TESTS 4-7/
          .LIST  BEX
5760
5761          034360          .EVEN
5762
5763 034360          ENDSFT
5764
5765 034360          L10072:          .EVEN
5766
5767          ;*****
5768          ;*****
5769          ;          PATCH AREA
5770
5771 034360 000140          PATCH:: .BLKW  96.
5772          ;*****
5773          ;*****
5774 034660          LASTAD
5775
5776 034660 034674          .EVEN
5777 034662 000004          .WORD  T$FREE
5778 034664          .WORD  T$SIZE
5779
5780 034664          L$LAST::
5781
5782          ENDMOD
5783
5784          .SBTTL  HARD CODED P-TABLE
5785
5786          ;**
5787          ;DIAG  IS PRE-PARAMETERIZED PER TBL
5788          ;--
5789          BGNSETUP 1
5790          BGNPTAB
5791
          .WORD  0
          .WORD  L10075-.

```

K11

PARAMETER CODING MACY11 30(1046) 09-APR-84 14:40 PAGE 142
CZTSIC.P11 09-APR-84 14:37 HARD CODED P-TABLE

SEQ 0140

5792	034670		L10073:	
5793	034670	172522		172522
5794	034672	000224		224
5795	034674			ENDPTAB
5796	034674		L10075:	
5797	034674			ENDSETUP
5798				
5799	000001		.END	

ACDATA	002362	G	BPE	=	020000	G	COD16B	025137	C\$ESEG	=	000005	DIAPKT	002250	G		
ACK.C	=	100000	G	CH.EAI	=	000040	G	COD16C	025221	C\$ESUB	=	000003	DROPU	014654	G	
ACORDY	002354	G	CH.ERI	=	000020	G	COD17	025301	C\$ETST	=	000001	DTKIDN	002416	G		
ACROML	002370	G	CH.ESS	=	000200	G	COD17A	025405	C\$EXIT	=	000032	DTRCHK	020452			
ACTRAC	002360	G	CH9.DA	=	000010	G	COD17B	025464	C\$GETB	=	000026	EEM1A	006206			
ACTRDD	002364	G	CH9.OR	=	000001	G	COD20	025544	C\$GETW	=	000027	EEM2A	006362			
ACTRK1	002406	G	CH9.TA	=	000004	G	COD20A	025652	C\$GMAN	=	000043	EEM3A	006534			
ACTRK2	002410	G	CH9.TD	=	000020	G	COD21	025752	C\$GPHR	=	000042	EEM4A	006706			
ACTRK3	002412	G	CH9.OD	=	000040	G	COD21A	026060	C\$GPLO	=	000030	EF.CON	=	000036	G	
ACTRK4	002414	G	CH9.ID	=	000002	G	COD22	026106	C\$GPRI	=	000040	EF.NEW	=	000035	G	
ACTTBL	002354	G	CMDPKT	002230	G	COD23	026143	C\$INIT	=	000011	EF.PWR	=	000034	G		
ACODTR	002366	G	CMD.CO	=	000001	G	COD24	026200	C\$INLP	=	000020	EF.RES	=	000037	G	
AC1DTR	002356	G	CMD.C1	=	000002	G	COD24A	026306	C\$MANI	=	000050	EF.STA	=	000040	G	
ADR	=	000020	G	CMD.C2	=	000004	G	COD25	026344	C\$MEM	=	000031	ENBMSG	034300		
ASSEMB	=	000010	G	CMD.C3	=	000010	G	COD25A	026452	C\$MSG	=	000023	ENDTBL	=	177777	G
AUTJDM	030064		G	CMD.C4	=	000020	G	COD25B	026526	C\$OPEN	=	000034	ERRFLG	002436	G	
A0	=	000001	G	CMPFLG	002212	G	COD26	026623	C\$PNTB	=	000014	EVL	=	000004	G	
