

RK611

UNIBUS DRIVE PART 2
CZR6IE0

AH-9122E-MC

COPYRIGHT © 76-78

FICHE 1 OF 2

MAR 1978

digital

MADE IN USA

RK611

UNIBUS DRIVE PART 2
CZR61E0

AH-9122E-MC

COPYRIGHT © 76-78

FICHE 2 OF 2

MAR 1978

digital

MADE IN USA

801

EDF:02R6HESEG
P00000001
CZ6IE.P11

00010000 780223
CZ6IED UNIBUS RK6 DR PRT2
10-JAN-78 09:43

POP10 411 43HOR1CZ6IESEG
MACY11 30A(1052) 10-JAN-78 11:25 PAGE 1

00010000 780223

REM %

IDENTIFICATION

PRODUCT CODE: AC-9120E-MC
PRODUCT NAME: CZ6IED UNIBUS RK6 DR PRT 2
DATE: FEB 1978
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: GARY PAPAIZIAN

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.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1976, 1978 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	HARDWARE
2.2	PRELIMINARY TESTING & PROGRAMS
3.0	PROGRAM CONSIDERATIONS
3.1	PDP-11 FAMILY COMPATIBILITY
3.2	XXDP
3.3	ACT/APT
3.3.1	APT ETABLE DEFINITIONS
3.4	DUAL ACCESS
3.5	MEMORY MANAGEMENT
3.6	PARITY CHECK ENABLED
3.7	BAD SECTORS
3.8	EXECUTION TIME
3.9	FAULT ISOLATION
3.10	ERROR CORRECTION & FAILURE RATE ANALYSIS
3.11	DEFAULT UNIBUS ADDRESSES & VECTORS
4.0	OPERATING PROCEDURE & CONTROL FUNCTIONS
4.1	PROGRAM LOADING
4.2	STARTING LOCATIONS
4.3	CONSOLE SWITCH REGISTERS
4.4	SOFTWARE SWITCH REGISTER
4.5	INPUT DIALOGUE
4.6	PROGRAM EXAMPLE
4.7	HALTING THE PROGRAM
5.0	DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION
5.1	GENERAL
5.2	TEST DESCRIPTIONS
6.0	ERROR REPORTING
6.1	ERROR INTERPRETATION
6.2	ERROR PRINTOUT EXAMPLE

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

1.0 ABSTRACT

THIS PROGRAM PERFORMS PART 2 OF THE DRIVE DIAGNOSTICS TO INSURE THAT THE DISK IS CAPABLE OF PERFORMING READ AND WRITE DATA OPERATIONS IN BOTH 20 AND 22 SECTOR FORMATS. WORST CASE PATTERNS, SPIRAL WRITING AND READING, AND ALL OFFSET OPERATIONS ARE PERFORMED. ERROR DETECTION LOGIC IS CHECKED BY SOFTWARE ERROR FORCING.

AFTER A SUCCESSFUL RUN (WITH NO ERRORS) OF PART 2, THE DRIVE IS READY FOR PART 3 OF THE DRIVE DIAGNOSTICS.

TESTING IS BASED ON A HIERARCHY APPROACH STARTING WITH BASIC LOGIC TESTS AND PROCEEDING THRU DYNAMIC TESTING. THE TESTS WILL BE KEPT SMALL TO FACILITATE SCOPING LOOPS.

THIS PROGRAM WILL TEST RK06 & RK07 WITHOUT NEED OF OPERATOR INTERVENTION.

*****CAUTION*****

HALTING THIS PROGRAM ANYWHERE BUT AT THE END OF A PASS, MAY LEAVE THE HEADERS IN THE DISK CARTRIDGE IN AN UNDETERMINED STATE.

2.0 REQUIREMENTS

2.1 HARDWARE

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DISK DIAGNOSTIC:

PDP-11
CONSOLE TELETYPE
16K MEMORY
KW11-L OR KW11-P CLOCK
RK06 UNIBUS CONTROLLER (RK611)
1 TO 8 RK06/RK07 DRIVES

- NOTES: 1. IF NEITHER KW11-L OR P CLOCK IS USED, ALL TIMING TESTS WILL BE BYPASSED. A MESSAGE AT THE BEGINNING OF THE TESTS WILL CONFIRM THIS.
2. A 22 SECTOR FORMATTED PACK IS REQ'D, BUT WILL BE A RESULT OF RUNNING DRIVE DIAGNOSTOC PART 1 (SEE BELOW).

2.2 PRELIMINARY TESTING & PROGRAMS

THE RK611 DISKLESS CONTROLLER DIAGNOSTICS (ALL PARTS) SHOULD FIRST RUN SUCCESSFULLY FOLLOWED BY THE RK06 DRIVE DIAGNOSTIC- PART 1.

3.0 PROGRAM CONSIDERATIONS

3.1 PDP-11 FAMILY COMPATIBILITY

THIS PROGRAM CAN BE USED BY THE PDP-11/04,05,10,20,
34,35,40,45,50, & 70.

IT IS COMPATABLE WITH THE LSI-11 INSTRUCTION SET AND CAN TEST
THE RK06 ONLY IF THE DRIVE CONTROLLER FOR THE LSI-11 IS
DESIGNED TO BE DIAGNOSTICALLY COMPATABLE WITH THE RK611.

3.2 XXDP

THIS PROGRAM CAN BE CHAINED BY XXDP & WILL NOT OVERLAY THE
LOADER.

CHAIN MODE OPERATION (MONITOR)

1. THE INPUT DIALOGUE IS BYPASSED.
2. THE BUSS ADDRESS & CONTROLLER INTERRUPT VECTOR IS
DEFAULTED.
3. DRIVE 0 WILL NOT BE TESTED.
4. ALL OTHER DRIVES IN THE 'DRIVE PRESENT' CONDITION WILL
BE TESTED.

NOTE: THE DRIVE PRESENT CONDITION IS:

- A. HEADS MANUALLY LOADED
- B. CORRECT PORT SELECTED
- C. WRITE LOCK DISABLED
- D. DRIVE READY INDICATOR ON

DUMP MODE OPERATION (MANUAL)

1. INPUT DIALOGUE IF STARTED FROM 220.
2. DRIVE 0 CAN BE TESTED, BUT THE OPERATOR IS FIRST GIVEN
A MESSAGE TO REPLACE THE PACK IN DRO WITH A SCRATCH
PACK & TYPE <CR> WHEN DONE.

3.3 ACT/APT

THIS PROGRAM IS ACT COMPATIBLE. IT IS APT
COMPATIBLE TO THE EXTENT THAT APT HOOKS WILL BE IN THE
PROGRAM & WILL WORK THRU THE 'UPTON INTERFACE'.

FOR OTHER INTERFACES, APT MAY ONLY LOAD & START THE PROGRAM.
I.E. LOAD & DUMP MODE.

AUTOMATIC MODE (MONITOR)

1. THE INPUT DIALOGUE IS BYPASSED.
2. THE BUSS ADDRESS & CONTROLLER INTERRUPT VECTOR IS

- 203
204
205
206
207
208
209
210
211
212
213
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
3. DEFAULTED.
ALL DRIVES IN THE 'DRIVE PRESENT' CONDITION WILL BE TESTED.

NOTE: THE DRIVE PRESENT CONDITION IS:

- A. HEADS MANUALLY LOADED
B. CORRECT PORT SELECTED
C. WRITE LOCK DISABLED
D. DRIVE READY INDICATOR ON

DUMP MODE (MANUAL): INPUT DIALOGUE IF STARTED FROM 220.

3.3.1 APT ETABLE DEFINITIONS

THE FOLLOWING DEFINITIONS ARE VALID FOR SPECIFYING APT ENVIRONMENTAL TABLE (ETABLE) ENTRIES. VIA RUNNING THE APT UTILITY PROGRAM "TSP":

1. SOFTWARE ENVIRONMENT:
=1 IF APT SCRIPT MODE
=0 IF STANDALONE MODE
2. ENVIRONMENT MODE: BYTE
BIT 7 = 1 ETABLE DOES SIZING
= 0 PROGRAM DOES SIZING
BIT 6 = 1 SPOOL MESSAGES TO APT IF SCRIPT MODE
= 0 DON'T SPOOL TO APT
BIT 5 = 1 SUPPRESS CONSOLE OUTPUT
= 0 ALLOW CONSOLE OUTPUT
BITS 4-0 NOT USED
3. SWITCH 1 (SOFTWARE SWITCH REGISTER)
IF ENVIRONMENT MODE BIT 7 (SIZING BIT) IS SET TO 1, THE SOFTWARE SWITCH REGISTER WILL BE USED, INSTEAD OF THE HARDWARE CONSOLE SWITCH REGISTER. REGARDLESS OF WHICH ONE IS USED, ALL BITS DEFINED IN SECTIONS 4.3 & 4.4 (SWITCH REGISTER OPTIONS) MAY USED WHEN RUNNING IN STANDALONE MODE.
IN APT SCRIPT MODE, HOWEVER, BIT 14 (LOOP ON TEST) MUST ALWAYS BE SET TO 0.
4. SWITCH 2 (USER SWITCH REGISTER)
NOT USED
5. CPU OPTIONS:
NOT USED
6. MEMORY TYPES 1-4 AND MAX MEMORY ADDRESSES
NOT USED
7. INTERRUPT VECTOR 1:
USED WHEN ENVIRONMENT MODE BIT 7=1. DEFAULT = 210
8. BUS PRIORITY 1:
USED WHEN ENVIRONMENT MODE BIT 7=1. DEFAULT = 5

- 259
260
261
262
263
264
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
306
307
308
309
310
311
312
313
314
9. INTERRUPT VECTOR 2:
NOT USED
 10. BUS PRIORITY 2:
NOT USED
 11. BASE ADDRESS:
USED WHEN ENVIRONMENT MODE BIT 7 = 1 DEFAULT = 177440
 12. DEVICE MAP:
USED WHEN ENVIRONMENT MODE BIT 7 = 1. EACH BIT SET TO
1 IN BITS 0-7 WILL SELECT THE CORRESPONDING DRIVE TO BE
TESTED. BITS 8-15 ARE NOT USED.
 13. CONTROLLER DESCRIPTOR WORDS:
NOT USED
 14. DEVICE DESCRIPTOR CODES (IN WORDS):
NOT USED

3.4 DUAL ACCESS

THIS PROGRAM WILL NOT TEST OR SUPPORT DUAL-ACCESS. A DRIVE
EQUIPED WITH DUAL ACCESS MUST BE SWITCHED TO THE PORT UNDER
TEST TO PREVENT CONTENTION WITH THE OTHER PORT.

DUAL ACCESS TESTS WILL BE INCORPORATED IN A SEPARATE PROGRAM
AT A LATER DATE.

3.5 MEMORY MANAGEMENT

MEMORY MANAGEMENT NOT USED

3.6 PARITY CHECK ENABLED

IF THE MEMORY PARITY CHECK OPTION IS AVAILABLE ON THE SYSTEM,
THE PROGRAM WILL RUN WITH MEMORY CHECK ENABLED.

3.7 BAD SECTOR

THE PROGRAM WILL COMPARE DATA ERRORS WITH THE BAD SECTOR
INFORMATION CONTAINED ON CYLINDER 410, HEAD 2. PRINTOUTS
OF DATA ERRORS DUE TO BAD SECTORS/TRACKS WILL BE MASKED OUT.

3.8 EXECUTION TIME

THE EXECUTION TIMES SHOWN BELOW ARE BASED ON THE PDP 11/50.

TOTAL TIME: 1 MIN, 30 SEC

3.9 FAULT ISOLATION
TO BE DETERMINED.

3.10 ERROR CORRECTION AND FAILURE RATE ANALYSIS
THIS PROGRAM WILL NOT DO ERROR CORRECTION OR FAILURE RATE
ANALYSIS.

3.11 DEFAULT UNIBUS ADDRESSES & VECTORS
THE FOLLOWING IS A LIST OF ALL DEFAULT ADDRESSES &
VECTORS OF ALL HARDWARE TO BE USED & THEIR MEMORY
ADDRESSES WHERE THEY CAN BE CHANGED.

	LOCATION	DEFAULT CONTENTS
RK06 BUSS ADDRESS	1264	177440
CONTROLLER INTERRUPT VECTOR	1314	210
CONTROLLER PRIORITY	1316	240
P-CLOCK STATUS REG	1320	172540
P-CLOCK SET BUFFER	1322	172542
P-CLOCK READ BUFFER	1324	172544
L-CLOCK STATUS REG	1326	177546
L-CLOCK INTERRUPT VECTOR	1330	100
P-CLOCK INTERRUPT VECTOR	1332	104
TTY KB STATUS REG	1144	177560
TTY KB BUFFER	1146	177562
TTY PRINTER STATUS REG	1150	177564
TTY PRINTER BUFFER	1152	177566

4.0 OPERATING PROCEDURE & CONTROL FUNCTIONS

4.1 PROGRAM LOADING
THE PROGRAM CAN BE LOADED FROM PAPER TAPE USING STANDARD
PROCEDURE FOR ABSOLUTE LOADER TAPES; OR FROM ANY MEDIA
SUPPORTED BY XXDP.

4.1.1 LOAD THE STARTING ADDRESS (SEE SEC 4.2).

4.1.2 SET SWITCH REGISTERS AS DESIRED (SEE SEC 4.3).

4.1.3 SET DRIVES TO BE TESTED IN THE 'LOAD' CONDITION & WITH THE
APPROPRIATE PORT SELECTED & WRITE LOCK DISABLED. DRIVES
NOT TO BE TESTED MUST HAVE BOTH PORTS DESELECTED.

NOTE: THE DRIVE WILL NOT RESPOND TO THE 'START SPINDLE'
COMMAND IF THE RUN/STOP SWITCH IS IN THE 'STOP'
POSITION.

4.1.4 PRESS 'START'

THE PROGRAM WILL IDENTIFY ITSELF AND WILL BEGIN A
DIALOGUE WITH THE OPERATOR TO DETERMINE DRIVES TO BE TESTED
(SEE SEC 4.5).

THE PROGRAM BEGINS TESTING ONLY THOSE DRIVES SPECIFIED BY
THE INPUT DIALOGUE. IF A SPECIFIED DRIVE CANNOT BE FOUND BY
THE PROGRAM IT WILL BE FLAGGED AS AN ERROR THAT THE DRIVE
WAS NOT AVAILABLE. THEN BEGINNING WITH THE LOWEST NUMERICAL
DRIVE AND PROCEEDING IN SEQUENTIAL ORDER, ALL VALID DRIVES
WILL BE TESTED. ONE PASS THROUGH THE TEST SEQUENCE WILL BE
PERFORMED ON EACH DRIVE BEFORE MOVING TO THE NEXT DRIVE
IN SEQUENCE. THE DRIVE TO BE TESTED WILL BE TYPED AT THE
BEGINNING OF EACH PASS. "END OF PASS" WILL BE TYPED AFTER
TESTING ALL DRIVES.

4.2 STARTING LOCATIONS

LOCATION 200 - STARTING ADDRESS TO DEFAULT THE BUSS
ADDRESS & THE CONTROLLER INTERRUPT VECTOR
& TEST ALL DRIVES IN THE 'DRIVE PRESENT'
CONDITION.

NOTE: THE DRIVE PRESENT CONDITION IS:

- A. HEADS MANUALLY LOADED
- B. CORRECT PORT SELECTED
- C. WRITE LOCK DISABLED
- D. DRIVE READY INDICATOR ON

LOCATION 220 - STARTING ADDRESS TO INPUT TESTING PARAMETERS
VIA THE INPUT DIALOGUE. BUSS ADDRESS &
CONT. INTERRUPT VECTOR INPUTTED ONLY ON
1ST PASS.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT
CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

ALSO, THE PROGRAM WILL AUTOMATICALLY DETERMINE WHETHER
THE DRIVE IS AN RK06 OR RK07.

4.3 SWITCH REGISTER

THE SWITCHES ARE USED TO PROVIDE CONTROL FUNCTIONS.

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
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

SWITCH	FUNCTION
-----	-----
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUT
12	BYPASS DRIVE AFTER 20 ERRORS
11	INHIBIT ITERATION
10	BELL ON ERROR
9	LOOP ON ERROR
8	LOOP ON TEST IN SW<07:00>

4.3.1 SW<15>

THE PROGRAM HALTS ON ENCOUNTERING AN ERROR, AFTER TYPING OUT THE ERROR MESSAGE AND PERTINENT INFORMATION. PRESSING "CONTINUE" CONTINUES NORMAL OPERATION OF THE PROGRAM.

4.3.2 SW<14>

THE PROGRAM LOOPS ON THE TEST THAT IS BEING EXECUTED WHEN THE SWITCH IS PUT ON. THIS SWITCH IS NORMALLY USED ALONG WITH SW15.

4.3.3 SW<13>

THIS SWITCH INHIBITS ALL ERROR MESSAGES. NORMALLY USED WHEN LOOPING ON TEST (SW14) OR LOOPING ON ERROR (SW9). WITH SWITCH <13> SET, SWITCH <15> SHOULD NOT BE SET.

4.3.4 SW<12>

THIS SWITCH BYPASSES A GIVEN DRIVE AFTER 20 ERRORS HAVE BEEN DETECTED.

4.3.5 SW<11>

EACH TEST WILL BE EXECUTED ONLY ONCE. NORMALLY AFTER THE FIRST PASS, EACH SUBTEST IS ITERATED A NUMBER OF TIMES (USUALLY 50, 5 IN SOME CASES). SETTING THIS SWITCH INHIBITS ITERATIONS, SO THAT QUICK PASSES CAN BE MADE.

4.3.6 SW<10>

RINGS A BELL ON ERROR. USEFUL WHEN ERROR TYPEOUT IS INHIBITED.

4.3.7 SW<09>

THIS SWITCH PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP FOR ERRORS. IF THE PROGRAM DETECTS AN ERROR, IT WILL LOOP BACK TO THE BEGINNING OF TEST.

4.3.8 SW<08>

THIS SWITCH IS USED TO SELECT A PARTICULAR TEST (AS PER SW<00-7>) FOR EXECUTION AND SUBSEQUENT LOOPING. THUS IF TEST 15 IS TO BE SELECTED THE SWITCH SETTING WOULD BE 000415. IT SHOULD BE NOTED THAT BEFORE SELECTING & LOOPING TEST 15, ALL THE PREVIOUS TESTS (1-14) WILL BE EXECUTED.

4.4 'SOFTWARE' SWITCH REGISTER

IF THE PROGRAM IS BEING RUN ON A SWITCHLESS PROCESSOR (I.E. AN 11/04 OR 11/34) THE PROGRAM WILL DETERMINE THAT THE HARDWARE SWITCH REGISTER IS NOT PRESENT AND WILL USE A 'SOFTWARE' SWITCH REGISTER. THE 'SOFTWARE' SWITCH REGISTER IS LOCATED AT LOCATION 176 (8). THE SETTINGS OF THE 'SOFTWARE' SWITCHES ARE CONTROLLED THROUGH A KEYBOARD ROUTINE WHICH IS CALLED BY TYPING A 'CONTROL G'. THE PROGRAM WILL RECOGNIZE THE 'CONTROL G' AT ANY TIME EXCEPT WHEN THE PROGRAM IS AT A HIGHER PRIORITY PROCESSING AN RK06 INTERRUPT. THE 'SOFTWARE' SWITCH VALUES ARE ENTERED AS AN OCTAL NUMBER IN RESPONSE TO THE PROMPT FROM THE SWITCH ENTRY ROUTINE:

SWR = NNNNNN NEW =

EACH TIME SWITCH SETTING ARE ENTERED, THE ENTIRE SWITCH REGISTER IMAGE MUST BE ENTERED. LEADING ZEROS ARE NOT REQUIRED. 'RUBOUT' AND 'CONTROL U' FUNCTIONS MAY BE USED TO CORRECT TYPING ERRORS DURING SWITCH ENTRY.

ON PROCESSORS WITH HARDWARE SWITCH REGISTERS, THE 'SOFTWARE' SWITCH REGISTER MAY BE USED. IF THE PROGRAM FINDS ALL 16 SWITCHES IN THE 'UP' POSITION, ALL SWITCH REGISTER REFERENCES WILL BE TO THE 'SOFTWARE' REGISTER AND THE PROCEDURES DESCRIBED ABOVE MUST BE FOLLOWED.

4.5 INPUT DIALOGUE

THE DIALOGUE WILL BE DONE INTERACTIVELY. THE PROGRAM WILL REQUEST A PARAMETER BY CONSOLE TYPEOUT. THE PARAMETER MAY THEN BE ENTERED AS SPECIFIED BELOW OR ALLOWED TO DEFAULT BY A CARRIAGE RETURN. UNRECOGNIZED OR ILLEGAL RESPONSES WILL BE ECHOED BACK FOLLOWED BY "?". THE PROPER RESPONSE MAY THEN BE ENTERED.

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT CONSIDERATIONS IN SECTIONS 3.2 & 3.4.

483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
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

4.5.1 DRIVE SELECTION

THE REQUEST WILL BE:

DRIVES TO BE TESTED:

THE DEFAULT RESPONSE IS CARRIAGE RETURN TO TEST ALL DRIVES
IN THE 'DRIVE PRESENT' CONDITION.

THE OPERATOR CAN ALSO TYPE IN THE SPECIFIC DRIVE NUMBERS
TO BE TESTED, SEPARATED BY COMMAS & TERMINATED BY A CARRIAGE
RETURN.

E.G. DRIVES TO BE TESTED: 1,2,4,6

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT
CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

4.5.2 BUS ADDRESS

THE REQUEST WILL BE:

TYPE IN BUSS ADDRESS IF NOT 177440

THE DEFAULT IS A CARRIAGE RETURN

4.5.3 CONTROLLER INTERRUPT VECTOR

THE REQUEST WILL BE:

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210

THE DEFAULT IS A CARRIAGE RETURN.

4.5.4 EXAMPLE OF PROGRAM DIALOGUE

THE EXAMPLE SHOWN IS FOR A PROGRAM STARTED AT ADDRESS 220.
ALL OPERATOR RESPONSES ARE UNDERLINED.

UNIBUS RK06 DRIVE DIAGNOSTIC
PART 2
CZR6IEO

DRIVES TO BE TESTED: 1,3<CR>

TYPE IN BUSS ADDRESS IF NOT 177440 <CR>

TYPE IN CONTROLLER INTERRUPT VECTOR IF NOT 210 <CR>

WILL TEST DRIVES:

539
540
541
542
543
544
545
546
547
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

1
3

DRIVE 1

(THE REST IS IDENTICAL TO THE EXAMPLE SHOWN IN 4.6 BELOW)

4.6 PROGRAM EXAMPLE

THE FOLLOWING IS AN EXAMPLE OF A PROGRAM STARTED AT THE
DEFAULT ADDRESS (200) & WITH 2 DRIVES ON THE LINE.

UNIBUS RK06 DRIVE DIAGNOSTIC
PART 2

CZR6IE0

WILL TEST DRIVES:

0
1

DRIVE 0

DRIVE SERIAL NO. AAA
CARTRIDGE SERIAL NO. BBB

DRIVE 1

DRIVE SERIAL NO. CCC
CARTRIDGE SERIAL NO. DDD

END PASS #1

WILL TEST DRIVES:

0
1

DRIVE 0

DRIVE 1

END PASS # 2

(ETC)

THE ABOVE ASSUMES NO ERRORS DETECTED.
THE NUMBER OF PASSES IS DETERMINED BY ACT/APT/XXDP

IMPORTANT: FOR VARIATIONS OF THE ABOVE, SEE XXDP, ACT/APT
CONSIDERATIONS IN SECTIONS 3.2 & 3.3.

4.7 HALTING THE PROGRAM

THE PROGRAM PROVIDES A METHOD OF HALTING ITSELF SUCH THAT
THE CARTRIDGE AND/OR DRIVE IS NOT LEFT IN ON UNDETERMINED

595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
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

STATE; IE: HEADS UNLOADED OR INVALID FORMAT.

TO PROPERLY HALT, TYPE CONTROL-C (↑C) ON THE CONSOLE.

IF HEADS ARE LOADED & FORMATTING IS VALID,
THE PROGRAM WILL:

1. ECHO ↑C
2. TYPE "CPU HALTED"
3. HALT THE PROGRAM

IF HEADS ARE NOT LOADED AND/OR FORMATTING IS INVALID,
THE PROGRAM WILL:

1. ECHO ↑C
2. TYPE 'HALT PENDING, PLEASE WAIT'
3. DO THE TEST(S) THAT LOADS HEADS AND/OR FORMATS
THE INVALID CYLINDERS
4. TYPE 'CPU HALTED'
5. HALT THE PROGRAM

NOTES:

1. THE ABOVE EXAMPLE IS FOR THE PROGRAM RUNNING IN DUMP
MODE (MANUAL). IF THE PROGRAM IS RUNNING IN CHAIN/AUTO
MODE VIA XXDP,ACT,APT; IT WILL FIRST LOAD HEADS
AND/OR FORMAT CORRECTLY, IF REQ'D, THEN IT WILL
JUMP ON TO THE MONITOR WHERE THE NEXT PROGRAM CAN BE
CALLED IN.

THE TYPEOUTS WILL BE "ABORT PENDING - PLEASE WAIT"
& "PROGRAM ABORTING"
2. OPERATING THE 'CONTINUE' SWITCH ON THE CPU CONSOLE WILL RETURN THE
PROGRAM TO TEST 1 WHERE TESTING WILL BEGIN WITH THE 1'ST DRIVE AGAIN.

5.0 DRIVE DIAGNOSTIC FUNCTIONAL DESCRIPTION

5.1 GENERAL

A. WRITE TESTS

THESE TESTS CHECK THE ABILITY OF THE DRIVE TO WRITE & READ
WORSE CASE PATTERNS; PERFORM ALL OFFSETS & PERFORM ALL
SPIRAL WRITING.

B. SERVO & SPINDLE TIMING TESTS

THESE TESTS CHECK & TYPE HEAD LOAD, UNLOAD & INDEX TIMING,
ALSO MIN, MAX, AND AVERAGE SEEK TIMES, AND MAX VELOCITY
OF THE HEADS ARE MEASURED & TYPED.

5.2 TEST DESCRIPTIONS

BASIC CONTROLLER TESTS, SIZING & SETUP

TEST 1 REFERENCE ALL CONTROLLER REGISTERS

THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS
CAN BE ACCESSED. THE INABILITY TO BE ACCESSED WILL
RESULT IN A TIMEOUT TRAP WITH AN ERROR MESSAGE. ANY
ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
TESTS AND JUMPING TO 'END OF PASS'

TEST 2 SIZE THE BUSS

THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
MANUAL MODE.
EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
DRIVE WILL BE TESTED AS AN RK06. IF SET, THE PROGRAM WILL BYPASS
TESTING THAT DRIVE ONLY IF THE ERROR WAS A RESULT OF
MDS, UFE OR NED BEING SET; OR BOTH NED & DRA RESET IN-
DICATING THE OTHER PORT IS ACCESSED.
IF CERR IS DUE TO DTYE, THE DRIVE WILL BE TESTED AS 4 RK07

TEST 3 VERIFY OPERATOR DRIVE SELECTIONS

THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
DEFAULTED. EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06.
IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
NED & DRA RESET (WRONG PORT). IF CERR IS A RESULT OF
NED ONLY, IT IS CHECKED AGAINST THE INPUTTED INFOR TO
VERIFY IT WAS NOT SPECIFIED.
IF CERR IS DUE TO DTYE, THE DRIVE WILL BE TESTED AS
AN RK07.

TEST 4 FIND NEXT DRIVE TO BE TESTED

THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
ADDRESS IN 'DRVAD' & \$TMP4 IS SET TO CDT IF THE DRIVE
DRIVE IS AN RK07.
THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
THE DRIVE WHOSE ADDRESS IS IN 'DRVAD'.

TEST 5 PRINT DRIVE SERIAL NUMBER

707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762

763
764
765
766
767
768
769
770
771
772
773
774
775
776
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

THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
IN DECIMAL & IS PERFORMED ON THE 1ST PASS ONLY

TEST 6 SET VV WITH PACK COMMAND

IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.

TEST 7 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL

THIS TEST VERIFIES THAT CYL 632 (1456 FOR RK07), TRACK 2 CAN BE READ.
THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
FACTORY DURING MANF. ALL BAD SECTOR INFO (BSE) WILL BE STORED
AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
A MESSAGE WILL BE TYPED INDICATING THAT ALL
FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING

THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.

WRITE TESTS

TEST 10 BASIC WRITE DATA TEST; 1 WORD

THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE JUST ONE WORD,
ALL SECTORS ON CYL 0 ARE GIVEN IDENTICAL HEADERS &
A WRITE COMMAND IS ISSUED. READ & WRITE CHECK COMMANDS ARE NOT
PERFORMED. THIS TEST PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP
FOR A WRITE ERROR.

TEST 11 BASIC WRITE DATA TEST; FULL SECTOR

THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE
A FULL SECTOR. ALL ZEROS ARE WRITTEN BY THE WRITE DATA COMMAND
& CHECKED BY A RD DATA COMMAND. A FURTHER CHECK IS PERFORMED
BY THE WRT CHK COMMAND.
THE ABOVE IS REPEATED FOR AN ALL ONES PATTERN.

TEST 12 20 SECTOR FORMAT TEST

DATA IS WRITTEN ON A FULL TRACK IN 20 SECTOR FORMAT.
MSG B0,B1 ARE CHECKED FOR ANY ERROR CONDITION.
CYLINDER, TRACK, SECTOR 0 IS USED.

TEST 13 TEST OFFSET & RTC LOGIC

819
820
821
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

THE HEADS ARE FIRST OFFSET BY OFFSET COMMANDS.
THIS TEST CHECKS THE RTC LOGIC BY VERIFYING THAT THE
'OFFSET ON' BIT (MSG A,00) RESETS AND THE OFFSET REG

BECOMES THE CYL DIFF INFO WHEN A SEEK CMD TO A
DIFFERENT CYLINDER IS ISSUED
IT ALSO TESTS THAT DRIVE CLEAR & SEEK TO SELF WILL NOT
CLEAR THE 'OFFSET ON' BIT OR THE OFFSET REG.
ALL OFFSET POSITIONS IN BOTH DIRECTIONS ARE CHECKED

TEST 14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

THIS TEST VERIFIES THAT THE HEAD OFFSET LOGIC IS OPERATIONAL BY
WRITING ALL 1'S PATTERNS ON CYLINDER 0, HEAD 0. THEN
PERFORMING READ DATA FROM CENTERLINE AND MOVING OUT + AND - OFFSET
POSITIONS UNTIL A FAILURE OCCURES. THE OFFSET POSITIONS
ARE TYPED OUT.
OFFSET CODES ARE ALSO VERIFIED BY READING MSG A, STATUS 00 & 10.
ALL HEADS ARE TESTED AT CYL 0.

IF THERE ARE NO FAILURES AT ALL, THIS INDICATES THAT

OR A. HEADS DID NOT MOVE AT ALL
B. THE COMBINATION OF DISC SURFACE, HEADS, R/W AMP
ARE EXCEPTIONALLY GOOD.

NOTE THAT THE OFFSET FAILURE IS NOT AN ERROR,
BUT AN INDICATION OF SURFACE, HEAD, & R/W ELECTRONICS QUALITY ONLY

TEST 15 WRITE WITH HEADS OFFSET

THIS TEST VERIFIES THAT WHEN ATTEMPTING TO
WRITE WITH HEADS OFFSET THAT THE OFFSET WILL CLEAR
& THE DRIVE WILL WRITE
SINCE THE WRITE COMMAND HAS AN IMPLIED RTC.
THIS TEST IS PERFORMED FOR MAX POS & NEG OFFSETS ONLY

TEST 16 TEST CURRENT CROSS-OVER CYLINDERS

THIS TEST VERIFIES THAT THE DRIVE CAN WRITE & READ OFF
CURRENT CHANGE CYLINDERS X & Y IN THE FOLLOWING WAY:

SPIRAL WRITING IS PERFORMED FROM CYLINDER X TO CYLINDER Y
WITH A DATA PATTERN FILLING THE ENTIRE 2 CYLINDERS.

A WRITE CHECK IS THEN PERFORMED TO VERIFY DATA WAS PROPERLY WRITTEN.
THIS TEST IS PERFORMED FOR ALL 3 HEADS.

CYLINDER X: 63 127 191 255 319 383 RK06
CYLINDER Y: 64 128 192 256 320 384 RK06

CYLINDER X: 127 255 383 511 639 767 RK07
CYLINDER Y: 128 256 384 512 640 768 RK07

THE ABOVE CYL NUMBERS ARE IN DECIMAL.

TEST 17 TEST HEAD SWITCHING TIME

TESTS THE ABILITY TO SWITCH HEADS IN LESS THEN 10MS WHEN HEADS SPIRAL.

1. SECTOR 23(8) IS FIRST LOCATED AND A WRITE DATA COMMAND OF 512 WORDS TO SECTOR 25(8) IS ISSUED.
2. THE PROGRAM NOW KNOWS THAT THE DRIVE WILL NOT HAVE TO TRAVEL A FULL REVOLUTION BEFORE FINDING SECTOR 25(8).
3. SINCE EACH SECTOR TAKES APPROX. 1.2MS, THE TIME BETWEEN THE START OF THE WRITE COMMAND (FROM SECTOR 25(8), HEAD 0; TO SECTOR 0, HEAD 1) AND CONTROLLER READY SHOULD BE APPROX 6MS

THE ABOVE IS REPEATED FOR HEAD SWITCHING BETWEEN 1 TO 2

THIS TEST IS BYPASSED IF NEITHER L OR P CLOCK IS PRESENT

TEST 20 DRIVE OFF TRACK TEST

THIS TEST CHECKS FOR SERVO OSCILLATIONS DURING SETTLING TIME BEYOND THE ALLOTTED 3MS.

1. INITIALLY, EVERY CYLINDER IS FORMATTED WITH IDENTICAL HEADERS (UNIQUE TO EACH CYLINDER)
2. A FULL SECTOR WRITE COMMAND IS ISSUED BY A SINGLE CYL SEEK FROM 0 TO AS HEADERS ARE IDENTICAL, THE NEXT SECTOR TO COME UNDER THE HEADS WILL IMMEDIATELY BE WRITTEN.
3. IF THERE IS OSCILLATION SENSED BY READING THE TRIBITS, DRIVE OFF TRACK ERROR WILL SET.

IN THIS MANNER OSCILLATING SEEKS ARE PERFORMED BETWEEN ALL MAJOR CYLINDER 100 OSCILLATIONS ARE PERFORMED AT EACH MAJOR CYLINDER BEFORE DOING THE NEXT CYLINDER

ANY TEST THAT MODIFIES STANDARD FORMATTING IS FOLLOWED BY A 'CLEAN UP' TEST TO PUT THOSE CYLINDERS BACK TO STANDARD FORMAT.

6.0 ERROR REPORTING

6.1 ERROR INTERPRETATION

WHENEVER AN ERROR MESSAGE IS PRINTED OUT, ALL REGISTERS AND OTHER DATA PERTAINING TO THE ERROR ARE ALSO GIVEN. MSG A(00), MSG B(01), RKER, RKBA...ETC, INDICATE THE CONTENTS OF THE CORRESPONDING REGISTERS AT THE TIME OF ERROR.

EVERY ERROR MESSAGE CONTAINS A PC. THIS PC INDICATES THE POSITION IN PROGRAM WHERE THE ERROR CALL IS LOCATED. THE

875
876
877
878
879
880
881
882
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

ERROR MESSAGE, BECAUSE OF PRACTICAL CONSIDERATIONS IS MADE SHORT AND MEANINGFUL. THE USER IS ADVISED TO LOOK UP THE PC IN THE PROGRAM LISTING, WHERE HE WILL FIND MORE INFORMATION ABOUT THE ERROR. IN MANY INSTANCES, A SINGLE FAULT WILL GIVE RISE TO MORE THAN ONE ERROR REPORT. A LITTLE DELIBERATION AND CAREFUL EXAMINATION OF THE DATA GIVEN WILL BE CERTAINLY VERY HELPFUL IN PINPOINTING THE FAULT. A BRIEF EXPLANATION OF WHAT IS BEING CHECKED IN THE TEST IS GIVEN AT THE BEGINNING OF EVERY TEST. ALL THE NUMBERS GIVEN WITH ERROR MESSAGES ARE IN OCTAL.

NOTE

NO ERROR LOGGING OR OPERATION HISTORY IS PROVIDED.

6.2 ERROR PRINTOUT EXAMPLES:

EXAMPLE #1

MESSAGE AD ERROR
AFTER START SPINDLE CMD & FWD SET

TEST NO. PC							
000014 016530							
EX-ECT							
AO	B0	A1	B1	A2	B2	B3	
030144	100000	013704	000001				
ACTUAL							
140144	100000	101744	000001				
RKCS1	RKCS2	RKASOF	RKER	RKDS	RKDC		
040200	000100	010000	000000	000000	000000		

THE ABOVE EXAMPLE SHOWS EXPECTED & ACTUAL DATA FOR MESSAGE REGISTERS AO, B0, A1 & B1.

MESSAGES A2, B2 & B3 WILL BE TYPED OUT ONLY AS REQUIRED IF THE CYLINDER DIFFERENCE/OFFSET, CYLINDER ADDRESS & HEAD & SECTOR INFORMATION IS A VARIABLE PARAMETER OF THE TEST.

EXAMPLE #2:

NO ATTN IN RKASOF
AFTER UNLOAD COMMAND

TEST NO. PC							
000003 014330							
RKMR2	RKMR3	RKER	RKDS	RKCS1	RKCS2	RKASOF	
000144	100000	000000	100101	000206	000104	000000	

G02

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 19

SEQ 0019

987
988
989
990

[END OF DOCUMENT]

%

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 20

SEQ 0020

```

991
992
993
994      :*** PGM REV 032 ***
995      :NLIST  CND,MC,MD
996      :LIST   ME
997      :ENABL  ABS,AMA
998
999      ;DEFINE SYSMAC MACROS
1000
1001      167400      $SWR= 167400      ;DEFINE SWITCHES 15,14,13,11,10,9,8
1002      000001      $TN= 1          ;SET FIRST TEST NO. TO 1
1003
1004
1005
1006      .TITLE  CZR6IED UNIBUS RK6 DR PRT2
1007      :*COPYRIGHT (C) 1976,1977
1008      :*DIGITAL EQUIPMENT CORP.
1009      :*MAYNARD, MASS. 01754
1010      :*
1011      :*PROGRAM BY GARY PAPAZIAN
1012      :*
1013      :*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
1014      :*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
1015      :*
1016
1017      .SBTTL  OPERATIONAL SWITCH SETTINGS
1018      :*
1019      :*      SWITCH      USE
1020      :*      -----
1021      :*      15      HALT ON ERROR
1022      :*      14      LOOP ON TEST
1023      :*      13      INHIBIT ERROR TYPEOUTS
1024      :*      12      ABORT DRIVE AFTER 20 ERRORS
1025      :*      11      INHIBIT ITERATIONS
1026      :*      10      BELL ON ERROR
1027      :*      9       LOOP ON ERROR
1028      :*      8       LOOP ON TEST IN SWR<7:0>
1029
1030
1031      .SBTTL  SUMMARY OF STARTING LOCATIONS
1032      :*
1033      :*
1034      :*      200      DEFAULT PARAMETERS
1035      :*      204      DEFAULT PARAMETERS & BYPASS WRITE TESTS
1036      :*      214      DEFAULT PARAMETERS & BYPASS TIMING TESTS
1037      :*      220      INPUT PARAMETERS
1038      :*      224      INPUT PARAMETERS & BYPASS WRITE TESTS
1039      :*      230      INPUT PARAMETERS & BYPASS TIMING TESTS
1040      :*      240      ODT11
1041      :*

```

```

1042      .SBTTL  BASIC DEFINITIONS
1043
1044      ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
1045      001100  STACK= 1100
1046      .EQUIV  EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
1047      .EQUIV  IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
1048
1049      ;*MISCELLANEOUS DEFINITIONS
1050      000011  HT= 11          ;;CODE FOR HORIZONTAL TAB
1051      000012  LF= 12          ;;CODE FOR LINE FEED
1052      000015  CR= 15          ;;CODE FOR CARRIAGE RETURN
1053      000200  CRLF= 200       ;;CODE FOR CARRIAGE RETURN-LINE FEED
1054      177776  PS= 177776      ;;PROCESSOR STATUS WORD
1055      .EQUIV  PS,PSW
1056      177774  STKLMT= 177774  ;;STACK LIMIT REGISTER
1057      177772  PIRQ= 177772    ;;PROGRAM INTERRUPT REQUEST REGISTER
1058      177570  DSWR= 177570    ;;HARDWARE SWITCH REGISTER
1059      177570  ODISP= 177570   ;;HARDWARE DISPLAY REGISTER
1060
1061      ;*GENERAL PURPOSE REGISTER DEFINITIONS
1062      000000  R0= %0           ;;GENERAL REGISTER
1063      000001  R1= %1           ;;GENERAL REGISTER
1064      000002  R2= %2           ;;GENERAL REGISTER
1065      000003  R3= %3           ;;GENERAL REGISTER
1066      000004  R4= %4           ;;GENERAL REGISTER
1067      000005  R5= %5           ;;GENERAL REGISTER
1068      000006  R6= %6           ;;GENERAL REGISTER
1069      000007  R7= %7           ;;GENERAL REGISTER
1070      000006  SP= %6          ;;STACK POINTER
1071      000007  PC= %7          ;;PROGRAM COUNTER
1072
1073      ;*PRIORITY LEVEL DEFINITIONS
1074      000000  PR0= 0           ;;PRIORITY LEVEL 0
1075      000040  PR1= 40          ;;PRIORITY LEVEL 1
1076      000100  PR2= 100        ;;PRIORITY LEVEL 2
1077      000140  PR3= 140        ;;PRIORITY LEVEL 3
1078      000200  PR4= 200        ;;PRIORITY LEVEL 4
1079      000240  PR5= 240        ;;PRIORITY LEVEL 5
1080      000300  PR6= 300        ;;PRIORITY LEVEL 6
1081      000340  PR7= 340        ;;PRIORITY LEVEL 7
1082
1083      ;*"SWITCH REGISTER" SWITCH DEFINITIONS
1084      100000  SW15= 100000
1085      040000  SW14= 40000
1086      020000  SW13= 20000
1087      010000  SW12= 10000
1088      004000  SW11= 4000
1089      002000  SW10= 2000
1090      001000  SW09= 1000
1091      000400  SW08= 400
1092      000200  SW07= 200
1093      000100  SW06= 100
1094      000040  SW05= 40
1095      000020  SW04= 20
1096      000010  SW03= 10
1097      000004  SW02= 4

```

```

1098      000002      SW01= 2
1099      000001      SW00= 1
1100      .EQUIV      SW09,SW9
1101      .EQUIV      SW08,SW8
1102      .EQUIV      SW07,SW7
1103      .EQUIV      SW06,SW6
1104      .EQUIV      SW05,SW5
1105      .EQUIV      SW04,SW4
1106      .EQUIV      SW03,SW3
1107      .EQUIV      SW02,SW2
1108      .EQUIV      SW01,SW1
1109      .EQUIV      SW00,SW0
1110
1111      .*DATA BIT DEFINITIONS (BIT00 TO BIT15)
1112      100000      BIT15= 100000
1113      040000      BIT14= 40000
1114      020000      BIT13= 20000
1115      010000      BIT12= 10000
1116      004000      BIT11= 4000
1117      002000      BIT10= 2000
1118      001000      BIT09= 1000
1119      000400      BIT08= 400
1120      000200      BIT07= 200
1121      000100      BIT06= 100
1122      000040      BIT05= 40
1123      000020      BIT04= 20
1124      000010      BIT03= 10
1125      000004      BIT02= 4
1126      000002      BIT01= 2
1127      000001      BIT00= 1
1128      .EQUIV      BIT09,BIT9
1129      .EQUIV      BIT08,BIT8
1130      .EQUIV      BIT07,BIT7
1131      .EQUIV      BIT06,BIT6
1132      .EQUIV      BIT05,BIT5
1133      .EQUIV      BIT04,BIT4
1134      .EQUIV      BIT03,BIT3
1135      .EQUIV      BIT02,BIT2
1136      .EQUIV      BIT01,BIT1
1137      .EQUIV      BIT00,BIT0
1138
1139      .*BASIC "CPU" TRAP VECTOR ADDRESSES
1140      000004      ERRVEC= 4      ; TIME OUT AND OTHER ERRORS
1141      000010      RESVEC= 10     ; RESERVED AND ILLEGAL INSTRI TIONS
1142      000014      TBITVEC=14    ; "T" BIT
1143      000014      TRTVEC= 14     ; TRACE TRAP
1144      000014      BPTVEC= 14     ; BREAKPOINT TRAP (BPT)
1145      000020      IOTVEC= 20     ; INPUT/OUTPUT TRAP (IOT) **SCOPE**
1146      000024      PWRVEC= 24     ; POWER FAIL
1147      000030      EMTVEC= 30     ; EMULATOR TRAP (EMT) **ERROR**
1148      000034      TRAPVEC=34     ; "TRAP" TRAP
1149      000060      TKVEC= 60      ; TTY KEYBOARD VECTOR
1150      000064      TPVEC= 64      ; TTY PRINTER VECTOR
1151      000240      PIRQVEC=240    ; PROGRAM INTERRUPT REQUEST VECTOR
1152
1153      .SBTTL RK06 CONTROLLER REGISTER DEFINITION

```

K02

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 23
RK06 CONTROLLER REGISTER DEFINITION