A1	=	000002	G	CODTBL	021314		COD26A	026706	C\$PNTF	=	000017	EVNPAR	015644			
A2	=	000004	G	CODXXX	027336		COD26B	026762	C\$PNTS	=	000016	EXDATA	002344	G		
A3	=	000010	G	COD00	021474		COD27	027057	C\$PNTX	=	000015	EXORDY	002336	G		
A4	=	000020	G	COD00A	021602		COD27A	027165	C\$QIO	=	000377	EXPTBL	002336	G		
A5	=	000040	G	COD00B	021667		COD27B	027241	C\$RDBU	=	000007	EXROML	002352	G		
A6	=	000100	G	COD01	021751		COD337	027374	C\$REFG	=	000047	EXTRAC	002342	G		
A7	=	000200	G	COD01A	022050		COUNTE	=	050024	C\$RESE	=	000033	EXTRDD	002346	G	
A8	=	000400	G	COD02	022072		CP\$ADH	=	000004	C\$REVI	=	000003	EXODTR	002350	G	
A9	=	001000	G	COD03	022165		CP\$ADL	=	000002	C\$RFLA	=	000021	EX1DTR	002340	G	
BIT0	=	000001	G	COD03A	022273		CP\$CMD	=	000000	C\$RPT	=	000025	E\$END	=	002100	
BIT00	=	000001	G	COD04	022351		CP\$CNT	=	000006	C\$SEFG	=	000046	E\$LOAD	=	000035	
BIT01	=	000002	G	COD04A	022457		CRMSG	005765	C\$SPRI	=	000041	FCMASK	=	177717	G	
BIT02	=	000004	G	COD05	022506		CTLFLG	002437	C\$SVEC	=	000037	FC.DAT	=	000010	G	
BIT03	=	000010	G	COD06	022605		CVC.C	=	040000	C\$TPRI	=	000013	FC.FLO	=	000100	G
BIT04	=	000020	G	COD07	022702		C\$AU	=	000052	C18\$DA	=	000010	FC.NRZ	=	000001	G
BIT05	=	000040	G	COD10	022773		C\$AUTO	=	000061	C18\$OR	=	000004	FC.PRE	=	000004	G
BIT06	=	000100	G	COD10A	023101		C\$BRK	=	000022	C18\$TA	=	000007	FC.RD	=	000200	G
BIT07	=	000200	G	COD10B	023203		C\$BSEG	=	000004	C18\$TD	=	000011	FC.VCO	=	000002	G
BIT08	=	000400	G	COD11	023234		C\$BSUB	=	000002	C18\$OD	=	000012	FIXDIA	015756	G	
BIT09	=	001000	G	COD11A	023317		C\$CEFG	=	000045	C18\$1D	=	000005	FMCTLO	=	000004	G
BIT1	=	000002	G	COD11B	023403		C\$CLCK	=	000062	DAEM	007472		FMTCH4	005330	G	
BIT10	=	002000	G	COD12	023503		C\$CLEA	=	000012	DAEMA	007545		FMTCH5	005253	G	
BIT11	=	004000	G	COD12A	023606		C\$CLOS	=	000035	DAEMB	007614		FMTCH6	005176	G	
BIT12	=	010000	G	COD13	023666		C\$CLP1	=	000006	DAEMC	007703		FMTCTR	005064	G	
BIT13	=	020000	G	COD13A	023765		C\$CVEC	=	000036	DAEMD	007765		FMTSEL	020556		
BIT14	=	040000	G	COD13B	024037		C\$DCLN	=	000044	DASKDD	003340	G	FMT.CO	=	000040	G
BIT15	=	100000	G	COD14	024117		C\$DODU	=	000051	DATBLD	016212	G	FMT.C1	=	000100	G
BIT2	=	000004	G	COD14A	024221		C\$DRPT	=	000024	DATER	003245	G	F\$AU	=	000015	
BIT3	=	000010	G	COD14B	024301		C\$DU	=	000053	DDER	003312	G	F\$AUTO	=	000020	
BIT4	=	000020	G	COD14C	024354		C\$EDIT	=	000003	DESCM	002442	G	F\$BGN	=	000040	
BIT5	=	000040	G	COD15	024434		C\$ERDF	=	000055	DFPTBL	002204	G	F\$CLEA	=	000007	
BIT6	=	000100	G	COD15A	024535		C\$ERHR	=	000056	DIA	=	100006	F\$DU	=	000016	
BIT7	=	000200	G	COD15B	024621		C\$ERRO	=	000060	DIABLK	002316	G	F\$END	=	000041	
BIT8	=	000400	G	COD15C	024675		C\$ERSF	=	000054	DIAEXE	016006	G	F\$HARD	=	000004	
BIT9	=	001000	G	COD16	024755		C\$ERSO	=	000057	DIAEXT	=	000020	F\$HW	=	000013	
BOE	=	000400	G	COD16A	025057		C\$ESCA	=	000010	DIAGMC	=	000000	F\$INIT	=	000006	

F\$JMP = 000050	I\$DU = 000041	L\$LADP 002026 G	L10042 030744	OPP.C = 020000 G
F\$MOD = 000000	I\$HRD = 000041	L\$LAST 034664 G	L10043 031452	ORDATA 002400 G
F\$MSG = 000011	I\$INIT= 000041	L\$LOAD 002100 G	L10044 031024	ORDTBL 002372 G
F\$PROT= 000021	I\$MOD = 000041	L\$LUN 002074 G	L10045 031324	ORORDY 002372 G
F\$PWR = 000017	I\$MSG = 000041	L\$MREV 002050 G	L10046 032360	ORTRAC 002376 G
F\$RPT = 000012	I\$PROT= 000040	L\$NAME 002000 G	L10047 031616	ORTRDD 002402 G
F\$SEG = 000003	I\$PTAB= 000041	L\$PRIO 002042 G	L10050 031772	ORODTR 002404 G
F\$SOFT= 000005	I\$PWR = 000041	L\$PROT 027426 G	L10051 032130	OR1DTR 002374 G
F\$SRV = 000010	I\$RPT = 000041	L\$PRT 002112 G	L10052 032312	O\$APTS= 000000
F\$SUB = 000002	I\$SEG = 000041	L\$REPP 002062 G	L10053 033022	O\$AU = 000001
F\$SW = 000014	I\$SETU= 000041	L\$REV 002010 G	L10054 032476	O\$BGNR= 000000
F\$TEST= 000001	I\$SFT = 000041	L\$SOFT 034270 G	L10055 032562	O\$BGNS= 000001
GES = 100017 G	I\$SRV = 000041	L\$SPC 002056 G	L10056 032660	O\$DU = 000001
G\$CNT0= 000200	I\$SUB = 000041	L\$SPCP 002020 G	L10057 032754	O\$ERRT= 000000
G\$DELM= 000372	I\$TST = 000041	L\$SPTP 002024 G	L10060 033404	O\$GNSW= 000001
G\$DISP= 000003	J\$JMP = 000167	L\$STA 002030 G	L10061 033206	O\$POIN= 000001
G\$EXCP= 000400	LOE = 040000 G	L\$SW 002212 G	L10062 033342	O\$SETU= 000001
G\$HILI= 000002	LONMSG 006060 G	L\$TEST 002114 G	L10063 033766	PAIRST 020510
G\$LOLI= 000001	LOOKTB 021234	L\$TIML 002014 G	L10064 033570	PARGEN 015650
G\$NO = 000000	LOT = 000010 G	L\$UNIT 002012 G	L10065 033724	PATCH 034360 G
G\$OFFS= 000400	LUNIT 027712	L10000 002210	L10066 034122	PCHDR 011614
G\$OFSI= 000376	L\$ACP 002110 G	L10001 002214	L10067 034170	PCHDRA 011710
G\$PRMA= 000001	L\$APT 002036 G	L10002 006126	L10070 034212	PCHDRB 011760
G\$PRMD= 000002	L\$AU 030230 G	L10003 006302	L10071 034266	PCHLTB 011522
G\$PRML= 000000	L\$AUT 002070 G	L10004 006454	L10072 034360	PCHTAD 011516
G\$RADA= 000140	L\$AUTO 027716 G	L10005 006626	L10073 034670	PCHT50 012022
G\$RADB= 000000	L\$CCP 002106 G	L10006 007000	L10075 034674	PCHT51 012101