SEQ 0023

```

1154
1155
1156 ; $BASE=177440
1157 000000 RKCS1= 0 ; CONTROL AND STATUS REGISTER 1
1158 000002 RKWC= 2 ; WORD COUNT REGISTER
1159 000004 RKBA= 4 ; BUS ADDRESS REGISTER
1160 000006 RKDA= 6 ; DESIRED TRACK SECTOR REGISTER
1161 000010 RKCS2= 10 ; CONTROL AND STATUS REGISTER 2
1162 000012 RKDS= 12 ; DRIVE STATUS REGISTER
1163 000014 RKE= 14 ; ERROR REGISTER
1164 000016 RKASOF= 16 ; ATTENTION SUMMARY AND OFFSET REGISTER
1165 000020 RKDC= 20 ; DESIRED CYLINDER REGISTER
1166 000024 RKDB= 24 ; DATA BUFFER
1167 000026 RKMR1= 26 ; MAINTENANCE REGISTER 1
1168 000034 RKMR2= 34 ; MAINTENANCE REGISTER 2 (MESSAGE LINE A)
1169 000036 RKMR3= 36 ; MAINTENANCE REGISTER 3 (MESSAGE LINE B)
1170 000030 RKECPS= 30 ; ECC POSITION INFORMATION
1171 000032 RKECPT= 32 ; ECC PATTERN INFORMATION
1172
1173 .SBTTL CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)
1174
1175 ; DRIVE COMMANDS
1176
1177 000001 SELDRV= 1 ; SELECT DRIVE (GET STATUS)
1178 000003 PACK= 3 ; PACK ACKNOWLEDGE
1179 000005 CLEAR= 5 ; DRIVE CLEAR
1180 000007 UNLOAD= 7 ; UNLOAD
1181 000011 SRTSPL= 11 ; START SPINDLE
1182 000013 RECAL= 13 ; RECALIBRATE
1183 000015 OFFSET= 15 ; OFFSET
1184 000017 SEEK= 17 ; SEEK
1185 000021 RDDATA= 21 ; READ DATA
1186 000023 WRDATA= 23 ; WRITE DATA
1187 000025 RDHEAD= 25 ; READ HEADER
1188 000027 WRHEAD= 27 ; WRITE HEADER AND DATA
1189 000031 WRTCHK= 31 ; WRITE CHECK
1190
1191 000001 GO= BIT0 ; GO BIT
1192 000100 IE= BIT6 ; INTERRUPT ENABLE
1193 000200 RDY= BIT7 ; CONTROLLER READY
1194 000400 BA16= BIT8 ; BUS ADDRESS BIT 16
1195 001000 BA17= BIT9 ; BUS ADDRESS BIT 17
1196 002000 CDT= BIT10 ; CONTROLLER DRIVE TYPE (0=RK06,1=RK07)
1197 004000 CTO= BIT11 ; CONTROLLER TIMEOUT
1198 010000 CFMT= BIT12 ; CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1199 020000 DCPAR= BIT13 ; SERCON PARITY ERROR DETECTED BY CONTROLLER
1200 040000 DI= BIT14 ; DRIVE INTERRUPT
1201 100000 CERR= BIT15 ; CONTROLLER ERROR

```

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 24
CONTROL AND STATUS REGISTER 1 BITS (RKCS1:0)

SEQ 0024

1202	100000	CCLR=	BIT15	;CONTROLLER CLEAR
1203				
1204		.SBTTL	CONTROL AND STATUS REGISTER 2 BITS (RKCS2:10)	
1205				
1206	000007	DRVMSK=	7	;MASK FOR DRIVE SELECTION CODE
1207	000010	RLS=	BIT3	;DESELECT OR RELEASE DRIVE IN BITS 0-2
1208	000020	BAI=	BIT4	;BUS ADDRESS INCREMENT INHIBIT
1209	000040	SCLR=	BIT5	;SUBSYSTEM CLEAR CONTROLLER AND ALL DRIVES
1210	000100	IR=	BIT6	;INPUT READY
1211	000200	OR=	BIT7	;OUTPUT READY
1212	000400	UFE=	BIT8	;UNIT FIELD ERROR
1213	001000	MDS=	BIT9	;MULTIPLE DRIVE SELECT
1214	002000	PGE=	BIT10	;PROGRAMMING ERROR
1215	004000	NEM=	BIT11	;NON-EXISTENT MEMORY
1216	010000	NED=	BIT12	;NON-EXISTENT DRIVE
1217	020000	UPE=	BIT13	;UNIBUS PARITY ERROR
1218	040000	WCE=	BIT14	;WRITE CHECK ERROR
1219	100000	DLT=	BIT15	;DATA LATE ERROR
1220				
1221		.SBTTL	ERROR REGISTER BIT DEFINITION (RKER:14)	
1222				
1223	000001	ILF=	BIT0	;ILLEGAL FUNCTION CODE
1224	000002	SKI=	BIT1	;SEEK INCOMPLETE
1225	000004	NXF=	BIT2	;NON-EXECUTABLE FUNCTION
1226	000010	DRPAR=	BIT3	;DRIVE DETECTED SERCON PARITY ERROR
1227	000020	FMTE=	BIT4	;FORMAT ERROR
1228	000040	DTYPE=	BIT5	;DRIVE TYPE ERROR
1229	000100	ECH=	BIT6	;ECC HARD
1230	000200	BSE=	BIT7	;BAD SECTOR ERROR
1231	000400	HVRC=	BIT8	;HEADER VRC ERROR
1232	001000	COE=	BIT9	;CYLINDER ADDRESS OVERFLOW ERROR
1233	002000	IDAE=	BIT10	;INVALID DISK ADDRESS ERROR: HEAD/CYL
1234	004000	WLE=	BIT11	;WRITE LOCK ERROR
1235	010000	DTE=	BIT12	;DRIVE TIMING ERROR
1236	020000	OPI=	BIT13	;OPERATION (SEARCH) INCOMPLETE
1237	040000	UNS=	BIT14	;DRIVE UNSAFE
1238	100000	DCK=	BIT15	;DATA CHECK
1239				
1240		.SBTTL	STATUS REGISTER BIT DEFINITION (RKDS:12)	
1241				
1242	000001	DRA=	BIT0	;DRIVE AVAILABLE (CONTROLLER IS SET IF THIS BIT IS RESET)
1243				
1244	000004	OFST=	BIT2	;DRIVE OFFSET
1245	000010	ACLO=	BIT3	;AC LOW
1246	000020	DCLO=	BIT4	;DC LOW
1247	000040	DROT=	BIT5	;DRIVE OFF TRACK
1248	000100	VV=	BIT6	;VOLUME VALID
1249	000200	DRDY=	BIT7	;DRIVE READY
1250	000400	DDT=	BIT8	;DRIVE TYPE (0=RK06,1=RK07)
1251	004000	WRL=	BIT11	;WRITE LOCK
1252	020000	PIP=	BIT13	;POSITIONING IN PROGRESS
1253	040000	DSC=	BIT14	;DRIVE STATUS CHANGE
1254	100000	SVAL=	BIT15	;STATUS VALID
1255				
1256		.SBTTL	MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)	
1257				

M02

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 25
MAINTENANCE REGISTER 1 BIT DEFINITION (RKMR1:22)

SEQ 0025

1258	000017	MESMSK= 17	;MESSAGE MASK
1259	000020	PAT= BIT4	;FORCE EVEN PARITY ON SERCON MESSAGE LINES
1260	000040	DMD= BIT5	;DIAGNOSTIC MODE
1261	000100	MSP= BIT6	;MAINTENANCE SECTOR PULSE
1262	000200	MIND= BIT7	;MAINTENANCE INDEX
1263	000400	MCLK= BIT8	;MAINTENANCE CLOCK
1264	001000	MERD= BIT9	;MAINTENANCE ENCODED READ DATA
1265	002000	MEWD= BIT10	;MAINTENANCE ENCODED WRITE DATA
1266	004000	PCA= BIT11	;PRECOMPENSATION ADVANCE
1267	010000	PCD= BIT12	;PRECOMPENSATION DELAY
1268	020000	ECCW= BIT13	;ECC WORD IS BEING READ OR WRITTEN
1269	040000	WRTGAT= BIT14	;WRITE GATE
1270	100000	RDGATE= BIT15	;READ GATE
1271			
1272		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A (RKMR2:34)
1273			
1274	000040	D.DRA= BIT5	;DRIVE AVAILABLE
1275	000100	D.VV= BIT6	;VOLUME VALID
1276	000200	D.DRDY= BIT7	;DRIVE READY
1277	000400	D.DDT= BIT8	;DRIVE TYPE (0=RK06,1=RK07)
1278	001000	D.FORM= BIT9	;DRIVE FORMAT
1279	002000	D.OFF= BIT10	;OFFSET ON
1280	004000	D.WRL= BIT11	;WRITE LOCK
1281	010000	D.SPIN= BIT12	;SPINDLE ON
1282	020000	D.PIP= BIT13	;POSITIONING IN PROGRESS
1283	040000	D.DSC= BIT14	;DRIVE STATUS CHANGE
1284			
1285		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A (RKMR2:34)
1286			
1287	000020	D.SSP= BIT4	;SERVO SIG PRES
1288	000040	D.HDHM= BIT5	;HEADS HOME
1289	000100	D.BRHM= BIT6	;BRUSHES HOME
1290	000200	D.DOOR= BIT7	;DOOR INTERLOCKED
1291	000400	D.CART= BIT8	;CARTRIDGE INTERLOCK
1292	001000	D.SPOK= BIT9	;SPEED OK
1293	002000	D.FWD= BIT10	;FORWARD
1294	004000	D.REV= BIT11	;REVERSE
1295	010000	D.LOAD= BIT12	;HEADS LOADING
1296	020000	D.RTZ= BIT13	;RETURN TO ZERO
1297	040000	D.UNLD= BIT14	;HEADS UNLOADING
1298			
1299		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B (RKMR3:36)
1300			
1301	000040	D.IDAE= BIT5	;INVALID DISK ADDRESS ERROR:HEAD/CYL
1302	000100	D.ACLO= BIT6	;AC LOW
1303	000200	D.FLT= BIT7	;DRIVE FAULT
1304	000400	D.ILF= BIT8	;ILLEGAL FUNCTION CODE
1305	001000	D.PAR= BIT9	;DRIVE DETECTED SERCON PARITY ERROR
1306	002000	D.SKI= BIT10	;SEEK INCOMPLETE
1307	004000	D.WLE= BIT11	;WRITE LOCK ERROR
1308	010000	D.SPLS= BIT12	;SPEED LOSS
1309	020000	D.DROT= BIT13	;DRIVE OFF TRACK
1310	040000	D.UNS= BIT14	;R/W UNSAFE
1311			
1312		.SBTTL	DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)
1313			

N02

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 26
DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B (RKMR3:36)

SEQ 0026

1314 000020
1315 000040
1316 000100
1317 000200
1318 000400
1319 001000
1320 002000
1321 004000
1322 010000
1323 020000
1324 040000
1325
1326
1327
1328 000007
1329 077770
1330
1331
1332
1333 000003
1334 040000
1335 000760
1336 007000
1337 100000

D.SECT= BIT4 ;SECTOR ERROR
D.WCUR= BIT5 ;WRITE CURRENT AND NO WRITE GATE
D.WGAT= BIT6 ;WRITE GATE AND NO TRANSISTIONS
D.HDFL= BIT7 ;HEAD FAULT
D.MHD= BIT8 ;MULTIPLE HEAD SELECT
D.XERR= BIT9 ;INDEX ERROR
D.TIB= BIT10 ;TRIBIT ERROR
D.PLO= BIT11 ;PLO ERROR
D.NMOV= BIT12 ;SEEK AND NO MOTION
D.LIMD= BIT13 ;LIMIT DETECT ON SEEK
D.SUNS= BIT14 ;SERVO UNSAFE

.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE A (RKMR2:34)

M.DRV= 7 ;DRIVE CODE, ALL BYTES
M.SER= 77770 ;DRIVE SERIAL #, BYTE 11

.SBTTL COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)

M.ID= 3 ;BYTE ID, ALL BYTES
M.ALGN= BIT14 ;ALIGN SIGN, BYTE 10
M.SECT= 760 ;SECTOR COUNT, BYTE 11
M.HEAD= 7000 ;HEAD DECODE, BYTE 11
M.PAR= BIT15 ;PARITY, MESS A/B, ALL BYTES

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 27
COMMON MASKS AND OTHER BITS: MESSAGE B (RKMR3:36)

SEQ 0027

```

1338
1339
1340
1341      000000
1342
1343
1344
1345      000174
1346      000174 000000
1347      000176 000000
1348
1349      000200 000137 012556
1350      000220 000220
1351      000220 000137 012546
1352
1353      000240 000240
1354      000240 000137 055500
1355
1356
1357
1358
1359
1360      000244
1361      000046 000046
1362      000046 031600
1363      000052 000052
1364      000052 100000
1365      000244
1366      001000
1367
1368
1369
1370
1371
1372      001000
1373      000024 000024
1374      000024 000200
1375      000044 000044
1376      000044 001000
1377      001000
1378
1379
1380
1381
1382      001000
1383      001000 000000
1384      001002 001210
1385      001004 000430
1386      001006 001130
1387      001010 001130
1388      001012 000042
1389
1390
1391
1392
1393

```

```

.SBTTL TRAF CATCHER

      .=0
      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
      .=174
DISPREG: .WORD 0      ;; SOFTWARE DISPLAY REGISTER
SWREG:   .WORD 0      ;; SOFTWARE SWITCH REGISTER
.SBTTL STARTING ADDRESS(ES)
      JMP 2*START ;; JUMP TO STARTING ADDRESS OF PROGRAM
      .=220
      JMP PARSTR      ; INPUT ALL PARAMETERS & START TESTING
      .=240
      JMP 0.0DT        ; ENTER ODT11

.SBTTL ACT11 HOOKS
      ;*****
      ;HOOKS REQUIRED BY ACT11
      $SVPC=.          ; SAVE PC
      .=46
      SENDAD            ;; 1) SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
      .=52
      .WORD 100000      ;; 2) SET LOC.52 TO 100000
      .=$SVPC          ;; RESTORE PC
      .=1000
.SBTTL APT PARAMETER BLOCK
      ;*****
      ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
      ;*****
      .SX=.            ; SAVE CURRENT LOCATION
      .=24            ; SET POWER FAIL TO POINT TO START OF PROGRAM
      200             ; FOR APT START UP
      .=44            ; POINT TO APT INDIRECT ADDRESS PNTR.
      $APTHDR         ; POINT TO APT HEADER BLOCK
      .=$X            ; RESET LOCATION COUNTER
      ;*****
      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
      ;INTERFACE SPEC.

$APTHD:
$HIBTS: .WORD 0      ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL   ;; ADDRESS OF APT MAILBOX (BITS 0-15)
$TSTM:  .WORD 280.    ;; RUN TIM OF LONGEST TEST
$PASTM: .WORD 600.    ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 600.    ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
      .WORD SETEND-$MAIL/2 ;; LENGTH MAILBOX-ETABLE(WORDS)

      .LIST MD

```

CZR61EO UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 28
APT PARAMETER BLOCK

C03

SEQ 0028

;USE LOOP X TO OMIT SUBCLR
;

.MACRO LOOP A
SCOP1
MOV #STACK,SP ;RESTORE STK PTR

.IF B A
JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR

.ENDC
.ENDM LOOP

; THIS MACRO FILLS EXPECTED MSG AO, BO, A1, B1, A2, B2 & B3 WITH STANDARD BITS
; A=D.DSC AFTER ATTN OR 0 AFTER DRIVE CLEAR OR ANY IMPLIED SEEKS
; NOTE: A CAN BE ANY BIT COMBINATION DESIRED.

.MACRO F.EAB A

MOV #<A!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG AO
CLR E.B0 ;EXPECTED MSG BO
MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
CLR E.A2 ;EXPECTED MSG A2
MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3

.ENDM F.EAB

; THIS MACRO ASSUMES DRIVE MSG AO, BO, A1, B1 WILL ALWAYS BE TESTED
; USE A,C,D,E FOR MSG AO, BO, A1, B1 ERROR NUMBERS RESP.
; USE G=T.A2 TO READ MSG A2 & PUT INFO INTO 'CYLDIF'
; H=T.B2 TO READ MSG B2 & PUT INFO INTO 'CYLADD'
; I=T.B3 TO READ MSG B3 & PUT INFO INTO 'SECTOR' & 'HEADA'

; USE F=<ERROR DESCRIPTION>

.MACRO CHECK A,C,D,E,F,G,H,I

JSR PC,CHKMSG ;CHECK MSGS AO, BO, A1, B1
.WORD G!A!I ;& MSGS SPECIFIED HERE
ERROR A ;MSG AO ERROR F
ERROR C ;MSG BO ERROR
ERROR D ;MSG A1 ERROR
ERROR E ;MSG B1 ERROR

.ENDM CHECK

; A=CYL DIFF/OFFSET ERROR #
; B=CYL ADDR ERROR #
; C=<ERROR DESCRIPTION>

.MACRO CWD2 A,B,C,'D','E'

1394
1395
1396
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
1447
1448
1449

CZR61ED UNIBUS RK6 DR PR2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 29
APT PARAMETER BLOCK

003

SEQ 0029

```
1450      MOV      #2,RKMR1(R5)      ;SELECT WORD 2
1451      JSR      PC,GSTAT
1452      TST      CYLDIF              ;SEE IF MSG A2=0
1453      BEQ      D                    ;BR IF YES
1454      ERROR     A                    ;MSG A2 NOT CLEARED C
1455      D:      TST      CYLADD        ;SEE IF MSG B2=0
1456      BEQ      E                    ;BR IF YES
1457      ERROR     B                    ;MSG B2 NOT CLEARED C
1458      E:
1459      .ENDM     CWD2
1460
1461      .MACRO     DRCLR      ?A
1462
1463      MOV      #CCLR,RKCS1(R5)
1464      MOV      $UNIT,RKCS2(R5) ;DRIVE#
1465      MOV      #CLEAR,HCS1
1466      JSR      PC,DOCMD            ;DO DRIVE CLEAR CMD & GET CONTR RDY
1467      ERROR     151                ;NO RDY AFTER DRIVE CLEAR CMD
1468      JSR      PC,TSTATN           ;TEST FOR ATTN
1469      BR        A
1470      ERROR     154                ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
1471      A:
1472      .ENDM     DRCLR
1473
1474      ;
1475      ;USE CALIB      X      TO OMIT CHECKING MSGS A0, B0, A1, B1, A2 & B2
1476      ;
1477      .MACRO     CALIB      A,?C
1478
1479      MOV      #CCLR,RKCS1(R5)
1480      MOV      $UNIT,RKCS2(R5)
1481      MOV      #RECAL,HCS1
1482      JSR      PC,DOCMD            ;DO RECAL CMD & GET CONTR RDY
1483      ERROR     124                ;RDY NOT SET AFTER RECAL CMD
1484
1485      MOV      #1,RKMR1(R5)      ;SELECT WORD 1
1486      JSR      PC,GSTAT
1487      BIT      #D.RTZ,HMR2
1488      BNE      C
1489      ERROR     244                ;RTZ NOT SET DURING RECAL CMD
1490      C:      MOV      T10,TEMP2    ;SETUP TIMEOUT
1491      JSR      PC,FATT1            ;FIND ATTN
1492      ERROR     55                ;NO ATTN AFTER RECAL CMD
1493
1494      .IF      B
1495      F.EAB      DSC
1496      CHECK      221,275,222,276, <AFTER RECAL CMD>,T.A2,T.B2,T.B3
1497      CWD2      47,50, <AFTER RECAL CMD>
1498      .ENDC
1499      DRCLR
1500
1501      .ENDM     CALIB
1502
1503      ;
1504      ;QUICK START SPINDLE
1505
```

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 30
APT PARAMETER BLOCK

E03

SEQ 0030

```
:MACRO QKSRT
    JSR PC,SUBCLR          ;CERR AFTER SCLR
    ERROR 24
    MOV #SRTSPL,HCS1
    JSR PC,DOCMD           ;DO START SPINDLE CMD & GET CONTR RDY
    ERROR 121              ;RDY NOT SET AFTER ST SPIN CMD.
    MOV T100,TEMP2         ;SETUP TIMEOUT
    JSR PC,FATT1           ;FIND ATTN
    ERROR 74               ;NO ATTN AFTER ST SPIN CMD.
    CLR UNLD
.ENDM QKSRT

: A=WRHEAD/<CFMT!WRHEAD>
: USE WRHDR <A>,X TO OMIT CHECKING AO, BO, A1 & B1
:MACRO WRHDR A,C,K,?D
    MOV #<A>,HCS1
    JSR PC,DATA            ;DO DATA X FOR CMD & GET CONTR RDY
    ERROR 200              ;NO RDY AFTER WRITE HEADER CMD
    .IF B
        JSR PC,GSTAT       ;GET FRESH STATUS
    .ENDC
    .IF NB
        MOV #<CFMT!SELDRA>,HCS1
        JSR PC,DOCMD
        ERROR 117          ;NO RDY AFTER SELDRV CMD
    .ENDC
    BIT #CERR,HCS1
    BEQ D
    ERROR 201              ;CERR AFTER WRITE HEADER CMD
    TYPE MSG26             ;ABORTING BAL OF TESTS
    JMP $EOP
D:
    .IF B
        C
        F.EAB 0
        CHECK 277,267,300,270,<AFTER WRITE HEADER CMD>,0,0,0
    .ENDC
.ENDM WRHDR

: A=RDHEAD/<CFMT!RDHEAD>
: USE RDHDR <A>,X TO OMIT CHECKING AO, BO, A1, B1
:MACRO RDHDR A,C,?D,?E
    MOV #RHTAB,RO
    MOV #<A>,HCS1
```

F03

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 31
APT PARAMETER BLOCK

SEQ 0031

```

1562      JSR      PC,DATCMD      ;DO DATA X FOR CMD & GET CONTR RDY
1563      ERROR    171            ;NO RDY AFTER READ HEADER CMD
1564      BIT      #CERR,HCS1
1565      BEQ       D
1566      ERROR    174            ;CERR AFTER READ HEADER CMD
1567      TYPE      MSG26          ;ABORTING BAL OF TESTS
1568      JMP       $EOF
1569
1570 D:      MOV      RKDB(R5),(R0)+ ;1'ST WORD FROM SILO TO RHTAB
1571      MOV      RKDB(R5),(R0)+ ;2'ND WORD
1572      MOV      RKDB(R5),(R0)+ ;3'RD WORD
1573
1574
1575      BIT      #DLT,RKCS2(R5)
1576      BEQ       E
1577      JSR      PC,GSTAT
1578      ERROR    173            ;DLT AFTER READ HEADER CMD
1579      TYPE      MSG26          ;ABORTING BAL OF TESTS
1580      JMP       $EOF
1581
1582 E:      .IF      B             C
1583      F.EAB      0
1584      CHECK      301,271,302,272, 'AFTER READ HEADER CMD',T.A2,T.B2,0
1585      .ENDC
1586
1587      .ENDM      RDHDR
1588
1589
1590      .MACRO      HDCHK3      ?A
1591
1592      RDHDR      RDHEAD
1593      CMP        RHTAB,TOCYL    ;CHECK WORD 0 ONLY, CYL#
1594      BEQ        A              ;BR IF SAME
1595      ERROR      S1             ;WRONG CYL# ON HEADER
1596
1597 A:
1598
1599      .ENDM      HDCHK3
1600
1601      ; A=TOCYL/FRCYL , B=HEAD#, C = 0 FOR 22 SECTOR, 1 FOR 20 SECTOR
1602
1603      .MACRO      HDTBL      A,B,C
1604
1605      MOV        A,CALADD        ;SETUP
1606      MOV        #B,HEAD        ;TO FILL
1607      MOV        #C,FORMAT      ;HEADER
1608      JSR        PC,FHDTAB      ;TABLE
1609
1610      .ENDM      HDTBL
1611
1612      ; QUICK SEEK. ENTER WITH CYL# IN RKDC
1613
1614      .MACRO      QKSEEK      A
1615
1616      MOV        #SEEK,HCS1
1617

```

G03

CZR6IE0 UNIBUS RK6 DR PR12
CZR6IE.P11 10-JAN-78 09:43MACY11 30A 1052) 10-JAN-78 11:25 PAGE 32
APT PARAMETER BLOCK

SEQ 0032

```

1618          JSR      PC,DOCMD          ;DO SEEK CMD & GET CONTR READY
1619          ERROR    131                ;NO RDY AFTER SEEK CMD
1620
1621          MOV      T50000,TEMP1        ;SETUP TIMEOUT
1622          JSR      PC,FAT12            ;FIND ATTN
1623          ERROR    132                ;NO ATTN AFTER SEEK CMD
1624
1625          BIT      #CERR,HCS1
1626          BEQ      A
1627          ERROR    210                ;CERR AFTER SEEK CMD
1628
1629          A:
1630
1631          .ENDM      QKSEEK
1632
1633          ;A=WRDATA/<CFMT!WRDATA>
1634          ;C=ADDR TO JMP TO ATTEMPT TO WRITE ON ANOTHER SECTOR
1635          ;D=ADDR TO JMP TO BYPASS TEST
1636          ;E: IF BLANK WILL CHECK AD. B0, A1 & B1 AT THE END OF WRITING
1637          ;E: IF NON BLANK WILL OMIT CHECKING AD THRU B1
1638          ;
1639
1640          .MACRO      WDATA      A,C,D,E,J,K,?F,?G,?H,?I
1641
1642          MOV      #<A>,HCS1
1643          JSR      PC,DATCMD          ;DO DATA X FOR CMD & GET CONTR RDY
1644          ERROR    11                ;NO RDY AFTER WRITE DATA CMD
1645          .IF      B
1646          JSR      PC,GSTAT          ;GET FRESH STATUS
1647          .ENDC
1648          .IF      NB
1649          MOV      #<CFMT!SELDV>,HCS1
1650          JSR      PC,DOCMD
1651          ERROR    117                ;NO RDY AFTER SELDRV CMD
1652          .ENDC
1653          BIT      #CERR,HCS1
1654          BEQ      I                ;BR IF NO ERRORS
1655
1656          BIT      #BSE,HER          ;SEE IF BAD SECTOR FLAG
1657          BEQ      G                ;BR IF NO
1658          JSR      PC,TRUEERR        ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
1659          BR      H                ;RETURN HERE IF NO
1660          .IF      B
1661          INC      SECTOR            ;RETURN HERE IF YES
1662          CMP      SECTOR,#10.        ;ARE 10 CONSEC. SECTORS BAD
1663          BNE      F                ;BR IF NO
1664          ERROR    46                ;ABORTING TEST DETECTED 10 BAD SECTORS
1665          JMP      D                ;BYPASS TEST
1666          F:      MOV      #CCLR,RKCS1(R5) ;TRY ANOTHER SECTOR
1667          JMP      C
1668
1669          .ENDC
1670          .IF      NB
1671          JMP      J
1672          .IF      J
1673          JMP      3$                ;RET HERE IF YES

```

CZR6IEO UNIBUS Rk6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 -JA(1052) 10-JAN-78 11:25 PAGE 33
APT PARAMETER BLOCK

H03

SEQ 0033

1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729

```
.ENDC
G:  ERROR 12 ;CERR WITH WRITE DATA CMD
    F.EAB 0
    CHECK 52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
    TYPE MSG26 ;ABORTING BAL OF TESTS
    JMP $EOP
H:  ERROR 63 ;BAD SECTOR NOT LISTED IN TABLE
I:
.IF B E
    F.EAB 0
    CHECK 52,23,53,25,<AFTER WRITE DATA CMD>,T.A2,T.B2,0
.ENDC
.ENDM WDATA

;A=RDDATA/<CFMT!RDATA>
;USE RDATA <A>,X TO OMIT CHECKING AO, BU, AI & BI
;
.MACRO RDATA A,C,K,?D,?E,?F,?G,?H
    MOV #<A>,HCS1
    JSR PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
    ERROR 13 ;NO RDY AFTER READ DATA CMD
.IF B K
    JSR PC,GSTAT ;GET FRESH STATUS
.ENDC
.IF NB K
    MOV #<CFMT!SELDRV>,HCS1
    JSR PC,DOCMD
    ERROR 11? ;NO RDY AFTER SELDRV CMD
.ENDC
    BIT #CERR,HCS1
    BEQ G
    BIT #BSE,HER ;SEE IF BAD SECTOR
    BEQ E
    ERROR 65 ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
    BR H
D:  TYPE MSG26 ;ABORTING BAL OF TESTS
    JMP $EOP
E:  BIT #DCK,HER ;SEE IF DATA CHECK ERROR
    BEQ F
    ERROR 21 ;DATA CHECK ERROR AFTER READ CMD (ECC)
    BR H
F:  ERROR 14 ;CERR AFTER READ DATA CMD.
H:  F.EAB 0
    CHECK 54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
    BR D
G:
.IF B C
    F.EAB 0
    CHECK 54,26,56,30,<AFTER READ DATA CMD>,T.A2,T.B2,0
```

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 34
APT PARAMETER BLOCK

I03

SEQ 0034

```
.ENDC
.ENDM  RDATA

;A=WRCHK/(CFMT!WRCHK)
;C=EXPECTED DATA FOR TYPEOUT
;USE WRCHK      <A>,DATA0,X TO OMIT CHECKING A0, B0, A1 & B1
;

.MACRO  WRCHK  A,C,D,K,?E,?F

      MOV      #<A>,HCS1
      JSR      PC,DATCMD      ;DO DATA X FOR CMD & GET CONTR RDY
      ERROR    15              ;NO RDY AFTER WRITE CHECK CMD
      .IF      B              K
      JSR      PC,GSTAT      ;GET FRESH STATUS
      .ENDC
      .IF      NB            K
      MOV      #<CFMT!SELDRV>,HCS1
      JSR      PC,DOCMD
      ERROR    11?              ;NO RDY AFTER SELDRV CMD
      .ENDC
      BIT      #CERR,HCS1
      BEQ      F
      BIT      #WCE,HCS2      ;SEE IF WRITE CHECK ERROR
      BEQ      E
      MOV      RKDB(R5),WD1    ;ACTUAL WORD FOR PRINTOUT
      MOV      C,WD2           ;EXPECTED WORD FOR TYPEOUT
      ERROR    16              ;WCE AFTER WRITE CMD
      BR      F

E:      ERROR    22              ;CERR AFTER WRITE CHECK CMD
      F.EAB    0
      CHECK    57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
      TYPE     MSG26           ;ABORTING BAL OF TESTS
      JMP      $EOP

F:
      .IF      B              D
      F.EAB    0
      CHECK    57,31,60,32,<AFTER WRITE CHECK CMD>,T.A2,T.B2,0
      .ENDC
      .ENDM  WRCHK

.MACRO  OFFSET ?A

      MOV      #A,$ESCAPE
      MOV      #OFFSET,HCS1
      JSR      PC,DOCMD      ;DO RECAL CMD & GET CONTR RDY
      ERROR    33              ;NO RDY AFTER OFFSET CMD

      MOV      #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
      CLR      E.B0
      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
```

CZR61EO UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 35
APT PARAMETER BLOCK

SEQ 0035

```

1786      MOV      #1,E.B1
1787      CHECK    35,61,36,62,<DURING OFFSET CMD>.0,0,0
1788
1789      A:      CLR      $ESCAPE
1790      MOV      T5000,TEMP1      ;SETUP TIMEOUT
1791      JSR      PC,FATT2      ;FIND ATTN
1792      ERROR    34      ;NO ATTN AFTER OFFSET CMD
1793
1794      F.EAB     <D.DSC!D.OFF>
1795      CHECK    260,261,37,40,<AFTER OFFSET CMD>.T.A2,T.B2,0
1796
1797      .ENDM     OFFSET
1798
1799
1800      .MACRO    EOPGM
1801
1802      SCOPE
1803      CLR      $ESCAPE
1804      MOV      #1,$TIMES
1805      MOV      $STACK,SP
1806      INC      $DEVCT      ;INCR COUNT FOR # DRIVES CHECKED
1807      CMP      DRIVS,$DEVCT ;ARE ALL DRIVES PRESENT TESTED?
1808      BEQ      $EOP1+2      ;BR IF YES
1809      JMP      NUDRV      ;ELSE TEST NEXT DRIVE PRESENT
1810      $EOP1:    SCOPE
1811      .ENDM     EOPGM
1812
1813      ;A= ERROR #
1814      ;B = ERROR CONDITION
1815
1816      .MACRO    OFFDIR  A,B,?C,?D
1817      MOV      R2,RKASOF(R5) ;REFRESH RKASOF
1818
1819      BIT      #BIT7,R2
1820      BNE      C      ;BR IF NEG OFFSET
1821
1822      CMP      R2,CYLDIF ;CHECK POS OFFSET
1823      BEQ      D
1824      ERROR    A      ;OFFSET IN A2 NOT = RKASOF
1825      BR      D      ;B
1826
1827      C:      CMP      R1,CYLDIF ;CHECK NEG OFFSET
1828      BEQ      D
1829      ERROR    A      ;OFFSET IN A2 NOT = RKASOF
1830      ;B
1831      D:
1832      .ENDM     OFFDIR
1833
1834      .NLIST    MD
1835

```

K03

1836
1837
1838
1839
1840
1841
1842 001100
1843 001100
1844 001100 000000
1845 001102 000
1846 001103 000
1847 001104 000000
1848 001106 000000
1849 001110 000000
1850 001112 000000
1851 001114 000
1852 001115 001
1853 001116 000000
1854 001120 000000
1855 001122 000000
1856 001124 000000
1857 001126 000000
1858 001130 000000
1859 001132 000000
1860 001134 000
1861 001135 000
1862 001136 000000
1863 001140 177570
1864 001142 177570
1865 001144 177560
1866 001146 177562
1867 001150 177564
1868 001152 177566
1869 001154 000
1870 001155 002
1871 001156 012
1872 001157 000
1873 001160 000000
1874 001162 000000
1875 001164 000000
1876 001166 000000
1877 001170 000000
1878 001172 000000
1879 001174 000000
1880 001176 000000
1881 001200 177607 000377
1882 001204 077
1883 001205 015
1884 001206 000012
1885
1886
1887
1888
1889
1890 001210
1891 001210 000000

.SBTTL COMMON TAGS

;*****
;THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
;USED IN THE PROGRAM.

SCMTAG: . =1100

.WORD 0
\$TSTNM: .BYTE 0
\$ERFLG: .BYTE 0
\$ICNT: .WORD 0
\$LPADR: .WORD 0
\$LPERR: .WORD 0
\$ERTTL: .WORD 0
\$ITEMB: .BYTE 0
\$ERMAX: .BYTE 1
\$ERRPC: .WORD 0
\$GDADR: .WORD 0
\$BDADR: .WORD 0
\$GDDAT: .WORD 0
\$BDDAT: .WORD 0
\$AUTOB: .BYTE 0
\$INTAG: .BYTE 0
\$SWR: .WORD DSWR
DISPLAY: .WORD DDISP
\$TKS: 177560
\$TKB: 177562
\$TPS: 177564
\$TPB: 177566
\$NULL: .BYTE 0
\$FILLS: .BYTE 2
\$FILLC: .BYTE 12
\$TPFLG: .BYTE 0
\$TMP0: .WORD 0
\$TMP1: .WORD 0
\$TMP2: .WORD 0
\$TMP3: .WORD 0
\$TMP4: .WORD 0
\$TMP5: .WORD 0
\$TIMES: 0
\$ESCAPE: 0
\$BELL: .ASCIZ <207><377><377>
\$QUES: .ASCII /?/
\$CRLF: .ASCII <15>
\$LF: .ASCIZ <12>

;;START OF COMMON TAGS

;;CONTAINS THE TEST NUMBER
;;CONTAINS ERROR FLAG
;;CONTAINS SUBTEST ITERATION COUNT
;;CONTAINS SCOPE LOOP ADDRESS
;;CONTAINS SCOPE RETURN FOR ERRORS
;;CONTAINS TOTAL ERRORS DETECTED
;;CONTAINS ITEM CONTROL BYTE
;;CONTAINS MAX. ERRORS PER TEST
;;CONTAINS PC OF LAST ERROR INSTRUCTION
;;CONTAINS ADDRESS OF 'GOOD' DATA
;;CONTAINS ADDRESS OF 'BAD' DATA
;;CONTAINS 'GOOD' DATA
;;CONTAINS 'BAD' DATA
;;RESERVED--NOT TO BE USED
;;AUTOMATIC MODE INDICATOR
;;INTERRUPT MODE INDICATOR
;;ADDRESS OF SWITCH REGISTER
;;ADDRESS OF DISPLAY REGISTER
;;TTY KBD STATUS
;;TTY KBD BUFFER
;;TTY PRINTER STATUS REG. ADDRESS
;;TTY PRINTER BUFFER REG. ADDRESS
;;CONTAINS NULL CHARACTER FOR FILLS
;;CONTAINS # OF FILLER CHARACTERS REQUIRED
;;INSERT FILL CHARS. AFTER A "LINE FEED"
;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
;;USER DEFINED
;;USER DEFINED
;;USER DEFINED
;;USER DEFINED
;;USER DEFINED
;;USER DEFINED
;;MAX. NUMBER OF ITERATIONS
;;ESCAPE ON ERROR ADDRESS
;;CODE FOR BELL
;;QUESTION MARK
;;CARRIAGE RETURN
;;LINE FEED

;*****
.SBTTL APT MAILBOX-ETABLE;*****
\$EVEN
\$MAIL: ;APT MAILBOX
\$MSGTY: .WORD MSGTY ;MESSAGE TYPE CODE

```

AFATAL: .WORD AFATAL ;; FATAL ERROR NUMBER
$TESTN: .WORD ATESTN ;; TEST NUMBER
$PASS: .WORD APASS ;; PASS COUNT
$DEVCT: .WORD ADEVCT ;; DEVICE COUNT
$UNIT: .WORD AUNIT ;; I/O UNIT NUMBER
$MSGAD: .WORD AMSGAD ;; MESSAGE ADDRESS
$MSGLG: .WORD AMSGLG ;; MESSAGE LENGTH
$ETABLE: .WORD APTENV ;; APT ENVIRONMENT TABLE
$ENV: .BYTE AENV ;; ENVIRONMENT BYTE
$ENVM: .BYTE AENVM ;; ENVIRONMENT MODE BITS
$SWREG: .WORD ASWREG ;; APT SWITCH REGISTER
$USWR: .WORD AUSWR ;; USER SWITCHES
$CPUOP: .WORD ACPUOP ;; CPU TYPE, OPTIONS
; *
; *
; *
; *
; *
; *
$MAMS1: .BYTE AMAMS1 ;; HIGH ADDRESS, M.S. BYTE
$MTYP1: .BYTE AMTYP1 ;; MEM. TYPE, BLK#1
; *
; *
; *
; *
$MADR1: .WORD AMADR1 ;; HIGH ADDRESS, BLK#1
; *
$MAMS2: .BYTE AMAMS2 ;; HIGH ADDRESS, M.S. BYTE
$MTYP2: .BYTE AMTYP2 ;; MEM. TYPE, BLK#2
$MADR2: .WORD AMADR2 ;; MEM. LAST ADDRESS, BLK#2
$MAMS3: .BYTE AMAMS3 ;; HIGH ADDRESS, M.S. BYTE
$MTYP3: .BYTE AMTYP3 ;; MEM. TYPE, BLK#3
$MADR3: .WORD AMADR3 ;; MEM. LAST ADDRESS, BLK#3
$MAMS4: .BYTE AMAMS4 ;; HIGH ADDRESS, M.S. BYTE
$MTYP4: .BYTE AMTYP4 ;; MEM. TYPE, BLK#4
$MADR4: .WORD AMADR4 ;; MEM. LAST ADDRESS, BLK#4
$VECT1: .WORD AVECT1 ;; INTERRUPT VECTOR#1 BUS PRIORITY#1
$VECT2: .WORD AVECT2 ;; INTERRUPT VECTOR#2 BUS PRIORITY#2
$BASE: .WORD ABASE ;; BASE ADDRESS OF EQUIPMENT UNDER TEST
$DEV: .WORD ADEV ;; DEVICE MAP
$CDW1: .WORD ACDW1 ;; CONTROLLER DESCRIPTION WORD#1
$CDW2: .WORD ACDW2 ;; CONTROLLER DESCRIPTION WORD#2
$DDW0: .WORD ADDW0 ;; DEVICE DESCRIPTOR WORD#0
$DDW1: .WORD ADDW1 ;; DEVICE DESCRIPTOR WORD#1
$DDW2: .WORD ADDW2 ;; DEVICE DESCRIPTOR WORD#2
$DDW3: .WORD ADDW3 ;; DEVICE DESCRIPTOR WORD#3
$DDW4: .WORD ADDW4 ;; DEVICE DESCRIPTOR WORD#4
$DDW5: .WORD ADDW5 ;; DEVICE DESCRIPTOR WORD#5
$DDW6: .WORD ADDW6 ;; DEVICE DESCRIPTOR WORD#6
$DDW7: .WORD ADDW7 ;; DEVICE DESCRIPTOR WORD#7
$ETEND:
.MEXIT
ABASE= 177440 ; DEFAULT BUSS ADDRESS
RKVEC: 210 ; DEFAULT CONTROLLER INTERRUPT VECTOR
RKPRI: PR5 ; PRIORITY
PKS: 172540 ; P-CLOCK STATUS REG

```

M03

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 38
APT MAILBOX-ETABLE

SEQ 0038

1948	001322	172542	PKSB:	172542	;P-CLOCK SET BUFFER
1949	001324	172544	PKRB:	172544	;P-CLOCK READ BUFFER
1950	001326	177546	LKS:	177546	;L-CLOCK STATUS REG.
1951					
1952	001330	000100	LCVEC:	100	;L-CLOCK INTERRUPT VECTOR
1953	001332	000104	PCVEC:	104	;P-CLOCK INTERRUPT VECTOR.
1954					
1955		000114	MEMVEC=	114	;MEMORY PARITY VECTOR
1956		172100	MEMBAS=	172100	;MEMORY PARITY OPTION CSR START ADDR
1957	001334	000000	TRAPPC:	0	;PC FOR MEM CHECK ENABLE TRAP
1958					
1959	001336	000000	PARAM:	0	;1 FOR 220 START, NO DEFAULT
1960	001340	000000	FTITLE:	0	;FLAG FOR PRINTING OUT 1ST PROGRAM TITLE
1961					
1962	001342	000000	DRVPTR:	0	;CONTAINS THE POINTER TO THE DRIVE FLAG
1963					; (DRIVO-DRIV7) OF THE DRIVE TO BE CHECKED NEXT.
1964	001344	000000	FRCYL:	0	;FROM CYLINDER
1965	001346	000000	TOCYL:	0	;TO CYLINDER
1966	001350	000000	CCYL:	0	;CURRENT CYL, USED IN N SQUARE TEST
1967	001352	000000	PCYL:	0	;PREV CYL, USED IN N SQUARE TEST
1968	001354	000000	CALDIF:	0	;CALC CYL DIFF USED IN N SQUARE TEST
1969	001356	000000	CYLDIF:	0	;CYL DIFF, RIGHT JUSTIFIED FROM RKMR3
1970	001360	000000	CYLADD:	0	;CYL ADDR, RIGHT JUSTIFIED FROM RKMP3
1971	001362	000000	CALADD:	0	;CYL ADDR USED IN FHD TAB ROUTINE
1972					
1973	001364	000074	HZ:	60.	;60 FOR 60 CPS
1974					;50 FOR 50 CPS
1975	001366	000000	COUNT:	0	;LOADED TO 50 OR 60 TO COUNT TO 1 SEC
1976					;OR ANY OTHER NUMBER TO COUNT OFF FRACTIONAL SECOND
1977	001370	000000	SEC:	0	;SECOND COUNTER
1978	001372	000000	TIMUP:	0	;FLAG TO INDICATE TIME IS UP
1979	001374	000000	SECNT:	0	;SECTOR COUNT
1980	001376	000000	PSEC:	0	;PREVIOUS SECTOR
1981	001400	000000	ESEC:	0	;EXPECTED SECTOR
1982	001402	000000	SECTOR:	0	;SECTOR COUNT, RIGHT JUSTIFIED FROM RKMR3
1983					
1984	001404	000001	T1:	1	;TIMEOUT CONSTANTS
1985	001406	000012	T10:	10.	
1986	001410	000062	T50:	50.	
1987	001412	000764	T500:	500.	
1988	001414	000144	T100:	100.	
1989	001416	011610	T5000:	5000.	
1990	001420	141520	T50000:	50000.	
1991					
1992	001422	000077	CYL:	63.	;CYLINDER NUMBERS USED IN
1993	001424	000177		127.	;CURRENT CROSSOVER TEST FOR RK06
1994	001426	000277		191.	
1995	001430	000377		255.	
1996	001432	000477		319.	
1997	001434	000577		383.	
1998					
1999	001436	000177	CYL7:	127.	;FOR RK07
2000	001440	000377		255.	
2001	001442	000577		383.	
2002	001444	000777		511.	
2003	001446	001177		639.	

N03

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 39
APT MAILBOX-ETABLE

SEQ 0039

2004	001450	001377		767.	
2005					
2006	001452	000000	WD1:	0	;ACTUAL HEADER/DATA WORD
2007	001454	000000	WD2:	0	;EXPECTED DATA WORD
2008					
2009	001456	000000	OFFERR:	0	;SET WHEN WRITE CHECK ERROR ON OFFSET
2010					
2011					
2012	001460	000000	HEAD:	0	;HEAD NUMBER
2013	001462	000000	HEAD:	0	;HEAD # FROM H.B3, RT. JUSTIFIED
2014	001464	000000	HD1:	0	;SHIFTED HEAD# FOR FORMATTER ROUTINE
2015	001466	000000	FORMAT:	0	;FORMAT TYPE
2016	001470	000000	FMT1:	0	;SHIFTED FORMAT FOR FORMATTER ROUTINE
2017	001472	000000	WDCNT:	0	;WORD COUNT
2018					
2019	001474	000000	DATA0:	0	;ALL 0'S
2020	001476	052525	DATA01:	52525	;0101 PATT
2021	001500	177777	DATA1:	177777	;ALL 1'S
2022	001502	133467	DPAT1:	133467	
2023	001504	070627	DPAT2:	70627	
2024					
2025	001506	000000	WORD:	0	;HEADER/DATA WORD
2026	001510	000000	HOWD:	0	;HEADER WORD FROM RKDB
2027					
2028	001512	000000	BSERR:	0	;CANNOT READ BSE INFO WHEN SET
2029	001514	000000	LIMERR:	0	;LIMIT DETECT ERROR FLAG
2030	001516	000000	BYPCERR:	0	;SET TO 1 TO BYPASS CKCERR IN 'GSTAT1'
2031	001520	000000	CHKFLG:	0	;WORDS TO BE TESTED
2032					
2033	001522	000102	HDTAB:	.BLKW 66.	;CALCULATED HEADER WORD TABLE
2034	001726	000102	RHTAB:	.BLKW 66.	;FILLED AFTER READ HEADER CMD
2035	002132	000102	SRTTAB:	.BLKW 66.	;ABOVE RHTAB SORTED STARTING FORM
2036					;SECTOR 0 BY SORT ROUTINE
2037	002336	000400	BSE20H:	.BLKW 256.	;20 SECTOR HARDWARE BSE INFO
2038	003336	000400	BSE22H:	.BLKW 256.	;22 SECTOR HARDWARE BSE INFO.
2039	004336	000400	BSE20S:	.BLKW 256.	;20 SECTOR SOFTWARE BSE INFO.
2040	005336	000400	BSE22S:	.BLKW 256.	;22 SECTOR SOFTWARE BSE INFO.
2041	006336	000400	RDAB:	.BLKW 256.	;FILLED AFTER READ DATA CMD
2042					
2043	007336	000000	UNLD:	0	;SET TO 0 IF HEADS ARE LOADED
2044					;SET TO 1 IF HEADS UNLOADED
2045	007340	000000	BADHDR:	0	;SET TO 0 IF FORMATTING OK
2046					;SET TO 1 IF FORMATTING ALTERED
2047	007342	000000	HPEND:	0	;SET TO 0 IF HALT NOT PENDING
2048					;SET TO 1 IF HALT PENDING
2049					
2050					;THE ABOVE 3 FLAGS ARE USED
2051					;BY 'STOP' ROUTINE TO BRING
2052					;THE CPU TO A VALID HALT.
2053					
2054					
2055	007344	001	002	004	ATTN: .BYTE 1,2,4,10,20,40,100,200 ;ATN 0-7 RESP.
2056	007347	010	020	040	
2057	007352	100	200		
2058					.EVEN
2059					

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 40
APT MAILBOX-ETABLE

SEQ 0040

```

2060
2061
2062
2063
2064
2065 007354 000000
2066 007356 000000
2067 007360 000000
2068 007362 000000
2069 007364 000000
2070 007366 000000
2071 007370 000000
2072 007372 000000
2073 007374 000000
2074 007376 000000
2075 007400 000000
2076 007402 000000
2077 007404 000000
2078 007406 000000
2079 007410 000000
2080
2081
2082 007412 000000
2083 007414 000000
2084 007416 000000
2085 007420 000000
2086 007422 000000
2087
2088
2089
2090 007424 000000
2091 007426 000000
2092 007430 000000
2093 007432 000000
2094 007434 000000
2095 007436 000000
2096 007440 000000
2097 007442 000000
2098
2099
2100
2101 007444 000000
2102 007446 000000
2103 007450 000000
2104 007452 000000
2105 007454 000000
2106 007456 000000
2107 007460 000000
2108 007462 000000
2109
2110
2111
2112 000001
2113 000002
2114 000004
2115

```

```

;THE FOLLOWING ARE HOLDING REGISTERS FOR THE RK611 REGISTERS
;THEY ARE LOADED AFTER PDY IS REC'D FROM WRDY ROUTINE.
;
HCS1: 0 ;HOLD RKCS1
HCS2: 0 ;HOLD RKCS2
HWC: 0 ;HOLD RKWC
HBA: 0 ;ETC.
HDA: 0
HDS: 0
HER: 0
HASOF: 0
HDC: 0
HDB: 0
HMR1: 0
HMR2: 0
HMR3: 0
HPOS: 0
HPAT: 0

TEMP1: 0 ;TEMPORARY STORAGE.
TEMP2: 0
TEMP3: 0
TEMP4: 0
TEMP5: 0

;THE FOLLOWING ARE HOLDING REGISTERS FOR MSGA(0-3) & MSGB(0-3).
;
H.A0: 0
H.B0: 0
H.A1: 0
H.B1: 0
H.A2: 0
H.B2: 0
H.A3: 0
H.B3: 0

;THE FOLLOWING ARE 'EXPECTED' REGISTER FOR THE ABOVE.
;
E.A0: 0
E.B0: 0
E.A1: 0
E.B1: 0
E.A2: 0
E.B2: 0
E.A3: 0
E.B3: 0

;THE FOLLOWING ARE IDENTIFIERS FOR DRIVE MSG WORDS TO BE TESTED.
;
T.A2=BIT0 ;TEST MSG A2 IF SET
T.B2=BIT1
T.B3=BIT2

```

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 41
APT MAILBOX-ETABLE

C04

SEQ 0041

```
2116  
2117  
2118  
2119  
2120 007464 000000  
2121 007466 000000  
2122 007470 000000  
2123 007472 000000  
2124 007474 000000  
2125  
2126  
2127  
2128  
2129 007476 000000  
2130 007500 000000  
2131 007502 000000  
2132 007504 000000  
2133 007506 000000  
2134 007510 000000  
2135 007512 000000  
2136 007514 000000  
2137  
2138 007516 000000  
2139 007520 000000  
2140 007522 000000  
2141 007524 000000
```

;
; ALL THE FLAGS BELOW ARE CLEARED INITIALLY BY THE CLRFLG ROUTINE.
;
DDUMP: 0 ; FLAG - SET WHEN IN DDP DUMP MODE
DDPCH: 0 ; FLAG - SET WHEN IN DDP CHAIN MODE
ACT11: 0 ; FLAG - SET WHEN IN ACT11 MODE OF OPERATION
PPTP: 0 ; FLAG - SET WHEN PROGRAM LOADED BY PAPER TAPE
DRVS: 0 ; CONTAINS THE NUMBER OF DRIVES PRESENT

;
; THE FLAGS BELOW ARE SET TO 1 TO INDICATE THAT A PARTICULAR DRIVE
; IS PRESENT AND IS TO BE TESTED.
;
DRIVO: 0 ; FLAG SET TO 1 WHEN DRIVE 0 PRESENT
DRIV1: 0 ; FOR DRIVE 1
DRIV2: 0 ; FOR DRIVE 2
DRIV3: 0 ; FOR DRIVE 3
DRIV4: 0 ; FOR DRIVE 4
DRIV5: 0 ; FOR DRIVE 5
DRIV6: 0 ; FOR DRIVE 6
DRIV7: 0 ; FOR DRIVE 7

LCLKF: 0 ; L-CLOCK FLAG PRESENT FLAG
PCLKF: 0 ; P-CLOCK FLAG PRESENT FLAG
DOTIM: 0 ; SET IF EITHER CLOCK PRESENT FOR TIMING TESTS.
SIZFLG: 0 ; SET IF DEFAULT DO SIZING IN TEST 1

.SBTTL ERROR POINTER TABLE

;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;*LOCATION \$ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF \$ITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;* EM ;;POINTS TO THE ERROR MESSAGE
;* DH ;;POINTS TO THE DATA HEADER
;* DT ;;POINTS TO THE DATA
;* DF ;;POINTS TO THE DATA FORMAT

\$ERRTB:

;ERROR 1
EM2 ;DR # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RkMR2
DH1
DT1
DF1
;ERROR 2
EM5 ;DETECTED MDS
DH1
DT1
DF1
;ERROR 3
EM6 ;DETECTED UFE
DH1
DT1
DF1
;ERROR 4
EM7 ;DETECTED DRA & NED RESET (WRONG PORT SELECTED?)
DH1
DT1
DF1
;ERROR 5
0
0
0
0
;ERROR 6
EM9 ;DR NOT PRESENT BUT SPECIFIED BY OPERATOR
DH1
DT1
DF1
;ERROR 7
EM10 ;ABORT TEST, COULD NOT REFERENCE CONTROLLER REGISTER
DH1
DT1
DF1

2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156 007526
2157
2158
2159 007526 046213
2160 007530 051647
2161 007532 053732
2162 007534 054534
2163
2164
2165 007536 046432
2166 007540 051647
2167 007542 053732
2168 007544 054534
2169
2170
2171 007546 046453
2172 007550 051647
2173 007552 053732
2174 007554 054534
2175
2176
2177 007556 046474
2178 007560 051647
2179 007562 053732
2180 007564 054534
2181
2182 007566 000000
2183 007570 000000
2184 007572 000000
2185 007574 000000
2186
2187
2188 007576 046637
2189 007600 051647
2190 007602 053732
2191 007604 054534
2192
2193
2194 007606 046713
2195 007610 051647
2196 007612 053732
2197 007614 054534

E04

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 43
ERROR POINTER TABLE

SEQ 0043

2198				
2199			:ERROR 10	
2200	007616	046776	EM11	:DRA & NED BOTH SET
2201	007620	051647	DH1	
2202	007622	053732	DT1	
2203	007624	054534	DF1	
2204			:ERR 11	
2205	007626	047042	EM12	:NO RDY
2206	007630	052632	DH27	:AFTER WRITE DATA CMD
2207	007632	053732	DT1	
2208	007634	054664	DF10	
2209			:ERR 12	
2210	007636	047436	EM21	:CERR SET
2211	007640	052632	DH27	
2212	007642	053732	DT1	
2213	007644	054664	DF10	
2214			:ERR 13	
2215	007646	047042	EM12	:NO RDY
2216	007650	052602	DH26	:AFTER READ DATA CMD
2217	007652	053732	DT1	
2218	007654	054664	DF10	
2219			:ERR 14	
2220	007656	047436	EM21	:CERR SET
2221	007660	052602	DH26	
2222	007662	053732	DT1	
2223	007664	054664	DF10	
2224			:ERR 15	
2225	007666	047042	EM12	:NO RDY
2226	007670	052762	DH32	:AFTER WRITE CHECK CMD
2227	007672	053732	DT1	
2228	007674	054664	DF10	
2229			:ERR 16	
2230	007676	050624	EM80	:WRITE CHECK ERROR SET
2231	007700	052762	DH32	:AFTER WRITE CHECK CMD
2232	007702	054044	DT6	
2233	007704	054554	DF3	
2234			:ERR 17	
2235	007706	050663	EM81	:WRITE CHECK CMD NOT FUNCTIONING
2236	007710	053511	DH52	:WITH INTENTIONAL MISCOMPARE
2237	007712	053732	DT1	
2238	007714	054664	DF10	
2239			:ERR 20	
2240	007716	050727	EM82	:READ DATA NOT COMPARE WITH WRITE DATA
2241	007720	052602	DH26	:AFTER READ DATA CMD
2242	007722	054044	DT6	
2243	007724	054554	DF3	
2244			:ERR 21	
2245	007726	051001	EM83	:DATA CHECK ERROR
2246	007730	052602	DH26	
2247	007732	053732	DT1	
2248	007734	054664	DF10	
2249			:ERR 22	
2250	007736	047436	EM21	:CERR SET
2251	007740	052762	DH32	:AFTER WRITE CHECK CMD
2252	007742	053732	DT1	
2253	007744	054664	DF10	

F04

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 44
ERROR POINTER TABLE

SEQ 0044

2254			:ERR 23	EM18	:MSG B0 ERROR
2255	007746	047353		DH27	:AFTER WRITE DATA CMD
2256	007750	052632		DT13	
2257	007752	054270		DF21	
2258	007754	055014			
2259			:ERROR 24	EM21	:CERR SET
2260	007756	047436		DH21	:AFTER SCLR
2261	007760	052452		DT1	
2262	007762	053732		DF10	
2263	007764	054664			
2264			:ERR 25	EM20	:MSG B1 ERROR
2265	007766	047415		DH27	
2266	007770	052632		DT13	
2267	007772	054270		DF21	
2268	007774	055014			
2269			:ERR 26	EM18	
2270	007776	047353		DH26	:AFTER READ DATA CMD
2271	010000	052602		DT13	
2272	010002	054270		DF21	
2273	010004	055014			
2274			:ERROR 27	EM24	:VOL VALID NOT SET
2275				DH19	:AFTER PACK CMD
2276	010006	047665		DT1	
2277	010010	052374		DF10	
2278	010012	053732			
2279	010014	054664			
2280			:ERR 30	EM20	:MSG B1 ERROR
2281	010016	047415		DH26	:AFTER READ DATA CMD.
2282	010020	052602		DT13	
2283	010022	054270		DF21	
2284	010024	055014			
2285			:ERR 31	EM18	:MSG B0 ERROR
2286	010026	047353		DH32	:AFTER WRITE CHECK CMD
2287	010030	052762		DT13	
2288	010032	054270		DF21	
2289	010034	055014			
2290			:ERR 32	EM20	:MSG B1 ERROR
2291	010036	047415		DH32	
2292	010040	052762		DT13	
2293	010042	054270		DF21	
2294	010044	055014			
2295			:ERR 33	EM12	:CONTR NOT READY
2296	010046	047042		DH24	:AFTER OFFSET CMD
2297	010050	052532		DT1	
2298	010052	053732		DF10	
2299	010054	054664			
2300			:ERR 34	EM13	:NO ATTN
2301	010056	047100		DH24	
2302	010060	052532		DT1	
2303	010062	053732		DF10	
2304	010064	054664			
2305			:ERR 35	EM17	:MSG A0 ERROR
2306	010066	047332		DH53	:DURING OFFSET COMMAND
2307	010070	053545		DT13	
2308	010072	054270		DF21	
2309	010074	055014			

G04

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 45
ERROR POINTER TABLE

SEQ 0045

2310
2311 010076 047374
2312 010100 053545
2313 010102 054270
2314 010104 055014
2315
2316 010106 047374
2317 010110 052532
2318 010112 054270
2319 010114 055014
2320

:ERR 36

EM19
DH53
DT13
DF21

:MSG A1 ERROR

:ERR 37

EM19
DH24
DT13
DF21

:MSG A1 ERROR
:AFTER OFFSET CMD

:ERR 40

CZR6IEO UNIBUS RK6 DR PR12
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 46
ERROR POINTER TABLE

H04

SEQ 0046

2321	010116	047415	EM20	:MSG B1 ERROR
2322	010120	052243	DH24	
2323	010122	054270	DT13	
2324	010124	055014	DF21	
2325			:ERR 41	
2326	010126	047122	EM14	:UNEXP MEM PCRTY TRAP
2327	010130	052040	DH8	:TEST #, TRAP PC
2328	010132	053772	DT3	
2329	010134	054550	DF2	
2330			:ERR 42	
2331	010136	050340	EM41	:CYL ADDR IN B2 DID NOT REMAIN CLEARED
2332	010140	052532	DH24	
2333	010142	054350	DT14	
2334	010144	055050	DF22	
2335			:ERR 43	
2336	010146	051127	EM85	
2337	010150	052500	DH22	
2338	010152	053732	DT1	
2339	010154	054664	DF10	
2340			:ERR 44	
2341	010156	047160	EM15	:WCE AT CYL 411, TRK 2, SEC 21
2342	010160	051647	DH1	
2343	010162	053732	DT1	
2344	010164	054600	DF4	
2345			:ERR 45	
2346	010166	051127	EM85	:OFFSET BIT IN RKMR2 CLEARED
2347	010170	053456	DH51	:AFTER SEEK TO SELF
2348	010172	053732	DT1	
2349	010174	054664	DF10	
2350			:ERR 46	
2351	010176	047720	EM25	:DETECTED 10 BAD SECTORS
2352	010200	052632	DH27	:AFTER WRITE DATA CMD.
2353	010202	053732	DT1	
2354	010204	054664	DF10	
2355			:ERROR 47	
2356	010206	050233	EM39	:CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2357	010210	052350	DH17	:AFTER RECAL CMD
2358	010212	054350	DT14	
2359	010214	055050	DF22	
2360			:ERROR 50	
2361	010216	050302	EM40	:CYL ADDR IN RKMR3 NOT CLEARED
2362	010220	052350	DH17	:AFTER RECAL COMD
2363	010222	054350	DT14	
2364	010224	055050	DF22	
2365			:ERR 51	
2366	010226	051271	EM93	:WRONG CYL# IN HEADER WORD (MISPOSITION)
2367	010230	052557	DH25	:AFTER SEEK CMD
2368	010232	054224	DT9	
2369	010234	054770	DF20	
2370			:ERR 52	
2371	010236	047332	EM17	:MSG A0 ERROR
2372	010240	052632	DH27	:AFTER WRITE DATA CMD
2373	010242	054270	DT13	
2374	010244	055014	DF21	
2375			:ERR 53	
2376	010246	047374	EM19	:MSG A1 ERROR

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 47
ERROR POINTER TABLE

SEQ 0047

2377	010250	052632	DH27	
2378	010252	054270	DT13	
2379	010254	055014	DF21	
2380			:ERR 54	
2381	010256	047332	EM17	:MSG A0 ERROR
2382	010260	052602	DH26	:AFTER READ DATA CMD
2383	010262	054270	DT13	
2384	010264	055014	DF21	
2385			:ERROR 55	
2386	010266	047100	EM13	:NO ATTN
2387	010270	052350	DH17	:AFTER RECAL CMD
2388	010272	053732	DT1	
2389	010274	054664	DF10	
2390			:ERR 56	
2391	010276	047374	EM19	:MSG A1 ERROR
2392	010300	052602	DH26	
2393	010302	054270	DT13	
2394	010304	055014	DF21	
2395			:ERR 57	
2396	010306	047332	EM17	:MSG A0 ERROR
2397	010310	052762	DH32	:AFTER WRITE CHECK CMD
2398	010312	054270	DT13	
2399	010314	055014	DF21	
2400			:ERR 60	
2401	010316	047374	EM19	:MSG A1 ERROR
2402	010320	052762	DH32	
2403	010322	054270	DT13	
2404	010324	055014	DF21	
2405			:ERR 61	
2406	010326	047353	EM18	:MSG B0 ERROR
2407	010330	053545	DH53	:DURING OFFSET CMD
2408	010332	054270	DT13	
2409	010334	055014	DF21	
2410			:ERR 62	
2411	010336	047415	EM20	:MSG B1 ERROR
2412	010340	053545	DH53	
2413	010342	054270	DT13	
2414	010344	055014	DF21	
2415			:ERR 63	
2416	010346	047772	EM26	:BSE ERROR IN WRITE CMD NOT ON BSE TABLE
2417	010350	052632	DH27	:AFTER WRITE DATA CMD
2418	010352	053732	DT1	
2419	010354	054664	DF10	
2420			:ERR 64	
2421	010356	051230	EM88	:DID NOT FIND SECTOR 0 FROM INDEX
2422	010360	053573	DH54	:AFTER FORMAT CHANGE AND READY REC'D
2423	010362	053732	DT1	
2424	010364	054664	DF10	
2425			:ERR 65	
2426	010366	050051	EM27	:DETECTED BSE IN READ BUT NOT IN WRITE CMD.
2427	010370	051647	DH1	
2428	010372	053732	DT1	
2429	010374	054534	DF1	
2430			:ERR 66	
2431	010376	000000	0	
2432	010400	000000	0	

J04

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 48
ERROR POINTER TABLE

SEQ 0048

2433	010402	000000	0	
2434	010404	000000	0	
2435			;ERR 67	0
2436	010406	000000	0	
2437	010410	000000	0	
2438	010412	000000	0	
2439	010414	000000	0	
2440			;ERROR 70	0
2441	010416	000000	0	
2442	010420	000000	0	
2443	010422	000000	0	
2444	010424	000000	0	
2445			;ERR 71	0
2446	010426	000000	0	
2447	010430	000000	0	
2448	010432	000000	0	
2449	010434	000000	0	
2450			;ERROR 72	0
2451	010436	000000	0	
2452	010440	000000	0	
2453	010442	000000	0	
2454	010444	000000	0	
2455			;ERR 73	0
2456	010446	000000	0	
2457	010450	000000	0	
2458	010452	000000	0	
2459	010454	000000	0	
2460			;ERR 74	EM13 DH10 DT1 DF10
2461	010456	047100		;NO ATTN
2462	010460	052134		;AT END OF HEAD LOADING
2463	010462	053732		
2464	010464	054664		
2465			;ERR 75	EM22 DH1 DT1 DF1
2466	010466	047460		;NO DRIVS IN \$DEVN
2467	010470	051647		
2468	010472	053732		
2469	010474	054534		
2470			;ERR 76	EM23 DH1 DT1 DF1
2471	010476	047565		;NO DRIVS ON BUSS
2472	010500	051647		
2473	010502	053732		
2474	010504	054534		
2475			;ERR 77	0
2476	010506	000000	0	
2477	010510	000000	0	
2478	010512	000000	0	
2479	010514	000000	0	
2480			;ERR 100	0
2481	010516	000000	0	
2482	010520	000000	0	
2483	010522	000000	0	
2484	010524	000000	0	
2485			;ERROR 101	EM94 DH47 DT14
2486	010526	051352		;OFFSET NOT CLEARED
2487	010530	053331		;AFTER READ HEADER WITH MOVEMENT
2488	010532	054350		

K04

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 49
ERROR POINTER TABLE

SEQ 0049

2489	010534	055050	DF22	
2490			;ERROR 102	
2491	010536	051406	EM95	;FORMAT NOT SET
2492	010540	052632	DH27	;AFTER WRITE DATA CMD
2493	010542	053732	DT1	
2494	010544	05	DF10	
2495			;ERR 103	
2496	010546	051406	EM95	
2497	010550	052762	DH32	;AFTER WRITE CHECK CMD
2498	010552	053732	DT1	
2499	010554	054664	DF10	
2500			;ERR 104	
2501	010556	050233	EM39	;OFFSET NOT RESET
2502	010560	053671	DH57	;AFTER WRITE CMD WITH OFFSET
2503	010562	054350	DT14	
2504	010564	055050	DF22	
2505			;ERR 105	
2506	010566	050302	EM40	;CYL ADDR NOT 0
2507	010570	053671	DH57	
2508	010572	054350	DT14	
2509	010574	055050	DF22	
2510			;ERR 106	
2511	010576	051442	EM96	;CANNOT FIND SECTOR 23(8)
2512	010600	051647	DH1	
2513	010602	053732	DT1	
2514	010604	054534	DF1	
2515			;ERR 107	
2516	010606	051473	EM97	;HEAD SWITCHING TOO LONG
2517	010610	052632	DH27	;AFTER WRITE DATA CMD
2518	010612	053732	DT1	
2519	010614	054724	DF15	
2520			;ERR 110	
2521	010616	000000	0	
2522	010620	000000	0	
2523	010622	000000	0	
2524	010624	000000	0	
2525			;ERR 111	
2526	010626	000000	0	
2527	010630	000000	0	
2528	010632	000000	0	
2529	010634	000000	0	
2530			;ERR 112	
2531	010636	051560	EM100	;DRIVE OFF TRACK SET
2532	010640	052632	DH27	;AFTER WRITE DATA CMD
2533	010642	053776	DT4	
2534	010644	054640	DF6	
2535			;ERR 113	
2536	010646	050170	EM36	;CYL ADDR IN RKMR3 INCORRECT
2537	010650	052632	DH27	
2538	010652	053776	DT4	
2539	010654	054640	DF6	
2540			;ERROR 114	
2541	010656	051172	EM86	;OFFSET IN A2 NOT = RKASOF
2542	010660	052532	DH24	;AFTER OFFSET CMD
2543	010662	054350	DT14	
2544	010664	055050	DF22	

L04

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 50
ERROR POINTER TABLE

SEQ 0050

2545			;ERR 115	
2546	010666	051172	EM86	
2547	010670	052500	DH22	;AFTER DRIVE CLEAR CMD
2548	010672	054350	DT14	
2549	010674	055050	DF22	
2550			;ERROR 116	
2551	010676	047042	EM12	;CONT NOT RDY
2552	010700	052374	DH19	;AFTER PACK CMD
2553	010702	053732	DT1	
2554	010704	054664	DF10	
2555			;ERROR 117	
2556	010706	047042	EM12	;CONT NOT RDY
2557	010710	052417	DH20	;AFTER SEL DR CMD
2558	010712	053732	DT1	
2559	010714	054664	DF10	
2560			;ERROR 120	
2561	010716	047042	EM12	
2562	010720	052452	DH21	;AFTER SUBSYS CLEAR
2563	010722	053732	DT1	
2564	010724	054664	DF10	
2565			;ERROR 121	
2566	010726	047042	EM12	
2567	010730	052061	DH9	;AFTER START SPINDLE CMD
2568	010732	053732	DT1	
2569	010734	054664	DF10	
2570			;ERROR 122	
2571	010736	051615	EM101	;DID NOT GO TO CYL 10
2572	010740	052703	DH30	;AFTER READ HEADER CMD
2573	010742	054350	DT14	
2574	010744	055050	DF22	
2575			;ERROR 123	
2576	010746	051172	EM86	;A2 OFFSET NOT = RKASOF
2577	010750	053456	DH51	;AFTER SEEK TO SELF
2578	010752	054350	DT14	
2579	010754	055050	DF22	
2580			;ERROR 124	
2581	010756	047042	EM12	
2582	010760	052350	DH.7	;AFTER RECAL CMD
2583	010762	053732	DT1	
2584	010764	054664	DF10	
2585			;ERR 125	
2586	010766	050535	EM73	;CTO SET
2587	010770	051036	EM84	;WHILE WAITING FOR OR REC'D CONTR RDY. MSG A&B BAD
2588	010772	053732	DT1	
2589	010774	054614	DF5	
2590			;ERR 126	
2591	010776	050603	EM79	;NED SET
2592	011000	051036	EM84	
2593	011002	053732	DT1	
2594	011004	054614	DF5	
2595			;ERR 127	
2596	011006	046432	EM5	;MDS SET
2597	011010	051036	EM84	
2598	011012	053732	DT1	
2599	011014	054614	DF5	
2600			;ERROR 130	

M04

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 51
ERROR POINTER TABLE

SEQ 0051

2601	011016	000000	0	
2602	011020	000000	0	
2603	011022	000000	0	
2604	011024	000000	0	
2605			; ERROR 131	
2606	011026	047042	EM12	; NO RDY
2607	011030	052557	DH25	; AFTER SEEK CMD
2608	011032	053732	DT1	
2609	011034	054664	DF10	
2610			; ERROR 132	
2611	011036	047100	EM13	; NO ATTN
2612	011040	052557	DH25	
2613	011042	053732	DT1	
2614	011044	054664	DF10	
2615			; ERROR 133	
2616	011046	000000	0	
2617	011050	000000	0	
2618	011052	000000	0	
2619	011054	000000	0	
2620			; ERROR 134	
2621	011056	000000	0	
2622	011060	000000	0	
2623	011062	000000	0	
2624	011064	000000	0	
2625			; ERROR 135	
2626	011066	000000	0	
2627	011070	000000	0	
2628	011072	000000	0	
2629	011074	000000	0	
2630			; ERROR 136	
2631	011076	000000	0	
2632	011100	000000	0	
2633	011102	000000	0	
2634	011104	000000	0	
2635			; ERROR 137	
2636	011106	050233	EM39	; CYL DIFF/OFFSET IN RKMR2 NOT CLEARED
2637	011110	052557	DH25	
2638	011112	053732	DT1	
2639	011114	054664	DF10	
2640			; ERR 140	
2641	011116	047332	EM17	; MSG AD ERROR
2642	011120	053456	DH51	; AFTER SEEK TO SELF
2643	011122	054270	DT13	
2644	011124	055014	DF21	
2645			; ERR 141	
2646	011126	047353	EM18	
2647	011130	053456	DH51	
2648	011132	054270	DT13	
2649	011134	055014	DF21	
2650			; ERR 142	
2651	011136	047374	EM19	
2652	011140	053456	DH51	
2653	011142	054270	DT13	
2654	011144	055014	DF21	
2655			; ERR 143	
2656	011146	047415	EM20	

N04

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 52
ERROR POINTER TABLE

SEQ 0052

2657	011150	053456	DH51	
2658	011152	054270	DT13	
2659	011154	055014	DF21	
2660			; ERROR 144	
2661	011156	000000	0	
2662	011160	000000	0	
2663	011162	000000	0	
2664	011164	000000	0	
2665			; ERROR 145	
2666	011166	000000	0	
2667	011170	000000	0	
2668	011172	000000	0	
2669	011174	000000	0	
2670			; ERROR 146	
2671	011176	000000	0	
2672	011200	000000	0	
2673	011202	000000	0	
2674	011204	000000	0	
2675			; ERROR 147	
2676	011206	000000	0	
2677	011210	000000	0	
2678	011212	000000	0	
2679	011214	000000	0	
2680			; ERROR 150	
2681	011216	000000	0	
2682	011220	000000	0	
2683	011222	000000	0	
2684	011224	000000	0	
2685			; ERROR 151	
2686	011226	047042	EM12	
2687	011230	052500	DH22	
2688	011232	053732	DT1	
2689	011234	054664	DF10	
2690			; ERROR 152	
2691	011236	000000	0	
2692	011240	000000	0	
2693	011242	000000	0	
2694	011244	000000	0	
2695			; ERROR 153	
2696	011246	000000	0	
2697	011250	000000	0	
2698	011252	000000	0	
2699	011254	000000	0	
2700			; ERROR 154	
2701	011256	050406	EM55	
2702	011260	052500	DH22	
2703	011262	053732	DT1	
2704	011264	054664	DF10	
2705			; ERROR 155	
2706	011266	000000	0	
2707	011270	000000	0	
2708	011272	000000	0	
2709	011274	000000	0	
2710			; ERROR 156	
2711	011276	000000	0	
2712	011300	000000	0	

; NO RDY
; AFTER CLEAR CMD

; ATTN NOT CLEARED

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 53
ERROR POINTER TABLE

SEQ 0053

2713	011302	000000	0
2714	011304	000000	0
2715			; ERROR 157
2716	011306	000000	0
2717	011310	000000	0
2718	011312	000000	0
2719	011314	000000	0
2720			; ERROR 160
2721	011316	000000	0
2722	011320	000000	0
2723	011322	000000	0
2724	011324	000000	0
2725			; ERROR 161
2726	011326	000000	0
2727	011330	000000	0
2728	011332	000000	0
2729	011334	000000	0
2730			; ERROR 162
2731	011336	000000	0
2732	011340	000000	0
2733	011342	000000	0
2734	011344	000000	0
2735			; ERROR 163
2736	011346	000000	0
2737	011350	000000	0
2738	011352	000000	0
2739			
2740			
2741			
2742			
2743			
2744			
2745			
2746			
2747			
2748			
2749			
2750			
2751			
2752			
2753			
2754			
2755			
2756			
2757			
2758			
2759			
2760	011354	000000	0
2761			; ERROR 164
2762	011356	000000	0
2763	011360	000000	0
2764	011362	000000	0
2765	011364	000000	0
2766			; ERROR 165
2767	011366	000000	0
2768	011370	000000	0

2769	011372	000000	0	
2770	011374	000000	0	
2771			; ERROR 166	
2772	011376	000000	0	
2773	011400	000000	0	
2774	011402	000000	0	
2775	011404	000000	0	
2776			; ERROR 167	
2777	011406	000000	0	
2778	011410	000000	0	
2779	011412	000000	0	
2780	011414	000000	0	
2781			; ERROR 170	
2782	011416	000000	0	
2783	011420	000000	0	
2784	011422	000000	0	
2785	011424	000000	0	
2786			; ERROR 171	
2787	011426	047042	EM12	; NO RDY
2788	011430	052703	DH30	; AFTER READ HEADER CMD
2789	011432	053732	DT1	
2790	011434	054664	DF10	
2791			; ERROR 172	
2792	011436	000000	0	
2793	011440	000000	0	
2794	011442	000000	0	
2795	011444	000000	0	
2796			; ERROR 173	
2797	011446	050441	EM63	; DLT SET
2798	011450	052703	DH30	
2799	011452	053732	DT1	
2800	011454	054724	DF15	
2801			; ERROR 174	
2802	011456	047436	EM21	; CERR SET
2803	011460	052703	DH30	
2804	011462	053732	DT1	
2805	011464	054724	DF15	
2806			; ERROR 175	
2807	011466	050233	EM39	; CYL DIFF NOT CLEARED
2808	011470	052134	DH10	; AT END OF HEAD LOADING
2809	011472	053732	DT1	
2810	011474	054664	DF10	
2811			; ERROR 176	
2812	011476	050302	EM40	; CYL ADDR NOT CLEARED.
2813	011500	052134	DH10	
2814	011502	053732	DT1	
2815	011504	054664	DF10	
2816			; ERROR 177	
2817	011506	000000	0	
2818	011510	000000	0	
2819	011512	000000	0	
2820	011514	000000	0	
2821			; ERROR 200	
2822	011516	047042	EM12	; NO RDY
2823	011520	053014	DH39	; AFTER WRITE HEADER CMD
2824	011522	053732	DT1	

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 55
ERROR POINTER TABLE

SEQ 0055

2825	011524	054724	DF15	
2826			; ERROR 201	
2827	011526	047436	EM21	; CERR SET
2828	011530	053014	DM39	
2829	011532	053732	DT1	
2830	011534	054724	DF15	
2831			; ERROR 202	
2832	011536	050462	EM65	; READ HEADER ERROR
2833	011540	051647	DM1	
2834	011542	054110	DT7	
2835	011544	054704	DF14	
2836			; ERROR 203	
2837	011546	000000	0	
2838	011550	000000	0	
2839	011552	000000	0	
2840	011554	000000	0	
2841			; ERROR 204	
2842	011556	000000	0	
2843	011560	000000	0	
2844	011562	000000	0	
2845	011564	000000	0	
2846			; ERROR 205	
2847	011566	000000	0	
2848	011570	000000	0	
2849	011572	000000	0	
2850	011574	000000	0	
2851			; ERROR 206	
2852	011576	000000	0	
2853	011600	000000	0	
2854	011602	000000	0	
2855	011604	000000	0	
2856			; ERROR 207	
2857	011606	050170	EM36	; CYL ADDR IN RKMR3 INCORRECT
2858	011610	052557	DM25	; AFTER SEEK CMD
2859	011612	053776	DT4	
2860	011614	054640	DF6	
2861			; ERROR 210	
2862	011616	047436	EM21	; CERR SET
2863	011620	052557	DM25	
2864	011622	053732	DT1	
2865	011624	054664	DF10	
2866			; ERROR 211	
2867	011626	000000	0	
2868	011630	000000	0	
2869	011632	000000	0	
2870	011634	000000	0	
2871			; ERROR 212	
2872	011636	000000	0	
2873	011640	000000	0	
2874	011642	000000	0	
2875	011644	000000	0	
2876			; ERROR 213	
2877	011646	000000	0	
2878	011650	000000	0	
2879	011652	000000	0	
2880	011654	000000	0	

E05

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 56
ERROR POINTER TABLE

SEQ 0056

2881			:ERROR 214	
2882	011656	000000	0	
2883	011660	000000	0	
2884	011662	000000	0	
2885	011664	000000	0	
2886			:ERROR 215	
2887	011666	000000	0	
2888	011670	000000	0	
2889	011672	000000	0	
2890	011674	000000	0	
2891			:ERROR 216	
2892	011676	000000	0	
2893	011700	000000	0	
2894	011702	000000	0	
2895	011704	000000	0	
2896			:ERROR 217	
2897	011706	000000	0	
2898	011710	000000	0	
2899	011712	000000	0	
2900	011714	000000	0	
2901			:ERROR 220	
2902	011716	000000	0	
2903	011720	000000	0	
2904	011722	000000	0	
2905	011724	000000	0	
2906			:ERROR 221	
2907	011726	047332	EM17	:MSG A0 ERROR
2908	011730	052350	DH17	
2909	011732	054270	DT13	
2910	011734	055014	DF21	
2911			:ERROR 222	
2912	011736	047374	EM19	:MSG A1 ERROR
2913	011740	052350	DH17	
2914	011742	054270	DT13	
2915	011744	055014	DF21	
2916			:ERROR 223	
2917	011746	000000	0	
2918	011750	000000	0	
2919	011752	000000	0	
2920	011754	000000	0	
2921			:ERROR 224	
2922	011756	000000	0	
2923	011760	000000	0	
2924	011762	000000	0	
2925	011764	000000	0	
2926			:ERROR 225	
2927	011766	000000	0	
2928	011770	000000	0	
2929	011772	000000	0	
2930	011774	000000	0	
2931			:ERROR 226	
2932	011776	047042	EM12	:NO RDY
2933	012000	052602	DH26	:AFTER READ DATA CMD
2934	012002	053732	DT1	
2935	012004	054664	DF10	
2936			:ERROR 227	

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 57
ERROR POINTER TABLE

SEQ 0057

2937	012006	047436	EM21	:CERP SET
2938	012010	052602	DH26	
2939	012012	053732	DT1	
2940	012014	054724	DF15	
2941			:ERROR 230	
2942	012016	047267	EM16	:CANNOT READ BSE INFO
2943	012020	052200	DH13	:ON SEC 10, 12, 14, 16, 18, 20
2944	012022	053732	DT1	
2945	012024	054744	DF17	
2946			:ERROR 231	
2947	012026	047267	EM16	
2948	012030	052264	DH14	:ON SEC 11, 13, 15, 17, 19, 21
2949	012032	053732	DT1	
2950	012034	054744	DF17	
2951			:ERROR 232	
2952	012036	000000	0	
2953	012040	000000	0	
2954	012042	000000	0	
2955	012044	000000	0	
2956			:ERROR 233	
2957	012046	047267	EM16	:CANNOT READ BSE INFO
2958	012050	053130	DH42	:ON SECT 0,2,4,6,8
2959	012052	053732	DT1	
2960	012054	054744	DF17	
2961			:ERROR 234	
2962	012056	047267	EM16	
2963	012060	053201	DH43	:ON SECT 1,3,5,7,9
2964	012062	053732	DT1	
2965	012064	054744	DF17	
2966			:ERROR 235	
2967	012066	050504	EM69	:ALIGN CARTRIDGE USED
2968	012070	053252	DH44	:WILL BYPASS FORMAT & ALL R/W TESTS
2969	012072	053732	DT1	
2970	012074	054664	DF10	
2971			:ERROR 236	
2972	012076	000000	0	
2973	012100	000000	0	
2974	012102	000000	0	
2975	012104	000000	0	
2976			:ERROR 237	
2977	012106	000000	0	
2978	012110	000000	0	
2979	012112	000000	0	
2980	012114	000000	0	
2981			:ERROR 240	
2982	012116	000000	0	
2983	012120	000000	0	
2984	012122	000000	0	
2985	012124	000000	0	
2986			:ERROR 241	
2987	012126	000000	0	
2988	012130	000000	0	
2989	012132	000000	0	
2990	012134	000000	0	
2991			:ERROR 242	
2992	012136	000000	0	

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 58
ERROR POINTER TABLE

SEQ 0058

2993	012140	000000	0	
2994	012142	000000	0	
2995	012144	000000	0	
2996				
2997			: ERROR 243	
2998	012146	050170	EM36	: CYL ADDR IN RKMR3 INCORRECT
2999	012150	052557	DH25	: AFTER SEEK CMD
3000	012152	054156	DT8	
3001	012154	054640	DF6	
3002			: ERR 244	
3003	012156	050556	EM74	: RTZ NOT SET
3004	012160	053103	DH41	: DURING RECAL CMD
3005	012162	053732	DT1	
3006	012164	054664	DF10	
3007			: ERR 245	
3008	012166	000000	0	
3009	012170	000000	0	
3010	012172	000000	0	
3011	012174	000000	0	
3012			: ERR 246	
3013	012176	000000	0	
3014	012200	000000	0	
3015	012202	000000	0	
3016	012204	000000	0	
3017			: ERR 247	
3018	012206	000000	0	
3019	012210	000000	0	
3020	012212	000000	0	
3021	012214	000000	0	
3022			: ERR 250	
3023	012216	000000	0	
3024	012220	000000	0	
3025	012222	000000	0	
3026	012224	000000	0	
3027			: ERR 251	
3028	012226	000000	0	
3029	012230	000000	0	
3030	012232	000000	0	
3031	012234	000000	0	
3032			: ERR 252	
3033	012236	000000	0	
3034	012240	000000	0	
3035	012242	000000	0	
3036	012244	000000	0	
3037			: ERR 253	
3038	012246	000000	0	
3039	012250	000000	0	
3040	012252	000000	0	
3041	012254	000000	0	
3042			: ERR 254	
3043	012256	000000	0	
3044	012260	000000	0	
3045	012262	000000	0	
3046	012264	000000	0	
3047			: ERR 255	
3048	012266	000000	0	

H05

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MAC/11 30A(1052) 10-JAN-78 11:25 PAGE 59
ERROR POINTER TABLE

SEQ 0059

3049	012270	000000	0	
3050	012272	000000	0	
3051	012274	000000	0	
3052			:ERR 256	
3053	012276	000000	0	
3054	012300	000000	0	
3055	012302	000000	0	
3056	012304	000000	0	
3057			:ERR 257	
3058	012306	000000	0	
3059	012310	000000	0	
3060	012312	000000	0	
3061	012314	000000	0	
3062			:ERR 260	
3063	012316	047332	EM17	:MSG A0 ERROR
3064	012320	052532	DH24	:AFTER OFFSET CMD
3065	012322	054270	DT13	
3066	012324	055014	DF21	
3067			:ERR 261	
3068	012326	047353	EM18	:MSG B0 ERROR
3069	012330	052532	DH24	
3070	012332	054270	DT13	
3071	012334	055014	DF21	
3072			:ERR 262	
3073	012336	000000	0	
3074	012340	000000	0	
3075	012342	000000	0	
3076	012344	000000	0	
3077			:ERR 263	
3078	012346	000000	0	
3079	012350	000000	0	
3080	012352	000000	0	
3081	012354	000000	0	
3082			:ERR 264	
3083	012356	000000	0	
3084	012360	000000	0	
3085	012362	000000	0	
3086	012364	000000	0	
3087			:ERR 265	
3088	012366	047353	EM18	:MSG B0 ERROR
3089	012370	052500	DH22	:AFTER DRIVE CLEAR CMD
3090	012372	054270	DT13	
3091	012374	055014	DF21	
3092			:ERR 266	
3093	012376	047415	EM20	:MSG B1 ERROR
3094	012400	052500	DH22	
3095	012402	054270	DT13	
3096	012404	055014	DF21	
3097			:ERR 267	
3098	012406	047353	EM18	:MSG B0 ERROR
3099	012410	053014	DH39	:AFTER WRITE HEADER CMD
3100	012412	054270	DT13	
3101	012414	055014	DF21	
3102			:ERR 270	
3103	012416	047415	EM20	:MSG B1 ERROR
3104	012420	053014	DH39	

CZR61EO UNIBLS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78
ERROR POINTER TABLE

11:25 PAGE 60

SEQ 0060

3105	012422	054270	DT13	
3106	012424	055014	DF21	
3107			;ERR 271	
3108	012426	047353	EM18	
3109	012430	052703	DH30	;AFTER RD. HDR. CMD.
3110	012432	054270	DT13	
3111	012434	055014	DF21	
3112			;ERR 272	
3113	012436	047415	EM20	
3114	012440	052703	DH30	
3115	012442	054270	DT13	
3116	012444	055014	DF21	
3117			;ERR 273	
3118	012446	047332	EM17	;MSG A0 ERROR
3119	012450	052500	DH22	;AFTER DRV CLR CMD
3120	012452	054270	DT13	
3121	012454	055014	DF21	
3122			;ERR 274	
3123	012456	047374	EM19	;MSG A1 ERROR
3124	012460	052500	DH22	
3125	012462	054270	DT13	
3126	012464	055014	DF21	
3127			;ERR 275	
3128	012466	047353	EM18	;MSG B0 ERROR
3129	012470	052350	DH17	;AFTER RECAL CMD
3130	012472	054270	DT13	
3131	012474	055014	DF21	
3132			;ERR 276	
3133	012476	047415	EM20	;MSG B1 ERROR
3134	012500	052350	DH17	
3135	012502	054270	DT13	
3136	012504	055014	DF21	
3137			;ERR 277	
3138	012506	047332	EM17	;MSG A0 ERROR
3139	012510	053014	DH39	;AFTER WRITE HEADER CMD
3140	012512	054270	DT13	
3141	012514	055014	DF21	
3142			;ERR 300	
3143	012516	047374	EM19	;MSG A1 ERROR
3144	012520	053014	DH39	
3145	012522	054270	DT13	
3146	012524	055014	DF21	
3147			;ERR 301	
3148	012526	047332	EM17	
3149	012530	052703	DH30	;AFT RD HDR. CMD
3150	012532	054270	DT13	
3151	012534	055014	DF21	
3152			;ERR 302	
3153	012536	047374	EM19	
3154	012540	052703	DH30	
3155	012542	054270	DT13	
3156	012544	055014	DF21	
3157				

```

3158
3159 .SBTTL PROGRAM SETUP
3160
3161 012546 012737 000001 001336 PARSRT: MOV #1,PARAM ;SET FLAG FOR 220 START
3162 012554 000402 BR PRGSRT ;START PROGRAM
3163
3164 012556 005037 001336 START: CLR PARAM ;CLEAR FOR 200 START
3165 012562 000005 PRGSRT: RESET ;CLEAR ALL INT ENABLE & INIT
3166 012564 012706 001100 MOV #STACK,SP ;SETUP STACK POINTER
3167 012570 012746 000000 MOV #PRO,-(SP) ;PSW LOADED TO BE
3168 012574 012746 012602 MOV #1$,-(SP) ;LSI-11 COMPATABLE
3169 012600 000002 RTI ;ENABLE ALL INTERRUPTS
3170
3171 012602 004737 041112 1$: JSR PC,$TKINT ;SETUP KB VECTOR ADDR, PRIORITY 4
3172 ; & TURN ON KB INTERRUPT
3173
3174 ;*** CPU PRIORITY LEVEL NOW AT 0 ***
3175 ;*** ANY DEVICE WHICH SETS ITS ***
3176 ;*** INTERRUPT ENABLE BIT WILL ***
3177 ;*** SERVICED. ***
3178
3179 ;CLOCK INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 6 (IN 'STS')
3180 ;RK06 CONTROLLER INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 5 IN 'SETINT')
3181 ;KEYBOARD INTERRUPTS WILL CHANGE CPU PRIORITY TO LEVEL 4 (SEE ABOVE)
3182
3183 ;ALL 'SYSMAC' TRAPS WILL CHANGE CPU PRIORITY TO LEVEL 7 (SEE BELOW)
3184
3185
3186 ;SYSMAC 'SETUP'
3187 .SBTTL INITIALIZE THE COMMON TAGS
3188 ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3189 012606 012706 001100 MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
3190 012612 005026 CLR (R6)+ ;;CLEAR MEMORY LOCATION
3191 012614 022706 001140 CMP #SWR,R6 ;;DONE?
3192 012620 001374 BNE -6 ;;LOOP BACK IF NO
3193 012622 012706 001100 MOV #STACK,SP ;;SETUP THE STACK POINTER
3194 ;;INITIALIZE A FEW VECTORS
3195 012626 012737 037220 000020 MOV #SCOPE,$IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
3196 012634 012737 000340 000022 MOV #340,$IOTVEC+2 ;;LEVEL 7
3197 012642 012737 037500 000030 MOV #ERROR,$EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
3198 012650 012737 000340 000032 MOV #340,$EMTVEC+2 ;;LEVEL 7
3199 012656 012737 043330 000034 MOV #TRAP,$TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
3200 012664 012737 000340 000036 MOV #340,$TRAPVEC+2 ;;LEVEL 7
3201 012672 012737 036754 000024 MOV #PWRDN,$PWRVEC ;;POWER FAILURE VECTOR
3202 012700 012737 000340 000026 MOV #340,$PWRVEC+2 ;;LEVEL 7
3203 012706 013737 031546 031540 MOV $ENDCT,$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
3204 012714 005037 001174 CLS $TIMES ;;INITIALIZE NUMBER OF ITERATIONS
3205 012720 005037 001176 CLS $ESCAPE ;;CLEAR THE ESCAPE ON ERROR ADDRESS
3206 012724 112737 000001 001115 MOVB #1,$ERMAX ;;ALLOW ONE ERROR PER TEST
3207 012732 012737 012732 001106 MOV #,$SLPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
3208 012740 012737 012740 001110 MOV #,$SLPERR ;;SETUP THE ERROR LOOP ADDRESS
3209 ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
3210 ;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.
3211 012746 013746 000004 MOV $ERRVEC,-(SP) ;;SAVE ERROR VECTOR
3212 012752 012737 013006 000004 MOV #64,$ERRVEC ;;SET UP ERROR VECTOR
3213 012760 012737 177570 001140 MOV #DSWR,$SWR ;;SETUP FOR A HARDWARE SWICH REGISTER