G\$RADD= 000040	L\$CLEA 030170 G	L10007 007260	MICERA 014314	PCHT52 012143
G\$RADL= 000120	L\$CO 002032 G	L10010 010054	MICERB 014370	PCHT53 012217
G\$RADO= 000020	L\$DEPO 002011 G	L10011 010224	MICERR 014250 G	PCHT54 012301
G\$XFER= 000004	L\$DESC 002150 G	L10012 010414	MICROE 003563 G	PCHT55 012363
G\$YES = 000010	L\$DESP 002076 G	L10013 010530	MODUER 002502 G	PCHT56 012426
HELP = 000000	L\$DEVP 002060 G	L10014 010624	MOD.CO= 000400 G	PCHT57 012467
HOE = 100000 G	L\$DISP 002124 G	L10015 010770	MOD.C1= 001000 G	PCHT60 012530
IBE = 010000 G	L\$DLY 002116 G	L10016 011204	MCD.C2= 002000 G	PCHT61 012576
IDU = 000040 G	L\$DTP 002040 G	L10017 011342	MOD.C3= 004000 G	PCHT62 012640
IER = 020000 G	L\$DTYP 002034 G	L10020 011520	MOT = 000200 G	PCHT63 012710
IE.C = 000200 G	L\$DU 030212 G	L10021 014224	MSGADR 006124	PCHT64 012766
INITM 016056	L\$DUT 002072 G	L10022 014246	MSGDFL 002262 G	PCHT65 013044
INTPRI= 000340 G	L\$DVTY 002174 G	L10023 014452	MSGEND= 002276 G	PCHT66 013111
IOBRD 005650 G	L\$EF 002052 G	L10024 014462	MSGEXT= 000016 G	PCHT67 013157
IOSCO = 000014 G	L\$ENVI 002044 G	L10025 014472	MSGPKT 002260 G	PCHT70 013225
ISR = 000100 G	L\$ETP 002102 G	L10026 014502	MSORT1 017640 G	PCHT71 013277
IS.DAP= 000020 G	L\$EXP1 002046 G	L10027 014512	MSORT2 020334 G	PCHT72 013361
IS.IVP= 000100 G	L\$EXP4 002064 G	L10031 027714	MSORT3 021022 G	PCHT73 013437
IS.LRC= 000010 G	L\$EXP5 002066 G	L10032 030062	MS\$RFC= 000004 G	PCHT74 013521
IS.NRZ= 000040 G	L\$HARD 034216 G	L10033 030210	MS\$XS0= 000006 G	PCHT75 013575
IS.PAR= 000200 G	L\$HIME 002120 G	L10034 030226	MS\$XS1= 000010 G	PCHT76 013751
IS.WRF= 000004 G	L\$HPCP 002016 G	L10035 030244	MS\$XS2= 000012 G	PCHT77 014033
IXE = 004000 G	L\$HPTP 002022 G	L10036 030600	MS\$XS3= 000014 G	PCH75A 013664
I\$AU = 000041	L\$HW 002204 G	L10037 030326	NINTM 003505 G	PEDATA 016262 G
I\$AUTO= 000041	L\$ICP 002104 G	L10040 030434	ODDPAR 015632 G	PEERCK 016570 G
I\$CLN = 000041	L\$INIT 027434 G	L10041 030760	ONEFIL= 000001	PEINIT 016240 G

PELOAD	016420	G	SCHERM	014205		TERM14	004551	G	T\$SUBN=	000000	T7	033406	G
PNT	= 001000	G	SCHERN	010672		TERM15	004622	G	T\$TAGL=	177777	T7.1	033442	
POPJMI	= 000000	G	SCHERR	003051	G	TERM16	004707	G	T\$TAGN=	010076	T7.2	033572	
POPJLO	= 000033	G	SCHERO	014116	G	TERM17	005000	G	T\$TEMP=	000000	T8	033770	G
PRCHST	= 000014	G	SCHER1	010264	G	TIME	002426	G	T\$TEST=	000012	T9	034124	G
PRI	= 002000	G	SCHER2	010416	G	TKACER	016130	G	T\$TSTM=	177777	UAM	= 000200	G