```

K05

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 62
INITIALIZE THE COMMON TAGS

SEQ 0062

```

3214 012766 012737 177570 001142      MOV      #DDISP,DISPLAY ; AND A HARDWARE DISPLAY REGISTER
3215 012774 022777 177777 166136      CMP      #-1,SWR ; TRY TO REFERENCE HARDWARE SWR
3216 013002 001012          BNE      66$ ; BRANCH IF NO TIMEOUT TRAP OCCURRED
3217          ; AND THE HARDWARE SWR IS NOT = -1
3218 013004 000403          BR       65$ ; BRANCH IF NO TIMEOUT
3219 013006 012716 013014 64$:      MOV      #65$, (SP) ; SET UP FOR TRAP RETURN
3220 013012 000002          RTI
3221 013014 012737 000176 001140 65$:      MOV      #SWREG,SWR ; POINT TO SOFTWARE SWR
3222 013022 012737 000174 001142      MOV      #DISPREG,DISPLAY
3223 013030 012637 000004 66$:      MOV      (SP)+,2#ERRVEC ; RESTORE ERROR VECTOR
3224
3225 013034 005037 001216          CLR      $PASS ; CLEAR PASS COUNT
3226 013040 132737 000200 001231      BITB     #APTSIZE,$ENVM ; TEST USER SIZE UNDER APT
3227 013046 001403          BEQ      67$ ; YES, USE NON-APT SWITCH
3228 013050 012737 001232 001140      MOV      #SSWREG,SWR ; NO, USE APT SWITCH REGISTER
3229 013056
3230 013056 012737 013122 000004 67$:      MOV      #1$,ERRVEC ; SETUP TIMEOUT VECTOR
3231 013064 012737 000340 000006      MOV      #PR7,ERRVEC+2
3232
3233 013072 012701 172100          MOV      #MEMBAS,R1 ; ADDR OF MEM CSR
3234 013076 005011 3$:      CLR      (R1) ; SEE IF CAN REFERENCE
3235 013100 012711 000001          MOV      #1,(R1) ; SET ENABLE BIT IF YES
3236 013104 012737 036656 000114      MOV      #MEMERR,MEMVEC ; LOAD VECTOR IF NO TIMEOUT
3237 013112 012737 000340 000116      MOV      #PR7,MEMVEC+2
3238 013120 000401          BR       2$
3239
3240 013122 022626          1$:      CMP      (SP)+,(SP)+ ; ADJ STACK
3241 013124 062701 000002 2$:      ADD      #2,R1 ; TRY NEXT CSR
3242 013130 020127 172140          CMP      R1,#MEMBAS+40 ; SEE IF TRIED ALL
3243 013134 001360          BNE      3$ ; BR IF NO
3244 013136 012737 000006 000004      MOV      #ERRVEC+2,ERRVEC ; RESTORE TRAP CATCHER
3245 013144 005037 000006          CLR      ERRVEC+2
3246
3247 013150 004737 031666          JSR      PC,CLRFLG ; CLEAR DDUMP THRU SIZE LG
3248 013154 005037 001220          CLR      $DEVCT
3249 013160 005037 001222          CLR      $UNIT
3250
3251          ;
3252          ; FIND OUT IF XXDP, ACT, APT; CHAIN OR DUMP MODE
3253          ;
3254
3255 013164 005737 000042      START1: TST      42
3256 013170 001015          BNE      1$
3257 013172 004737 031706          JSR      PC,TITLE ; BR IF AUTO
3258 013176 123727 000041 000013      CMPB     41,#13 ; MANUAL, TYPE PROG ID
3259 013204 001011          BNE      2$ ; 13=LOADED BY XXDP
3260 013206 005237 007464          INC      DDUMP ; SET RK06 DUMP MODE FLAG
3261 013212 104401 044307          TYPE     ,MSG2 ; REPLACE DRD PACK W/SCRATCH & DO<CR>
3262 013216 000000          HALT ; HALT
3263 013220 000137 013234          JMP      ST2
3264 013224 000137 013300          1$:      JMP      ST3
3265 013230 005237 007472          2$:      INC      PPTP ; SET ACT/APT/PTP DUMP MODE FLAG
3266
3267          ;
3268          ; CHECK IF ALL PARAMETERS DEFAULTED. IF NOT, BEGIN INPUT DIALOGUE
3269          ; WITH OPERATOR. THE REPLY TO 'DRIVES TO BE TESTED' SHOULD BE

```

```

3270      ;DRIVE NOS. SEPERATED BY COMMAS & TERMINATED BY <CR>
3271      ;EX:      DRIVES TO BE TESTED: 1,2,4<CR>
3272      ;
3273      ;
3274      013234 005737 001336      ST2:   TST      PARAM
3275      013240 001002              BNE      1$
3276      013242 000137 013332      JMP      ST4      ;BR IF 220 START
3277      013246 104401 044370      1$:   TYPE     MSG3      ;200 START, DEFAULT & SIZE THE BUSS
3278      013252 004737 031766      JSR      PC, GDRVS  ;DRIVES TO BE TESTED
3279      013256 104401 044422      TYPE     MSG4      ;GET DR NOS.
3280      013262 004737 032126      JSR      PC, GBA      ;BUSS ADDR
3281      013266 104401 044467      TYPE     MSG5      ;GET BA
3282      013272 004737 032154      JSR      PC, GINT  ;CONT INT VECTOR
3283      013276 000427              BR       ST5      ;GET INT VECTOR
3284
3285      ;
3286      ;
3287      ;AUTO MODE
3288      ;CHECK IF LOADED BY XXDP OR OTHER. SET FLAGS & NO INPUT DIALOGUE.
3289      ;DEFAULT ALL PARAMETERS. TEST ONLY THOSE DRIVES THAT ARE READY
3290      ;ON THE BUSS
3291      ;
3292      013300 123727 000041 000013 ST3:   CMPB     41, #13      ;13=LOADED BY XXDP
3293      013306 001007              BNE      1$
3294      013310 005237 007466      INC      DDPCH      ;SET RK06 CHAIN MODE FLAG
3295      013314 004737 031706      JSR      PC, TITLE
3296      013320 104401 044604      TYPE     MSG7      ;DRO NOT TSTD
3297      013324 000402              BR       ST4
3298      013326 005237 007470      1$:   INC      ACT11      ;SET ACT AUTO FLAG.
3299
3300      013332 012737 177440 001264 ST4:   MOV      #177440, $BASE  ;DEFAULT VALUE
3301      013340 012737 000210 001314      MOV      #210, RKVEC  ;DEFAULT VALUE
3302      013346 004737 032206      JSR      PC, SETINT
3303      013352 005237 007524      INC      SIZFLG      ;DO "SIZE THE BUSS" TEST
3304
3305      013356 005037 007336      ST5:   CLR      UNLD      ;INITIALIZE FLAGS
3306      013362 005037 007340      CLR      BADHDR    ;USED IN 'STOP' ROUTINE
3307      013366 005037 007342      CLR      HPEND      ;FOR VALID PROGRAM HALTS
3308      013372 005037 001176      CLR      $ESCAPE
3309      013376 005037 001170      CLR      $TMP4
3310      013402 012737 007476 001342      MOV      #DRIVO, DRVPTR ;CLEAR RK07 FLAG
3311      013410 005037 001220      CLR      $DEVCT    ;SETUP
3312      013414 005037 001222      CLR      $UNIT      ;NO. OF DRVS DONE
3313      013420 012737 013466 000004      MOV      #1$, ERRVEC  ;CURRENT DRV UNDER TEST
3314      013426 005777 165674      TST      @LK$      ;SETUP TIMEOUT ERROR VECTOR
3315      013432 005237 007516      INC      LCLKF      ;SEE IF L-CLOCK THERE
3316      013436 013700 001330      MOV      LCVEC, RO  ;PRESENT, SET FLAG.
3317      013442 012737 013530 000004      MOV      #2$, ERRVEC ;VECTOR ADDR
3318      013450 005777 165644      TST      @PK$      ;SEE IF P-CLOCK THERE
3319      013454 005237 007520      INC      PCLKF      ;PRESENT, SET FLAG
3320      013460 013700 001332      MOV      PCVEC, RO  ;VECTOR ADDR
3321      013464 000412      BR
3322
3323      013466 022626 013534 000004 1$:   CMP      (SP)+, (SP)+ ;L-CLOCK NOT THERE, CLEAR STACK
3324      013470 012737 165616      MOV      #4$, ERRVEC
3325      013476 005777      TST      @PK$      ;SEE IF P-CLOCK THERE

```

M05

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 64
INITIALIZE THE COMMON TAGS

SEQ 0064

3326	013502	005237	007520		INC	PCLKF	;PRESENT SET FLAG
3327	013506	013700	001332		MOV	PCVEC,RO	;VECTOR ADDR
3328	013512	005237	007522	3\$:	INC	DOTIM	;INDICATES TIMING TESTS CAN BE DONE
3329	013516	012720	036100		MOV	#CLOCK,(RO)+	;SERVICE ROUTINE FOR CLOCKS
3330	013522	012710	000300		MOV	#PR6,(RO)	
3331	013526	000407			BR	TST1	;GO TO NEXT TEST
3332							
3333	013530	022626		2\$:	CMP	(SP)+,(SP)+	;P-CLOCK NOT THERE, CLEAR STACK
3334	013532	000767			BR	3\$	
3335							
3336	013534	022626		4\$:	CMP	(SP)+,(SP)+	;NEITHER CLOCK THERE, CLEAR STACK
3337	013536	005037	007522		CLR	DOTIM	;TIMING TESTS CANNOT BE DONE.
3338	013542	104401	045145		TYPE	.MSG13	;HEAD SW. TEST BYP SSED
3339							
3340							

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 65
BASIC CONTROLLER TESTS, SIZING & SETUP

SEQ 0065

.SBTTL BASIC CONTROLLER TESTS, SIZING & SETUP

TEST 1 REFERENCE ALL CONTROLLER REGISTERS

THIS TEST VERIFIES THAT ALL THE CONTROLLER REGISTERS
CAN BE ACCESSED. THE INABILITY TO BE ACCESSED WILL
RESULT IN A TIMEOUT TRAP WITH AN ERROR MESSAGE. ANY
ERROR IN THIS TEST WILL RESULT IN ABORTING ALL OTHER
TESTS AND JUMPING TO 'END OF PASS'

ST1: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
MOV #PRO,-(SP) ;RESET PSW TO PRIORITY 0
MOV #SS,-(SP) ;& MAKE IT LSI COMPATABLE
RTI

5\$:

3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353 013546 000004
3354 013550 012737 000001 001174
3355 013556 012706 001100
3356
3357 013562 012746 000000
3358 013566 012746 013574
3359 013572 000002
3360 013574
3361

B06

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 66
T1 REFERENCE ALL CONTROLLER REGISTERS

SEQ 0066

3362 013574 012737 013720 000004 MOV #15,ERRVEC ;SETUP TIMOUT ERROR VECTOR

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 67
T1 REFERENCE ALL CONTROLLER REGISTERS

SEQ 0067

```

3363 013602 013705 001264      MOV    $BASE,R5      ;SETUP INDEX REG.
3364 013606 005765 000000      TST    RKCS1(R5)    ;REFERENCE ALL THE
3365 013612 005765 000010      TST    RKCS2(R5)    ;CONTROLLER REGISTERS
3366 013616 005765 000002      TST    RKWC(R5)
3367 013622 005765 000004      TST    RKBA(R5)
3368 013626 005765 000006      TST    RKDA(R5)
3369 013632 005765 000012      TST    RKDS(R5)    ;TIMEOUTS IN THIS SECTION
3370 013636 005765 000014      TST    RKER(R5)    ;INDICATE THAT THE CONTROLLER
3371 013642 005765 000016      TST    RKASOF(R5)   ;REGISTERS CANNOT BE READ.
3372 013646 005765 000020      TST    RKDC(R5)    ;TESTING SHOULD NOT PROCEED
3373 013652 005765 000024      TST    RKDB(R5)    ;UNTIL THIS IS REMEDIED.
3374 013656 005765 000026      TST    RKMR1(R5)
3375 013662 005765 000034      TST    RKMR2(R5)
3376 013666 005765 000036      TST    RKMR3(R5)
3377 013672 005765 000030      TST    RKECPS(R5)
3378 013676 005765 000032      TST    RKECPT(R5)
3379
3380 013702 012737 036570 000004      MOV    #BADTMO,ERRVEC ;SETUP TIMEOUT HANDLER
3381 013710 012737 000340 000006      MOV    #PR7,ERRVEC+2
3382 013716 000404          BR      TST2      ;;GO TO NEXT TEST
3383
3384 013720 022626          15:  CMP     (SP)+,(SP)+ ;RESTORE STACK POINTER
3385 013722 104007          ERROR 7          ;ABORT-COULD NOT REFERENCE CONTROLLER REGISTER
3386 013724 000137 031512      JMP     $EOP1
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404 013730 000004          ;*****
3405 013732 012737 000001 001174      ;TEST 2      SIZE THE BUSS
3406 013740 012706 001100          ;*****
3407
3408 013744 005237 001516          *
3409
3410
3411 013750 132737 000200 001231      *
3412 013756 001002          *   THIS TEST IS ENTERED ONLY IF 'DRIVE SELECTION' IS DEFAULTED
3413 013760 000137 014100          *   EITHER BY RUNNING IN THE AUTO MODE OR A 200 START IN THE
3414
3415 013764 104401 044713          *   MANUAL MODE.
3416 013770 005037 007474          *   EVERY DRIVE FROM 0 THRU 7 IS ADDRESSED.
3417 013774 005000          *   CONTROLLER ERROR (CERR) IS EXAMINED AND IF NOT SET, THE
3418 013776 012701 007476          *   DRIVE WILL BE TESTED AS AN RK06. IF SET, THE PROGRAM WILL BYPASS
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
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
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626
3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638
3639
3640
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
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
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058
4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077
4078
4079
4080
4081
4082
4083
4084
4085
4086
4087
4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106
4107
4108
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119
4120
4121
4122
4123
4124
4125
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136
4137
4138
4139
4140
4141
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
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
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000

```

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 68
T2 SIZE THE BUSS

SEQ 0068

3419	014002	013702	001266		MOV	\$DEV0,R2	;APT DEVICE MAP
3420							
3421	014006	032702	000001	15:	BIT	#BIT0,R2	;SEE IF DRV IN DEVICE MAP
3422	014012	001410			BEQ	16:	;BR IF NO
3423	014014	005237	007474		INC	DRVS	;ELSE INCR DRIVE COUNT
3424	014020	005211			INC	(P1)	; & SET DRIVE PRESENT FLAG
3425	014022	104401	001205		TYPE	\$CALF	
3426	014026	010046			MOV	RO,-(SP)	;SAVE RO FOR TYPEOUT
3427							;TYPE DRIVE #
3428	014030	104403			TYPOS		;GO TYPE--OCTAL ASCII
3429	014032	001			.BYTE	1	;TYPE 1 DIGIT(S)
3430	014033	000			.BYTE	0	;SUPPRESS LEADING ZEROS
3431							
3432	014034	005721		16:	TST	(R1)+	;ADV POINTER TO NEXT FLAG
3433	014036	005200			INC	RO	;INC DRIVE #
3434	014040	022700	000010		CMP	#B.,RO	;ALL B TESTED?
3435	014044	001402			BEQ	17:	;BR IF YES
3436							
3437	014046	006002			ROR	R2	;ELSE GET NEXT BIT OFF DEVICE MAP
3438	014050	000756			BR	15:	; & TRY AGAIN
3439							
3440	014052	005737	007474	17:	TST	DRVS	;SEE IF MORE DRIVES PRESENT
3441	014056	001402			BEQ	18:	;BR IF NO
3442	014060	000137	014554		JMP	VERIFY	;ELSE EXIT TEST & SETUP FOR RK07'S
3443							
3444	014064	104075		18:	ERROR	75	;NO DRIVES FOUND IN \$DEV0
3445	014066	000000			HALT		;SETUP CORRECTLY & PRESS 'CONTINUE'
3446	014070	000137	013356		JMP	ST5	;TO TRY AGAIN
3447	014074	000137	014554	20:	JMP	VERIFY	;DO NOT SIZE, GO TO NEXT TEST
3448							
3449	014100	012765	000040	000010	12:	MOV	#SCLR,RKCS2(R5)
3450	014106	013737	001406	007412	MOV	T10,TEMP1	;SUBSYSTEM CLEAR
3451	014114	004737	032320		JSR	PC,FRDY	;SET TIMEOUT
3452	014120	104120			ERROR	120	;FIND RDY
3453	014122	005737	007524		TST	SIZFLG	;RDY NOT SET BY END OF SCLR
3454	014126	001762			BEQ	20:	;SIZE BUS?
3455	014130	104401	044713		TYPE	MSG10	;BR IF NO
3456	014134	005037	007474		CLR	DRVS	;WILL TEST DRIVES
3457	014140	005000			CLR	RO	;# OF DRIVES PRESENT
3458	014142	012701	007476		MOV	#DRIV0,R1	;DRV ADDR
3459	014146						;DRV FLAG
3460	014146	104415		15:	SCOP1		
3461	014150	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
3462							
3463	014154	012765	000040	000010	MOV	#SCLR,RKCS2(R5)	;SUBSYSTEM CLEAR
3464	014162	013737	001406	007412	MOV	T10,TEMP1	;SET TIMEOUT
3465	014170	004737	032320		JSR	PC,FRDY	;FIND RDY
3466	014174	104120			ERROR	120	;RDY NOT SET BY END OF SCLR
3467	014176	010065	000010		MOV	RO,RKCS2(R5)	;SELECT THE DRIVE ADDR
3468	014202	012737	000001	007354	MOV	#SELDRV,HCS1	
3469	014210	053737	001170	007354	BIS	STMP4,HCS1	;ADD CDT IF RK07
3470	014216	013765	007354	000000	MOV	HCS1,RKCS1(R5)	;GET STATUS
3471	014224	013737	001420	007412	MOV	T50000,TEMP1	
3472	014232	004737	033010		JSR	PC,DLY	;DO DELAY TO CATCH MDS
3473	014236	013737	001406	007412	MOV	T10,TEMP1	
3474	014244	004737	032320		JSR	PC,FRDY	;FIND RDY

```

3475 014250 104117 ERROR 117 ;NO RDY AFTER SELECT DRIVE CMD.
3476 014252 032737 11 000 007354 BIT #CERR,HCS1
3477 014260 001052 2$ BNE
3478 014262 013737 007402 007412 MOV HMR2,TEMP1
3479 014270 042737 177770 007412 BIC #1C<DRVMSK>,TEMP1
3480 014276 020037 007412 CMP RO,TEMP1 ;S/B SAME
3481 014302 001020 3$ BNE
3482 014304 005700 TST RO
3483 014306 001003 4$ BNE
3484 014310 005737 007466 TST DDPCH ;SEE IF XXDP CHAIN MODE
3485 014314 001016 5$ BNE
3486 014316 005237 007474 4$: INC DRIVS ;INC DRIVE COUNT.
3487 014322 005211 INC (R1) ;SET DRIVE PRESENT FLAG
3488 014324 053711 001170 BIS STMP4,(R1) ;ADD CDT IF RK07
3489 014330 104401 001205 TYPE $CRLF
3490 014334 010046 MOV RO,-(SP) ;SAVE RO FOR TYPEOUT
3491 ;TYPE DR #
3492 014336 104403 TYPOS ;GO TYPE--OCTAL ASCII
3493 014340 001 .BYTE 1 ;TYPE 1 DIGIT(S)
3494 014341 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
3495 014342 000403 BR 5$
3496
3497 014344 004737 033026 3$: JSR PC,BYP ;TYPE BYPASS DR #
3498 014350 104001 ERROR 1 ;WRITTEN DR # DOES NOT MATCH RKMR2 DR #
3499
3500 014352 005721 5$: TST (R1)+ ;SHIFT PTR TO NEXT DR. FLAG
3501 014354 005200 INC RO ;INC DR #
3502 014356 005037 001170 CLR STMP4 ;CLEAR RK07 FLAG
3503 014362 022700 000010 CMP #8.,RO
3504 014366 001267 1$ BNE ;MORE LEFT.
3505 014370 005737 007474 TST DRIVS
3506 014374 001065 BNE 10$
3507 014376 104076 ERROR 76 ;NO DRIVES FOUND ON BUSS
3508 014400 000000 HALT ;SETUP CORRECTLY
3509 014402 000137 013356 JMP ST5 ;AND PRESS 'CONTINUE'
3510 014406 032737 000040 007370 2$: BIT #DTYE,HER
3511 014414 001405 BEQ 13$
3512 014416 012737 002000 001170 MOV #CDT,STMP4 ;ADD CDT
3513 014424 000137 014146 JMP 1$ ;TRY AGAIN
3514
3515 014430 032737 001000 007356 13$: BIT #MDS,HCS2
3516 014436 001015 6$ BNE
3517 014440 032737 000400 007356 BIT #UFE,HCS2
3518 014446 001015 7$ BNE
3519 014450 032737 000001 007366 BIT #DRA,HDS
3520 014456 001015 8$ BNE
3521 014460 032737 010000 007356 BIT #NED,HCS2
3522 014466 001424 9$ BEQ
3523 014470 000730 BR 5$
3524
3525 014472 004737 033026 6$: JSR PC,BYP ;TYPE BYP DR #
3526 014476 104002 ERROR 2 ;MDS DETECTED
3527 014500 000724 BR 5$
3528
3529 014502 004737 033026 7$: JSR PC,BYP
3530 014506 104003 ERROR 3 ;UFE DETECTED

```

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 70
T2 SIZE THE BUSS

SEQ 0070

```

3531 014510 000720 BR 5$
3532
3533 014512 032737 010000 007356 8$: BIT #NED,HCS2
3534 014520 001676 BEQ 4$
3535 014522 104401 045266 TYPE MSG15 ;DRV#
3536 014526 010046 MOV RC,-(SP) ;SAVE RD FOR TYPEOUT
3537 ;TYPE DR#
3538 014530 104403 TYPOS ;GO TYPE--OCTAL ASCII
3539 014532 001 .BYTE 1 ;TYPE 1 DIGIT(S)
3540 014533 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
3541 014534 104010 ERROR 10 ;DRA & NED BOTH SET
3542 014536 000705 BR 5$
3543
3544 014540 004737 033026 9$: JSR PC,BYP
3545 014544 104004 ERROR 4 ;NO DRA & NO NED = OTHER PORT SELECTED
3546 014546 000701 BR 5$
3547 014550 000137 015146 10$: JMP NUDRV
3548
3549 014554
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565 014554 000004
3566 014556 012737 000001 001174
3567 014564 012706 001100
3568 014570 005000
3569 014572 012701 007476
3570 014576
3571 014576 104415
3572 014600 012706 001100
3573
3574 014604 012765 000040 000010
3575 014612 013737 001406 007412
3576 014620 004737 032320
3577 014624 104120
3578 014626 010065 000010
3579 014632 012737 000001 007354
3580 014640 053737 001170 007354
3581 014646 013765 007354 000000
3582 014654 013737 001420 007412
3583 014662 004737 033010
3584 014666 013737 001406 007412
3585 014674 004737 032320
3586 014700 104117

VERIFY:
*****
*TEST 3 VERIFY OPERATOR DRIVE SELECTIONS
*
* THIS TEST IS ENTERED ONLY IF DRIVE SELECTION IS NOT
* DEFAULTED. EVERY DRIVE FROM 0 TO 7 IS ADDRESSED &
* CONTROLLER ERROR (CERR) IS EXAMINED. IF NOT SET, THE
* PROGRAM WILL ASSUME THE DRIVE IS PRESENT AS AN RK06.
* IF CERR WAS SET, THAT DRIVE WILL BE BYPASSED
* ONLY IF THE ERROR WAS A RESULT OF MDS OR UFE SET OR BOTH
* NED & DRA RESET (WRONG PORT). IF CERR IS A RESULT OF
* NED ONLY, IT IS CHECKED AGAINST THE INPUTTED INFOR TO
* VERIFY IT WAS NOT SPECIFIED.
* IF CERR DUE TO DTYE, THE DRIVE WILL BE TESTED AS AN RK07.
*****
*ST3: SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR
CLR RD ;DRIVE ADDR
MOV #DRIVO,R1 ;DRIVE FLAG
1$: SCOP1
MOV #STACK,SP ;RESTORE STK PTR
MOV #SCLR,RKCS2(R5) ;SUBSYSTEM CLEAR
MOV T10,TEMP1 ;SET TIME OUT
JSR PC,FRDY ;FIND RDY
ERROR 120 ;NO RDY AFTER SCLR
MOV RD,RKCS2(R5) ;DRV ADDR
MOV #SELDRV,HCS1
BIS $TMP4,HCS1 ;ADD CDT IF RK07
MOV HCS1,RKCS1(R5) ;GET STATUS
MOV T50000,TEMP1
JSR PC,OLY ;DO DELAY TO CATCH MDS
MOV T10,TEMP1
JSR PC,FRDY ;FIND RDY
ERROR 117 ;NO RDY AFTER SELECT DRIVE CMD.

```

G06

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 71
T3 VERIFY OPERATOR DRIVE SELECTIONS

SEQ 0071

3587	014702	032737	100000	007354		BIT	#CERR,HCS1	
3588	014710	001036				BNE	2\$	
3589	014712	013737	007402	007412		MOV	HMR2,TEMP1	
3590	014720	042737	177770	007412		BIC	#IC(DRVMSK),TEMP1	
3591	014726	020037	007412			CMP	RO,TEMP1	;S/B SAME
3592	014732	001014				BNE	3\$	
3593	014734	005711			11\$:	TST	(R1)	
3594	014736	001402				BEQ	4\$	
3595	014740	053711	001170			BIS	STMP4,(R1)	;SET CDT IF RK07.
3596	014744	005721			4\$:	TST	(R1)+	;SHIFT PTR TO NEXT DR FLAG
3597	014746	005200				INC	RO	;INC DR#
3598	014750	005037	001170			CLR	STMP4	;CLEAR CDT FOR NEXT DRIVE
3599	014754	022700	000010			CMP	#8.,RO	
3600	014760	001306				BNE	1\$	;MORE LEFT
3601	014762	000475				BR	TST4	;GO TO NEXT TEST
3602								
3603	014764	004737	033026		3\$:	JSR	PC,BYP	;TRY BYPASS DRIVE#
3604	014770	104001				ERROR	1	;WRITTEN DR# DOES NOT MATCH RKMR2 DR#
3605	014772	005711				TST	(R1)	
3606	014774	001763				BEQ	4\$	;BRANCH IF NOT SPEC BY INPUT
3607	014776	005337	007474		12\$:	DEC	DRIVS	;DECREMENT TOTAL DRIVS
3608	015002	005011				CLR	(R1)	;CLEAR DRIVE FLAG
3609	015004	000757				BR	4\$	
3610								
3611	015006	032737	000040	007370	2\$:	BIT	#DTYE,HER	
3612	015014	001405				BEQ	13\$	
3613	015016	012737	002000	001170		MOV	#CDT,STMP4	;ADD CDT
3614	015024	000137	014576			JMP	1\$	;TRY AGAIN
3615								
3616	015030	032737	001000	007356	13\$:	BIT	#MDS,HCS2	
3617	015036	001027				BNE	6\$	
3618	015040	032737	000400	007356		BIT	#UFE,HCS2	
3619	015046	001027				BNE	7\$	
3620	015050	032737	000001	007366		BIT	#ORA,HDS	
3621	015056	001005				BNE	8\$	
3622	015060	032737	010000	007356		BIT	#NED,HCS2	
3623	015066	001423				BEQ	9\$	
3624	015070	000404				BR	10\$	
3625	015072	032737	010000	007356	8\$:	BIT	#NED,HCS2	
3626	015100	001715				BEQ	11\$	
3627	015102	005711			10\$:	TST	(R1)	
3628	015104	001717				BEQ	4\$	
3629								
3630	015106	004737	033026			JSR	PC,BYP	;TYPE BYPASS DRIVE#
3631	015112	104006				ERROR	6	
3632	015114	000730				BR	12\$	
3633								
3634	015116	004737	033026		6\$:	JSR	PC,BYP	;TYPE BYPASS DRIVE#
3635	015122	104002				ERROR	2	;MDS DETECTED
3636	015124	000724				BR	12\$	
3637								
3638	015126	004737	033026		7\$:	JSR	PC,BYP	
3639	015132	104003				ERROR	3	;UFE DETECTED
3640	015134	000720				BR	12\$	
3641								
3642	015136	004737	033026		9\$:	JSR	PC,BYP	

H06

CZR61EO UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN 78 11:25 PAGE 72
T3 VERIFY OPERATOR DRIVE SELECTIONS

SEQ 0072

3643 015142 104004
3644 015142 000714ERROR 4 ;DRA & NED RESET - OTHER PORT SELECTED
BR 125; THIS PART OF THE PROGRAM WILL BE REPEATED FOR EACH
; DRIVE PRESENT
; 'SUNIT' CONTAINS THE ADDRESS OF THE DRIVE CURRENTLY
; UNDER TEST3655 015146 005037 001516
3656
3657 015152 005037 001170
3658
3659NUDRV: CLR BYPCERR ; ENTER HERE FROM LAST TEST
CLR \$TMP4 ; ALLOW CHECKING CERR IN 'FRDY'
; CLEAR RK07 FLAG; *****
; *TEST 4 FIND NEXT DRIVE TO BE TESTED
; *; * THIS TEST FINDS THE NEXT DRIVE PRESENT & PUTS THAT
; * ADDRESS IN 'SUNIT' & \$TMP4 IS SET TO CDT IF DRIVE IS RK07.
; * THROUGHOUT THE FOLLOWING TESTS, THE DRIVE TESTED IS
; * THE DRIVE WHOSE ADDRESS IS IN 'SUNIT'.
; *; *****
; *ST4: SCOPE3668 015156 000004
3669 015160 012737 000001 001174
3670 015166 012706 001100
3671 015172 012737 000004 001214
3672 015200 012737 000004 001102
3673MOV #1,\$TIMES ; DO 1 ITERATION
MOV \$STACK,SP ; RESTORE STK PTR
MOV \$STN-1,\$TESTN
MOV \$STN-1,\$STSTNM3674 015206 005737 007474
3675 015212 001004
3676 015214 104401 045432
3677 015220 000137 031512
3678TST DRVS ; ANY DRIVES PRESENT?
BNE 4\$; YES BRANCH
TYPE MSG27 ; ALL DRIVES TESTED
JMP \$EOP1 ; NO, GO TO END3679 015224 013701 001342
3680 015230 005737 001220
3681 015234 001402
3682 015236 005237 001222
3683 015242 005711
3684 015244 001002
3685 015246 005721
3686 015250 000772
3687 015252 005737 007466
3688 015256 001405
3689 015260 005737 001222
3690 015264 001002
3691 015266 005721
3692 015270 000762
36934\$: MOV DRVPTR,R1 ; ADDR OF NEXT DRIVE FLAG
TST \$DEVCT ; IS FIRST DRIVE BEING CHECKED
BEQ 2\$; YES, BRANCH
1\$: INC \$SUNIT ; INCR DRIVE ADDR TO NEXT DRIVE
2\$: TST (R1) ; IS DRIVE PRESENT?
BNE 5\$; BR IF YES
TST (R1)+ ; ELSE FIND NEXT DRIVE
BR 1\$
5\$: TST DDPCH ; DDP CHAIN MODE?
BEQ 3\$; BR IF NO
TST \$SUNIT ; ELSE SEE IF DRV 0
BNE 3\$; BR IF NO
TST (R1)+ ; ELSE FIND NEXT DRIVE PRESENT
BR 1\$3694 015272 032721 002000
3695 015276 001403
3696 015300 012737 002000 001170
3697 015306 010137 001342
3698 015312 104401 0452663\$: BIT #CDT,(R1)+ ; SEE IF DRIVE UNDER TEST IS RK07
BEQ 6\$; BR IF NO
MOV #CDT,\$TMP4 ; ELSE SET RK07 FLAG
6\$: MOV R1,DRVPTR ; STORE POINTER TO NEXT DR FLAG
TYPE ,MSG15 ; "DRIVE"

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 73
T4 FIND NEXT DRIVE TO BE TESTED

SEQ 0073

```

3699 015316 013700 001222      MOV    $UNIT,RO
3700 015322 010046              MOV    RO,-(SP)          ;:SAVE RO FOR TYPEOUT
3701                                ;:DRIVE #
3702 015324 104403              TYP0S          ;:GO TYPE--OCTAL ASCII
3703 015326      001              .BYTE    1          ;:TYPE 1 DIGIT(S)
3704 015327      000              .BYTE    0          ;:SUPPRESS LEADING ZEROS
3705
3706                                ;
3707                                TYPE    ,SCLF
3708 015330 005737 001170      TST    $TMP4          ;:SEE IF RK07 UNDER TEST
3709 015334 001017              BNE     7$           ;:BR IF YES
3710 015336 012737 000632 015434      MOV    #632,LC          ;:ELSE LOAD RK06 PARAMETERS
3711 015344 005037 015444              CLR     E.DOT
3712 015350 012737 001000 015436      MOV    #1000,MC1
3713 015356 012737 000777 015440      MOV    #777,MASK
3714 015364 012737 160017 015442      MOV    #160017,MASK1
3715 015372 000425              BR      TST5          ;:GOTO NEXT TEST
3716
3717 015374 012737 001456 015434 7$:    MOV    #1456,LC          ;:LOAD RK07 PARAMETERS
3718 015402 012737 000400 015444      MOV    #0.DOT,E.DOT
3719 015410 012737 002000 015436      MOV    #2000,MC1
3720 015416 012737 001777 015440      MOV    #1777,MASK
3721 015424 012737 140017 015442      MOV    #140017,MASK1
3722 015432 000405              BR      TST5          ;:GOTO NEXT TEST
3723
3724 015434 000000              LC:      0          ;:LAST CYL
3725 015436 000000              MC1:    0          ;:MAJ CYL + 1 SHIFT
3726 015440 000000              MASK:   0
3727 015442 000000              MASK1:  0
3728 015444 000000              E.DOT:   0          ;:EXPECTED DRIVE TYPE TO E.A0
3729
3730 015446              PFSRT:          ;:ENTER HERE FOR POWER FAIL RESTART
3731              ;:*****
3732              ;:TEST 5          PRINT DRIVE SERIAL NUMBER
3733              ;:
3734              ;:THIS TEST READS & PRINTS THE DRIVE SERIAL # FROM MSG A, WORD 11
3735              ;:IN BCD & IS PERFORMED ON THE 1ST PASS ONLY
3736              ;:*****
3737
3738 015446 000004              TST5:    SCOPE
3739 015450 012737 000001 001174      MOV    #1,$TIMES          ;:DO 1 ITERATION
3740 015456 012706 001100              MOV    #STACK,SP          ;:RESTORE STK PTR
3741
3742 015462 005737 001216              TST    $PASS
3743 015466 001042              BNE     TST6          ;:GO TO NEXT IF NOT FIRST PASS
3744 015470 004737 034214              JSR    PC,SUBCLR          ;:DO SUBSYS CLEAR
3745 015474 104024              ERROR   24          ;:CERR AFTER SCLR
3746
3747 015476 104401 045300              TYPE    ,MSG16          ;:DRIVE SERIAL NO.
3748 015502 012765 000003 000026      MOV    #3,RKMR1(R5)      ;:SELECT BYTE 3
3749 015510 004737 033664              JSR    PC,GSTAT          ;:GET STATUS
3750 015514 013701 007402              MOV    HMA2,R1          ;:GET SERIAL #
3751 015520 012704 042614              MOV    #SOCTVL,R4          ;:GET ADDR CHAR BUFF
3752 015524 010446              MOV    R4,-(SP)          ;:STORE ON STACK FOR $SUPRS
3753 015526 012703 000003              MOV    #3,R3          ;:SETUP CHAR COUNT
3754 015532 006101              ROL     R1          ;:INITIALIZE BIT POSITIONS

```

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 74
TS PRINT DRIVE SERIAL NUMBER

SEQ 0074

3755 015534 006101
3756 015536 006101
3757 015540 006101
3758 015542 006101
3759 015544 006101
3760 015546 010100
3761 015550 042700 177760
3762 015554 052700 000060
3763 015560 110024
3764 015562 005303
3765 015564 001364
3766 015566 105014

3768 015570 004737 043062

3769
3770
3771
3772
3773
3774
3775
3776
3777
3778 015574 000004
3779 015576 012737 000001 001174
3780 015604 012706 001100
3781
3782 015610 004737 034214
3783 015614 104024
3784
3785 015616 032737 000100 007402
3786 015624 001021
3787
3788 015626 104415
3789 015630 012706 001100
3790
3791 015634 004737 034214
3792 015640 104024
3793
3794 015642 012737 000003 007354
3795 015650 004737 032224
3796 015654 104116
3797
3798 015656 032737 000100 007402
3799 015664 001001
3800 015666 104027

```

1$:  ROL    R1
      ROL    R1      ;GET NEXT 4 BITS
      ROL    R1
      ROL    R1
      ROL    R1
      MOV    R1,R0    ;GET WORKING COPY
      BIC    #177760,R0 ;CLEAR ALL BUT LOW 4 BITS
      BIS    #60,R0   ;CONVERT TO ASCII DIGIT
      MOVB   R0,(R4)+ ;PUT ASCII DIGIT INTO CHAR BUFF
      DEC    R3
      BNE    1$      ;BR IF ALL 3 CHARS NOT DONE
      CLRB   (R4)    ;ELSE INSERT NULL TERMINATOR

      JSR    PC,$SUPRS ;TYPE
      TYPE   $,SCLF
      TYPE   $,SCLF

```

```

*****
;TEST 6      SET VV WITH PACK COMMAND
;
; IF VV IS RESET, THE PACK COMMAND IS USED TO SET IT.
;
*****

```

```

1$T6:  SCOPE
      MOV    #1,$TIMES ;DO 1 ITERATION
      MOV    #STACK,SP ;RESTORE STK PTR

      JSR    PC,SUBCLR
      ERROR  24        ;CERR AFTER SCLR

      BIT    #D.VV,HMR2
      BNE    TST7      ;;GO TO NEXT TEST IF VV SET

      SCOPE
      MOV    #STACK,SP ;RESTORE STK PTR

      JSR    PC,SUBCLR
      ERROR  24        ;CERR AFTER SCLR

      MOV    #PACK,HCS1
      JSR    PC,DOCMD   ;DO PACK CMD & GET CONTR RDY
      ERROR  116        ;RDY NOT SET AFTER PACK CMD

      BIT    #D.VV,HMR2
      BNE    TST7      ;;GO TO NEXT TEST IF VV NOW SET
      ERROR  27        ;PACK DID NOT SET V.V.

```

```

*****
;TEST 7      READ & SAVE BAD SECTOR INFO - TYPE PACK SERIAL #
;
; THIS TEST VERIFIES THAT CYL 632 (1456 FOR RK07), TRACK 2 CAN BE READ.
; THIS AREA CONTAINS BAD SECTOR INFO WHICH IS WRITTEN BY THE
; FACTORY DURING MANF. ALL BAD SECTOR INFO (BSE) WILL BE STORED
; AT THIS TIME TO MASK FUTURE READ HEADER OR DATA ERROR PRINTOUTS.
;

```

K06

CZR61EO UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 75
T7 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

SEQ 0075

SECTORS 0,2,4,6,8 CONTAIN IDENTICAL INFO FOR 22 SECTOR HARDWARE DETECTED BAD SEC
SECTORS 10,12,14,16,18,20 CONTAIN IDENTICAL INFO FOR 22 SECTOR SOFTWARE DETECTED

SECTORS 1,3,5,7,9 CONTAIN IDENTICAL INFO FOR 20 SECTOR HARDWARE DETECTED BAD SEC
SECTORS 11,13,15,17,19,21 CONTAIN IDENTICAL INFO FOR 20 SECTOR SOFTWARE DETECTED

IF BSE INFO CANNOT BE READ, OR IF AFTER READING THE BSE INFO
IT IS DETERMINED THAT AN ALIGNMENT CARTRIDGE IS USED,
A MESSAGE WILL BE TYPED INDICATING THAT ALL
FUTURE FORMAT AND READ-WRITE TESTS WILL BE BYPASSED.
THIS IS DONE SO AS NOT TO DESTROY BSE INFO OR AN ALIGNMENT PACK BY WRITING

THE PACK SERIAL # IS TYPED IN OCTAL & FOR THE FIRST PASS ONLY.

ST7:

SCOPE
MOV #1,STIMES ;DO 1 ITERATION
MOV #STACK,SP ;RESTORE STK PTR

JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR

MOV #CCLR,RKCS1(R5)
MOV \$UNIT,RKCS2(R5)
MOV #RECAL,HCS1
JSR PC,DOCMD ;DO RECAL CMD & GET CONTR RDY
ERROR 124 ;RDY NOT SET AFTER RECAL CMD

MOV #1,RKMR1(R5) ;SELECT WORD 1
JSR PC,GSTAT
BIT #0,RTZ,HMR2
BNE 64\$
ERROR 244 ;RTZ NOT SET DURING RECAL CMD
MOV T10,TEMP2 ;SETUP TIMEOUT
JSR PC,FATT1 ;FIND ATTN
ERROR 55 ;NO ATTN AFTER RECAL CMD

MOV #CCLR,RKCS1(R5)
MOV \$UNIT,RKCS2(R5) ;DRIVE#
MOV #CLEAR,HCS1
JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
JSR PC,TSTATN ;TEST FOR ATTN
BR 65\$
ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD

65\$:

JSR PC,SUBCLR
ERROR 24 ;CERR AFTER SCLR

CLR TEMP2 ;SECTOR CTR
CLR TEMP3 ;0=22 SECTOR HARDWARE DETECTED TABLE

3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827 015670 000004
3828 015672 012737 000001 001174
3829 015700 012706 001:00
3830
3831 015704 004737 034214
3832 015710 104024
3833
3834
3835 015712 012765 100000 000000
3836 015720 013765 001222 000010
3837 015726 012737 000013 007354
3838 015734 004737 032224
3839 015740 104124
3840
3841 015742 012765 000001 000026
3842 015750 004737 033664
3843 015754 032737 020000 007402
3844 015762 001001
3845 015764 104244
3846 015766 013737 001406 007414 64\$:
3847 015774 004737 032634
3848 016000 104055
3849
3850 016002 012765 100000 000000
3851 016010 013765 001222 000010
3852 016016 012737 000005 007354
3853 016024 004737 032224
3854 016030 104151
3855 016032 004737 032602
3856 016036 000401
3857 016040 104154
3858 016042
3859
3860
3861
3862 016042 004737 034214
3863 016046 104024
3864
3865 016050 005037 007414
3866 016054 005037 007416

L06

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 76
T7 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

SEQ 0076

```

3867                                ;1=22 SECTOR SOFTWARE DETECTED TABLE
3868                                ;2=20 SECTOR HARDWARE DETECTED TABLE
3869                                ;3=20 SECTOR SOFTWARE DETECTED TABLE
3870 016060 012737 003336 007420      MOV    #BSE22H,TEMP4 ;STORE 22 SECTOR HARDWARE BSE ADDR.
3871 016066 013765 007420 000004      MOV    TEMP4,RKBA(R5)
3872 016074 012737 001000 007422      MOV    #1000,TEMP5 ;TRACK 2, SECTOR 0
3873 016102 013765 007422 000006      MOV    TEMP5,RKDA(R5)
3874
3875 016110 013765 015434 000020 1$:    MOV    LC,RKDC(R5) ;LAST CYL
3876 016116 012765 177400 000002      MOV    #-256,RKWC(R5) ;LOAD WORD CT
3877 016124 012737 000021 007354      MOV    #RDDATA,HCS1
3878 016132 004737 032262              JSR    PC,DATCMD ;DO READ DATA CMD & GET CONTR RDY
3879 016136 104226              ERROR 226 ;NO RDY AFTER READ DATA CMD
3880 016140 004737 033664              JSR    PC,GSTAT ;GET FRESH DATA
3881 016144 032737 100000 007354      BIT    #CERR,HCS1
3882 016152 001504              BEQ    8$
3883 016154 104227              ERROR 227 ;CERR AFTER READ DATA CMD
3884
3885 016156 012737 010340 007444      MOV    #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
3886 016164 005037 007446              CLR    E.B0 ;EXPECTED MSG B0
3887 016170 012737 001720 007450      MOV    #<0.SPOK!D.CART!D.DDOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
3888 016176 012737 000001 007452      MOV    #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
3889 016204 005037 007454              CLR    E.A2 ;EXPECTED MSG A2
3890 016210 012737 000002 007456      MOV    #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
3891 016216 012737 000003 007462      MOV    #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
3892
3893 016224 004737 033042              JSR    PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
3894 016230 000000              .WORD 0!0!0 ;& MSGS SPECIFIED HERE
3895 016232 104054              ERROR 54 ;MSG A0 ERROR AFTER READ DATA CMD
3896 016234 104026              ERROR 26 ;MSG B0 ERROR
3897 016236 104056              ERROR 56 ;MSG A1 ERROR
3898 016240 104030              ERROR 30 ;MSG B1 ERROR
3899
3900 016242 004737 034214              JSR    PC,SUBCLR
3901 016246 104024              ERROR 24 ;CERR AFTER SUBCLR
3902
3903 016250 005237 007414              INC    TEMP2
3904 016254 023727 007414 000005      CMP    TEMP2,#5 ;READ ALL 5 SECTORS?
3905 016262 001023              BNE    5$
3906 016264 005737 007416              TST    TEMP3
3907 016270 001002              BNE    2$
3908 016272 104233              ERROR 233 ;CANT READ SECTORS 0,2,4,6,8
3909 016274 000430              BR     3$
3910
3911 016276 023727 007416 000001 2$:    CMP    TEMP3,#1
3912 016304 001002              BNE    4$
3913 016306 104230              ERROR 230 ;CANT READ SECTORS 10,12...
3914 016310 000422              BR     3$
3915
3916 016312 023727 007416 000002 4$:    CMP    TEMP3,#2
3917 016314 001002              BNE    6$
3918 016322 104234              ERROR 234 ;CANT READ SECTORS 1,3,5 ...
3919 016324 000414              BR     3$
3920
3921 016326 104231              6$:    ERROR 231 ;CANT READ SECTORS 11,13,15 ...
3922 016330 000412              BR     3$

```

M06

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 77
T7 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

SEQ 0077

```

3923
3924 016332 013765 007420 000004 5$: MOV TEMP4,RKBA(R5) ;RESTORE TABLE ADDR
3925 016340 062737 000002 007422 ADD #2,TEMP5 ;READ 2 SECTORS FROM LAST
3926 016346 013765 007422 000006 MOV TEMP5,RKDA(R5)
3927 016354 000655 BR 1$
3928
3929 016356 005237 001512 3$: INC BSEERR ;SET BSE FLAG
3930 016362 000550 BR TST10 ;GO TO NEXT TEST
3931 016364 005737 003344 8$: TST BSE22H+6 ;TEST CARTRIDGE TYPE
3932 016370 001404 BEQ 9$ ;BRANCH IF DATA CARTRIDGE
3933 016372 104235 ERROR 235 ;ALIGNMENT CARTRIDGE USED
3934 016374 005237 001512 INC BSEERR ;SET BSE ERROR FLAG
3935 016400 000476 BR 10$
3936
3937 016402 005237 007416 9$: INC TEMP3
3938 016406 023727 007416 000001 CMP TEMP3,#1
3939 016414 001020 BNE 11$
3940 016416 005037 007414 CLR TEMP2 ;SECTOR CTR
3941 016422 012737 005336 007420 MOV #BSE22S,TEMP4 ;STORE 22 SECTOR SOFTWARE BSE ADDR
3942 016430 013765 007420 000004 MOV TEMP4,RKBA(R5)
3943 016436 012737 001012 007422 MOV #1012,TEMP5 ;TRACK 2, SECTOR 12(8)
3944 016444 013765 007422 000006 MOV TEMP5,RKDA(R5)
3945 016452 000137 016110 JMP 1$ ;REPEAT
3946
3947 016456 023727 007416 000002 11$: CMP TEMP3,#2
3948 016464 001020 BNE 12$
3949 016466 005037 007414 CLR TEMP2 ;SECTOR CTR
3950 016472 012737 002336 007420 MOV #BSE20H,TEMP4 ;STORE 20 SECTOR HARDWARE BSE ADDR.
3951 016500 013765 007420 000004 MOV TEMP4,RKBA(R5)
3952 016506 012737 001001 007422 MOV #1001,TEMP5 ;TRACK 2, SECTOR 1
3953 016514 013765 007422 000006 MOV TEMP5,RKDA(R5)
3954 016522 000137 016110 JMP 1$ ;REPEAT
3955
3956 016526 023727 007416 000003 12$: CMP TEMP3,#3
3957 016534 001020 BNE 10$
3958 016536 005037 007414 CLR TEMP2 ;SECTOR CTR
3959 016542 012737 004336 007420 MOV #BSE20S,TEMP4 ;STORE 20 SECTOR SOFTWARE BSE ADDR
3960 016550 013765 007420 000004 MOV TEMP4,RKBA(R5)
3961 016556 012737 001013 007422 MOV #1013,TEMP5 ;TRACK 2, SECTOR 13(8)
3962 016564 013765 007422 000006 MOV TEMP5,RKDA(R5)
3963 016572 000137 016110 JMP 1$ ;REPEAT
3964
3965 016576 005737 001216 10$: TST $PASS
3966 016602 001040 BNE TST10 ;GO TO NEXT TST IF NOT 1'ST PASS
3967 016604 104401 045324 TYPE MSG17 ;CART SERIAL #
3968 016610 012746 003336 MOV #BSE22H, -(SP)
3969 016614 004737 042512 JSR PC,$DB20 ;CONVERT DBL BINARY WORD TO OCTAL
3970 016620 004737 043062 JSR PC,$SUPRS ;TYPE SERIAL #
3971 016624 104401 001205 TYPE $CRLF
3972 016630 104401 001205 TYPE $CRLF
3973
3974 016634 004737 034214 JSR PC,SUBCLR
3975 016640 104024 ERROR 24 ;CERR AFTER SCLR
3976 ;GO BACK TO CYL 0
3977
3978 016642 012737 000017 007354 MOV #SEEK,HCS1

```

N06

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 78
T7 READ & SAVE BAD SECTOR INFO & TYPE PACK SERIAL #

SEQ 0078

```

3979 016650 004737 032224      JSR    PC,DOCMD      ;DO SEEK CMD & GET CONTR READY
3980 016654 104131              ERROR    131      ;NO RDY AFTER SEEK CMD
3981
3982 016656 013737 001420 007412  MOV    T50000,TEMP1    ;SETUP TIMEOUT
3983 016664 004737 032730      JSR    PC,FAT12      ;FIND ATTN
3984 016670 104132              ERROR    132      ;NO ATTN AFTER SEEK CMD
3985
3986 016672 032737 100000 007354  BIT     #CERR,HCS1
3987 016700 001401              BEQ     66$
3988 016702 104210              ERROR    210      ;CERR AFTER SEEK CMD
3989
3990 016704              66$:
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007 016704 000004              ;*****
4008 016706 012737 000001 001174  ;TEST 10      BASIC WRITE DATA TEST; 1 WORD
4009 016714 012706 001100      ;
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027
4028
4029
4030
4031
4032
4033
4034

```

THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE JUST ONE WORD,
ALL SECTORS ON CYL 0 ARE GIVEN IDENTICAL HEADERS &
A WRITE COMMAND IS ISSUED. READ & WRITE CHECK COMMANDS ARE NOT
PERFORMED. THIS TEST PROVIDES THE TIGHTEST POSSIBLE SCOPE LOOP
FOR A WRITE ERROR.

```

;*****
;ST10: SCOPE
;MOV    #1,$TIMES      ;DO 1 ITERATION
;MOV    #STACK,SP      ;RESTORE STK PTR
;TST    BSERR          ;SEE IF ALIGN CART
;BEQ     2$            ;BR IF NO
;TYPE    ,MSG40        ;BSE OR ALIGN CART USED
;TYPE    ,MSG26        ;ABORTING BAL OF TESTS
;JMP     $EOP
;2$: JSR    PC,SUBCLR    ;CERR AFTER SCLR
;ERROR    24
;INC     BADHDR        ;USED FOR VALID HALT
;MOV     #HDTAB,RO      ;MAKE ALL CYL 0 HEADERS IDENTICAL
;1$: CLR     (RO)+      ;HEADER WORD 0: CYL 0
;MOV     #140000,(RO)+  ;HEADER WORD 1: SECTOR 0
;MOV     #140000,(RO)+  ;HEADER WORD 2: XOR OF 1 & 2
;CMP     RO,#HDTAB+132. ;ALL HEADERS DONE? (22X6=132)
;BNE     1$            ;BR IF NO
;MOV     #HDTAB,RKBA(R5) ;HEADER TABLE
;MOV     #-66.,RKWC(R5)  ;WORD COUNT
;MOV     #<WRHEAD>,HCS1

```

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 79
T10 BASIC WRITE DATA TEST; 1 WORD

SEQ 0079

4035	017022	004737	032262		JSR	PC,DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
4036	017026	104200			ERROR	200	;NO RDY AFTER WRITE HEADER CMD
4037	017030	004737	033664		JSR	PC,GSTAT	;GET FRESH STATUS
4038	017034	032737	100000	007354	BIT	#CERR,HCS1	
4039	017042	001405			BEQ	64\$	
4040	017044	104201			ERROR	201	;CERR AFTER WRITE HEADER CMD
4041	017046	104401	045354		TYPE	MSG26	;ABORTING BAL OF TESTS
4042	017052	000137	031452		JMP	\$EOP	
4043	017056			64\$:			
4044							
4045							
4046	017056	104415			SCOP1		
4047	017060	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
4048							
4049	017064	004737	034214		JSR	PC,SUBCLR	
4050	017070	104024			ERROR	24	;CERR AFTER SCLR
4051							
4052	017072	005037	001402		CLR	SECTOR	
4053	017076	013765	001402	000006	MOV	SECTOR,RKDA(R5)	;TRACK/SECTOR #
4054	017104	012765	001500	000004	MOV	#DATA1,RKBA(R5)	;DATA TO BE ALL 1'S
4055	017112	012765	177777	000002	MOV	#-1,RKWC(R5)	;WORD COUNT=1
4056							
4057							
4058	017120	012737	000023	007354	MOV	#<WRDATA>,HCS1	
4059	017126	004737	032262		JSR	PC,DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
4060	017132	104011			ERROR	11	;NO RDY AFTER WRITE DATA CMD
4061	017134	004737	033664		JSR	PC,GSTAT	;GET FRESH STATUS
4062	017140	032737	100000	007354	BIT	#CERR,HCS1	
4063	017146	001465			BEQ	68\$	;BR IF NO ERRORS
4064							
4065	017150	032737	000200	007370	BIT	#BSE,HER	;SEE IF BAD SECTOR FLAG
4066	017156	001421			BEQ	66\$	;BR IF NO
4067	017160	004737	035700		JSR	PC,TRUERR	;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4068	017164	000455			BR	67\$	;RETURN HERE IF NO
4069							
4070	017166	005237	001402		INC	SECTOR	;RETURN HERE IF YES
4071	017172	023727	001402	000012	CMP	SECTOR,#10.	;ARE 10 CONSEC. SECTORS BAD
4072	017200	001003			BNE	65\$	;BR IF NO
4073	017202	104046			ERROR	46	;ABORTING TEST DETECTED 10 BAD SECTORS
4074	017204	000137	017406		JMP	5\$	;BYPASS TEST
4075	017210	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	;TRY ANOTHER SECTOR
4076	017216	000137	017076		JMP	3\$	
4077	017222	104012		66\$:	ERROR	12	;CERR WITH WRITE DATA CMD
4078							
4079	017224	012737	010340	007444	MOV	#<O!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4080	017232	005037	007446		CLR	E.B0	;EXPECTED MSG B0
4081	017236	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4082	017244	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4083	017252	005037	007454		CLR	E.A2	;EXPECTED MSG A2
4084	017256	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4085	017264	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4086							
4087	017272	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4088	017276	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4089	017300	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
4090	017302	104023			ERROR	23	;MSG B0 ERROR

C07

CZR61EO UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MAC/11 30A(1052) 10-JAN-78 11:25 PAGE 80
T10 BASIC WRITE DATA TEST; 1 WORD

SEQ 0080

```

4091 017304 104053          ERROR 53          ;MSG A1 ERROR
4092 017306 104025          ERROR 25          ;MSG B1 ERROR
4093 017310 104401 045354    TYPE MSG26      ;ABORTING BAL OF TESTS
4094 017314 000137 031452    JMP $EOP
4095 017320 104063          67$: ERROR 63      ;BAD SECTOR NOT LISTED IN TABLE
4096 017322
4097
4098 017322 012737 010340 007444    MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4099 017330 005037 007446    CLR E.B0          ;EXPECTED MSG B0
4100 017334 012737 001720 007450    MOV #<0.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
4101 017342 012737 000001 007452    MOV #1,E.B1          ;MSG ID FOR EXPECTED MSG B1
4102 017350 005037 007454    CLR E.A2          ;EXPECTED MSG A2
4103 017354 012737 000002 007456    MOV #2,E.B2          ;MSG ID FOR EXPECTED MSG B2
4104 017362 012737 000003 007462    MOV #3,E.B3          ;MSG ID FOR EXPECTED MSG B3
4105
4106 017370 004737 033042    JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4107 017374 000003          .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4108 017376 104052          ERROR 52          ;MSG A0 ERROR AFTER WRITE DATA CMD
4109 017400 104023          ERROR 23          ;MSG B0 ERROR
4110 017402 104053          ERROR 53          ;MSG A1 ERROR
4111 017404 104025          ERROR 25          ;MSG B1 ERROR
4112 017406
4113 5$: *****
4114 ;TEST 11 BASIC WRITE DATA TEST; FULL SECTOR
4115 ;
4116 ; THIS TEST VERIFIES THE ABILITY OF THE DRIVE TO WRITE
4117 ; A FULL SECTOR. ALL ZEROS ARE WRITTEN BY THE WRITE DATA COMMAND
4118 ; & CHECKED BY A RD DATA COMMAND. A FURTHER CHECK IS PERFORMED
4119 ; BY THE WRT CHK COMMAND.
4120 ; THE ABOVE IS REPEATED FOR AN ALL ONES PATTERN.
4121 ;
4122 ; *****
4123 ;ST11: SCOPE
4124 017406 000004          MOV #1,$TIMES ;DO 1 ITERATION
4125 017410 012737 000001 001174    MOV #STACK,SP ;RESTORE STK PTR
4126 017416 012706 001100
4127 017422 004737 034214    JSR PC,SUBCLR
4128 017426 104024          ERROR 24          ;CERR AFTER SCLR
4129
4130 017430 012765 001522 000004    MOV #HDTAB,RKBA(R5) ;RESTORE TO 22 SECTOR
4131 017436 012765 177676 000002    MOV #-66.,RKWC(R5) ;STANDARD FORMAT
4132 017444 005037 001346    CLR TOCYL
4133
4134 017450 013737 001346 001362    MOV TOCYL,CALADD ;SETUP
4135 017456 012737 000000 001460    MOV #0,HEAD ;TO FILL
4136 017464 012737 000000 001466    MOV #0,FORMAT ;HEADER
4137 017472 004737 035212    JSR PC,FHDTAB ;TABLE
4138
4139
4140 017476 012737 000027 007354    MOV #<WRHEAD>,HCS1
4141 017504 004737 032262    JSR PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
4142 017510 104200          ERROR 200        ;NO RDY AFTER WRITE HEADER CMD
4143 017512 004737 033664    JSR PC,GSTAT ;GET FRESH STATUS
4144 017516 032737 100000 007354    BIT #CERR,HCS1
4145 017524 001405          BEQ 64$
4146 017526 104201          ERROR 201          ;CERR AFTER WRITE HEADER CMD

```

D07

CZR61ED UNIBUS RK6 DR PR*2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 81
T11 BASIC WRITE DATA TEST; FULL SECTOR

SEQ 0081

4147	017530	104401	045354			TYPE	MSG26		;ABORTING BAL OF TESTS
4148	017534	000137	031452			JMP	\$EOP		
4149	017540				64\$:				
4150									
4151	017540	005037	007340			CLR	BADHDR		;USED FOR VALID HALT
4152	017544	104415				SCOP1			
4153	017546	012706	001100			MOV	#STACK,SP		;RESTORE STK PTR
4154									
4155	017552	004737	034214			JSR	PC,SUBCLR		
4156	017556	104024				ERROR	24		;CERR AFTER SCLR
4157									
4158	017560	005037	001402			CLR	SECTOR		
4159	017564	013765	001402	000006	8\$:	MOV	SECTOR,RKDA(R5)		;SETUP SECTOR
4160	017572	012765	001474	000004		MOV	#DATA0,RKBA(R5)		;WRITE ALL 0'S
4161	017600	013700	001474			MOV	DATA0,R0		
4162	017604	052765	000020	000010	1\$:	BIS	#BAI,RKCS2(R5)		
4163	017612	012765	177400	000002		MOV	#-256.,RKWC(R5)		
4164									
4165	017620	012737	000023	007354		MOV	#(WRDATA),HCS1		
4166	017626	004737	032262			JSR	PC,DATCMD		;DO DATA X FOR CMD & GET CONTR RDY
4167	017632	104011				ERROR	11		;NO RDY AFTER WRITE DATA CMD
4168	017634	004737	033664			JSR	PC,GSTAT		;GET FRESH STATUS
4169	017640	032737	100000	007354		BIT	#CERR,HCS1		
4170	017646	001465				BEQ	68\$		;BR IF NO ERRORS
4171									
4172	017650	032737	000200	007370		BIT	#BSE,HER		;SEE IF BAD SECTOR FLAG
4173	017656	001421				BEQ	66\$		;BR IF NO
4174	017660	004737	035700			JSR	PC,TRUERR		;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4175	017664	000455				BR	67\$		;RETURN HERE IF NO
4176									
4177	017666	005237	001402			INC	SECTOR		;RETURN HERE IF YES
4178	017672	023727	001402	000012		CMP	SECTOR,#10.		;ARE 10 CONSEC. SECTORS BAD
4179	017700	001003				BNE	65\$		;BR IF NO
4180	017702	104046				ERROR	46		;ABORTING TEST DETECTED !. BAD SECTORS
4181	017704	000137	021156			JMP	7\$		;BYPASS TEST
4182	017710	012765	100000	000000	65\$:	MOV	#CCLR,RKCS1(R5)		;TRY ANOTHER SECTOR
4183	017716	000137	017564			JMP	8\$		
4184	017722	104012			66\$:	ERROR	12		;CERR WITH WRITE DATA CMD
4185									
4186	017724	012737	010340	007444		MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0		;EXPECTED MSG A0
4187	017732	005037	007446			CLR	E.B0		;EXPECTED MSG B0
4188	017736	012737	001720	007450		MOV	#(0.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1		;EXPECTED A1
4189	017744	012737	000001	007452		MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
4190	017752	005037	007454			CLR	E.A2		;EXPECTED MSG A2
4191	017756	012737	000002	007456		MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
4192	017764	012737	000003	007462		MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
4193									
4194	017772	004737	033042			JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1
4195	017776	000003				.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE
4196	020000	104052				ERROR	52		;MSG A0 ERROR AFTER WRITE DATA CMD
4197	020002	104023				ERROR	23		;MSG B0 ERROR
4198	020004	104053				ERROR	53		;MSG A1 ERROR
4199	020006	104025				ERROR	25		;MSG B1 ERROR
4200	020010	104401	045354			TYPE	MSG26		;ABORTING BAL OF TESTS
4201	020014	000137	031452			JMP	\$EOP		
4202	020020	104063			67\$:	ERROR	63		;BAD SECTOR NOT LISTED IN TAB E

```

4203 020022 68$:
4204
4205 020022 012737 010340 007444 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4206 020030 005037 007446 CLR E.B0 ;EXPECTED MSG B0
4207 020034 012737 001720 007450 MOV #<0.D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4208 020042 012737 000001 007452 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4209 020050 005037 007454 CLR E.A2 ;EXPECTED MSG A2
4210 020054 012737 000002 007456 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4211 020062 012737 000003 007462 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4212
4213 020070 004737 033042 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4214 020074 000003 .WORD T.A2!T.B2!0 ;8 MSGS SPECIFIED HERE
4215 020076 104052 ERROR 52 ;MSG A0 ERROR AFTER WRITE DATA CMD
4216 020100 104023 ERROR 23 ;MSG B0 ERROR
4217 020102 104053 ERROR 53 ;MSG A1 ERROR
4218 020104 104025 ERROR 25 ;MSG B1 ERROR
4219 020106 104415 SCOP1
4220 020110 012706 001100 MOV #STACK,SP ;RESTORE STK PTR
4221
4222 020114 004737 034214 JSR PC,SUBCLR
4223 020120 104024 ERROR 24 ;CERR AFTER SCLR
4224
4225 020122 013765 001402 000006 MOV SECTOR,RKDA(R5) ;SETUP SECTOR
4226 020130 012765 006336 000004 MOV #RDTAB,RKBA(R5)
4227 020136 012765 177400 000002 MOV #-256.,RKWC(R5)
4228
4229
4230 020144 012737 000021 007354 MOV #<RDDATA>,HCS1
4231 020152 004737 032262 JSR PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
4232 020156 104013 ERROR 13 ;NO RDY AFTER READ DATA CMD
4233 020160 004737 033664 JSR PC,GSTAT ;GET FRESH STATUS
4234 020164 032737 100000 007354 BIT #CERR,HCS1
4235 020172 001454 BEQ 72$
4236 020174 032737 000200 007370 BIT #BSE,HER ;SEE IF BAD SECTOR
4237 020202 001406 BEQ 70$
4238 020204 104065 ERROR 65 ;DETECTED BSE IN READ BUT NOT IN WRITE CMD.
4239 020206 000413 BR 73$
4240 020210 104401 045354 69$: TYPE MSG26 ;ABORTING BAL OF TESTS
4241 020214 000137 031452 JMP $EOP
4242
4243 020220 032737 100000 007370 70$: BIT #DCK,HER ;SEE IF DATA CHECK ERROR
4244 020226 001402 BEQ 71$
4245 020230 104021 ERROR 21 ;DATA CHECK ERROR AFTER READ CMD (ECC)
4246 020232 000401 BR 73$
4247
4248 020234 104014 71$: ERROR 14 ;CERR AFTER READ DATA CMD.
4249
4250 020236 73$:
4251
4252 020236 012737 010340 007444 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4253 020244 005037 007446 CLR E.B0 ;EXPECTED MSG B0
4254 020250 012737 001720 007450 MOV #<0.D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4255 020256 012737 000001 007452 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4256 020264 005037 007454 CLR E.A2 ;EXPECTED MSG A2
4257 020270 012737 000002 007456 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4258 020276 012737 000003 007462 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3

```

F07

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN 78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 83
T11 BASIC WRITE DATA TEST; FULL SECTOR

SEQ 0083

```

4259
4260 020304 004737 033042      JSR      PC,CHKMSG      ;CHECK MSGS AO, BO, A1, B1
4261 020310 000003              .WORD      T.A2!T.B2!0      ;& MSGS SPECIFIED HERE
4262 020312 104054              ERROR      54              ;MSG AO ERROR AFTER READ DATA CMD
4263 020314 104026              ERROR      26              ;MSG BO ERROR
4264 020316 104056              ERROR      56              ;MSG A1 ERROR
4265 020320 104030              ERROR      30              ;MSG B1 ERROR
4266 020322 000732              BR          69$
4267 020324
4268
4269 020324 012737 010340 007444  72$:  MOV      #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG AO
4270 020332 005037 007446          CLR      E.B0              ;EXPECTED MSG BO
4271 020336 012737 001720 007450  MOV      #<0.SPOK!D.CAR!D.DOOB!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
4272 020344 012737 000001 007452  MOV      #1,E.B1              ;MSG ID FOR EXPECTED MSG B1
4273 020352 005037 007454          CLR      E.A2              ;EXPECTED MSG A2
4274 020356 012737 000002 007456  MOV      #2,E.B2              ;MSG ID FOR EXPECTED MSG B2
4275 020364 012737 000003 007462  MOV      #3,E.B3              ;MSG ID FOR EXPECTED MSG B3
4276
4277 020372 004737 033042      JSR      PC,CHKMSG      ;CHECK MSGS AO, BO, A1, B1
4278 020376 000003              .WORD      T.A2!T.B2!0      ;& MSGS SPECIFIED HERE
4279 020400 104054              ERROR      54              ;MSG AO ERROR AFTER READ DATA CMD
4280 020402 104026              ERROR      26              ;MSG BO ERROR
4281 020404 104056              ERROR      56              ;MSG A1 ERROR
4282 020406 104030              ERROR      30              ;MSG B1 ERROR
4283
4284 020410 012701 006336          MOV      #RDTAB,R1
4285 020414 011137 001452  2$:  MOV      (R1),W01          ;ACTUAL WORD FOR TYPEOUT
4286 020420 010037 001454          MOV      R0,W02          ;EXPECTED DATA FOR TYPEOUT
4287
4288 020424 020021              CMP      R0,(R1)+          ;COMPARE READ DATA TABLE AGAINST
4289 020426 001401              BEQ      3$              ;THE 'SHOULD BE' VALUE
4290 020430 104020              ERROR      20              ;READ DATA DID NOT COMPARE WITH WRITE DATA
4291
4292 020432 020127 007336  3$:  CMP      R1,#RDTAB+512.      ;SEE IF REACHED END OF TABLE
4293 020436 001366              BNE      2$              ;BR IF NO & DO NEXT WORD
4294
4295 020440 020037 001474          CMP      R0,DATA0          ;SEE IF DID ALL 0'S
4296 020444 001401              BEQ      4$              ;BR IF YES
4297 020446 000412
4298
4299 020450 012765 001500 000004  4$:  MOV      #DATA1,RKBA(R5) ;WRITE ALL 1'S
4300 020456 013700 001500          MOV      DATA1,R0
4301 020462 013765 001402 000006  MOV      SECTOR,RKDA(R5)
4302 020470 000137 017604          JMP      1$
4303
4304 020474
4305 020474 104415  5$:  SCOP1
4306 020476 012706 001100          MOV      #STACK,SP      ;RESTORE STK PTR
4307
4308 020502 004737 034214      JSR      PC,SUBCLR
4309 020506 104024              ERROR      24              ;CERR AFTER SCLR
4310
4311 020510 052765 000020 000010  BIS      #BA1,RKCS2(R5) ;THIS PORTION OF THE TEST CHECKS
4312 020516 012765 001500 000004  MOV      #DATA1,RKBA(R5) ;OUT THE WRITE CHECK CMD
4313 020524 012765 177400 000002  MOV      #-256.,RKWC(R5) ;ALL 1'S WERE PREVIOUSLY WRITTEN
4314 020532 013765 001402 000006  MOV      SECTOR,RKDA(R5)

```

G07

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 84
T11 BASIC WRITE DATA TEST; FULL SECTOR

SEQ 0084

4315							
4316	020540	012737	000031	007354	MOV	#(WRTCHK),HCS1	
4317	020546	004737	032262		JSR	PC,DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
4318	020552	104015			ERROR	15	;NO RDY AFTER WRITE CHECK CMD
4319	020554	004737	033664		JSR	PC,GSTAT	;GET FRESH STATUS
4320	020560	032737	100000	007354	BIT	#CERR,HCS1	
4321	020566	001453			BEQ	75\$	
4322	020570	032737	040000	007356	BIT	#WCE,HCS2	;SEE IF WRITE CHECK ERROR
4323	020576	001410			BEQ	74\$	
4324	020600	016537	000024	001452	MOV	RKDB(R5),WD1	;ACTUAL WORD FOR PRINTOUT
4325	020606	013737	001500	001454	MOV	DATA1,WD2	;EXPECTED WORD FOR TYPEOUT
4326	020614	104016			ERROR	16	;WCE AFTER WRITE CMD
4327	020616	000437			BR	75\$	
4328							
4329	020620	104022			74\$: ERROR	22	;CERR AFTER WRITE CHECK CMD
4330							
4331	020622	012737	010340	007444	MOV	#(D.D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0
4332	020630	005037	007446		CLR	E.B0	;EXPECTED MSG B0
4333	020634	012737	001720	007450	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1
4334	020642	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4335	020650	005037	007454		CLR	E.A2	;EXPECTED MSG A2
4336	020654	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4337	020662	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4338							
4339	020670	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4340	020674	000003			.WORD	T.A2!T.B2!0	& MSGS SPECIFIED HERE
4341	020676	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
4342	020700	104031			ERROR	31	;MSG B0 ERROR
4343	020702	104060			ERROR	60	;MSG A1 ERROR
4344	020704	104032			ERROR	32	;MSG B1 ERROR
4345	020706	104401	045354		TYPE	MSG26	;ABORTING BAL OF TESTS
4346	020712	000137	031452		JMP	\$EOP	
4347							
4348	020716				75\$:		
4349							
4350	020716	012737	010340	007444	MOV	#(D.D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0
4351	020724	005037	007446		CLR	E.B0	;EXPECTED MSG B0
4352	020730	012737	001720	007450	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1
4353	020736	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4354	020744	005037	007454		CLR	E.A2	;EXPECTED MSG A2
4355	020750	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4356	020756	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4357							
4358	020764	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4359	020770	000003			.WORD	T.A2!T.B2!0	& MSGS SPECIFIED HERE
4360	020772	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
4361	020774	104031			ERROR	31	;MSG B0 ERROR
4362	020776	104060			ERROR	60	;MSG A1 ERROR
4363	021000	104032			ERROR	32	;MSG B1 ERROR
4364							
4365	021002	104415			SCOP1		
4366	021004	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
4367							
4368	021010	004737	034214		JSR	PC,SUBCLR	
4369	021014	104024			ERROR	24	;CERR AFTER SCLR
4370							

H07

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 85
T11 BASIC WRITE DATA TEST; FULL SECTOR

SEQ 0085

```

4371 021016 012765 001474 000004      MOV      #DATA0,RKBA(R5)  SETUP TO CHECK AGAINST WRONG DATA
4372 021024 012765 177400 000002      MOV      #-256.,RKWC(R5)
4373 021032 013765 001402 000006      MOV      SECTOR,RKDA(R5)
4374 021040 012737 000031 007354      MOV      #WATCH,HCS1
4375 021046 004737 032262              JSR      PC,DATCHD      ;DO WRITE CHECK CMD. & GET CONTR RDY.
4376 021052 104015              ERROR      15      ;NO RDY AFTER WRITE CHECK CMD
4377 021054 004737 033664              JSR      PC,GSTAT      ;GET FRESH STATUS
4378 021060 032737 040000 007356      BIT      #WCE,HCS2      ;EXPECT MISCOMPARE
4379 021066 001001              BNE      6$
4380 021070 104017              ERROR      17      ;WRITE CHECK CMD NOT FUNCTIONING
4381                                ;WITH INTENTIONAL MISCOMPARE
4382 021072              6$:
4383
4384 021072 012737 010340 007444      MOV      #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4385 021100 005037 007446              CLR      E.B0      ;EXPECTED MSG B0
4386 021104 012737 001720 007450      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
4387 021112 012737 000001 007452      MOV      #1,E.B1      ;MSG ID FOR EXPECTED MSG B1
4388 021120 005037 007454              CLR      E.A2      ;EXPECTED MSG A2
4389 021124 012737 000002 007456      MOV      #2,E.B2      ;MSG ID FOR EXPECTED MSG B2
4390 021132 012737 000003 007462      MOV      #3,E.B3      ;MSG ID FOR EXPECTED MSG B3
4391
4392 021140 004737 033042              JSR      PC,CHKMSG      ;CHECK MSGS A0, B0, A1, B1
4393 021144 000000              .WORD      0!0!0      ;& MSGS SPECIFIED HERE
4394 021146 104057              ERROR      57      ;MSG A0 ERROR AFT WRT CHK CMD
4395 021150 104031              ERROR      31      ;MSG B0 ERROR
4396 021152 104060              ERROR      60      ;MSG A1 ERROR
4397 021154 104032              ERROR      32      ;MSG B1 ERROR
4398
4399              7$:
4400
4401      ;*****
4402      ;TEST 12      20 SECTOR FORMAT TEST
4403      ;
4404      ;ALL 1'S ARE WRITTEN ON A FULL SECTOR IN 20 SECTOR FORMAT.
4405      ;MSG B0,B1 ARE CHECKED FOR ANY ERROR CONDITION.
4406      ;CYL 0, TRACK 0, & SECTOR 0 IS USED.
4407      ;*****
4408      ;ST12: SCOPE
4409      ;MOV      #1,STIMES      ;DO 1 ITERATION
4410      ;MOV      #STACK,SP      ;RESTORE STK PTR
4411
4412      ;JSR      PC,SUBCLR      ;CERR AFTER SCLR
4413      ;ERROR      24
4414
4415      ;MOV      #HDTAB,RKBA(R5) ;HEADER WORD TABLE
4416      ;MOV      #-60.,RKWC(R5) ;WORD COUNT FOR 20 SECTOR FMT
4417      ;CLR      TOCYL
4418      ;INC      BADHDR      ;USED FOR VALID HALT
4419
4420
4421      ;MOV      TOCYL,CALADD      ;SETUP
4422      ;MOV      #0,HEAD      ;TO FILL
4423      ;MOV      #1,FORMAT      ;HEADER
4424      ;JSR      PC,FHDTAB      ;TABLE
4425
4426

```

CZR6IEO UNIBUS RK6 DR PRT2 MACY11 30A(1052) 10-JAN-78 11:25 PAGE 86
 CZR6IE.P11 10-JAN-78 09:43 T12 20 SECTOR FORMAT TEST

SEQ 0086

4427	021252	012737	010027	007354	MOV	#(CFMT!WRHEAD),HCS1	
4428	021260	004737	032262		JSR	PC,DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
4429	021264	104200			ERROR	200	;NO RDY AFTER WRITE HEADER CMD
4430	021266	012737	010001	007354	MOV	#(CFMT!SELDRV),HCS1	
4431	021274	004737	032224		JSR	PC,DOCMD	
4432	021300	104117			ERROR	117	;NO RDY AFTER SELDRV CMD
4433	021302	032737	100000	007354	BIT	#CERR,HCS1	
4434	021310	001405			BEQ	64\$	
4435	021312	104201			ERROR	201	;CERR AFTER WRITE HEADER CMD
4436	021314	104401	045354		TYPE	MSG26	;ABORTING BAL OF TESTS
4437	021320	000137	031452		JMP	\$EOP	
4438	021324						
4439					64\$:		
4440	021324	104415			SCOP1		
4441	021326	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
4442							
4443	021332	004737	034214		JSR	PC,SUBCLR	
4444	021336	104024			ERROR	24	;CERR AFTER SCLR
4445							
4446	021340	005037	001402		CLR	SECTOR	
4447	021344	013765	001402	000006	MOV	SECTOR,RKDA(R5)	
4448	021352	012765	001500	000004	MOV	#DATA1,RKBA(R5)	;WRITE ALL 1'S
4449	021360	052765	000020	000010	BIS	#BA1,RKCS2(R5)	;BUSS ADDR INCR INHIBIT
4450	021366	012765	177400	000002	MOV	#-256.,RKWC(R5)	;DO FULL SECTOR
4451							
4452							
4453	021374	012737	010023	007354	MOV	#(CFMT!WRDATA),HCS1	
4454	021402	004737	032262		JSR	PC,DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
4455	021406	104011			ERROR	11	;NO RDY AFTER WRITE DATA CMD
4456	021410	012737	010001	007354	MOV	#(CFMT!SELDRV),HCS1	
4457	021416	004737	032224		JSR	PC,DOCMD	
4458	021422	104117			ERROR	117	;NO RDY AFTER SELDRV CMD
4459	021424	032737	100000	007354	BIT	#CERR,HCS1	
4460	021432	001465			BEQ	68\$	;BR IF NO ERRORS
4461							
4462	021434	032737	000200	007370	BIT	#BSE,HER	;SEE IF BAD SECTOR FLAG
4463	021442	001421			BEQ	66\$	;BR IF NO
4464	021444	004737	035700		JSR	PC,TRUEERR	;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4465	021450	000455			BR	67\$	;RETURN HERE IF NO
4466							
4467	021452	005237	001402		INC	SECTOR	;RETURN HERE IF YES
4468	021456	003727	001402	000012	CMP	SECTOR,#10.	;ARE 10 CONSEC. SECTORS BAD
4469	021464	001003			BNE	65\$	;BR IF NO
4470	021466	104046			ERROR	46	;ABORTING TEST DETECTED 10 BAD SECTORS
4471	021470	000137	022374		JMP	3\$	;BYPASS TEST
4472	021474	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	;TRY ANOTHER SECTOR
4473	021502	000137	021344		JMP	4\$	
4474	021506	104012			ERROR	12	;CERR WITH WRITE DATA CMD
4475					66\$:		
4476	021510	012737	010340	007444	MOV	#(O!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0
4477	021516	005037	007446		CLR	E.B0	;EXPECTED MSG B0
4478	021522	012737	001720	007450	MOV	#(O.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1
4479	021530	012737	000001	007452	MOV	#1,E.B1	;MSG 1D FOR EXPECTED MSG B1
4480	021536	005037	007454		CLR	E.A2	;EXPECTED MSG A2
4481	021542	012737	000002	007456	MOV	#2,E.B2	;MSG 2D FOR EXPECTED MSG B2
4482	021550	012737	000003	007462	MOV	#3,E.B3	;MSG 3D FOR EXPECTED MSG B3

```
4483
4484 021556 004737 033042 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4485 021562 000003 .WORD T.A2!T.P2!0 ;& MSGS SPECIFIED HERE
4486 021564 104052 ERROR 52 ;MSG A0 ERROR AFTER WRITE DATA CMD
4487 021566 104023 ERROR 23 ;MSG B0 ERROR
4488 021570 104053 ERROR 53 ;MSG A1 ERROR
4489 021572 104025 ERROR 25 ;MSG B1 ERROR
4490 021574 104401 045354 TYPE MSG26 ;ABORTING BAL OF TESTS
4491 021600 000137 031452 JMP $EOP
4492 021674 104063 67$: ERROR 63 ;BAD SECTOR NOT LISTED IN TABLE
4493 021606 68$:
4494 021606 012737 010001 007354 MOV #<CFMT!SELDRV>,HCS1
4495 021614 004737 032224 JSR PC,DOCMD
4496 021620 104117 ERROR 117 ;RDY NOT FOUND AFTER SELDRV CMD
4497 021622 032737 001000 007402 BIT #D.FORM,HMR2
4498 021630 001001 BNE 15
4499 021632 104102 ERROR 102 ;FORMAT NOT SET
4500
4501 021634 15:
4502
4503 021634 012737 010340 007444 MOV #<O!D.SPIN!D.DRDY!D.VV!D.DRA>.E.A0 ;EXPECTED MSG A0
4504 021642 005037 007446 CLR E.B0 ;EXPECTED MSG B0
4505 021646 012737 001720 007450 MOV #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>.E.A1 ;EXPECTED A1
4506 021654 012737 000001 007452 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4507 021662 005037 007454 CLR E.A2 ;EXPECTED MSG A2
4508 021676 012737 000002 007456 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4509 021674 012737 000003 007462 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4510
4511 021702 004737 033042 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
4512 021706 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
4513 021710 104052 ERROR 52 ;MSG A0 ERROR AFTER WRITE DATA CMD
4514 021712 104023 ERROR 23 ;MSG B0 ERROR
4515 021714 104053 ERROR 53 ;MSG A1 ERROR
4516 021716 104025 ERROR 25 ;MSG B1 ERROR
4517 021720 012765 177400 000002 MOV #-256.,RKWC(R5)
4518 021726 012765 001500 000004 MOV #DATA1,RKBA(R5)
4519 021734 052765 000020 000010 BIS #BA1,RKCS2(R5)
4520 021742 013765 001402 000006 MOV SECTOR,RKDA(R5)
4521
4522 021750 012737 010031 007354 MOV #<CFMT!WRTCHK>,HCS1
4523 021756 004737 032262 JSR PC,DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
4524 021762 104015 ERROR 15 ;NO RDY AFTER WRITE CHECK CMD
4525 021764 012737 010001 007354 MOV #<CFMT!SELDRV>,HCS1
4526 021772 004737 032224 JSR PC,DOCMD
4527 021776 104117 ERROR 117 ;NO RDY AFTER SELDRV CMD
4528 022000 032737 100000 007354 BIT #CERR,HCS1
4529 022006 001453 BEQ 70$
4530 022010 032737 040000 007356 BIT #WCE,HCS2 ;SEE IF WRITE CHECK ERROR
4531 022016 001410 BEQ 69$
4532 022020 016537 000024 001454 MOV RKDB(R5),WD1 ;ACTUAL WORD FOR PRINTOUT
4533 022026 013737 001500 001454 MOV DATA1,WD2 ;EXPECTED WORD FOR TYPEOUT
4534 022034 104016 ERROR 16 ;WCE AFTER WRITE CMD
4535 022036 000437 BR 70$
4536
4537 022040 104022 69$: ERROR 22 ;CERR AFTER WRITE CHECK CMD
4538
```

K07

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 88
T12 20 SECTOR FORMAT TEST

SEQ 0088

4539	022042	012737	010340	007444	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4540	022050	005037	007446		CLR	E.B0	;EXPECTED MSG B0
4541	022054	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4542	022062	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4543	022070	005037	007454		CLR	E.A2	;EXPECTED MSG A2
4544	022074	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4545	022102	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4546							
4547	022110	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4548	022114	000003			.WORD	T.A2!T.B2!0	;8 MSGS SPECIFIED HERE
4549	022116	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
4550	022120	104031			ERROR	31	;MSG B0 ERROR
4551	022122	104060			ERROR	60	;MSG A1 ERROR
4552	022124	104032			ERROR	32	;MSG B1 ERROR
4553	022126	104401	045354		TYPE	MSG26	;ABORTING BAL OF TESTS
4554	022132	000137	031452		JMP	\$EOP	
4555							
4556	022136						
4557	022136	012737	010001	007354	MOV	#<CFMT!SELDRV>,HCS1	
4558	022144	004737	032224		JSR	PC,DOCMD	
4559	022150	104117			ERROR	117	;NO RDY AFTER SELDRV CMD
4560	022152	032737	001000	007402	BIT	#D.FORM,HMR2	
4561	022160	001001			BNE	2\$	
4562	022162	104103			ERROR	103	;FORMAT NOT SET
4563							
4564	022164						
4565							
4566	022164	012737	010340	007444	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4567	022172	005037	007446		CLR	E.B0	;EXPECTED MSG B0
4568	022176	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4569	022204	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4570	022212	005037	007454		CLR	E.A2	;EXPECTED MSG A2
4571	022216	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4572	022224	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4573							
4574	022232	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4575	022236	000000			.WORD	0!0!0	;8 MSGS SPECIFIED HERE
4576	022240	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
4577	022242	104031			ERROR	31	;MSG B0 ERROR
4578	022244	104060			ERROR	60	;MSG A1 ERROR
4579	022246	104032			ERROR	32	;MSG B1 ERROR
4580	022250	104415			SCOP1		
4581	022252	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
4582							
4583	022256	004737	034214		JSR	PC,SUBCLR	
4584	022262	104024			ERROR	24	;CERR AFTER SCLR
4585							
4586	022264	012765	001522	000004	MOV	#HDTAB,RKBA(R5)	;RESTORE CYL 0 TO 22 SECTOR FMT
4587	022272	012765	177676	000002	MOV	#-66.,RKWC(R5)	
4588	022300	005037	001346		CLR	TOCYL	
4589							
4590							
4591	022304	013737	001346	001362	MOV	TOCYL,CALADD	;SETUP
4592	022312	012737	000000	001460	MOV	#0,HEAD	;TO FILL
4593	022320	012737	000000	001466	MOV	#0,FORMAT	;HEADER
4594	022326	004737	035212		JSR	PC,FHDTAB	;TABLE

```

4595
4596
4597 022332 012737 000027 007354      MOV      #<WRHEAD>,HCS1
4598 022340 004737 032262                JSR      PC,DATCMD      ;DO DATA X FOR CMD & GET CONTR RDY
4599 022344 104200                ERROR    200                ;NO RDY AFTER WRITE HEADER CMD
4600 022346 004737 033664                JSR      PC,GSTAT      ;GET FRESH STATUS
4601 022352 032737 100000 007354      BIT      #CERR,HCS1
4602 022360 001405                BEQ      71$
4603 022362 104201                ERROR    201                ;CERR AFTER WRITE HEADER CMD
4604 022364 104401 045354                TYPE     MSG26                ;ABORTING BAL OF TESTS
4605 022370 000137 031452                JMP      $EOP
4606 022374
4607
4608 022374 005037 007340      71$:      3$:      CLR      BADHDR      ;USED FOR VALID HALT
4609
4610      ;*****
4611      ;TEST 13      TEST OFFSET & RTC LOGIC
4612      ;
4613      ;      THE HEADS ARE FIRST OFFSET BY OFFSET COMMANDS.
4614      ;      THIS TEST CHECKS THE RTC LOGIC BY VERIFYING THAT THE
4615      ;      'OFFSET ON' BIT (MSG A,00) RESETS AND THE OFFSET REG
4616      ;      BECOMES THE CYL DIFF INFO WHEN A SEEK CMD TO A
4617      ;      DIFFERENT CYLINDER IS ISSUED
4618      ;      IT ALSO TESTS THAT DRIVE CLEAR & SEEK TO SELF WILL NOT
4619      ;      CLEAR THE 'OFFSET ON' BIT OR THE OFFSET REG.
4620      ;      ALL OFFSET POSITIONS IN BOTH DIRECTIONS ARE CHECKED
4621      ;
4622      ;*****
4623 022400 000004                ST13:    SCOPE
4624 022402 012737 000001 001174      MOV      #1,$TIMES      ;DO 1 ITERATION
4625 022410 012706 001100      MOV      #STACK,SP      ;RESTORE STK PTR
4626
4627 022414 012702 000001      MOV      #1,R2      ;MIN POS OFFSET
4628
4629 022420 004737 034214      1$:      JSR      PC,SUBCLR
4630 022424 104024                ERROR    24      ;CERR AFTER SCLR
4631
4632 022426 010265 000016      MOV      R2,RK750F(R5) ;SET OFFSET
4633
4634 022432 012737 022520 001176      MOV      #6,$$ESCAPE
4635 022440 012737 000015 007354      MOV      #OFFSET,HCS1
4636 022446 004737 032224      JSR      PC,DOCMD      ;DO RECAL CMD & GET CONTR RDY
4637 022452 104033                ERROR    33      ;NO RDY AFTER OFFSET CMD
4638
4639 022454 012737 032140 007444      MOV      #<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4640 022462 005037 007446      CLR      E.B0
4641 022466 012737 001720 007450      MOV      #<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1
4642 022474 012737 000001 007452      MOV      #1,E.B1
4643
4644 022502 004737 033042      JSR      PC,CHKMSG      ;CHECK MSGS A0, B0, A1, B1
4645 022506 000000                .WORD    0!0!0      ;8 MSGS SPECIFIED HERE
4646 022510 104035                ERROR    35      ;MSG A0 ERROR DURING OFFSET CMD
4647 022512 104061                ERROR    61      ;MSG B0 ERROR
4648 022514 104036                ERROR    36      ;MSG A1 ERROR
4649 022516 104062                ERROR    62      ;MSG B1 ERROR
4650

```

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 90
T13 TEST OFFSET & RTC LOGIC