PRI00	= 000000	G	SCHER3	010532	G	TRAPD4	030166		T\$TSTS=	000001	UINTM	003522	G
PRI01	= 000040	G	SCHER4	010626	G	TRAP4	030160		T\$AU=	010035	UNIT	002422	G
PRI02	= 000100	G	SCHER5	010772	G	TSBA	002220	G	T\$AUT=	010032	WAITEM	015542	
PRI03	= 000140	G	SCHER6	011206	G	TSBAHI	002222	G	T\$CLE=	010033	WAITMR	015230	
PRI04	= 000200	G	SCHER7	011344	G	TSDB	002214	G	T\$DAT=	010075	WAITMS	015504	
PRI05	= 000240	G	SCHEXE	016032	G	TSDBHI	002216	G	T\$DU=	010034	WAITMT	015260	G
PRI06	= 000300	G	SCHEXT	= 000010	G	TSSR	002224	G	T\$HAR=	010071	WAITSR	014702	G
PRI07	= 000340	G	SCHPKT	002240	G	TSVCT	002226	G	T\$HW=	010000	WRAPR1	006130	G
RDCHP3	005523	G	SFPTBL	002212	G	TS.NBA=	002000	G	T\$INI=	010031	WRAPR2	006304	G
RDCH1	005454	G	SHWRAP	016104	G	TS.SPE=	020000	G	T\$MSG=	010023	WRAPR3	006456	G
RDCH2	005405	G	SIP	= 040000	G	TS.SSR=	000200	G	T\$PC=	000001	WRAPR4	006630	G
RDCSEL	020700		SKDAEM	007262	G	TS.XA0=	000400	G	T\$PRO=	010030	WRPER1	002710	G
RDCTLO	= 000020	G	SKEWER	003272	G	TS.XA1=	001000	G	T\$PTA=	010074	WRPER2	002742	G
RD.MAI	= 000100	G	SPECON	003424	G	TS11BD	005041	G	T\$SEG=	010000	WRPER3	003017	G
RD.REV	= 000200	G	SSROFF	002570	G	TS4ADR	034242		T\$SOF=	010072	WRPHI	= 000200	G
RD.SKP	= 000020	G	SSRON	002625	G	TS4CL	002424	G	T\$SRV=	010027	WRPLO	= 000005	G
RD.SPC	= 000040	G	SVCGBL	= 000000		TS4INT	002306	G	T\$SUB=	010065	WT4SSR	014514	G
RD.07	= 000000	G	SVCINS	= 000001		TS4INO	014454	G	T\$SW=	010001	XCORDA	017160	
RD.110	= 000007	G	SVCSUB	= 000000		TS4IN0	014464	G	T\$TES=	010070	XINCOR	017246	
RD.12	= 000001	G	SVCTAG	= 000000		TS4IN1	014474	G	T1	030246	XMULT	017330	
RD.20	= 000002	G	SVCTST	= 000000		TS4IN2	014474	G	T1DATA	030334	XPREAM	017376	
RD.40	= 000003	G	SWB.C	= 010000	G	TS4IN3	014504	G	T1WRAP	030450	XRDFMK	017542	
RD.68	= 000004	G	S\$LSYM=	010000		TS4VCT	034257		T1.1	030246	XSTAT0	002266	G
RD.75	= 000005	G	TAEM	007002	G	T\$ARGC=	000001		T1.2	030372	XSTAT1	002270	G
RD.90	= 000006	G	TAEMA	007102		T\$CODE=	031004		T10	034172	XSTAT2	002272	G
RFC	002264	G	TAEMB	007153		T\$ERRN=	000022		T2	030602	XSTAT3	002274	G
ROMCK	017614	G	TAEMC	007207		T\$EXCP=	000000		T2SHFT	030752	X\$ALWA=	000000	
ROMEM	010056	G	TAERR	003222	G	T\$FLAG=	000041		T2WRAP	030606	X\$FALS=	000040	
ROMEMA	010122		TAER1	003102	G	T\$FREE=	034674		T2.1	030626	X\$OFFS=	000400	
ROMER	003375	G	TAER2	003152	G	T\$GMAN=	000000		T3	030762	X\$TRUE=	000020	
ROMEX	017106	G	TALOAD	015712	G	T\$HILI=	000776		T3.1	030762	XOOF9	017264	
ROMLER	014226	G	TCFCMK	= 177701	G	T\$LAST=	000001		T3.2	031026	X8OF9	017112	