SEQ 0090

4651	022520	005037	001176		64\$:	CLR	\$ESCAPE	
4652	022524	013737	001416	007412		MOV	15000,TEMP1	;SETUP TIMEOUT
4653	022532	004737	032730			JSR	PC,FATT2	;FIND ATTN
4654	022536	104034				ERROR	34	;NO ATTN AFTER OFFSET CMD
4655								
4656								
4657	022540	012737	052340	007444		MOV	#<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
4658	022546	005037	007446			CLR	E.B0	;EXPECTED MSG B0
4659	022552	012737	001720	007450		MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
4660	022560	012737	000001	007452		MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
4661	022566	005037	007454			CLR	E.A2	;EXPECTED MSG A2
4662	022572	012737	000002	007456		MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4663	022600	012737	000003	007462		MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4664								
4665	022606	004737	033042			JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4666	022612	000003				.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4667	022614	104260				ERROR	260	;MSG A0 ERROR AFTER OFFSET CMD
4668	022616	104261				ERROR	261	;MSG B0 ERROR
4669	022620	104037				ERROR	37	;MSG A1 ERROR
4670	022622	104040				ERROR	40	;MSG B1 ERROR
4671								
4672	022624	005737	001360			TST	CYLADD	
4673	022630	001401				BEQ	17\$	
4674	022632	104042				ERROR	42	;CYL ADDR IN B2 WAS NOT 0
4675								;AFTER OFFSET CMD FROM CYL 0
4676	022634	042737	001000	001356	17\$:	BIC	#1000,CYLDIF	;GET RID OF HIGH BIT
4677	022642	010265	000016			MOV	R2,RKASOF(R5)	;REFRESH RKASOF
4678								
4679	022646	032702	000200			BIT	#BIT7,R2	
4680	022652	001005				BNE	65\$	;BR IF NEG OFFSET
4681								
4682	022654	020237	001356			CMP	R2,CYLDIF	;CHECK POS OFFSET
4683	022660	001406				BEQ	66\$	
4684	022662	104114				ERROR	114	;OFFSET IN A2 NOT = RKASOF
4685	022664	000404				BR	66\$	;AFTER OFFSET CMD
4686								
4687	022666	020137	001356		65\$:	CMP	R1,CYLDIF	;CHECK NEG OFFSET
4688	022672	001401				BEQ	66\$	
4689	022674	104114				ERROR	114	;OFFSET IN A2 NOT = RKASOF
4690								;AFTER OFFSET CMD
4691	022676				66\$:			
4692								
4693	022676	012765	100000	000000		MOV	#CLR,RKCS1(R5)	
4694	022704	013765	001222	000010		MOV	\$UNIT,RKCS2(R5)	;DRIVE#
4695	022712	012737	000005	007354		MOV	#CLEAR,HCS1	
4696	022720	004737	032224			JSR	PC,DOCMD	;DO DRIVE CLEAR CMD & GET CONTR RDY
4697	022724	104151				ERROR	151	;NO RDY AFTER DRIVE CLEAR CMD
4698	022726	004737	032602			JSR	PC,TSTATN	;TEST FOR ATTN
4699	022732	000401				BR	67\$	
4700	022734	104154				ERROR	154	;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
4701	022736				67\$:			
4702								
4703	022736	012765	100000	000000		MOV	#CLR,RKCS1(R5)	
4704	022744	004737	033664			JSR	PC,GSTAT	
4705	022750	032737	002000	007402		BIT	#D.OFF,HMR2	
4706	022756	001001				BNE	4\$	

NO7

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 91
T13 TEST OFFSET & RTC LOGIC

SEQ 0091

C
C

4707	022760	104J43			ERROR	43		; OFFSET BIT IN RKMR2 CLEARED
4708								; AFTER DRIVE CLEAR CMD & SELECT DRV CMD
4709	022762	012737	012340	007444	4\$: MOV	#<D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		; EXPECTED A0
4710	022770	005037	007446		CLR	E.B0		
4711	022774	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		
4712	023002	012737	000001	007452	MOV	#1,E.B1		
4713								
4714	023010	004737	033042		JSR	PC,CHKMSG		; CHECK MSGS A0, B0, A1, B1
4715	023014	000003			.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE
4716	023016	104273			ERROR	273		; MSG A0 ERROR AFTER DRIVE CLEAR CMD
4717	023020	104265			ERROR	265		; MSG B0 ERROR
4718	023022	104274			ERROR	274		; MSG A1 ERROR
4719	023024	104266			ERROR	266		; MSG B1 ERROR
4720	023026	042737	001000	001356	BIC	#1000,CYLDIF		; GET RID OF HIGH BIT
4721	023034	010265	000016		MOV	R2,RKASOF(R5)		; REFRESH RKASOF
4722								
4723	023040	032702	000200		BIT	#BIT7,R2		
4724	023044	001005			BNE	68\$		; BR IF NEG OFFSET
4725								
4726	023046	020237	001356		CMP	R2,CYLDIF		; CHECK POS OFFSET
4727	023052	001406			BEQ	69\$		
4728	023054	104115			ERROR	115		; OFFSET IN A2 NOT = RKASOF
4729	023056	000404			BR	69\$		; AFTER DRIVE CLEAR CMD
4730								
4731	023060	020137	001356		68\$: CMP	R1,CYLDIF		; CHECK NEG OFFSET
4732	023064	001401			BEQ	69\$		
4733	023066	104115			ERROR	115		; OFFSET IN A2 NOT = RKASOF
4734								; AFTER DRIVE CLEAR CMD
4735	023070				69\$:			
4736								
4737	023070	012737	000017	007354	MOV	#SEEK,HCS1		
4738	023076	004737	032224		JSR	PC,NOCMD		; DO SEEK CMD & GET CONTR READY
4739	023102	104131			ERROR	131		; NO RDY AFTER SEEK CMD
4740								
4741	023104	013737	001420	007412	MOV	T50000,TEMP1		; SETUP TIMEOUT
4742	023112	004737	032730		JSR	PC,FAT12		; FIND ATTN
4743	023116	104132			ERROR	132		; NO ATTN AFTER SEEK CMD
4744								
4745	023120	032737	100000	007354	BIT	#CERR,HCS1		
4746	023126	001401			BEQ	70\$		
4747	023130	104210			ERROR	210		; CERR AFTER SEEK CMD
4748								
4749	023132				70\$:			
4750								
4751								
4752	023132	032737	002000	007402	BIT	#D.OFF,HMR2		
4753	023140	001001			BNE	7\$		
4754	023142	104045			ERROR	45		; OFFSET BIT CLEARED IN RKMR2 AFTER SEEK TO SELF.
4755								
4756	023144				7\$:			
4757								
4758	023144	012737	052340	007444	MOV	#<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		; EXPECTED MSG A0
4759	023152	005037	007446		CLR	E.B0		; EXPECTED MSG B0
4760	023156	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		; EXPECTED A1
4761	023164	012737	000001	007452	MOV	#1,E.B1		; MSG ID FOR EXPECTED MSG B1
4762	023172	005037	007454		CLR	E.A2		; EXPECTED MSG A2

4763	023176	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
4764	023204	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
4765							
4766	023212	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
4767	023216	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
4768	023220	104140			ERROR	140	;MSG A0 ERROR AFTER SEEK TO SELF
4769	023222	104141			ERROR	141	;MSG B0 ERROR
4770	023224	104142			ERROR	142	;MSG A1 ERROR
4771	023226	104143			ERROR	143	;MSG B1 ERROR
4772	023230	042737	001000	001356	BIC	#1000,CYLDIF	;GET RID OF HIGH BIT
4773	023236	010265	000016		MOV	R2,RKASOF(R5)	;REFRESH RKASOF
4774							
4775	023242	032702	000200		BIT	#BIT7,R2	
4776	023246	001005			BNE	71\$	;BR IF NEG OFFSET
4777							
4778	023250	020237	001356		CMP	R2,CYLDIF	;CHECK POS OFFSET
4779	023254	001406			BEQ	72\$	
4780	023256	104123			ERROR	123	;OFFSET IN A2 NOT = RKASOF
4781	023260	000404			BR	72\$	;AFTER SEEK TO SELF
4782							
4783	023262	020137	001356		71\$: CMP	R1,CYLDIF	;CHECK NEG OFFSET
4784	023266	001401			BEQ	72\$	
4785	023270	104123			ERROR	123	;OFFSET IN A2 NOT = RKASOF
4786							;AFTER SEEK TO SELF
4787	023272				72\$:		
4788							
4789	023272	004737	034214		JSR	PC,SUBCLR	
4790	023276	104024			ERROR	24	;CERR AFTER SCLR
4791							
4792	023300	012737	000012	001346	MOV	#10.,TOCYL	;SETUP CYL 10
4793	023306	012765	000012	000020	MOV	#10.,RKDC(R5)	;DO ACTUAL IMPLIED SEEK TO CYL 10 TO VERIFY
4794							;OFFSET BIT IN RKMR2 CLEARED
4795							
4796							
4797							
4798	023314	012700	001726		MOV	#RHTAB,R0	
4799	023320	012737	000025	007354	MOV	#<RDHEAD>,HCS1	
4800	023326	004737	032262		JSR	PC,DATCMD	;DO DATA X FOR CIL & GET CONTR RDY
4801	023332	104171			ERROR	171	;NO RDY AFTER READ HEADER CMD
4802	023334	032737	100000	007354	BIT	#CERR,HCS1	
4803	023342	001405			BEQ	74\$	
4804	023344	104174			ERROR	174	;CERR AFTER READ HEADER CMD
4805	023346	104401	045354		TYPE	MSG26	;ABORTING BAL OF TESTS
4806	023352	000137	031452		JMP	\$EOP	
4807							
4808	023356	016520	000024		74\$: MOV	RKDB(R5),(R0)+	;1'ST WORD FROM SILO TO RHTAB
4809	023362	016520	000024		MOV	RKDB(R5),(R0)+	;2'ND WORD
4810	023366	016520	000024		MOV	RKDB(R5),(R0)+	;3'RD WORD
4811							
4812							
4813	023372	032765	100000	000010	BIT	#DLT,RKCS2(R5)	
4814	023400	001407			BEQ	75\$	
4815	023402	004737	033664		JSR	PC,GSTAT	
4816	023406	104173			ERROR	173	;DLT AFTER READ HEADER CMD
4817	023410	104401	045354		TYPE	MSG26	;ABORTING BAL OF TESTS
4818	023414	000137	031452		JMP	\$EOP	

```

4819 023420          75$:
4820
4821 023420 012737 010340 007444      MOV      #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
4822 023426 005037 007446              CLR      E.B0 ;EXPECTED MSG B0
4823 023432 012737 001720 007450      MOV      #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
4824 023440 012737 000001 007452      MOV      #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
4825 023446 005037 007454              CLR      E.A2 ;EXPECTED MSG A2
4826 023452 012737 000002 007456      MOV      #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
4827 023460 012737 000003 007462      MOV      #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
4828
4829 023466 004737 033042              JSR      PC,CHKMSG ;CHECK MSGS A0,B0,A1,B1
4830 023472 000003              .WORD      T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
4831 023474 104301              ERROR      301 ;MSG A0 ERROR AFTER READ HEADER CMD
4832 023476 104271              ERROR      271 ;MSG B0 ERROR
4833 023478 104302              ERROR      302 ;MSG A1 ERROR
4834 023480 104272              ERROR      272 ;MSG B1 ERROR
4835
4836 023504 023737 001726 001346      CMP      RHTAB,TOCYL ;CHECK WORD 0 ONLY, CYL#
4837 023512 001401              BEQ       73$ ;BR IF SAME
4838 023514 104051              ERROR      51 ;WRONG CYL# ON HEADER
4839
4840          73$:
4841
4842 023516 032737 002000 007402      BIT      #0.OFF,HMR2
4843 023524 001401              BEQ       9$
4844 023526 104101              ERROR      101 ;OFFSET NOT CLEARED AFTER READ HEADER WITH MOVEMENT
4845
4846 023530 023727 001360 000012 9$:      CMP      CYLADD,#10.
4847 023536 001401              BEQ       10$
4848 023540 104122              ERROR      122 ;DID NOT GO TO CYL 10
4849
4850 023542 005737 001356          10$:      TST      CYLDIF
4851 023546 001401              BEQ       16$
4852 023550 104101              ERROR      101 ;OFFSET NOT CLEARED IN RKMR2
4853
4854 023552 004737 034214          16$:      JSR      PC,SUBCLR
4855 023556 104024              ERROR      24 ;CERR AFTER SCLR
4856
4857 023560 012737 000017 007354      MOV      #SEEK,HCS1
4858 023566 004737 032224              JSR      PC,DOCMD ;DO SEEK CMD & GET CONTR READY
4859 023572 104131              ERROR      131 ;NO RDY AFTER SEEK CMD
4860
4861 023574 013737 001420 007412      MOV      T50000,TEMP1 ;SETUP TIMEOUT
4862 023602 004737 032730              JSR      PC,FATT2 ;FIND ATTN
4863 023606 104132              ERROR      132 ;NO ATTN AFTER SEEK CMD
4864
4865 023610 032737 100000 007354      BIT      #CERR,HCS1
4866 023616 001401              BEQ       76$
4867 023620 104210              ERROR      210 ;CERR AFTER SEEK CMD
4868
4869 023622          76$:
4870
4871
4872 023622 032702 000200          BIT      #BIT7,R2 ;SEE IF DOING NEG OFFSETS
4873 023626 001014              BNE       18$ ;BR IF YES
4874

```

D08

CZR61EO UNIBUS RK6 DR PR2
CZR61E.P11 10-JAN- 8 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 94
T13 TEST OFFSET & RTC LOGIC

SEQ 0094

4875 023630 005202
4876 023632 020227 000061
4877 023636 001402
4878 023640 000137 022420
4879
4880 023644 012702 000201
4881 023650 012701 000101
4882 023650 000137 022420
4883
4884 023660 005201
4885 023662 005202
4886 023664 020227 000261
4887 023670 001402
4888 023672 000137 022420
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911 023676 000004
4912 023700 012737 000001 001174
4913 023706 012706 001100
4914
4915 023712 004737 034214
4916 023716 104024
4917
4918 023720 005037 001402
4919 023724 013765 001402 000006
4920 023732 012765 001500 000004
4921 023740 052765 000020 000010
4922 023746 012765 177400 000002
4923
4924
4925
4926 023754 012737 000023 007354
4927 023762 004737 032262
4928 023766 104011
4929 023770 004737 033664
4930 023774 032737 100000 007354

INC R2
CMP R2, #61 ;SEE IF JUST DID MAX POS OFFSET
BEQ 205 ;BR IF YES
JMP 15 ;ELSE DO NEXT POS OFFSET
205: MOV #201, R2 ;SETUP NEG OFFSET FOR RKASOF
MOV #101, R1 ;SETUP NEG OFFSET OFOR MSG A
JMP 15 ;DO NEG OFFSET
185: INC R1
INC R2
CMP R2, #261 ;SEE IF ALL NEG OFFSETS DONE
BEQ TST14 ;GO TO NEXT TST
JMP 15 ;DO ANOTHER

TEST 14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

THIS TEST VERIFIES THAT THE HEAD OFFSET LOGIC IS OPERATIONAL BY
WRITING ALL 1'S PATTERNS ON CYLINDER 0, HEAD 0. THEN
PERFORMING READ DATA FROM CENTERLINE AND MOVING OUT + AND - OFFSET
POSITIONS UNTIL A FAILURE OCCURES. THE FAILING OFFSET POSITIONS
ARE TYPED OUT.
OFFSET CODES ARE ALSO VERIFIED BY READING MSG A, STATUS 00 & 10.
ALL HEADS ARE TESTED AT CYLINDER 0
IF THERE ARE NO FAILURES AT ALL, THIS INDICATES THAT
OR A. HEADS DID NOT MOVE AT ALL
B. THE COMBINATION OF DISC SURFACE, HEADS, R/W AMP
ARE EXCEPTIONALL' GOOD.

NOTE THAT THE OFFSET FAILURE IS NOT AN ERROR,
BUT AN INDICATION OF SURFACE, HEAD & R/W ELECTRONICS QUALITY ONLY.

TST14: SCOPE
MOV #1, \$TIMES ;DO 1 ITERATION
MOV #STACK, SP ;RESTORE STK PTR
JSR PC, SUBCLR
ERROR 24 ;CERR AFTER SCLR

115: CLR SECTOR
MOV SECTOR, RKDA(R5)
MOV #DATA1, RKBA(R5) ;WRITE ALL 1'S
BIS #BA1, RKCS2(R5) ;BUS ADDR INCR INHIB
MOV #-256, RKWC(R5) ;SECTOR 0 ONLY
;WILL DO IMPLIED SEEK TO CYL 0
;WAS ON CYL 1 FROM LAST TEST

MOV #<WRDATA>, HCS1
JSR PC, DATCMD ;DO DATA X FOR CMD & GET CONTR RDY
ERROR 11 ;NO RDY AFTER WRITE DATA CMD
JSR PC, STAT ;GET FRESH STATUS
BIT #CERR, HCS1

E08

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 95
T14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

SEQ 0095

4931	024002	001465			BEQ	67\$		;BR IF NO ERRORS
4932								
4933	024004	032737	000200	007370	BIT	#BSE,HEP		;SEE IF BAD SECTOR FLAG
4934	024012	001421			BEQ	65\$		;BR IF NO
4935	024014	004737	035700		JSR	PC,TRUERR		;ELSE SEE IF SECTOR LISTED IN BSE TABLE
4936	024020	000455			BR	66\$		;RETURN HERE IF NO
4937								
4938	024022	005237	001402		INC	SECTOR		;RETURN HERE IF YES
4939	024026	023727	001402	000012	CMP	SECTOR,#10.		;ARE 10 CONSEC. SECTORS BAD
4940	024034	001003			BNE	64\$		;BR IF NO
4941	024036	104046			ERROR	46		;ABORTING TEST DETECTED 10 BAD SECTORS
4942	024040	000137	025436		JMP	10\$		;BYPASS TEST
4943	024044	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	64\$:	;TRY ANOTHER SECTOR
4944	024052	000137	023724		JMP	11\$		
4945	024056	104012			ERROR	12	65\$:	;CERR WITH WRITE DATA CMD
4946								
4947	024060	012737	010340	007444	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
4948	024066	005037	007446		CLR	E.B0		;EXPECTED MSG B0
4949	024072	012737	001720	007450	MOV	#<0.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		;EXPECTED A1
4950	024100	012737	000001	007452	MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
4951	024106	005037	007454		CLR	E.A2		;EXPECTED MSG A2
4952	024112	012737	000002	007456	MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
4953	024120	012737	000003	007462	MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
4954								
4955	024126	004737	033042		JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1
4956	024132	000003			.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE
4957	024134	104052			ERROR	52		;MSG A0 ERROR AFTER WRITE DATA CMD
4958	024136	104023			ERROR	23		;MSG B0 ERROR
4959	024140	104053			ERROR	53		;MSG A1 ERROR
4960	024142	104025			ERROR	25		;MSG B1 ERROR
4961	024144	104401	045354		TYPE	MSG26		;ABORTING BAL OF TESTS
4962	024150	000137	031452		JMP	\$EOP		
4963	024154	104063			ERROR	63	66\$:	;BAD SECTOR NOT LISTED IN TABLE
4964	024156						67\$:	
4965								
4966	024156	012737	010340	007444	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
4967	024164	005037	007446		CLR	E.B0		;EXPECTED MSG B0
4968	024170	012737	001720	007450	MOV	#<0.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		;EXPECTED A1
4969	024176	012737	000001	007452	MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
4970	024204	005037	007454		CLR	E.A2		;EXPECTED MSG A2
4971	024210	012737	000002	007456	MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
4972	024216	012737	000003	007462	MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
4973								
4974	024224	004737	033042		JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1
4975	024230	000003			.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE
4976	024232	104052			ERROR	52		;MSG A0 ERROR AFTER WRITE DATA CMD
4977	024234	104023			ERROR	23		;MSG B0 ERROR
4978	024236	104053			ERROR	53		;MSG A1 ERROR
4979	024240	104025			ERROR	25		;MSG B1 ERROR
4980	024242	012765	001500	000004	MOV	#DATA1,RKBA(R5)		
4981	024250	052765	000020	000010	BIS	#BA1,RKCS2(R5)		
4982	024256	012765	177400	000002	MOV	#-256.,RKWC(R5)		
4983	024264	013765	001402	000006	MOV	SECTOR,RKDA(R5)		
4984								
4985	024272	012737	000031	007354	MOV	#<WRTCHK>,HCS1		
4986	024300	004737	032262		JSR	PC,DATCMD		;DO DATA X FOR CMD & GET CONTR RDY

F08

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 96
T14 TEST READ DATA AT ALL HEAD OFFSET POSITIONS

SEQ 0096

4987	024304	104015			ERROR	15	;NO RDY AFTER WRITE CHECK CMD
4988	024306	004737	033664		JSR	PC, GSTAT	;GET FRESH STATUS
4989	024312	032737	100000	007354	BIT	#CERR, HCS1	
4990	024320	001453			BEQ	69\$	
4991	024322	032737	040000	007356	BIT	#WCE, HCS2	;SEE IF WRITE CHECK ERROR
4992	024330	001410			BEQ	68\$	
4993	024332	016537	000024	001452	MOV	RK06(R5), WD1	;ACTUAL WORD FOR PRINTOUT
4994	024340	013737	001500	001454	MOV	DATA1, WD2	;EXPECTED WORD FOR TYPEOUT
4995	024346	104016			ERROR	16	;WCE AFTER WRITE CMD
4996	024350	000437			BR	69\$	
4997							
4998	024352	104022			68\$: ERROR	22	;CERR AFTER WRITE CHECK CMD
4999							
5000	024354	012737	010340	007444	MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA), E.A0	;EXPECTED MSG A0
5001	024362	005037	007446		CLR	E.B0	;EXPECTED MSG B0
5002	024366	012737	001720	007450	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP), E.A1	;EXPECTED A1
5003	024374	012737	000001	007452	MOV	#1, E.B1	;MSG ID FOR EXPECTED MSG B1
5004	024402	005037	007454		CLR	E.A2	;EXPECTED MSG A2
5005	024406	012737	000002	007456	MOV	#2, E.B2	;MSG ID FOR EXPECTED MSG B2
5006	024414	012737	000003	007462	MOV	#3, E.B3	;MSG ID FOR EXPECTED MSG B3
5007							
5008	024422	004737	033042		JSR	PC, CHKMSG	;CHECK MSGS A0, B0, A1, B1
5009	024426	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5010	024430	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
5011	024432	104031			ERROR	31	;MSG B0 ERROR
5012	024434	104060			ERROR	60	;MSG A1 ERROR
5013	024436	104032			ERROR	32	;MSG B1 ERROR
5014	024440	104401	045354		TYPE	MSG26	;ABORTING BAL OF TESTS
5015	024444	000137	031452		JMP	\$EOP	
5016							
5017	024450				69\$:		
5018							
5019	024450	012737	010340	007444	MOV	#(0!D.SPIN!D.DRDY!D.VV!D.DRA), E.A0	;EXPECTED MSG A0
5020	024456	005037	007446		CLR	E.B0	;EXPECTED MSG B0
5021	024462	012737	001720	007450	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP), E.A1	;EXPECTED A1
5022	024470	012737	000001	007452	MOV	#1, E.B1	;MSG ID FOR EXPECTED MSG B1
5023	024476	005037	007454		CLR	E.A2	;EXPECTED MSG A2
5024	024502	012737	000002	007456	MOV	#2, E.B2	;MSG ID FOR EXPECTED MSG B2
5025	024510	012737	000003	007462	MOV	#3, E.B3	;MSG ID FOR EXPECTED MSG B3
5026							
5027	024516	004737	033042		JSR	PC, CHKMSG	;CHECK MSGS A0, B0, A1, B1
5028	024522	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5029	024524	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
5030	024526	104031			ERROR	31	;MSG B0 ERROR
5031	024530	104060			ERROR	60	;MSG A1 ERROR
5032	024532	104032			ERROR	32	;MSG B1 ERROR
5033							
5034	024534	104401	044641		TYPE	MSG8	;READ WITH OFFSET TEST
5035	024540	005001			CLR	A1	;HEAD #
5036							
5037	024542	012700	000001		9\$: MOV	#1, R0	;INIT OFFSET COUNTER
5038	024546	104401	044700		TYPE	MSG9	;HEAD #
5039	024552	010146			MOV	A1, -(SP)	;SAVE R1 FOR TYPEOUT
5040							;TYPE HEAD #
5041	024554	104403			TYPOS		;GO TYPE--OCTAL ASCII
5042	024556	001			.BYTE	1	;TYPE 1 DIGIT(S)

5043	024557	000			.BYTE	0		::SUPPRESS LEADING ZEROS
5044	024560	104401	001205		TYPE	,SCLF		
5045								
5046	024564	005037	001456		1\$: CLR	OFFERR		;WRITE CHECK ERROR FLAG
5047								
5048	024570	004737	034214		JSR	PC,SUBCLR		
5049	024574	104024			ERROR	24		;CERR AFTER SCLR
5050								
5051	024576	010065	000016		MOV	RD,RKASOF(R5)		;OFFSET VALUE
5052	024602	000301			SWAB	R1		
5053	024604	010165	000006		MOV	R1,RKDA(R5)		;HEAD NO.
5054	024610	000301			SWAB	R1		
5055								
5056	024612	012737	024700	001176	MOV	#70\$, \$ESCAPE		
5057	024620	012737	000015	007354	MOV	#OFFSET, HCS1		
5058	024626	004737	032224		JSR	PC,DOCMD		;DO RECAL CMD & GET CONTR RDY
5059	024632	104033			ERROR	33		;NO RDY AFTER OFFSET CMD
5060								
5061	024634	012737	032140	007444	MOV	#<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
5062	024642	005037	007446		CLR	E.B0		
5063	024646	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		
5064	024654	012737	000001	007452	MOV	#1,E.B1		
5065								
5066	024662	004737	033042		JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1
5067	024666	000000			.WORD	0!0!0		; & MSGS SPECIFIED HERE
5068	024670	104035			ERROR	35		;MSG A0 ERROR DURING OFFSET CMD
5069	024672	104061			ERROR	61		;MSG B0 ERROR
5070	024674	104036			ERROR	36		;MSG A1 ERROR
5071	024676	104062			ERROR	62		;MSG B1 ERROR
5072								
5073	024700	005037	001176		70\$: CLR	\$ESCAPE		
5074	024704	013737	001416	007412	MOV	T5000,TEMP1		;SETUP TIMEOUT
5075	024712	004737	032730		JSR	PC,FATT2		;FIND ATTN
5076	024716	104034			ERROR	34		;NO ATTN AFTER OFFSET CMD
5077								
5078								
5079	024720	012737	052340	007444	MOV	#<D.DSC!D.OFF!D.SPIN!D.DROY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
5080	024726	005037	007446		CLR	E.B0		;EXPECTED MSG B0
5081	024732	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		;EXPECTED A1
5082	024740	012737	000001	007452	MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
5083	024746	005037	007454		CLR	E.A2		;EXPECTED MSG A2
5084	024752	012737	000002	007456	MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
5085	024760	012737	000003	007462	MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
5086								
5087	024766	004737	033042		JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1
5088	024772	000003			.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE
5089	024774	104260			ERROR	260		;MSG A0 ERROR AFTER OFFSET CMD
5090	024776	104261			ERROR	261		;MSG B0 ERROR
5091	025000	104037			ERROR	37		;MSG A1 ERROR
5092	025002	104040			ERROR	40		;MSG B1 ERROR
5093								
5094								
5095	025004	012765	100000	000000	MOV	#CLR,RKCS1(R5)		
5096	025012	013765	001222	000010	MOV	\$UNIT,RKCS2(R5)		;DRIVE#
5097	025020	012737	000005	007354	MOV	#CLEAR,HCS1		
5098	025026	004737	032224		JSR	PC,DOCMD		;DO DRIVE CLEAR CMD & GET CONTR RDY

5099	025032	104151		ERROR	151	;NO RDY AFTER DRIVE CLEAR CMD
5100	025034	004737	032602	JSR	PC, TSTATN	;TEST FOR ATTN
5101	025040	000401		BR	71\$	
5102	025042	104154		ERROR	154	;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5103	025044			71\$:		
5104						
5105	025044	012765	001500	MOV	#DATA1, RKBA(R5)	
5106	025052	052765	000020	BIS	#BA1, RKCS2(R5)	
5107	025060	012765	177400	MOV	#-256, RKWC(R5)	
5108	025066	013765	001402	MOV	SECTOR, RKDA(R5)	
5109	025074	012737	000031	MOV	#WATCH, HCS1	
5110	025102	004737	032262	JSR	PC, DATCMD	;DO WRITE CHECK CMD. & GET CONTR RDY.
5111	025106	104015		ERROR	15	;NO RDY AFTER WRITE CHECK CMD
5112	025110	004737	033664	JSR	PC, GSTAT	;GET FRESH STATUS
5113	025114	032737	040000	BIT	#WCE, HCS2	
5114	025122	001421		BEQ	2\$	
5115						
5116	025124	016537	000024	MOV	RKDB(R5), WD1	;GET MISCOMPARED WORD
5117	025132	005237	001456	INC	OFFERR	;BAD WRITE CHK ERROR=SET ERR FLG.
5118						
5119	025136	005737	001456	TST	OFFERR	
5120	025142	001411		BEQ	2\$	
5121	025144	104401	045620	TYPE	MSG39	;WRITE CHECK FAILURE AT OFFSET
5122	025150	010046		MOV	RO, -(SP)	;SAVE RO FOR TYPEOUT
5123						;TYPE 0: SET VALUE
5124	025152	104403		TYPOS		;GO TYPE--OCTAL ASCII
5125	025154	006		.BYTE	6	;TYPE 6 DIGITS
5126	025155	000		.BYTE	0	;SUPPRESS LEADING ZEROS
5127	025156	104401	001205	TYPE	, \$CRLF	
5128	025162	104401	001205	TYPE	, \$CRLF	
5129						
5130	025166	032700	000200	2\$:	BIT	#BIT7, RO
5131	025172	001023		BNE	5\$	;SEE IF OFFSET IS + OR -
5132						;BR IF - OFFSET
5133	025174	020027	000060	CMP	RO, #60	
5134	025200	001412		BEQ	4\$	
5135	025202	005737	001456	TST	OFFERR	
5136	025206	001404		BEQ	3\$	
5137	025210	012700	000200	8\$:	MOV	#200, RO
5138	025214	000137	024564	JMP	1\$	;SETUP FOR NEG OFFSET
5139						
5140	025220	005200		3\$:	INC	RO
5141	025222	000137	024564	JMP	1\$	
5142						
5143	025226	005737	001456	4\$:	TST	OFFERR
5144	025232	001366		BNE	8\$	;DO NEG OFFSETS
5145	025234	104401	045462	TYPE	, MSG37	;NO WRITE CHECK ERROR AT MAX POS OFFSET
5146						;NOTE! EITHER HEADS DID NOT MOVE
5147						;OR READ/WRITE AMP IS EXCEPTIONALLY GOOD.
5148	025240	000763		BR	8\$	;DO NEG OFFSETS
5149						
5150	025242	020027	000260	5\$:	CMP	RO, #260
5151	025246	001404		BEQ	6\$	
5152	025250	005737	001456	TST	OFFERR	
5153	025254	001072		BNE	TST15	::GO TO NEXT TST
5154	025256	000760		BR	3\$	

```

5155
5156 025260 005737 001456      6$: TST OFFERR
5157 025264 001002              BNE 7$
5158 025266 104401 045540      TYPE ,MSG38
5159                                     ;NO WRITE CHECK ERROR AT MAX NEG OFFSET
5160                                     ;NOTE! EITHER HEADS DID NOT MOVE
5161                                     ;OR READ/WRITE AMP IS EXCEPTIONALLY GOOD.
5162
5163 025272
5164 025272 012765 100000 000000      MOV #CCLR,RKCS1(R5)
5165 025300 013765 001222 000010      MOV $UNIT,RKCS2(R5)
5166 025306 012737 000013 007354      MOV #RECAL,HCS1
5167 025314 004737 032224              JSR PC,DOCMD ;DO RECAL CMD & GET CONTR RDY
5168 025320 104124              ERROR 124 ;RDY NOT SET AFTER RECAL CMD
5169
5170 025322 012765 000001 000026      MOV #1,RKMR1(R5) ;SELECT WORD 1
5171 025330 004737 033664              JSR PC,GSTAT
5172 025334 032737 020000 007402      BIT #D.RTZ,HMR2
5173 025342 001001              BNE 72$
5174 025344 104244              ERROR 244 ;RTZ NOT SET DURING RECAL CMD
5175 025346 013737 001406 007414 72$: MOV T10,TEMP2 ;SETUP TIMEOUT
5176 025354 004737 032634              JSR PC,FATT1 ;FIND ATTN
5177 025360 104055              ERROR 55 ;NO ATTN AFTER RECAL CMD
5178
5179 025362 012765 100000 000000      MOV #CCLR,RKCS1(R5)
5180 025370 013765 001222 000010      MOV #UNIT,RKCS2(R5) ;DRIVE#
5181 025376 012737 000005 007354      MOV #CLEAR,HCS1
5182 025404 004737 032224              JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
5183 025410 104151              ERROR 31 ;NO RDY AFTER DRIVE CLEAR CMD
5184 025412 004737 032602              JSR PC,TSTATN ;TEST FOR ATTN
5185 025416 000401              BR 73$
5186 025420 104154              ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5187
5188
5189 025422 005201              INC R1 ;HEAD CTR
5190 025424 020127 000003      CMP R1,#3 ;SEE IF ALL HEADS DONE
5191 025430 001402              BEQ 10$ ;BR IF YES
5192 025432 000137 024542      JMP 9$ ;ELSE REPEAT ALL FOR NEXT HEAD
5193 025436 104401 045000      TYPE ,MSG12 ;OFFSET FAILURES ARE NOT ERRORS
5194
5195 *****
5196 *TEST 15 WRITE WITH HEADS OFFSET
5197 *
5198 * THIS TEST VERIFIES THAT WHEN ATTEMPTING TO
5199 * WRITE WITH HEADS OFFSET THAT THE OFFSET WILL CLEAR
5200 * & THE DRIVE WILL WRITE
5201 * SINCE THE WRITE COMMAND HAS AN IMPLIED RTC.
5202 * THIS TEST IS PERFORMED FOR MAX POS & NEG OFFSETS ONLY
5203 *****
5204 025442 000004      ST15: SCOPE
5205 025444 012737 000001 001174      MOV #1,$TIMES ;DO 1 ITERATION
5206 025452 012706 001100              MOV #STACK,SP ;RESTORE STK PTR
5207
5208 025456 012700 000260              MOV #260,R0 ;MAX NEG OFFSET
5209
5210 025462 004737 034214      1$: JSR PC,SUBCLR

```

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 100
T15 WRITE WITH HEADS OFFSET

SEQ 0100

5211	025466	104024		ERROR	24		;CERR AFTER SCLR
5212							
5213	025470	010065	000016	MOV	RD,RKASCF(R5)		;SET OFFSET
5214							
5215	025474	012737	025562	MOV	#E4\$,SESCAPE		
5216	025502	012737	000015	MOV	#C,FSET,HCS1		
5217	025510	004737	032224	JSR	PC,DOCMD		;DO RECAL CMD & GET CONTR RDY
5218	025514	104033		ERROR	33		;NO RDY AFTER OFFSET CMD
5219							
5220	025516	012737	032140	MOV	#<D.PIP!D.SPIN!D.OFF!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
5221	025524	005037	007446	CLR	E.B0		
5222	025530	012737	001720	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		
5223	025536	012737	000001	MOV	#1,E.B1		
5224							
5225	025544	004737	033042	JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1
5226	025550	000000		.WORD	0!0!0		; & MSGS SPECIFIED HERE
5227	025552	104035		ERROR	35		;MSG A0 ERROR DURING OFFSET CMD
5228	025554	104061		ERROR	61		;MSG B0 ERROR
5229	025556	104036		ERROR	36		;MSG A1 ERROR
5230	025560	104062		ERROR	62		;MSG B1 ERROR
5231							
5232	025562	005037	001176	CLR	SESCAPE		
5233	025566	013737	001416	MOV	T5000,TEMP1		;SETUP TIMEOUT
5234	025574	004737	032730	JSR	PC,FATT2		;FIND ATTN
5235	025600	104034		ERROR	34		;NO ATTN AFTER OFFSET CMD
5236							
5237							
5238	025602	012737	052340	MOV	#<D.DSC!D.OFF!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
5239	025610	005037	007446	CLR	E.B0		;EXPECTED MSG B0
5240	025614	012737	001720	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		;EXPECTED A1
5241	025622	012737	000001	MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
5242	025630	005037	007454	CLR	E.A2		;EXPECTED MSG A2
5243	025634	012737	000002	MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2
5244	025642	012737	000003	MOV	#3,E.B3		;MSG ID FOR EXPECTED MSG B3
5245							
5246	025650	004737	033042	JSR	PC,CHKMSG		;CHECK MSGS A0, B0, A1, B1
5247	025654	000003		.WORD	T.A2!T.B2!0		; & MSGS SPECIFIED HERE
5248	025656	104260		ERROR	260		;MSG A0 ERROR AFTER OFFSET CMD
5249	025660	104261		ERROR	261		;MSG B0 ERROR
5250	025662	104037		ERROR	37		;MSG A1 ERROR
5251	025664	104040		ERROR	40		;MSG B1 ERROR
5252							
5253							
5254	025666	012765	100000	MOV	#CCLR,RKCS1(R5)		
5255	025674	013765	001222	MOV	\$UNIT,RKCS2(R5)		;DRIVE#
5256	025702	012737	000005	MOV	#CLEAR,HCS1		
5257	025710	004737	032224	JSR	PC,DOCMD		;DO DRIVE CLEAR CMD & GET CONTR RDY
5258	025714	104151		ERROR	151		;NO RDY AFTER DRIVE CLEAR CMD
5259	025716	004737	032602	JSR	PC,TSTATN		;TEST FOR ATTN
5260	025722	000401		BR	65\$		
5261	025724	104154		ERROR	154		;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5262	025726						
5263							
5264							
5265	025726	005037	001402	CLR	SECTOR		
5266	025732	013765	001402	MOV	SECTOR,RKDA(R5)		

5267	025740	012765	001474	000004	MOV	#DATA0,RKBA(R5)	;WRITE ALL 0'S
5268	025746	052765	000020	000010	BIS	#BAI,RKCS2(R5)	;BUS ADDR INCR INH
5269	025754	012765	177400	000002	MOV	#-256.,RKWC(R5)	;FULL SECTOR
5270							
5271	025762	012737	000023	007354	MOV	#<WRDATA>,HCS1	
5272	025770	004737	032262		JSR	PC.DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
5273	025774	104011			ERROR	11	;NO RDY AFTER WRITE DATA CMD
5274	025776	004737	033664		JSR	PC.GSTAT	;GET FRESH STATUS
5275	026002	032737	100000	007354	BIT	#CERR,HCS1	
5276	026010	001465			BEQ	69\$	;BR IF NO ERRORS
5277							
5278	026012	032737	000200	007370	BIT	#BSE,HER	;SEE IF BAD SECTOR FLAG
5279	026020	001421			BEQ	67\$	;BR IF NO
5280	026022	004737	035700		JSR	PC.TRUERR	;ELSE SEE IF SECTOR LISTED IN BSE TABLE
5281	026026	000455			BR	68\$	;RETURN HERE IF NO
5282							
5283	026030	005237	001402		INC	SECTOR	;RETURN HERE IF YES
5284	026034	023727	001402	000012	CMP	SECTOR,#10.	;ARE 10 CONSEC. SECTORS BAD
5285	026042	001003			BNE	66\$	;BR IF NO
5286	026044	104046			ERROR	46	;ABORTING TEST DETECTED 10 BAD SECTORS
5287	026046	000137	026756		JMP	3\$	;BYPASS TEST
5288	026052	012765	100000	000000	MOV	#CCLR,RKCS1(R5)	;TRY ANOTHER SECTOR
5289	026060	000137	025732		JMP	4\$	
5290	026064	104012			ERROR	12	;CERR WITH WRITE DATA CMD
5291							
5292	026066	012737	010340	007444	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5293	026074	005037	007446		CLR	E.B0	;EXPECTED MSG B0
5294	026100	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5295	026106	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5296	026114	005037	007454		CLR	E.A2	;EXPECTED MSG A2
5297	026120	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5298	026126	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5299							
5300	026134	004737	033042		JSR	PC.CHKMSG	;CHECK MSGS A0, B0, A1, B1
5301	026140	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5302	026142	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5303	026144	104023			ERROR	23	;MSG B0 ERROR
5304	026146	104053			ERROR	53	;MSG A1 ERROR
5305	026150	104025			ERROR	25	;MSG B1 ERROR
5306	026152	104401	045354		TYPE	MSG26	;ABORTING BAL OF TESTS
5307	026156	000137	031452		JMP	\$EOP	
5308	026162	104063			ERROR	63	;BAD SECTOR NOT LISTED IN TABLE
5309	026164						
5310							
5311	026164	012737	010340	007444	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5312	026172	005037	007446		CLR	E.B0	;EXPECTED MSG B0
5313	026176	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5314	026204	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5315	026212	005037	007454		CLR	E.A2	;EXPECTED MSG A2
5316	026216	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5317	026224	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5318							
5319	026232	004737	033042		JSR	PC.CHKMSG	;CHECK MSGS A0, B0, A1, B1
5320	026236	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5321	026240	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5322	026242	104023			ERROR	23	;MSG B0 ERROR

L08

CZR6IED UNIBUS RK6 DR PR2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 102
T15 WRITE WITH HEADS OFFSET

SEQ 0102

5323	026244	104053			ERROR	53	;MSG A1 ERROR
5324	026246	104025			ERROR	25	;MSG B1 ERROR
5325							
5326	026250	012765	000002	000026	MOV	#2,RKMR1(R5)	;SELECT WORD 2
5327	026256	004737	033664		JSR	PC,GSTAT	
5328	026262	005737	001356		TST	CYLDIF	;SEE IF MSG A2=0
5329	026266	001401			BEQ	70\$	;BR IF YES
5330	026270	104104			ERROR	104	;MSG A2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
5331	026272	005737	001360	70\$:	TST	CYLADD	;SEE IF MSG B2=0
5332	026276	001401			BEQ	71\$	;BR IF YES
5333	026300	104105			ERROR	105	;MSG B2 NOT CLEARED AFTER WRITE CMD WITH OFFSET
5334	026302			71\$:			
5335							
5336	026302	104415			SCOP1		
5337	026304	012706	001100		MOV	#STACK,SP	;RESTORE STK PTR
5338							
5339	026310	004737	034214		JSR	PC,SUBCLR	
5340	026314	104024			ERROR	24	;CERR AFTER SCLR
5341							
5342							
5343	026316	012765	001474	000004	MOV	#DATA0,RKBA(R5)	
5344	026324	052765	000020	000010	BIS	#BA1,RKCS2(R5)	
5345	026332	012765	177400	000002	MOV	#-256.,RKWC(R5)	
5346	026340	013765	001402	000006	MOV	SECTOR,RKDA(R5)	
5347							
5348	026346	012737	000031	007354	MOV	#(WRTCHK),HCS1	
5349	026354	004737	032262		JSR	PC,DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
5350	026360	104015			ERROR	15	;NO RDY AFTER WRITE CHECK CMD
5351	026362	004737	033664		JSR	PC,GSTAT	;GET FRESH STATUS
5352	026366	032737	100000	007354	BIT	#CERR,HCS1	
5353	026374	001453			BEQ	73\$	
5354	026376	032737	040000	007356	BIT	#WCE,HCS2	;SEE IF WRITE CHECK ERROR
5355	026404	001410			BEQ	72\$	
5356	026406	016537	000024	001452	MOV	RKDB(R5),WD1	;ACTUAL WORD FOR PRINTOUT
5357	026414	013737	001474	001454	MOV	DATA0,WD2	;EXPECTED WORD FOR TYPEOUT
5358	026422	104016			ERROR	16	;WCE AFTER WRITE CMD
5359	026424	000437			BR	73\$	
5360							
5361	026426	104022		72\$:	ERROR	22	;CERR AFTER WRITE CHECK CMD
5362							
5363	026430	012737	010340	007444	MOV	#(O!D.SPIN!D.DRDY!D.VV!D.DRA),E.A0	;EXPECTED MSG A0
5364	026436	005037	007446		CLR	E.B0	;EXPECTED MSG B0
5365	026442	012737	001720	007450	MOV	#(D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP),E.A1	;EXPECTED A1
5366	026450	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5367	026456	005037	007454		CLR	E.A2	;EXPECTED MSG A2
5368	026462	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5369	026470	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5370							
5371	026476	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5372	026502	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5373	026504	104057			ERROR	57	;MSG A0 ERROR AFTER WRITE CHECK CMD
5374	026506	104031			ERROR	31	;MSG B0 ERROR
5375	026510	104060			ERROR	60	;MSG A1 ERROR
5376	026512	104032			ERROR	32	;MSG B1 ERROR
5377	026514	104401	045354		TYPE	_MSG26	;ABORTING BAL OF TESTS
5378	026520	000137	031452		JMP	\$EOP	

M08

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 103
T15 WRITE WITH HEADS OFFSET

SEQ 0103

```

5379
5380 026524 73$:
5381
5382 026524 012737 010340 007444 MOV #0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5383 026532 005037 007446 CLR E.B0 ;EXPECTED MSG B0
5384 026536 012737 001720 007450 MOV #0!D.SPOK!D.CART!D.DOOR!D.BRAH!D.SSP>,E.A1 ;EXPECTED A1
5385 026544 012737 000001 007452 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5386 026552 005037 007454 CLR E.A2 ;EXPECTED MSG A2
5387 026556 012737 000002 007456 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5388 026564 012737 000003 007462 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5389
5390 026572 004737 033042 JSR PC,CHKMSG ;CHECK MSGS A0 B0 A1 B1
5391 026576 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5392 026600 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
5393 026602 104031 ERROR 31 ;MSG B0 ERROR
5394 026604 104060 ERROR 60 ;MSG A1 ERROR
5395 026606 104032 ERROR 32 ;MSG B1 ERROR
5396
5397 026610 020027 000260 CMP R0,#260
5398 026614 001004 BNE 2$ ;BR IF JUST DID POS OFFSET
5399 026616 012700 000060 MOV #60,R0 ;ELSE SETUP FOR POS OFFSET
5400 026622 000137 025462 JMP 1$
5401
5402 026626 2$:
5403
5404 026626 012765 100000 000000 MOV #CCLR,RKCS1(R5)
5405 026634 013765 001222 000010 MOV $UNIT,RKCS2(R5)
5406 026642 012737 000013 007354 MOV #RECAL,HCS1
5407 026650 004737 032224 JSR PC,DOCMD ;DO RECAL CMD & GET CONTR RDY
5408 026654 104124 ERROR 124 ;RDY NOT SET AFTER RECAL CMD
5409
5410 026656 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
5411 026664 004737 033664 JSR PC,GSTAT
5412 026670 032737 020000 007402 BIT #D.RTZ,HMR2
5413 026676 001001 BNE 74$
5414 026700 104244 ERROR 244 ;RTZ NOT SET DURING RECAL CMD
5415 026702 013737 001406 007414 74$: MOV T10,TEMP2 ;SETUP TIMEOUT
5416 026710 004737 032634 JSR PC,FATT1 ;FIND ATTN
5417 026714 104055 ERROR 55 ;NO ATTN AFTER RECAL CMD
5418
5419 026716 012765 100000 000000 MOV #CCLR,RKCS1(R5)
5420 026724 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
5421 026732 012737 000005 007354 MOV #CLEAR,HCS1
5422 026740 004737 032224 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
5423 026744 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
5424 026746 004737 032602 JSR PC,TSTATN ;TEST FOR ATTN
5425 026752 000401 BR 75$
5426 026754 104154 ERROR 154 ;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD
5427 026756 75$:
5428
5429
5430 026756 3$:
5431

```

```
*****
TEST 16      TEST CURRENT CROSS-OVER CYLINDERS
*****
```

```

THIS TEST VERIFIES THAT THE DRIVE CAN WRITE & READ OFF
CURRENT CHANGE CYLINDERS X & Y IN THE FOLLOWING WAY:

SPIRAL WRITING IS PERFORMED FROM CYLINDER X TO CYLINDER Y
WITH A DATA PATTERN FILLING THE ENTIRE 2 CYLINDERS.

A WRITE CHECK IS THEN PERFORMED TO VERIFY DATA WAS PROPERLY WRITTEN.
THIS TEST IS PERFORMED FOR ALL 3 HEADS.

CYLINDER X:  63  127  191  255  319  383  RK06
CYLINDER Y:  64  128  192  256  320  384  RK06

CYLINDER X:  127 255  383  511  639  767  RK07
CYLINDER Y:  128 256  384  512  640  768  RK07

ALL ABOVE CYLINDERS IN DECIMAL

```

```
*****
```

```

TST16:  SCOPE
        MOV     #1,STIMES      ;;DO 1 ITERATION
        MOV     #STACK,SP
        TST     $TMP4          ;SEE IF RK07
        BEQ     $$             ;BR IF NO
        MOV     #CYL7,RO       ;ELSE LOAD UP RK07 TABLE
        BR      1$
        MOV     #CYL,RO        ;RK06 CYL ADDR TABLE
        JSR     PC,SUBCLR      ;CERR AFTER SCLR
        ERROR   24
        MOV     (RO),RKDC(R5)  ;CYL #
        MOV     #DPAT1,RKBA(R5);DATA PATTERN
        BIS     #BAI,RKCS2(R5);BUSS ADDR INCREMENT INHIBIT
        MOV     #-6*22,*256.,RKWC(R5);WORD COUNT TO SPIRAL & FILL 2 CYLINDERS

        MOV     #<WRDATA>,HCS1
        JSR     PC,DATCMD      ;DO DATA X FOR CMD & GET CONTR RDY
        ERROR   11            ;NO RDY AFTER WRITE DATA CMD
        JSR     PC,GSTAT       ;GET FRESH STATUS
        BIT     #CERR,HCS1
        BEQ     67$           ;BR IF NO ERRORS

        BIT     #BSE,HER       ;SEE IF BAD SECTOR FLAG
        BEQ     65$           ;BR IF NO
        JSR     PC,TRUERR      ;ELSE SEE IF SECTOR LISTED IN BSE TABLE
        BR      66$           ;RETURN HERE IF NO
        JMP     3$             ;RET HERE IF YES
        ERROR   12            ;CERR WITH WRITE DATA CMD
        65$:  MOV     #<O!D.SPIN!D.DRDY!D.VV!D.DRA>,E.AO ;EXPECTED MSG AO
        CLR     E.B0           ;EXPECTED MSG B0

```

```

5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454 026756 000004
5455 026760 012737 000001 001174
5456 026766 012706 001100
5457
5458 026772 005737 001170
5459 026776 001403
5460 027000 012700 001436
5461 027004 000402
5462 027006 012700 001422
5463
5464 027012 004737 034214
5465 027016 104024
5466
5467 027020 011065 000020
5468 027024 012765 001502 000004
5469 027032 052765 000020 000010
5470 027040 012765 076000 000002
5471
5472 027046 012737 000023 007354
5473 027054 004737 032262
5474 027060 104011
5475 027062 004737 033664
5476 027066 032737 100000 007354
5477 027074 001451
5478
5479 027076 032737 000200 007370
5480 027104 001405
5481 027106 004737 035700
5482 027112 000441
5483 027114 000137 027574
5484 027120 104012
5485
5486 027122 012737 010340 007444
5487 027130 005037 007446

```

5488	027134	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5489	027142	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5490	027150	005037	007454		CLR	E.A2	;EXPECTED MSG A2
5491	027154	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5492	027162	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5493							
5494	027170	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5495	027174	000003			WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5496	027176	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5497	027200	104023			ERROR	23	;MSG B0 ERROR
5498	027202	104053			ERROR	53	;MSG A1 ERROR
5499	027204	104025			ERROR	25	;MSG B1 ERROR
5500	027206	104401	045354		TYPE	MSG26	;ABORTING BAL OF TESTS
5501	027212	000137	031452		JMP	\$EOP	
5502	027216	104063			ERROR	63	;BAD SECTOR NOT LISTED IN TABLE
5503	027220						
5504							
5505	027220	012737	010340	007444	MOV	#<O!D.SPIN!D.DRD!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5506	027226	005037	007446		CLR	E.B0	;EXPECTED MSG B0
5507	027232	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5508	027240	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5509	027246	005037	007454		CLR	E.A2	;EXPECTED MSG A2
5510	027252	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5511	027260	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5512							
5513	027266	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5514	027272	000003			WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5515	027274	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5516	027276	104023			ERROR	23	;MSG B0 ERROR
5517	027300	104053			ERROR	53	;MSG A1 ERROR
5518	027302	104025			ERROR	25	;MSG B1 ERROR
5519							
5520	027304	011065	000020		MOV	(R0),RKDC(R5)	
5521	027310	012765	001502	000004	MOV	#DPAT1,RKBA(R5)	
5522	027316	052765	000020	000010	BIS	#BAI,RKCS2(R5)	
5523	027324	012765	076000	000002	MOV	#-6*22.*256.,RKWC(R5)	
5524							
5525	027332	012737	000031	007354	MOV	#<WRTCHK>,HCS1	
5526	027340	004737	032262		JSR	PC,DATCMD	;DO DATA X FOR CMD & GET CONTR RDY
5527	027344	104015			ERROR	15	;NO RDY AFTER WRITE CHECK CMD
5528	027346	004737	033664		JSR	PC,GSTAT	;GET FRESH STATUS
5529	027352	032737	100000	007354	BIT	#CERR,HCS1	
5530	027360	001453			BEQ	69\$	
5531	027362	032737	040000	007356	BIT	#WCE,HCS2	;SEE IF WRITE CHECK ERROR
5532	027370	001410			BEQ	68\$	
5533	027372	016537	000024	001452	MOV	RKDB(R5),WD1	;ACTUAL WORD FOR PRINTOUT
5534	027400	013737	001502	001454	MOV	DPAT1,WD2	;EXPECTED WORD FOR TYPEOUT
5535	027406	104016			ERROR	16	;WCE AFTER WRITE CMD
5536	027410	000437			BR	69\$	
5537							
5538	027412	104022			ERROR	22	;CERR AFTER WRITE CHECK CMD
5539							
5540	027414	012737	010340	007444	MOV	#<O!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5541	027422	005037	007446		CLR	E.B0	;EXPECTED MSG B0
5542	027426	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1	;EXPECTED A1
5543	027434	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1

CZR61EO UNICUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 106
T16 TEST CURRENT CROSS-OVER CYLINDERS

SEQ 0106

```

5544 027442 005037 007454 CLR E.A2 ;EXPECTED MSG A2
5545 027446 012737 000002 007456 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5546 027454 012737 000003 007462 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5547
5548 027462 004737 033042 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5549 027466 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5550 027470 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
5551 027472 104031 ERROR 31 ;MSG B0 ERROR
5552 027474 104060 ERROR 60 ;MSG A1 ERROR
5553 027476 104032 ERROR 32 ;MSG B1 ERROR
5554 027500 104401 045354 TYPE MSG26 ;ABORTING BAL OF TESTS
5555 027504 000137 031452 JMP $EOP
5556
5557 027510 69$:
5558
5559 027510 012737 010340 007444 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5560 027516 005037 007446 CLR E.B0 ;EXPECTED MSG B0
5561 027522 012737 001720 007450 MOV #<0.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1 ;EXPECTED A1
5562 027530 012737 000001 007452 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5563 027536 005037 007454 CLR E.A2 ;EXPECTED MSG A2
5564 027542 012737 000002 007456 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5565 027550 012737 000003 007462 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5566
5567 027556 004737 033042 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5568 027562 000003 .WORD T.A2!T.B2!0 ;& MSGS SPECIFIED HERE
5569 027564 104057 ERROR 57 ;MSG A0 ERROR AFTER WRITE CHECK CMD
5570 027566 104031 ERROR 31 ;MSG B0 ERROR
5571 027570 104060 ERROR 60 ;MSG A1 ERROR
5572 027572 104032 ERROR 32 ;MSG B1 ERROR
5573 027574 022027 000577 3$: CMP (R0)+, #383. ;ALL CYLINDERS DONE?
5574 027600 001402 BEQ 4$ ;BR IF YES
5575 027602 000137 027012 JMP 1$ ;ELSE REPEAT
5576
5577 027606 4$:
5578
5579 027606 012765 100000 000000 MOV #CLR,RKCS1(R5)
5580 027614 013765 001222 000010 MOV $UNIT,RKCS2(R5)
5581 027622 012737 000013 007354 MOV #RECAL,HCS1
5582 027630 004737 032224 JSR PC,DOCMD ;DO RECAL CMD & GET CONTR RDY
5583 027634 104124 ERROR 124 ;RDY NOT SET AFTER RECAL CMD
5584
5585 027636 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
5586 027644 004737 033664 JSR PC,GSTAT
5587 027650 032737 020000 007402 BIT #0,RTZ,HMR2
5588 027656 0010C1 BNE 70$
5589 027660 104244 ERROR 244 ;RTZ NOT SET DURING RECAL CMD
5590 027662 013737 001406 007414 70$: MOV T10,TEMP2 ;SETUP TIMEOUT
5591 027670 004737 032634 JSR PC,FATT1 ;FIND ATTN
5592 027674 104055 ERROR 55 ;NO ATTN AFTER RECAL CMD
5593
5594 027676 012765 100000 000000 MOV #CLR,RKCS1(R5)
5595 027704 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;DRIVE#
5596 027712 012737 000005 007354 MOV #CLEAR,HCS1
5597 027720 004737 032224 JSR PC,DOCMD ;DO DRIVE CLEAR CMD & GET CONTR RDY
5598 027724 104151 ERROR 151 ;NO RDY AFTER DRIVE CLEAR CMD
5599 027726 004737 032602 JSR PC,TSTATN ;TEST FOR ATTN

```

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 107
T16 TEST CURRENT CROSS-OVER CYLINDERS

SEQ 0107

5600 027732 000401
5601 027734 104154
5602 027736

BR 715
ERROR 154

;ATTN NOT CLEARED AFTER DRIVE CLEAR CMD

715:

5603
5604
5605 027736
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624

25:

TEST 17 TEST HEAD SWITCHING TIME

TESTS THE ABILITY TO SWITCH HEADS IN LESS THEN 10MS WHEN HEADS SPIRAL.

1. SECTOR 23(8) IS FIRST LOCATED AND A WRITE DATA COMMAND OF 512 WORDS TO SECTOR 25(8) IS ISSUED.

2. THE PROGRAM NOW KNOWS THAT THE DRIVE WILL NOT HAVE TO TRAVEL A FULL REVOLUTION BEFORE FINDING SECTOR 25(8).

3. SINCE EACH SECTOR TAKES APPROX. 1.2MS, THE TIME BETWEEN THE START OF THE WRITE COMMAND (FROM SECTOR 25(8), HEAD 0; TO SECTOR 0, HEAD 1) AND CONTROLLER READY SHOULD BE APPROX 6MS

THE ABOVE IS REPEATED FOR HEAD SWITCHING BETWEEN 1 TO 2

THIS TEST IS BYPASSED IF NEITHER L OR P CLOCK IS PRESENT

ST17: SCOPE

5625 027736 000004
5626 027740 012737 J00001 001174
5627 027746 012706 001100
5628
5629 027752 005737 007522

MOV #1,STIMES ;;DO 1 ITERATION
MOV #STACK,SP

TST DOTIM ;BYPASS THIS TEST IF

E09

CZK6IED UNIBUS RK6 DR PRT2
CZK6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 108
T17 TEST HEAD SWITCHING TIME

SEQ 0108

5630 027756 001001
5631 027760 000526

BNE 15
BR TST20

;NEITHER L OR P CLOCK PRESENT
;;GO TO NEXT TEST

```

5632
5633 027762 012737 000025 007422 1$: MOV #25,TEMP5 ;HEAD 0, SECTOR 21 TO BE PUT IN RKDA
5634 027770 004737 034214 2$: JSR PC,SUBCLR ;CERR AFTER SCLR
5635 027774 104024 ERROR 24
5636
5637 027776 004737 035000 JSR PC,FSEC23 ;FIND SECTOR 23
5638 030002 104106 ERROR 106 ;CANNOT FIND SECTOR 23
5639 030004 000514 BR TST20 ;GO TO NEXT TEST
5640
5641 030006 012765 001476 000004 MOV #DATA01,RKBA(R5) ;DATA 0101
5642 030014 052765 000020 000010 BIS #BA1,RKCS2(R5) ;BUSS ADDR INCREMENT INHIBIT
5643 030022 012765 177000 000002 MOV #512,RKWC(R5) ;WORD COUNT
5644 030030 013765 007422 000006 MOV TEMF5,RKDA(R5) ;HEAD & SECTOR
5645 030036 012737 000023 007354 MOV #WRDATA,HCS1
5646 030044 053737 001170 007354 BIS STMP4,HCS1
5647 030052 013765 707354 000000 MOV HCS1,RKCS1(R5) ;DO WRITE DATA CMD
5648
5649 030060 012737 004000 007412 MOV #4000,TEMP1 ;2000 IS IN VER 30
5650 ;CHANGE TO 4000 7-DEC-77
5651
5652 030066 004737 033010 JSR PC,DLY ;DO DELAY
5653
5654 030072 032765 000200 000000 7$: BIT #RDY,RKCS1(R5) ;LOOK FOR CONTROLLER READY
5655 030100 001006 BNE 8$
5656 030102 004737 032320 JSR PC,FRDY ;FIND RDY AND GET FRESH STATUS
5657 030106 104011 ERROR 11 ;NO RDY AFTER SEL DRV CMD
5658 030110 004737 033664 JSR PC,GSTAT
5659 030114 104107 ERROR 107 ;HEAD SWITCHING LONGER THAN DELAY
5660
5661 030116 032737 100000 007354 8$: BIT #CERR,HCS1
5662 030124 001444 BEQ 3$
5663 030126 104012 ERROR 12 ;CERR AFTER WRITE DATA
5664
5665 030130 012737 010340 007444 MOV #<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0 ;EXPECTED MSG A0
5666 030136 005037 007446 CLR E.B0 ;EXPECTED MSG B0
5667 030142 012737 001720 007450 MOV #<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1 ;EXPECTED A1
5668 030150 012737 000001 007452 MOV #1,E.B1 ;MSG ID FOR EXPECTED MSG B1
5669 030156 005037 007454 CLR E.A2 ;EXPECTED MSG A2
5670 030162 012737 000002 007456 MOV #2,E.B2 ;MSG ID FOR EXPECTED MSG B2
5671 030170 012737 000003 007462 MOV #3,E.B3 ;MSG ID FOR EXPECTED MSG B3
5672
5673 030176 004737 033042 JSR PC,CHKMSG ;CHECK MSGS A0, B0, A1, B1
5674 030202 000000 .WORD 0!0!0 ;& MSGS SPECIFIED HERE
5675 030204 104052 ERROR 52 ;MSG A0 ERROR AFTER WRITE DATA CMD
5676 030206 104023 ERROR 23 ;MSG B0 ERROR
5677 030210 104053 ERROR 53 ;MSG A1 ERROR
5678 030212 104025 ERROR 25 ;MSG B1 ERROR
5679
5680 030214 023727 007422 000425 CMP TEMP5 #425 ;HEAD 1, SECTOR 21 DONE?
5681 030222 001405 BEQ TST20 ;GO TO NEXT TEST
5682 030224 012737 000425 007422 MOV #425,TEMP5
5683 030232 000137 027770 JMP 2$ ;ELSE REPEAT FOR HEAD 1, SECTOR 21
5684
5685 3$:
5686 ;*****
5687 ;*TEST 20 DRIVE OFF TRACK TEST

```

G09

5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723
5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743

030236	000004		
030240	012737	000001	001174
030246	012706	001100	
030252	005237	007340	
030256	005037	001346	
030262	012737	100000	007422
030270	004737	034214	
030274	104024		
030276	012700	001522	
030302	013720	001346	
030306	012720	140000	
030312	012710	140000	
030316	053720	001346	
030322	020027	001726	
030326	001365		
030330	012765	001522	000004
030336	012765	177676	030002
030344	013765	001346	000020
030352	012737	000027	007354
030360	004737	032262	
030364	104200		
030366	004737	033654	
030372	032737	100000	007354
030400	001405		
030402	104201		
030404	104401	045354	
030410	000137	031452	
030414			
030414	006137	007422	
030420	006137	001346	
030424	023737	001346	015436

```

*
* THIS TEST CHECKS FOR SERVO OSCILLATIONS DURING SETTLING TIME BEYOND
* THE ALLOTTED 3MS.
*
* 1. INITIALLY, EVERY CYLINDER IS FORMATTED WITH IDENTICAL HEADERS
*    (UNIQUE TO EACH CYLINDER)
* 2. A FULL SECTOR WRITE COMMAND IS ISSUED BY A SINGLE CYL SEEK FROM 0 TO 1.
*    AS HEADERS ARE IDENTICAL, THE NEXT SECTOR TO COME UNDER THE
*    HEADS WILL IMMEDIATELY BE WRITTEN.
* 3. IF THERE IS OSCILLATION SENSED BY READING THE TRIBITS,
*    DRIVE OFF TRACK ERROR WILL SET.
*
* IN THIS MANNER OSCILLATING SEEKS ARE PERFORMED BETWEEN ALL MAJOR CYLINDERS.
* 100 OSCILLATIONS ARE PERFORMED AT EACH MAJOR CYLINDER
* BEFORE DOING THE NEXT CYLINDER
*
*****
T20:  SCOPE
      MOV    #1,STIMES      ;DO 1 ITERATION
      MOV    #STACK,SP     ;RESTORE STK PTR
      INC    BADHDR        ;USED FOR VALID HALT
      CLR    TOCYL
      MOV    #BIT15,TEMPS
1$:   JSR    PC,SUBCLR      ;CERR AFTER SCLR
      ERROR  24
      MOV    #HDTAB,RO     ;FORMAT HEADERS ON ALL MAJOR CYL.
2$:   MOV    TOCYL,(RO)+   ;HEADER WORD 0: CYL #
      MOV    #140000,(RO)+ ;HEADER WORD 1: ALL SECTOR 0
      MOV    #140000,(RO)  ;HEADER WORD 2: XOR OF 0 & 1
      BIS    TOCYL,(RO)+   ;ADD CYL # TO WORD 2
      CMP    RO,#HDTAB+132 ;ALL 22 SECTORS DONE? (22X6=132)
      BNE    2$           ;BR IF NO
      MOI    #HDTAB,RKBA(R5)
      MOV    #-66,RKWC(R5)
      MOV    TOCYL,RKDC(R5)
      MOV    #<WRHEAD>,HCS1
      JSR    PC,DATCMD     ;DO DATA X FOR CMD & GET CONTR RDY
      ERROR  200          ;NO RDY AFTER WRITE HEADER CMD
      JSR    PC,GSTAT      ;GET FRESH STATUS
      BIT    #CERR,HCS1
      BEQ    64$
      ERROR  201          ;CERR AFTER WRITE HEADER CMD
      TYPE   MSG26        ;ABORTING BAL OF TESTS
      JMP    $EOP
64$:
      ROL    TEMPS        ;SET CARRY ONLY ONCE
      ROL    TOCYL        ;SELECT NEXT MAJOR CYL
      CMP    TOCYL,MC1    ;ALL MAJOR CYL FORMATTED?

```

5744	030432	001316			BNE	1\$		;BR IF NO
5745	030434	005065	000020		CLR	RKDC(R5)		;SETUP TO RETURN TO CYL 0
5746								
5747	030440	012737	000017	007354	MOV	#SEEK,HCS1		
5748	030446	004737	032224		JSR	PC,DOCMD		;DO SEEK CMD & GET CONTR READY
5749	030452	104131			ERROR	131		;NO RDY AFTER SEEK CMD
5750								
5751	030454	013737	001420	007412	MOV	T50000,TEMP1		;SETUP TIMEOUT
5752	030462	004737	032730		JSR	PC,FATT2		;FIND ATTN
5753	030466	104132			ERROR	132		;NO ATTN AFTER SEEK CMD
5754								
5755	030470	032737	100000	007354	BIT	#CERR,HCS1		
5756	030476	001401			BEQ	65\$		
5757	030500	104210			ERROR	210		;CERR AFTER SEEK CMD
5758								
5759	030502						65\$:	
5760								
5761	030502	005000			CLR	RO		;ITERATION COUNTER
5762	030504	012737	000001	001346	MOV	#1,TOCYL		;SETUP TO CYL #
5763	030512	005037	001344		CLR	FRCYL		
5764								
5765	030516	104415			SCOP1			
5766	030520	012706	001100		MOV	#STACK,SP		;RESTORE STK PTR
5767								
5768	030524	013737	001346	001354	MOV	TOCYL,CALDIF		;SETUP FOR ERROR PRINTOUT
5769								
5770	030532	004737	034214		JSR	PC,SUBCLR		
5771	030536	104024			ERROR	24		;CERR AFTER SCLR
5772								
5773	030540	012737	031174	001176	MOV	#FORM,\$ESCAPE		
5774	030546	013765	001346	000020	MOV	TOCYL,RKDC(R5)		;GO TO CYL #
5775	030554	012765	001500	000004	MOV	#DATA1,RKBA(R5)		;ALL 1'S
5776	030562	052765	000020	000010	BIS	#BA1,RKCS2(R5)		
5777	030570	012765	177400	000002	MOV	#-256.,RKWC(R5)		;SECTOR TO BE ALL 1'S
5778								
5779	030576	012737	000023	007354	MOV	#WRDATA,HCS1		
5780	030604	004737	032262		JSR	PC,DATCMD		;DO WRITE DATA CMD & GET CONTR RDY
5781	030610	104011			ERROR	11		;NO RDY AFTER WRITE DATA CMD.
5782								
5783	030612	004737	033664		JSR	PC,GSTAT		;GET FRESH STATUS
5784	030616	032737	020000	007404	BIT	#D.DROT,HMR3		;SEE IF DRIVE OFF TRACK
5785	030624	001401			BEQ	5\$		
5786	030626	104112			ERROR	112		;DRIVE OFF TRACK AFTER WRITE DATA CMD
5787								
5788	030630	032737	100000	007354	BIT	#CERR,HCS1		
5789	030636	001401			BEQ	6\$		
5790	030640	104012			ERROR	12		;CERR SET AFTER WRITE DATA CMD
5791								
5792	030642						6\$:	
5793								
5794	030642	012737	010340	007444	MOV	#<D!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0		;EXPECTED MSG A0
5795	030650	005037	007446		CLR	E.B0		;EXPECTED MSG B0
5796	030654	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRHM!D.SSP>,E.A1		;EXPECTED A1
5797	030662	012737	000001	007452	MOV	#1,E.B1		;MSG ID FOR EXPECTED MSG B1
5798	030670	005037	007454		CLR	E.A2		;EXPECTED MSG A2
5799	030674	012737	000002	007456	MOV	#2,E.B2		;MSG ID FOR EXPECTED MSG B2

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 112
T20 DRIVE OFF TRACK TEST

SEQ 0112

5800	030702	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5801							
5802	030710	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5803	030714	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5804	030716	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5805	030720	104023			ERROR	23	;MSH B0 ERROR
5806	030722	104053			ERROR	53	;MSG A1 ERROR
5807	030724	104025			ERROR	25	;MSG B1 ERROR
5808	030726	023737	001360	001346	CMP	CYLADD,TOCYL	
5809	030734	001401			BEQ	7\$	
5810	030736	104113			ERROR	113	;CYL ADDR IN RKMR3 NOT = RKDC
5811							
5812	030740				7\$: SCOP1		
5813	030740	104415			MOV	#STACK,SP	;RESTORE STK PTR
5814	030742	012706	001100				
5815							
5816	030746	004737	034214		JSR	PC,SUBCLR	
5817	030752	104024			ERROR	24	;CERR AFTER SCLR
5818							
5819							;RETURN TO CYL 0
5820	030754	012765	001500	000004	MOV	#DATA1,RKBA(R5)	
5821	030762	052765	000020	000010	BIS	#BA1,RKCS2(R5)	
5822	030770	012765	177400	000002	MOV	#-256.,RKWC(R5)	
5823							
5824	030776	012737	000023	007354	MOV	#WRDATA,HCS1	
5825	031004	004737	032262		JSR	PC,DATCMD	;DO WRITE DATA CMD & GET CONTR RDY
5826	031010	104011			ERROR	11	;NO RDY AFTER WRITE DATA CMD
5827							
5828	031012	004737	033664		JSR	PC,GSTAT	;GET FRESH STATUS
5829	031016	032737	020000	007404	BIT	#D.DROT,HMR3	
5830	031024	001401			BEQ	8\$	
5831	031026	104112			ERROR	112	;DRIVE OFF TRACK AFTER WRITE DATA CMD
5832							
5833	031030	032737	100000	007354	8\$: BIT	#CERR,HCS1	
5834	031036	001401			BEQ	9\$	
5835	031040	104012			ERROR	12	;CERR AFTER WRITE DATA CMD
5836							
5837	031042				9\$:		
5838							
5839	031042	012737	010340	007444	MOV	#<0!D.SPIN!D.DRDY!D.VV!D.DRA>,E.A0	;EXPECTED MSG A0
5840	031050	005037	007446		CLR	E.B0	;EXPECTED MSG B0
5841	031054	012737	001720	007450	MOV	#<D.SPOK!D.CART!D.DOOR!D.BRM!D.SSP>,E.A1	;EXPECTED A1
5842	031062	012737	000001	007452	MOV	#1,E.B1	;MSG ID FOR EXPECTED MSG B1
5843	031070	005037	007454		CLR	E.A2	;EXPECTED MSG A2
5844	031074	012737	000002	007456	MOV	#2,E.B2	;MSG ID FOR EXPECTED MSG B2
5845	031102	012737	000003	007462	MOV	#3,E.B3	;MSG ID FOR EXPECTED MSG B3
5846							
5847	031110	004737	033042		JSR	PC,CHKMSG	;CHECK MSGS A0, B0, A1, B1
5848	031114	000003			.WORD	T.A2!T.B2!0	; & MSGS SPECIFIED HERE
5849	031116	104052			ERROR	52	;MSG A0 ERROR AFTER WRITE DATA CMD
5850	031120	104023			ERROR	23	;MSH B0 ERROR
5851	031122	104053			ERROR	53	;MSG A1 ERROR
5852	031124	104025			ERROR	25	;MSG B1 ERROR
5853	031126	005737	001360		TST	CYLADD	
5854	031132	001401			BEQ	10\$	
5855	031134	104042			ERROR	42	;NOT BACK TO CYL 0

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 113
T20 DRIVE OFF TRACK TEST

SEQ 0113

5856									
5857	031136	005200			10\$:	INC	RO		
5858	031140	020027	000144			CMP	RO, #100.		; ALL ITERATIONS DONE?
5859	031144	001402				BEG	13\$		; BR IF YES
5860	031146	000137	030532			JMP	3\$		; ELSE DO AGAIN
5861									
5862	031152	005000			13\$:	CLR	RO		; RESET ITERATION CTR
5863	031154	006337	001346			ASL	TOCYL		
5864	031160	023737	001346	015436		CMP	TOCYL, MC1		; ALL MAJOR CYL DONE?
5865	031166	001402				BEG	FORM		; BR IF YES
5866	031170	000137	030532			JMP	3\$		; ELSE DO NEXT CYL
5867									
5868	031174	005037	001176		FORM:	CLR	\$ESCAPE		
5869	031200	005037	001346			CLR	TOCYL		; RESTORE TO ORIG 22 SECTOR FORMAT.
5870	031204	012737	100000	007422		MOV	#BIT15, TEMP5		
5871									
5872	031212	004737	034214		12\$:	JSR	PC, SUBCLR		
5873	031216	104024				ERROR	24		; CERR AFTER SCLR
5874									
5875	031220	012765	001522	000004		MOV	#HDTAB, RKBA(R5)		; REFORMAT ALL MAJOR CYLINDERS
5876	031226	012765	177676	000002		MOV	#-66., RKWC(R5)		
5877	031234	013765	001346	000020		MOV	TOCYL, RKDC(R5)		
5878									
5879	031242	013737	001346	001362		MOV	TOCYL, CALADD		; SETUP
5880	031250	012737	000000	001460		MOV	#0, HEAD		; TO FILL
5881	031256	012737	000000	001466		MOV	#0, FORMAT		; HEADER
5882	031264	004737	035212			JSR	PC, FHDTAB		; TABLE
5883									
5884									
5885	031270	012737	000027	007354		MOV	#<WRHEAD>, HCS1		
5886	031276	004737	032262			JSR	PC, DATCMD		; DO DATA X FOR CMD & GET CONTR RDY
5887	031302	104200				ERROR	200		; NO RDY AFTER WRITE HEADER CMD
5888	031304	004737	033664			JSR	PC, GSTAT		; GET FRESH STATUS
5889	031310	032737	100000	007354		BIT	#CERR, HCS1		
5890	031316	001405				BEG	64\$		
5891	031320	104201				ERROR	201		; CERR AFTER WRITE HEADER CMD
5892	031322	104401	045354			TYPE	MSG26		; ABORTING BAL OF TESTS
5893	031326	000137	031452			JMP	\$EOP		
5894	031332				64\$:				
5895									
5896									
5897	031332	006137	007422			ROL	TEMP5		
5898	031336	006137	001346			ROL	TOCYL		
5899	031342	023737	001346	015436		CMP	TOCYL, MC1		; ALL MAJOR CYL REFORMATTED?
5900	031350	001320				BNE	12\$		; BR IF NO
5901									
5902	031352	005065	000020			CLR	RKDC(R5)		; SETUP TO RETURN TO CYL 0
5903	031356	005037	001176			CLR	\$ESCAPE		
5904									
5905	031362	012737	000017	007354		MOV	#SEEK, HCS1		
5906	031370	004737	032224			JSR	PC, DOCMD		; DO SEEK CMD & GET CONTR READY
5907	031374	104131				ERROR	131		; NO RDY AFTER SEEK CMD
5908									
5909	031376	013737	001420	007412		MOV	T50000, TEMP1		; SETUP TIMEOUT
5910	031404	004737	032730			JSR	PC, FAT12		; FIND ATTN
5911	031410	104132				ERROR	132		; NO ATTN AFTER SEEK CMD

K09

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 114
T20 DRIVE OFF TRACK TEST

SEQ 0114

5912							
5913	031412	032737	100000	007354	BIT	#CERR,HCS1	
5914	031420	001401			BEQ	65\$	
5915	031422	104210			ERROR	210	;CERR AFTER SEEK CMD
5916							
5917	031424				65\$:		
5918							
5919							
5920	031424	005737	007342		TST	HPEND	:SEE IF HALT PENDING
5921	031430	001406			BEQ	4\$	:BR IF NO
5922	031432	005037	007342		CLR	HPEND	:CLEAR FOR FUTURE FORMATTING
5923	031436	005037	007340		CLR	BADHDR	:HEADERS NOW OK
5924	031442	000137	036262		JMP	STOP	:GO BACK & HALT CPU
5925							
5926	031446	005037	007340		4\$:	CLR	:HEADERS NOW OK
5927							

L09

```

5928 .SBTTL END OF PASS ROUTINE
5929
5930 *****
5931 ;*INCREMENT THE PASS NUMBER ($PASS)
5932 ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
5933 ;*IF THERES A MONITOR GO TO IT
5934 ;*IF THERE ISN'T JUMP TO STAPT
5935
5936 031452 $EOP:
5937
5938 031452 000004 SCOPE
5939 031454 005037 001176 CLR $ESCAPE
5940 031460 012737 000001 001174 MOV #1,$TIMES
5941 031466 012706 001100 MOV #STACK,$P
5942 031472 005237 001220 INC $DEVCT ; INCR COUNT FOR # DRIVES CHECKED
5943 031476 023737 007474 001220 CMP DRVS,$DEVCT ; ARE ALL DRIVES PRESENT TESTED?
5944 031504 001403 BEQ $EOP1+2 ; BR IF YES
5945 031506 000137 015146 JMP NUDRV ; ELSE TEST NEXT DRIVE PRESENT
5946 031512 000004 $EOP1: SCOPE
5947 031514 005037 001102 CLR $TSTNM ; ZERO THE TEST NUMBER
5948 031520 005037 001174 CLR $TIMES ; ZERO THE NUMBER OF ITERATIONS
5949 031524 005237 001216 INC $PASS ; INCREMENT THE PASS NUMBER
5950 031530 042737 100000 001216 BIC #100000,$PASS ; DON'T ALLOW A NEG. NUMBER
5951 031536 005327 DEC (PC)+ ; LOOP?
5952 031540 000001 $EOPCT: .WORD 1
5953 031542 003022 BGT $DOAGN ; YES
5954 031544 012737 MOV (PC)+,$(PC)+ ; RESTORE COUNTER
5955 031546 000001 $ENDCT: .WORD 1
5956 031550 031540 $EOPCT
5957 031552 104401 031617 TYPE $ENDMG ; TYPE "END PASS #"
5958 031556 013746 001216 MOV $PASS,-(SP) ; SAVE $PASS FOR TYPEOUT
5959 031562 104405 TYPDS ; GO TYPE--DECIMAL ASCII WITH SIGN
5960 031564 104401 031614 TYPE $ENULL ; TYPE A NULL CHARACTER
5961 031570 013700 000042 $GET42: MOV #42,R0 ; GET MONITOR ADDRESS
5962 031574 001405 BEQ $DOAGN ; BRANCH IF NO MONITOR
5963 031576 000005 RESET ; CLEAR THE WORLD
5964 031600 004710 $ENDAD: JSR PC,(R0) ; GO TO MONITOR
5965 031602 000240 NOP ; SAVE ROOM
5966 031604 000240 NOP ; FOR
5967 031606 000240 NOP ; ACT11
5968 031610 $DOAGN:
5969 031610 000137 JMP $(PC)+ ; RETURN
5970 031612 031634 $RTNAD: .WORD STAPT
5971 031614 377 377 000 $ENULL: .BYTE -1,-1,0 ; NULL CHARACTER STRING
5972 031617 015 042412 042116 $ENDMG: .ASCIZ <15><12>/END PASS #/
5973 031624 050040 051501 020123
5974 031632 000043
5975
5976 031634 122737 000001 001230 ;ADD WAITING LOOP FOR THE APT MODE
5977 031642 001007 STAPT: CMPB #APTENV,$ENV
5978 031644 023727 001216 000002 BNE 2$
5979 031652 103403 CMP $PASS,#2 ; TWO PASS DONE ?,AS REQUIRED BY
5980 031654 005237 001102 BLO 2$ ; NOT YET
5981 031660 000775 1$: INC $TSTNM ; INCREMENT THE TEST NUMBER
5982 031662 000137 013356 BR 1$ ; WAITING LOOP FOR APT TO DUMP NEXT PROG
JMP ST$ ; RETURN TO MAIN LOOP

```

SUBROUTINES

.SBTTL SUBROUTINES

;SUBROUTINE TO CLEAR ALL FLAGS FROM DDUMP THRU DOTIM

```

CLRFLG: MOV    #DDUMP, R0
          MOV    #-17, R1
1$:      CLR    (R0)+
          INC    R1
          BNE    1$
          RTS    PC

```

;TYPE PROGRAM ID IF FTITLE=0

```

TITLE:   TST     FTITLE
          BNE     1$
          INC     FTITLE
          TYPE    MSG1
.SBTTL   GET VALUE FOR SOFTWARE SWITCH REGISTER
          TST     0#42
          BNE     64$
          CMPB    $ENV, #1
          BEQ     64$
          CMP     SWR, #SWREG
          BNE     65$
          GTSWR
          BR      65$
64$:     MOVB     #1, $AUTOB
          RTS     PC
          ;PROGRAM ID
          ;ARE WE RUNNING UNDER XXDP/ACT?
          ;BRANCH IF YES
          ;ARE WE RUNNING UNDER APT?
          ;BRANCH IF YES
          ;SOFTWARE SWITCH REG SELECTED?
          ;BRANCH IF NO
          ;GET SOFT-SWR SETTINGS
          ;;SET AUTO-MODE INDICATOR

```

```

;ROUTINE TO INPUT DRIVE NOS. TYPED IN & SET
;DRIVS, DRIVO-DRIV7 REGISTERS APPROPRIATELY

```

```

GDRVS:   RDLIN
          MOV     (SP)+, R0
          MOV     #-8, R1
          MOV     (R0)+, R2
          BIC     #177400, R2
          MOV     #DRIVO, R3
          MOV     #60, R4
          ;GET STARTING ADDR OF ASCII STRING
          ;SET UP COUNT
          ;GET ASCII CHAR
          ;MASK HI BYTE
          ;DRIVE FLAG ADDR

2$:      CMP     R4, R2
          BEQ     3$
          TST     (R3)+
          INC     R4
          CMP     R4, #70
          BNE     2$
          TST     R2
          BNE     4$
          CMP     R1, #-F.
          BEQ     6$
          ;WAS TYPED CHAR 0 THRU 7?
          ;BRANCH IF YES
          ;NO, INCREMENT DR FLAG ADDR
          ;S/B 0-7 OR TERMINATOR
          ;DEFAULT ALL DRIVES

```

```

5983
5984
5985
5986
5987
5988 031666 012700 007464
5989 031672 012701 177757
5990 031676 005020
5991 031700 005201
5992 031702 001375
5993 031704 000207
5994
5995
5996
5997
5998
5999 031706 005737 001340
6000 031712 001024
6001 031714 005237 001340
6002 031720 104401 043420
6003
6004 031724 005737 000042
6005 031730 001012
6006 031732 123727 001230 000001
6007 031740 001406
6008 031742 023727 001140 000176
6009 031750 001005
6010 031752 104406
6011 031754 000403
6012 031756 112737 000001 001134
6013 031764
6014 031764 000207
6015
6016
6017
6018
6019
6020
6021 031766 104411
6022 031770 012600
6023 031772 012701 177770
6024 031776 112002
6025 032000 042702 177400
6026 032004 012703 007476
6027 032010 012704 000060
6028
6029 032014 020402
6030 032016 001415
6031 032020 005723
6032 032022 005204
6033 032024 020427 000070
6034 032030 001371
6035 032032 005702
6036 032034 001022
6037 032036 020127 177770
6038 032042 001426

```

```

6039 032044 005037 007524      7$: CLR      SIZEFLG      ;BYPASS TEST 1 (SIZING)
6040 032050 000207              RTS      PC      ;FOUND TERMINATOR, EXIT
6041
6042 032052 005213              3$: INC      DR3              ;SET UP FLAG FOR THE DRIVE
6043 032054 005237 007474      INC      DRVS      ;INCREMENT TOTAL # DRIVES TO BE TESTED
6044 032060 112002              MOV      (P0)+,R2    ;GET NEXT ASCII CHAR.
6045 032062 042702 177400      BIC      #177400,R2   ;MASK
6046 032066 022702 000054      CMP      #54,R2      ;IS IT A COMMA?
6047 032072 001407              BEQ      5$          ;YES, GO TO NEXT WORD.
6048 032074 005702              TST      R2          ;NO, IS IT A TERMINATOR?
6049 032076 001001              BNE      4$          ;IF NOT, SOMETHING WRONG.
6050 032100 000761              BR       7$          ;FOUND TERMINATOR, EXIT
6051
6052 032102 104401 046137      4$: TYPE      EMI              ;ONLY 0-7 ALLOWED.
6053 032106 000137 012562      JMP      PRGSRT      ;START ALL OVER
6054
6055 032112 005201              5$: INC      R1              ;S/B NO MORE THAN 8 DIFF
6056 032114 001330              BNE      1$          ;DRIVES TYPED IN.
6057 032116 000771              BR       4$          ;IF MORE, HAVE ERROR.
6058
6059 032120 005237 007524      6$: INC      SIZEFLG      ;DO TEST 1 (SIZING)
6060 032124 000207              RTS      PC      ;EXIT.
6061
6062      ;ROUTINE TO INPUT RKBAS OR DEFAULT.
6063      ;
6064
6065      GBA: RDOCT
6066      MOV      (SP)+,R0      ;GET LOW ORDER FROM STACK
6067 032130 012600              TST      R0
6068 032132 005700              BEQ      1$          ;BRANCH IF DEFAULT.
6069 032134 001403              MOV      R0,$BASE
6070 032136 010037 001264      RTS      PC
6071 032142 000207              1$: MOV      #177440,$BASE ;DEFAULT VALUE
6072 032144 012737 177440 001264 RTS      PC
6073 032152 000207
6074
6075      ;ROUTINE TO INPUT RKVEC OR DEFAULT
6076      ;
6077
6078      GINT: RDOCT
6079      MOV      (SP)+,R0      ;GET LOW ORDER FROM STACK
6080 032154 012600              TST      R0
6081 032156 005700              BEQ      1$          ;BRANCH IF DEFAULT
6082 032162 001405              MOV      R0,RKVEC
6083 032164 010037 001314      JSR      PC,SETINT
6084 032170 004737 032206      RTS      PC
6085 032174 000207              2$: MOV      #210,RKVEC ;DEFAULT VALUE
6086 032176 012737 000210 001314 BR       2$
6087 032204 000771
6088
6089      ;ROUTINE TO SETUP INTERRUPT VECTOR & PRIORITY
6090      ;
6091
6092      SETINT: MOV      RKVEC,R0
6093 032206 013700 001314      MOV      #INTER,(R0)+ ;INTER ADDR TO RKVEC
6094 032212 012720 036720

```

```

6095 032216 013710 001316      MOV      RKPRI,(R0)      ;PRS TO RKVEC+2
6096 032222 000207              RTS      PC
6097
6098
6099      ; THIS ROUTINE CDT IN RKCS1 IF DRIVE UNDER TEST IS AN RK07.
6100      ; ENTER WITH COMMAND IN HCS1
6101
6102 032224 053737 001170 007354 DOCMD: BIS      $TMP4,HCS1      ;SET CDT IF RK07
6103 032232 013765 007354 000000      MOV      HCS1,RKCS1(R5) ;DO COMMAND
6104 032240 013737 001406 007412      MOV      T10,TEMP1
6105 032246 004737 032320              JSR      PC,FRDY      ;FIND CONTR READY
6106 032252 000207              RTS      PC      ;SET HERE IF NOT RDY
6107 032254 062716 000002              ADD      #2,(SP)      ;ELSE SKIP OVER ERROR
6108 032260 000207              RTS      PC
6109
6110      ; THIS ROUTINE IS SIMILAR TO THE ABOVE BUT IS USED FOR DATA TRANSFERS
6111      ; & REQUIRES A LONGER TIMEOUT
6112
6113 032262 053737 001170 007354 DATCMD: BIS      $TMP4,HCS1      ;SET CDT IF RK07
6114 032270 013765 007354 000000      MOV      HCS1,RKCS1(R5) ;DO CMD
6115 032276 013737 001420 007412      MOV      T50000,TEMP1
6116 032304 004737 032320              JSR      PC,FRDY      ;FIND CONTR RDY
6117 032310 000207              RTS      PC
6118 032312 062716 000002              ADD      #2,(SP)
6119 032316 000207              RTS      PC
6120
6121      ; ROUTINE TO FIND CONTROLLER READY (RDY) DURING A DELAY
6122      ; ENTER WITH A COUNT IN TEMP1
6123      ; RETURN IF RDY NOT PRESENT (ERROR CONDITION)
6124      ; RETURN +2 IF RDY PRESENT (SKIP OVER ERROR)
6125      ; STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE
6126
6127
6128 032320 032765 000200 000000 FRDY:  BIT      #RDY,RKCS1(R5)
6129 032326 001010              BNE      1$
6130 032330 005337 007412              DEC      TEMP1
6131 032334 001371              BNE      FRDY
6132 032336 004737 032454              JSR      PC,HOLD      ;STORE ALL RK611 REGS IN HOLDING REGS.
6133 032342 004737 033602              JSR      PC,CKCERR      ;CHECK FOR SPECIAL CERR
6134 032346 000207              RTS      PC      ;NO RDY, EXIT
6135 032350 062716 000002 1$:      ADD      #2,(SP)      ;SKIP OVER ERROR
6136 032354 004737 032454              JSR      PC,HOLD
6137 032360 004737 033602              JSR      PC,CKCERR      ;CHECK FOR SPECIAL CERR
6138 032364 000207              RTS      PC
6139
6140      ; ROUTINE TO FIND CONTROLLER READY AND STORE DRIVE REGS ONLY
6141
6142 032366 032765 000200 000000 FRDY1: BIT      #RDY,RKCS1(R5)
6143 032374 001014              BNE      1$
6144 032376 005337 007412              DEC      TEMP1
6145 032402 001371              BNE      FRDY1
6146 032404 016537 000034 007402      MOV      RKMR2(R5),HMR2
6147 032412 016537 000036 007404      MOV      RKMR3(R5),HMR3
6148 032420 004737 033602              JSR      PC,CKCERR      ;CHECK FOR SPECIAL CERR CONDITIONS
6149 032424 000207              RTS      PC      ;NO RDY, EXIT
6150 032426 062716 000002 1$:      ADD      #2,(SP)      ;SKIP OVER ERROR

```

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 119
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0119

6151	032432	016537	000034	007402	MOV	RKMR2(R5),HMR2	
6152	032440	016537	000036	007404	MOV	RKMR3(R5),HMR3	
6153	032446	004737	033602		JSR	PC,CKCEPR	;CHECK FOR SPECIAL CERR CONDITIONS
6154	032452	000207			RTS	PC	
6155							
6156							
6157							
6158							
6159							
6160	032454	016537	000007	007354	HOLD: MOV	R CS1(R5),HCS1	
6161	032462	016537	000010	007356	MOV	RKCS2(R5),HCS2	
6162	032470	016537	000002	007360	MOV	RKWC(R5),HWC	
6163	032476	016537	000004	007362	MOV	RKBA(R5),HBA	
6164	032504	016537	000006	007364	MOV	RKDA(R5),HDA	
6165	032512	016537	000012	007366	MOV	RKDS(R5),HDS	
6166	032520	016537	000014	007370	MOV	RKER(R5),HER	
6167	032526	016537	000016	007372	MOV	RKASOF(R5),HASOF	
6168	032534	016537	000020	007374	MOV	RKDC(R5),HDC	
6169	032542	016537	000026	007400	MOV	RKMR1(R5),HMR1	
6170	032550	016537	000034	007402	MOV	RKMR2(R5),HMR2	
6171	032556	016537	000036	007404	MOV	RKMR3(R5),HMR3	
6172	032564	016537	000030	007406	MOV	RKECPS(R5),HPOS	
6173	032572	016537	000032	007410	MOV	RKECPT(R5),HPAT	
6174	032600	000207			RTS	PC	
6175							
6176							
6177							
6178							
6179							
6180							
6181	032602	010446			STATN: MOV	R4, -(SP)	;SAV R4
6182	032604	013704	001222		MOV	\$UNIT,R4	
6183	032610	136437	007344	007373	BITB	ATTN(R4),HASOF+1	
6184	032616	001404			BEG	1\$	;BRANCH IF ATTN NOT PRESENT
6185	032620	012604			MOV	(SP)+,R4	;RESTOR R4
6186	032622	062716	000002		ADD	#2,(SP)	;INCR RET ADDR TO JUMP OVER ERROR.
6187	032626	000207			RTS	PC	
6188	032630	012604			1\$: MOV	(SP)+,R4	;RESTOR R4
6189	032632	000207			RTS	PC	
6190							
6191							
6192							
6193							
6194							
6195							
6196							
6197							
6198							
6199	032634	010446			FATT1: MOV	R4, -(SP)	;SAV R4
6200	032636	012737	177777	007412	3\$: MOV	#-1,TEMP1	
6201	032644	013704	001222		MOV	\$UNIT,R4	
6202	032650	136465	007344	000017	1\$: BITB	ATTN(R4),RKASOF+1(R5)	;FIND CORRECT ATTN
6203	032656	001014			BNE	2\$	
6204	032660	005337	007412		DEC	TEMP1	
6205	032664	001371			BNE	1\$	
6206	032666	005337	007414		DEC	TEMP2	

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 120
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0120

D10

6207 032672 001361
6208 032674 005065 000026
6209 032700 004737 033664
6210 032704 012604
6211 032706 000207
6212 032710 005065 000026
6213 032714 004737 033664
6214 032716 012604
6215 032722 062716 000002
6216 032726 000207
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226 032730 010446
6227 032732 013704 001222
6228 032736 136465 007344 000017
6229 032744 001011
6230 032746 005337 007412
6231 032752 001367
6232 032754 005065 000026
6233 032760 004737 033664
6234 032764 012604
6235 032766 000207
6236 032770 005065 000026
6237 032774 004737 033664
6238 033000 012604
6239 033002 062716 000002
6240 033006 000207
6241
6242
6243
6244
6245
6246 033010 005737 007412
6247 033014 001403
6248 033016 005337 007412
6249 033022 000772
6250 033024 000207
6251
6252
6253
6254
6255 033026 104401 045243
6256 033032 010046
6257
6258 033034 104403
6259 033036 001
6260 033037 000
6261 033040 000207
6262

```
BNE 3$  
CLR RKMR1(R5) ;SET WORD 0  
JSR PC,GSTAT ;GET LATEST STATUS  
MOV (SP)+,R4 ;RESTOR R4  
RTS PC  
2$: CLR RKMR1(R5)  
JSR PC,GSTAT ;GET STATUS AFTER ATTN SEEN  
MOV (SP)+,R4 ;RESTOR R4  
ADD #2,(SP) ;SKIP OVER ERROR  
RTS PC  
  
;ROUTINE TO FIND ATTN WITHIN 1 SEC  
;ENTER WITH COUNT IN TEMP1  
;RETURN IF NO ATTN (ERROR)  
;RETURN +2 IF ATTN FOUND  
;STATUS IS OBTAINED BEFORE THE RETURN FOR EITHER CASE  
;  
FATT2: MOV R4,-(SP) ;SAV R4  
2$: MOV $UNIT,R4  
BITB ATTN(R4),RKASOF+1(R5) ;FIND CORRECT ATTN  
BNE 1$  
DEC TEMP1  
BNE 2$  
CLR RKMR1(R5) ;SELECT WORD 0  
JSR PC,GSTAT ;GET LATEST STATUS.  
MOV (SP)+,R4 ;RESTOR R4  
RTS PC  
1$: CLR RKMR1(R5)  
JSR PC,GSTAT  
MOV (SP)+,R4 ;RESTOR R4  
ADD #2,(SP) ;SKIP OVER ERROR  
RTS PC  
  
;ENTER WITH A COUNT IN TEMP1  
;THE DELAY IS APPROX 17 US/ITERATION + 12 US TO EXIT  
;WHEN COUNT IS 0. BASED ON AN 11/05  
DLY: TST TEMP1 ;5.6 US  
BEQ 1$ ;1.9 US  
DEC TEMP1 ;6.8 US  
BR DLY ;2.5 US  
1$: RTS PC ;3.8 US  
  
;THIS ROUTINE TYPES BYPASSED DRIVE#. ENTER WITH DRIVE# IN R0  
;  
BYP: TYPE MSG14 ;BYPASS DRIVE  
MOV R0,-(SP) ;SAVE R0 FOR TYPEOUT  
;TYPE DR#  
;GO TYPE--OCTAL ASCII  
;TYPE 1 DIGIT(S)  
;SUPPRESS LEADING ZEROS  
;TYPE 1  
;BYTE 0  
RTS PC
```

E10

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 121
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0121

; THIS ROUTINE READS ALL MSG A & B WORDS & CHECKS THEM AS REQ'D.