ROMLKI	002420	G	TEMPO	002430	G	T\$LOLI=	000060		T4	031454	X9OF9	017302	
ROMLOK	017006	G	TEMP1	002432	G	T\$LSYM=	010000		T4S4	002440	\$BGNLE=	177777	
ROM\$LK	= 000013	G	TEMP2	002434	G	T\$LTNO=	000012		T4S4MG	005775	\$ERFLG=	000400	
SCH	= 140004	G	TERMA	010226		T\$NEST=	177777		T4.1	031464	\$F\$AND=	000310	
SCHBLK	002276	G	TERMTB	014616		T\$NS0=	000000		T4.2	031620	\$F\$BAD=	000401	
SCHERA	010330		TERM01	003632	G	T\$NS1=	000005		T4.3	031774	\$F\$BLA=	000170	
SCHERB	010462		TERM02	003653	G	T\$NS2=	000002		T4.4	032132	\$F\$CAS=	000150	
SCHERC	010576		TERM03	003716	G	T\$NS3=	000003		T5	032362	\$F\$DEC=	000220	
SCHERD	010720		TERM05	003777	G	T\$PCNT=	000000		T5.1	032416	\$F\$DO=	000340	
SCHERE	011036		TERM06	004060	G	T\$PTAB=	010074		T5.2	032500	\$F\$FAL=	000405	
SCHERF	011122		TERM07	004157	G	T\$PTHV=	000001		T5.3	032564	\$F\$G00=	000400	
SCHERG	011252		TERM10	004233	G	T\$PTNU=	000001		T5.4	032662	\$F\$IF=	000110	
SCHERM	011274		TERM11	004310	G	T\$SAVL=	177777		T6	033024	\$F\$INC=	000210	
SCHERI	011434		TERM12	004413	G	T\$SEGL=	177777		T6.1	033060	\$F\$L00=	000200	
SCHERL	014162		TERM13	004500	G	T\$SEKO=	010000		T6.2	033210	\$F\$NAM=	000160	

\$F\$NO = 000403	\$LOCTA= 177777	\$TAGNU= 050205	\$\$\$ERFL= 000000	\$\$\$TAG= 050000
\$F\$OR = 000320	\$LSTCN= 177777	\$TEMP = 050204	\$\$\$FLAG= 000001	. = 034674
\$F\$RTN= 000300	\$LSTIN= 000001	\$TSKO = 050204	\$\$\$FROM= 000000	.CORD = 000004 G
\$F\$SEL = 000140	\$LSTST= 177777	\$TSK1 = 050203	\$\$\$LOC = 034104	.INCOR= 000002 G
\$F\$THE = 000330	\$LSTTA= 000001	\$TSK2 = 177777	\$\$\$LOCN= 000000	.MULT = 000200 G
\$F\$TRU= 000404	\$MCALL= 000000	\$TSK3 = 177777	\$\$\$REG = 177777	.PREAM= 000040 G
\$F\$LNT = 000130	\$NESTL = 177777	\$TSK4 = 050137	\$\$\$RETU= 000000	.RDFMK= 000100 G
\$F\$WHI= 000120	\$NSKO = 000110	\$TSK5 = 050145	\$\$\$RTN1= 000000	.OOF9 = 000010 G
\$F\$YES= 000402	\$NSK1 = 000110	\$TSK6 = 050147	\$\$\$RTN2= 000000	.8OF9 = 000001 G
\$IFLEV= 177777	\$NSK2 = 000110	\$\$\$ARGC= 000000	\$\$\$SRC = 000027	.9OF9 = 000020 G
\$ISK0 = 000001	\$NSK3 = 000130	\$\$\$BYTE= 000402	\$\$\$TGSV= 050137	
\$ISK1 = 000001	\$SAVLE= 177777	\$\$\$CASE= 000404	\$\$\$TGS1= 000001	
\$ISK2 = 000001	\$SSKO = 050175	\$\$\$DST = 000003	\$\$\$TGS2= 000000	
\$ISK3 = 000001	\$TAGLE = 177777	\$\$\$ELOC= 000402	\$\$\$TO = 000000	

. ABS. 034674 000

ERRORS DETECTED: 0

CZTSIC,CZTSIC/SOL/EQ:ONEFILE=SVC.SML,SPMAC.SML,CZTSIC.P11
RUN-TIME: 74 73 .8 SECONDS
RUN-TIME RATIO: 224/148=1.5
CORE USED: 31K (62 PAGES)