6263									
6264									
6265	033042	017637	000000	001520	CHKMSG: MOV	2(SP),CHKFLG		; PASS MSGS TO BE TESTED	
6266	033050	062716	000002		ADD	#2,(SP)		; BUMP RETURN ADDR TO 1ST ERROR	
6267	033054	004737	033720		JSR	PC,GSTAT1		; GET ALL ACTUAL DRIVE & CONTR STATUS	
6268									
6269	033060	053737	001222	007444	BIS	\$UNIT,E.A0		; SET UNIT #	
6270	033066	053737	001222	007450	BIS	\$UNIT,E.A1			
6271	033074	053737	001222	007454	BIS	\$UNIT,E.A2			
6272	033102	053737	001222	007460	BIS	\$UNIT,E.A3			
6273	033110	053737	015444	007444	BIS	E.DOT,E.A0		; ADD DRIVE TYPE	
6274									
6275	033116	013746	007412		MOV	TEMP1,-(SP)		; SAVE TEMP1	
6276									
6277	033122	013737	007444	007412	MOV	E.A0,TEMP1			
6278	033130	004737	036156		JSR	PC,S0PAR		; GET PARITY FOR MSG A0	
6279	033134	013737	007412	007444	MOV	TEMP1,E.A0			
6280									
6281	033142	013737	007450	007412	MOV	E.A1,TEMP1			
6282	033150	004737	036156		JSR	PC,S0PAR		; GET PARITY FOR MSG A1	
6283	033154	013737	007412	007450	MOV	TEMP1,E.A1			
6284									
6285	033162	013737	007454	007412	MOV	E.A2,TEMP1			
6286	033170	004737	036156		JSR	PC,S0PAR		; GET PARITY FOR MSG A2	
6287	033174	013737	007412	007454	MOV	TEMP1,E.A2			
6288									
6289	033202	013737	007446	007412	MOV	E.B0,TEMP1			
6290	033210	004737	036156		JSR	PC,S0PAR		; GET PARITY FOR MSG B0	
6291	033214	013737	007412	007446	MOV	TEMP1,E.B0			
6292									
6293	033222	013737	007452	007412	MOV	E.B1,TEMP1			
6294	033230	004737	036156		JSR	PC,S0PAR		; GET PARITY FOR MSG B1	
6295	033234	013737	007412	007452	MOV	TEMP1,E.B1			
6296									
6297	033242	013737	007456	007412	MOV	E.B2,TEMP1			
6298	033250	004737	036156		JSR	PC,S0PAR		; GET PARITY FOR MSG B2	
6299	033254	013737	007412	007456	MOV	TEMP1,E.B2			
6300									
6301	033262	013737	007462	007412	MOV	E.B3,TEMP1			
6302	033270	004737	036156		JSR	PC,S0PAR		; GET PARITY FOR MSG B3	
6303	033274	013737	007412	007462	MOV	TEMP1,E.B3			
6304									
6305	033302	012637	007412		MOV	(SP)+,TEMP1		; RESTORE TEMP1	
6306	033306	013737	001176	001172	MOV	\$ESCAPE,\$TMP5		; SAVE ESCAPE	
6307									
6308	033314	023737	007424	007444	CMP	H.A0,E.A0		; TEST MSG A0	
6309	033322	001411			BEQ	2\$		; BR IF OK	
6310	033324	012737	033336	001176	MOV	#1\$,\$ESCAPE		; ELSE SETUP ESCAPE	
6311	033332	011646			MOV	(SP),-(SP)		; COPY RET ADDR.	
6312	033334	000207			RTS	PC		; & RETURN TO MAINLINE ERROR	
6313									
6314	033336	032777	001000	145574	1\$: BIT	#SW9,\$SWR		; RET HERE FROM MAINLINE ERROR	
6315	033344	001107			BNE	20\$		; & BR IF LOOP ON ERROR	
6316	033346	062716	000002		2\$: ADD	#2,(SP)		; BUMP RET ADDR TO NEXT ERROR	
6317									
6318	033352	023737	007426	007446	CMP	H.B0,E.B0		; TEST MSG B0	

CZR61EO UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 122
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0122

```

6319 033360 001411      BEQ      5$      ;BR IF OK
6320 033362 012737 033374 001176      MOV      4$, $ESCAPE ;ELSE SETUP ESCAPE
6321 033370 011646      MOV      (SP), -(SP) ;COPY RET ADDR
6322 033372 000207      RTS      PC      ;& RETURN TO MAINLINE ERROR
6323
6324 033374 032777 001000 145536 4$:      BIT      #SW9, $SWR      ;RETURN HERE FROM MAINLINE ERROR
6325 033402 001070      BNE      20$      ;& BR IF LOOP ON ERROR
6326 033404 062716 000002      5$:      ADD      #2, (SP)      ;BUMP RET ADDR TO NEXT ERROR
6327
6328 033410 023737 007430 007450      CMP      H.A1, E.A1      ;TEST MSG A1
6329 033416 001411      BEQ      8$      ;BR IF OK
6330 033420 012737 033432 001176      MOV      #7$, $ESCAPE
6331 033426 011646      MOV      (SP), -(SP)
6332 033430 000207      RTS      PC
6333
6334 033432 032777 001000 145500 7$:      BIT      #SW9, $SWR
6335 033440 001051      BNE      20$
6336 033442 062716 000002      8$:      ADD      #2, (SP)
6337
6338 033446 023737 007432 007452      CMP      H.B1, E.B1      ;TEST MSG B1
6339 033454 001411      BEQ      11$      ;BR IF OK
6340 033456 012737 033470 001176      MOV      #10$, $ESCAPE
6341 033464 011646      MOV      (SP), -(SP)
6342 033466 000207      RTS      PC
6343
6344 033470 032777 001000 145442 10$:      BIT      #SW9, $SWR
6345 033476 001032      BNE      20$
6346 033500 062716 000002      11$:      ADD      #2, (SP)
6347
6348 033504 032737 000001 001520 12$:      BIT      #T.A2, CHKFLG      ;TEST MSG A2?
6349 033512 001402      BEQ      13$      ;BR IF NO
6350 033514 004737 034604      JSR      PC, RCYLD      ;PUT INFO CYLDIF, DO NOT CHECK
6351 033520 032737 000002 001520 13$:      BIT      #T.B2, CHKFLG      ;TEST MSG B2?
6352 033526 001402      BEQ      14$      ;BR IF NO
6353 033530 004737 034656      JSR      PC, RCYLA      ;PUT INFO IN CYLADD. DO NOT CHECK
6354
6355 033534 032737 000004 001520 14$:      BIT      #T.B3, CHKFLG      ;TEST MSG B3?
6356 033542 001404      BEQ      15$
6357 033544 004737 034714      JSR      PC, RSEC      ;PUT INFO IN SECTOR, DO NOT CHECK
6358 033550 004737 034752      JSR      PC, RHEAD      ;PUT INFO IN HEADA. DO NOT CHECK
6359
6360 033554 013737 001172 001176 15$:      MOV      $TMP5, $ESCAPE ;RESTORE ESCAPE
6361 033562 000207      RTS      PC
6362
6363 033564 012706 001100      20$:      MOV      #STACK, SP      ;RESET STACK PTR
6364 033570 013737 001172 001176      MOV      $TMP5, $ESCAPE ;RESTORE ESCAPE
6365 033576 000177 145306      JMP      $SLPEAR
6366
6367      ; THIS ROUTINE CHECKS FOR CERTAIN ERROR CONDITIONS ONLY
6368      ; I.E.: IF NED, CTO OR MDS SET MESSAGE A & B ARE INVALID
6369
6370 033602 005737 001516      CKCERR: TST      BYPCERR
6371 033606 001025      BNE      4$
6372 033610 032737 100000 007354      BIT      #CERR, HCS1
6373 033616 001001      BNE      1$      ;BR IF CERR
6374 033620 000207      RTS      PC

```

```

6375
6376 033622 032737 004000 007354 1$: BIT #CTO,HCS1
6377 033630 001402 BEQ 2$ ;BR IF NOT CTO
6378 033632 104125 ERROR 125 ;CTO ERROR, MSG A & B INVALID
6379 033634 000207 RTS PC
6380
6381 033636 032737 010000 007356 2$: BIT #NED,HCS2
6382 033644 001401 BEQ 3$ ;BR IF NOT NED
6383 033646 104126 ERROR 126 ;NED ERROR, MSG A & B INVALID
6384
6385 033650 032737 001000 007356 3$: BIT #MDS,HCS2
6386 033656 001401 BEQ 4$
6387 033660 104127 ERROR 127 ;MDS ERROR, MSG A & B INVALID
6388
6389 033662 000207 4$: RTS PC
6390
6391 ;
6392 ; THIS ROUTINE DOES THE SELECT DRIVE COMMAND TO GET STATUS
6393 ; IT THEN WAITS FOR CONTROLLER READY
6394 ; IF RDY NOT RECEIVED BY THE TIMEOUT, AN ERROR IS FLAGGED
6395 ;
6396
6397 033664 013746 007412 GSTAT: MOV TEMP1,-(SP) ;SAVE TEMP1
6398 033670 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;CURRENT DRIVE #
6399 033676 012737 000001 007354 MOV #SELDV,HCS1
6400 033704 004737 032224 JSR PC,DCCMD ;DO SELDRV (STATUS) CMD & GET CONTR RDY
6401 033710 104117 ERROR 117 ;RDY NOT SET BY END OF SELECT DRIVE CMD
6402 033712 012637 007412 MOV (SP)+,TEMP1 ;RESTOR TEMP1.
6403 033716 000207 RTS PC
6404
6405 ;
6406 ; THIS ROUTINE GETS STATUS OF ALL DRIVE REGISTERS (MSG A0-A3, B0-B3)
6407 ; & ALL CONTROLLER REGISTERS.
6408
6408 033720 013746 007412 GSTAT1: MOV TEMP1,-(SP) ;SAVE TEMP1
6409 033724 004737 032454 JSR PC,HOLD ;GET ALL CONTR REG
6410 033730 012765 100000 000000 MOV #CCLR,RKCS1(R5) ;CLEAR CONTR
6411 033736 013765 001222 000010 MOV $UNIT,RKCS2(R5) ;CURRENT DRIVE #
6412 033744 012765 000003 000026 MOV #3,RKMR1(R5) ;SELECT WORD 3
6413 033752 004737 034150 JSR PC,GSTAT2
6414 033756 104117 ERROR 117 ;RDY NOT SET BY END OF SELECT DRV CMD
6415 033760 013737 007402 007440 MOV HMR2,H.A3 ;STORE MSG A3
6416 033766 013737 007404 007442 MOV HMR3,H.B3 ;STORE MSG B3
6417
6418 033774 012765 100000 000000 MOV #CCLR,RKCS1(R5)
6419 034002 013765 001222 000010 MOV $UNIT,RKCS2(R5)
6420 034010 012765 000002 000026 MOV #2,RKMR1(R5) ;SELECT WORD 2
6421 034016 004737 034150 JSR PC,GSTAT2
6422 034022 104117 ERROR 117 ;RDY NOT SET BY END OF SELECT DRV CMD
6423 034024 013737 007402 007434 MOV HMR2,H.A2 ;STORE MSG A2
6424 034032 013737 007404 007436 MOV HMR3,H.B2 ;STORE MSG B2
6425
6426 034040 012765 100000 000000 MOV #CCLR,RKCS1(R5)
6427 034046 013765 001222 000010 MOV $UNIT,RKCS2(R5)
6428 034054 012765 000001 000026 MOV #1,RKMR1(R5) ;SELECT WORD 1
6429 034062 004737 034150 JSR PC,GSTAT2
6430 034066 104117 ERROR 117 ;RDY NOT SET BY END OF SELECT DRV CMD

```

H10

CZR61ED UNIBUS RK6 DR PRT2
CZR61E P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 124
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0124

```

6431 034070 013737 007402 007430      MOV      HMR2,H.A1      ;STORE MSG A1
6432 034076 013737 007404 007432      MOV      HMR3,H.B1      ;STORE MSG B1
6433
6434 034104 012765 100000 000000      MOV      #CCLR,RKCS1(R5)
6435 034112 013765 001222 000010      MOV      $UNIT,RKCS2(R5)
6436 034120 004737 034150      JSR      PC,GSTAT2
6437 034124 104117      ERROR      117      ;RDY NOT SET BY END OF SEL DRV CMD
6438 034126 013737 007402 007424      MOV      HMR2,H.A0      ;STORE MSG A0
6439 034134 013737 007404 007426      MOV      HMR3,H.B0      ;STORE MSG B0
6440
6441 034142 012637 007412      MOV      (SP)+,TEMP1      ;RESTORE TEMP1
6442 034146 000207      RTS      PC
6443
6444 034150 012737 000001 007354  GSTAT2: MOV      #SELDLV,HCS1
6445 034156 053737 001170 007354      BIS      $TMP4,HCS1      ;RET CDT IF RK07
6446 034164 013765 007354 000000      MOV      HCS1,RKCS1(R5) ;GET STATUS
6447 034172 013737 001406 007412      MOV      T10,TEMP1
6448 034200 004737 032366      JSR      PC,FRDY1      ;FIND CONTR RDY & STORE DRIVE REGS ONLY
6449 034204 000207      RTS      PC      ;RET HERE IF NOT RDY
6450 034206 062716 000002      ADD      #2,(SP)      ;RET HERE IF OK
6451 034212 000207      RTS      PC
6452
6453
6454 ; THIS ROUTINE DOES A SUBSYSTEM CLEAR & WAITS FOR CONTROLLER READY
6455 ; IF RDY IS NOT RECEIVED BY THE END OF THE TIMEOUT, AN ERROR IS FLAGGED.
6456 ; THE ROUTINE THEN GETS CURRENT STATUS & CHECKS FOR CONTROLLER ERROR (CERR)
6457 ; RETURN IF CERR SET
6458 ; RETURN +2 IF CERR CLEAR
6459
6460 034214 012765 000040 000010  SUBCLR: MOV      #SCLR,RKCS2(R5) ;SUBSYS CLEAR
6461 034222 013737 001406 007412      MOV      T10,TEMP1
6462 034230 004737 032320      JSR      PC,FRDY      ;FIND RDY
6463 034234 104120      ERROR      120      ;RDY NOT SET BY END OF SCLR
6464 034236 013765 001222 000010      MOV      $UNIT,RKCS2(R5) ;CURRENT DRIVE #
6465 034244 005065 000026      CLR      RKMR1(R5)      ;SELECT WORD 0
6466 034250 004737 033664      JSR      PC,GSTAT      ;GET STATUS
6467 034254 032737 100000 007354      BIT      #CERR,HCS1      ;CHECK FOR CONT ERROR
6468 034262 001401      BEQ      1$
6469 034264 000207      RTS      PC
6470 034266 062716 000002 1$:      ADD      #2,(SP)      ;SKIP OVER ERROR
6471 034272 000207      RTS      PC
6472
6473 ; READ THE SECTOR COUNT IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
6474
6475 RDSEC: MOV      #3,RKMR1(R5) ;WORD 3
6476 034274 012765 000003 000026      JSR      PC,GSTAT
6477 034302 004737 033664      MOV      HMR3,SECTOR
6478 034306 013737 007404 001402      BIC      #C<M,SECT>,SECTOR
6479 034314 042737 177017 001402      ASR      SECTOR      ;RIGHT JUSTIFY
6480 034322 006237 001402      ASR      SECTOR      ;SECTOR
6481 034326 006237 001402      ASR      SECTOR      ;INFO
6482 034332 006237 001402      ASR      SECTOR
6483 034336 006237 001402      ASR      SECTOR
6484 034342 000207      RTS      PC
6485
6486 ; READ THE CYL DIFF/OFFSET IN RKMR2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'

```

```

6487
6488 034344 012765 000002 000026 RDCYLD: MOV #2,RKMR1(R5) ;WORD 2
6489 034352 004737 033664 JSR PC,GSTAT
6490 034356 013737 007402 001356 MOV HMR2,CYLDIF
6491 034364 043737 015442 001356 BIC MASK1,CYLDIF
6492 034372 006237 001356 ASR CYLDIF ;RIGHT JUSTIFY
6493 034376 006237 001356 ASR CYLDIF ;CYL DIFF/OFFSET
6494 034402 006237 001356 ASR CYLDIF ;INFO
6495 034406 006237 001356 ASR CYLDIF
6496 034412 023737 001356 015440 CMP CYLDIF,MASK ;CHK TO SEE IF RET IN COMPL. FORM
6497 034420 001002 BNE 1$ ;BR IF NOT
6498 034422 005037 001356 CLR CYLDIF ;CLR IF YES
6499 034426 000207 1$: RTS PC
6500
6501 ;
6502 ;QUICK SELECT DRIVE COMMAND TO OBTAIN CYL DIFF
6503
6504 034430 013746 007412 QKCYLD: MOV TEMP1,-(SP) ;SAVE TEMP1
6505 034434 012765 000002 000026 MOV #2,RKMR1(R5) ;SELECT WORD 2
6506 034442 012737 000001 007354 MOV #SELDRV,HCS1 ;SELECT DRIVE CMD
6507 034450 053737 001170 007354 BIS STMP4,HCS1
6508 034456 013765 007354 000000 MOV HCS1,RKCS1(R5)
6509 034464 013737 001406 007412 MOV T10,TEMP1
6510 034472 032765 000200 000000 1$: BIT #RDY,RKCS1(R5) ;TEST FOR CONT RDY
6511 034500 001004 BNE 2$ ;BR IF THERE
6512 034502 005337 007412 DEC TEMP1
6513 034506 001371 BNE 1$
6514 034510 104117 ERROR 117 ;NO RDY AFTER SEL DRV CMD
6515
6516 034512 016537 000034 001356 2$: MOV RKMR2(R5),CYLDIF
6517 034520 043737 015442 001356 BIC MASK1,CYLDIF ;GET CYL DIFF ONLY (NO SHIFTING)
6518 034526 012637 007412 MOV (SP)+,TEMP1 ;RESTORE TEMP1
6519 034532 000207 RTS PC
6520
6521 ;READ THE CYL ADDR IN RKMR3, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
6522
6523 034534 012765 000002 000026 RDCYLA: MOV #2,RKMR1(R5) ;WORD 2
6524 034542 004737 033664 JSR PC,GSTAT
6525 034546 013737 007404 001360 MOV HMR3,CYLADD
6526 034554 043737 015442 001360 BIC MASK1,CYLADD
6527 034562 006237 001360 ASR CYLADD ;RIGHT JUSTIFY
6528 034566 006237 001360 ASR CYLADD ;CYL ADDR
6529 034572 006237 001360 ASR CYLADD ;INFO
6530 034576 006237 001360 ASR CYLADD
6531 034602 000207 RTS PC
6532
6533 ;READ THE CYL DIFF/OFFSET IN H.A2, RIGHT JUSTIFY IT & STORE IT IN 'CYLDIF'
6534
6535 034604 013737 007434 001356 RDCYLD: MOV H.A2,CYLDIF
6536 034612 043737 015442 001356 BIC MASK1,CYLDIF ;CLEAR UNWANTED INFO
6537 034620 006237 001356 ASR CYLDIF ;RIGHT JUSTIFY
6538 034624 006237 001356 ASR CYLDIF
6539 034630 006237 001356 ASR CYLDIF
6540 034634 006237 001356 ASR CYLDIF
6541 034640 023737 001356 015440 CMP CYLDIF,MASK ;CHK TO SEE IF RET IN COMPL. FORM
6542 034646 001002 BNE 1$ ;BR IF NO

```

```

6543 034650 005037 001356
6544 034654 000207
6545
6546
6547
6548 034656 013737 007436 001360
6549 034664 043737 015442 001360
6550 034672 006237 001360
6551 034676 006237 001360
6552 034702 006237 001360
6553 034706 006237 001360
6554 034712 000207
6555
6556
6557
6558 034714 013737 007442 001402
6559 034722 042737 177017 001402
6560 034730 006237 001402
6561 034734 006237 001402
6562 034740 006237 001402
6563 034744 006237 001402
6564 034750 000207
6565
6566
6567
6568 034752 013737 007442 001462
6569 034760 042737 170777 001462
6570 034766 006237 001462
6571 034772 000337 001462
6572 034776 000207
6573
6574
6575
6576
6577
6578 035000 013737 001416 007412
6579 035006 004737 034274
6580 035012 023727 001402 000023
6581 035020 001014
6582
6583 035022 004737 034274
6584 035026 023727 001402 000023
6585 035034 001412
6586 035036 004737 034274
6587 035042 023727 001402 000023
6588 035050 001404
6589
6590 035052 005337 007412
6591 035056 001353
6592 035060 000207
6593
6594 035062 062716 000004
6595 035066 000207
6596
6597
6598

      CLR      CYLDIF      ;ELSE CLEAR
1$:      RTS      PC
      ;READ THE CYL ADDR IN H.B2, RIGHT JUSTIFY IT & STORE IT IN 'CYLADD'
      RCYLA:  MOV      H.B2,CYLADD
      BIC      MASK1,CYLADD ;CLEAR UNWANTED INFO
      ASR      CYLADD      ;RIGHT JUSTIFY
      ASR      CYLADD
      ASR      CYLADD
      ASR      CYLADD
      RTS      PC
      ;READ THE SECTOR COUNT IN H.B3, RIGHT JUSTIFY IT & STORE IT IN 'SECTOR'
      RSEC:  MOV      H.B3,SECTOR
      BIC      #1<M.SECT>,SECTOR ;CLEAR UNWANTED INFO
      ASR      SECTOR      ;RIGHT JUSTIFY
      ASR      SECTOR
      ASR      SECTOR
      ASR      SECTOR
      RTS      PC
      ;READ THE HEAD ADDR IN H.B3, RIGHT IT & STORE IT IN 'HEADA'
      RHEAD: MOV      H.B3,HEADA
      BIC      #1<M.HEAD>,HEADA ;CLEAR UNWANTED INFO
      ASR      HEADA      ;RIGHT JUSTIFY IT
      SWAB     HEADA
      RTS      PC
      ;FIND SECTOR 23
      ;RETURN IF NOT FOUND
      ;RETURN +4 IF FOUND
      FSEC23: MOV      T5000,TEMP1 ;SETUP TIMEOUT
1$:      JSR      PC,RDSEC ;READ SECTOR
      CMP      SECTOR,#23 ;TEST FOR SECTOR 23(8)
      BNE      2$ ;BR IF NOT 23(8)
      JSR      PC,RDSEC
      CMP      SECTOR,#23
      BEQ      3$ ;BR IF READ SAME TWICE
      JSR      PC,RDSEC ;ELSE TRY 1 MORE TIME
      CMP      SECTOR,#23
      BEQ      3$ ;BR IF 23(8)
2$:      DEC      TEMP1
      BNE      1$ ;TRY AGAIN
      RTS      PC
3$:      ADD      #4,(SP) ;SKIP OVER ERROR
      RTS      PC
      ;ROUTINE TO FIND HEADS HOME IN RKMR2 WORD 1 BEFORE SPECIFIED DELAY
      ;ENTER WITH TIME IN SECONDS IN TEMP2

```

```

6599      ;RETURN IF NOT FOUND
6600      ;RETURN+2 IF FOUND - SKIP OVER ERROR
6601
6602      035070 012737 177777 007412 FHDHM: MOV     #1,TEMP1      ;ALL 1'S
6603      035076 012765 000001 000026      MOV     #1,RKMR1(R5) ;WORD 1
6604      035104 004737 033664      JSR     PC,GSTAT
6605      035110 032737 000040 007402      BIT     #D.FDHM,HMR2
6606      035116 001007      BNE     2$
6607      035120 005337 007412      DEC     TEMP1
6608      035124 001367      BNE     1$
6609      035126 005337 007414      DEC     TEMP2
6610      035132 001356      BNE     FHDHM
6611      035134 000207      RTS     PC
6612      035136 062716 000002      2$:  ADD     #2,(SP)      ;SKIP OVER ERROR
6613      035142 000207      RTS     PC
6614
6615      ;ROUTINE TO FIND LOAD HEADS IN RKMR2 WORD 1 BEFORE THE TIMEOUT
6616      ;RETURN IF NOT FOUND
6617      ;RETURN+2 IF FOUND: SKIP OVER ERROR
6618
6619      035144 012737 000372 007412 FLOAD: MOV     #250,TEMP1
6620      035152 012765 000001 000026      MOV     #1,RKMR1(R5) ;WORD 1
6621      035160 004737 033664      JSR     PC,GSTAT
6622      035164 032737 010000 007402      BIT     #D.LOAD,HMR2
6623      035172 001004      BNE     2$
6624      035174 005337 007412      DEC     TEMP1
6625      035200 001367      BNE     1$
6626      035202 000207      RTS     PC
6627      035204 062716 000002      2$:  ADD     #2,(SP)      ;SKIP OVER ERROR
6628      035210 000207      RTS     PC
6629
6630      ;FILL HEADER TABLE WITH 66 WORDS OF VALID HEADERS
6631      ;ENTER WITH CYL # IN 'CALADD'
6632      ;ENTER WITH HEAD # IN 'HEAD'
6633      ;ENTER WITH FORMAT IN 'FORMAT'
6634
6635      035212 010046      FHD TAB: MOV     R0,-(SP)      ;SAV R0
6636      035214 010146      MOV     R1,-(SP)      ;SAV R1
6637      035216 012700 001522      MOV     #HDTAB,R0      ;HEADER WORD TABLE ADDR
6638      035222 005001      CLR     R1      ;SECTOR COUNTER
6639      035224 013737 001460 001464      MOV     HEAD,HD1
6640      035232 006337 001464      ASL     HD1
6641      035236 006337 001464      ASL     HD1
6642      035242 006337 001464      ASL     HD1
6643      035246 006337 001464      ASL     HD1
6644      035252 006337 001464      ASL     HD1      ;SETUP HEAD # FOR WORD 2 OF HEADER
6645      035256 013737 001466 001470      MOV     FORMAT,FMT1
6646      035264 000337 001470      SWAB    FMT1
6647      035270 006337 001470      ASL     FMT1      ;SETUP FORMAT FOR WORD 2 OF HEADER
6648
6649      035274 013720 001362      1$:  MOV     CALADD,(R0)+ ;HEADER WORD 1-CYL ADDR
6650      035300 010110      MOV     R1,(R0)      ;HEADER WORD 2-SECTOR NO
6651      035302 053710 001464      BIS     HD1,(R0)      ;-HEAD NO
6652      035306 053710 001470      BIS     FMT1,(R0)      ;-FORMAT
6653      035312 004737 035372      JSR     PC,SECFLG
6654

```

```

6655 035316 013737 001362 007412 MOV CALADD,TEMP1
6656 035324 011037 007414 MOV (R0),TEMP2
6657 035330 043737 001362 007414 BIC CALADD,TEMP2
6658 035336 042037 007412 BIC (R0)+,TEMP1
6659 035342 053737 007412 007414 BIS TEMP1,TEMP2
6660 035350 013720 007414 MOV TEMP2,(R0)+ ;HEADER WORD 3-HEADER CHECK
6661
6662 035354 005201 INC R1 ;SECTOR CTR
6663 035356 020127 000026 CMP R1,#22. ;ALL 22 SECTORS DONE? (66 WORDS)
6664 035362 001344 BNE 1$ ;BR IF NO
6665
6666 035364 012601 MOV (SP)+,R1 ;RESTOR R1
6667 035366 012600 MOV (SP)+,R0 ;RESTOR R0
6668 035370 000207 RTS PC
6669
6670 ; THIS ROUTINE GETS INFORMATION FROM THE BAD SECTOR TABLE FILLED BY A PREVIOUS
6671 ; TEST & SETS BITS 14 & 15 APPROPRIATLY.
6672
6673 035372 010246 SECFLG: MOV R2,-(SP) ;SAVE R2
6674 035374 005737 001466 TST FORMAT
6675 035400 001016 BNE 1$ ;BR IF 20 SECTOR FORMAT
6676 035402 012702 003346 MOV #BSE22H+8.,R2
6677 035406 004737 035472 JSR PC,FLGTST ;GET HARDWARE DETECTED FLAG
6678 035412 052710 100000 BIS #BIT15,(R0) ;RETURN HERE IF GOOD SECTOR
6679
6680 035416 012702 005346 MOV #BSE22S+8.,R2 ;ELSE RETURN HERE
6681 035422 004737 035472 JSR PC,FLGTST ;GET SOFTWARE DETECTED FLAG
6682 035426 052710 040000 BIS #BIT14,(R0) ;RETURN HERE IF GOOD SECTOR
6683
6684 035432 012602 MOV (SP)+,R2 ;ELSE RETURN HERE
6685 035434 000207 RTS PC
6686
6687 035436 012702 002346 1$: MOV #BSE20H+8.,R2
6688 035442 004737 035472 JSR PC,FLGTST ;GET HARDWARE DETECTED FLAG
6689 035446 052710 100000 BIS #BIT15,(R0) ;RETURN HERE IF GOOD SECTOR
6690
6691 035452 012702 004346 MOV #BSE20S+8.,R2
6692 035456 004737 035472 JSR PC,FLGTST ;GET SOFTWARE DETECTED FLAG
6693 035462 052710 040000 BIS #BIT14,(R0) ;RETURN HERE IF GOOD SECTOR
6694
6695 035466 012602 MOV (SP)+,R2 ;RESTORE R2
6696 035470 000207 RTS PC
6697
6698 ; THIS ROUTINE DOES THE ACTUAL SCANNING OF THE BAD SECTOR TABLES
6699 ; ENTER WITH THE ADDRESS OF TABLE (BSE22H, BSE22S, ETC.) IN TEMP1
6700 ; RETURN IF NO COMPARE
6701 ; RETURN+4 IF COMPARE
6702
6703 FLGTST: MOV R3,-(SP) ;SAVE R3
6704 035472 010346 1$: CMP (R2),#-1 ;SEE IF ALL 1'S
6705 035474 021227 177777 BNE 2$ ;BR IF NO
6706 035500 001002 MOV (SP)+,R3 ;RESTORE R3
6707 035502 012603 RTS PC
6708 035504 000207
6709
6710

```

6711	035506	022237	001362	2\$:	CMP	(R2)+,CALADD	;SEE IF=CYL # & ADR PTR TO TRK/SECTOR WORD
6712	035512	001403			BEG	3\$	
6713	035514	062702	000002		ADD	#2,R2	;GO TO NEXT CYL WORD IN TABLE
6714	035520	000765			BR	1\$	
6715							
6716	035522	013703	001460	3\$:	MOV	HEAD,R3	;GET HEAD # FROM FHDTAB ROUTINE
6717	035526	000303			SWAB	R3	
6718	035530	050103			BIS	R1,PC	;ADD SECTOR # FROM FHDTAB ROUTINE
6719	035532	022203			CMP	(R2)+,R3	;SEE IF SECTOR/HEAD COMPARE
6720							; & INCR PTR TO NEXT CYL WORD
6721	035534	001401			BEG	4\$	;BR IF COMPARE
6722	035536	000756			BR	1\$	;ELSE TRY NEXT CYL
6723							
6724	035540	012603		4\$:	MOV	(SP)+,R3	;RESTORE R3
6725	035542	062716	000004		ADD	#4,(SP)	;INCREMENT RET ADDR
6726	035546	000207			RTS	PC	
6727							
6728							
6729							
6730							
6731							
6732	035550	010046					
6733	035552	010146					
6734	035554	004737	034274				
6735	035560	062737	000001	001402			
6736	035566	004737	035656				
6737							
6738	035572	012700	000204				
6739	035576	163700	001402				
6740	035602	010037	001402				
6741	035606	062737	001726	001402			
6742							
6743	035614	062700	001726				
6744	035620	012701	002132				
6745							
6746	035624	012021					
6747	035626	020027	002132				
6748	035632	001374					
6749							
6750	035634	012700	001726				
6751	035640	012021					
6752	035642	020037	001402				
6753	035646	001374					
6754							
6755	035650	012601					
6756	035652	012600					
6757	035654	000207					
6758							
6759							
6760							
6761							
6762	035656	006337	001402				
6763	035662	013746	001402				
6764	035666	006337	001402				
6765	035672	062637	001402				
6766	035676	000207					

```

; THIS ROUTINE SORTS THE RHTAB TABLE FROM WHATEVER SECTOR IT BEGINS
; WITH AND RE-WITES THE INFO IN SRTTAB TABLE TO BEGIN WITH SECTOR 0
SORT:  MOV  RO, -(SP)      ;SAVE RO
        MOV  R1, -(SP)    ;SAVE R1
        JSR  PC,ROSEC
        ADD  #1,SECTOR
        JSR  PC,MULT6      ;MULT SECTOR BY 6

        MOV  #132,RO
        SUB  SECTOR,RO     ;RO-SECTOR TO RO = INDEX
        MOV  RO,SECTOR
        ADD  #RHTAB,SECTOR ;SAVE INDEX

        ADD  #RHTAB,RO     ;INDEX TO BOT HALF OF RHTAB
        MOV  #SRTTAB,R1    ;INDEX TO TOP HALF OF SRTTAB

1$:     MOV  (RO)+,(R1)+    ;PUT BOTTOM OF RHTAB TO TOP OF SRTTAB
        CMP  RO,#RHTAB+132.
        BNE  1$

2$:     MOV  #RHTAB,RO
        MOV  (RO)+,(R1)+    ;PUT TOP OF RHTAB TO BOT OF SRTTAB
        CMP  RO,SECTOR
        BNE  2$

        MOV  (SP)+,R1      ;RESTOR R1
        MOV  (SP)+,RO      ;RESTOR RO
        RTS  PC

;MULT BY 6. ENTER WITH DESIRED # IN 'SECTOR'
MULT6:  ASL  SECTOR        ;2 X SECTOR
        MOV  SECTOR,-(SP)
        ASL  SECTOR        ;4 X SECTOR
        ADD  (SP)+,SECTOR  ;(4 X 5)+(2 X 5) = 6 X SECTOR
        RTS  PC

```

N10

```

6767
6768
6769
6770
6771
6772
6773
6774
6775
6776 035700 010446
6777
6778 035702 032737 010000 007354
6779 035710 001014
6780
6781 035712 012704 003346
6782 035716 004737 036000
6783 035722 000422
6784
6785 035724 012704 005346
6786 035730 004737 036000
6787 035734 000415
6788
6789 035736 012604
6790 035740 000207
6791
6792 035742 012704 002346
6793 035746 004737 036000
6794 035752 000406
6795
6796 035754 012704 004346
6797 035760 004737 036000
6798 035764 000401
6799 035766 000763
6800
6801 035770 012604
6802 035772 062716 000002
6803 035776 000207
6804

; THIS ROUTINE IS ENTERED ONLY IF THERE IS A BSE ERROR AFTER A WRITE DATA
; CMD. IT VERIFIES THAT THE BAD SECTOR IS LISTED IN THE BSE INFORMATION
; CYLINDER AT CYL 410, TRACK 2.
; RETURN IF SECTOR NOT LISTED IN BSE TABLE, ERROR CONDITION.
; RETURN+2 IF LISTED, SKIP OVER ERROR
TRUERR: MOV R4, -(SP) ;SAVE R4
        BIT #CFMT, HCS1 ;CHECK FORMAT
        BNE 2$ ;BR FOR 20 SECTOR FORMAT
        MOV #BSE22H+B., R4
        JSR PC, TERR1 ;SEE IF ON HARDWARE DETECTED TABLE
        BR 3$ ;RETURN HERE IF YES
        MOV #BSE22S+B., R4 ;ELSE RETURN HERE
        JSR PC, TERR1 ;& SEE IF ON SOFTWARE DETECTED TABLE
        BR 3$ ;RETURN HERE IF YES
1$: MOV (SP)+, R4 ;RESTORE R4
   RTS PC ;RETURN WITHOUT JUMPING OVER ERROR
2$: MOV #BSE20H+B., R4
   JSR PC, TERR1 ;SEE IF ON HARDWARE DETECTED TABLE
   BR 3$ ;RETURN HERE IF YES
        MOV #BSE20S+B., R4 ;ELSE RETURN HERE
        JSR PC, TERR1 ;SEE IF ON SOFTWARE DETECTED TABLE
        BR 3$ ;RETURN HERE IF YES
        BR 1$ ;RETURN HERE IF NO
3$: MOV (SP)+, R4 ;RESTORE R4
   ADD #2, (SP) ;SKIP OVER ERROR ON RETURN
   RTS PC

```

B11

CZR61EO UNIBJS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 131
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0131

6805
6806
6807
6808
6809
6810
6811 036000 021427 177777

; THIS ROUTINE DOES THE ACTUAL COMPARING OF CYLINDER, HEAD & TRACK AGAINST
; THE BSE TABLE FOR THE ABOVE SUBROUTINE.
; RETURN IF FOUND ON TABLE
; RETURN+2 IF NOT FOUND
;ERR1: CMP (R4),#-1 ;SEE IF ALL 1'S

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 132
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0132

6812	036004	001405		BEQ	1\$		;BR IF YES, NOT ON TABLE
6813	036006	022437	007374	CMP	(R4)+,HDC		;SEE IF CYL MATCH
6814	036012	001405		BEQ	2\$		;BR IF YES
6815	036014	005724		TST	(R4)+		;ELSE ADV TO NEXT CYL WORD
6816	036016	000770		BR	TERR1		; & TRY AGAIN.
6817							
6818	036020	062716	000002	1\$: ADD	#2,(SP)		
6819	036024	000207		RTS	PC		
6820							
6821	036026	022437	007364	2\$: CMP	(R4)+,HDA		;SEE IF SECTOR & TRACK MATCH
6822	036032	001401		BEQ	3\$		;BR IF YES
6823	036034	000761		BR	TERR1		;OR TRY AGAIN
6824							
6825	036036	000207		3\$: RTS	PC		
6826							
6827							
6828							
6829	036040	005037	001372	CLKON: CLR	TIMUP		
6830	036044	005737	007520	TST	PCLKF		

;ROUTINE TO TURN L OR P CLOCK INTERRUPT ON

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 133
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0133

```

6831 036050 001004          RNE      1$          ;BRANCH IF P-CLOCK PRESENT
6832 036052 012777 000100 143246 MOV     #100,ALKS      ;L-CLOCK, ENABLE INT
6833 036060 000207          RTS      PC
6834 036062 012777 177777 143232 1$: MOV     #-1,APKSB      ;P-CLOCK, ALL 1'S
6835 036070 012777 000135 143222 MOV     #135,APKS      ;ENABLE INT, CT UP, REP INT
6836 036076 000207          RTS      PC          ;LINE FREQ & RUN
6837
6838          ;KW11-L & KW11-P INTERRUPT HANDLER
6839
6840 036100 005037 001372  CLOCK: CLR      TIMUP
6841 036104 005337 001366          DEC     COUNT
6842 036110 001010          BNE      1$
6843 036112 013737 001364 001366 MOV     HZ,COUNT
6844 036120 005337 001370          DEC     SEC
6845 036124 001002          BNE      1$
6846 036126 005237 001372          INC     TIMUP      ;SORRY, TIME IS UP
6847 036132 000002 1$:      RTI
6848
6849          ;ROUTINE TO TURN L OR P CLOCK INTERRUPT OFF
6850
6851 036134 005737 007520  CLKOF: TST      PCLKF
6852 036140 001003          BNE      1$          ;BRACH IF P-CLOCK PRESENT
6853 036142 005077 143160          CLR     ALKS      ;L-CLOCK, CLEAR INTERRUPT
6854 036146 000207          RTS      PC
6855 036150 005077 143144 1$:      CLR     APKS      ;P-CLOCK, CLEAR INTERRUPT
6856 036154 000207          RTS      PC
6857
6858          ;THIS ROUTINE GENERATES PARITY FOR THE EXPECTED MESSAGE
6859          ;ENTER WITH THE EXPECTED WORD IN TEMP1
6860          ;TEMP1 IS ROTATED LEFT 17 TIMES. EACH TIME THE CARRY BIT IS SET,
6861          ;R1 IS INCREMENTED. AT THE END OF 17 ROTATES (TEMP1 BACK TO ORIG),
6862          ;R1 BIT 0 IS EXAMINED. IF IT IS SET, INDICATING AN ODD # OF 1'S,
6863          ;THE PARITY BIT IS NOT SET IN B
6864          ;IF IT IS NOT SET, INDICATING AN EVEN # OF 1'S ,THE PARITY BIT IS
6865          ;SET IN TEMP1
6866
6867          SBPAR: MOV     R0,-(SP)      ;SAVE R0
6868 036156 010046          MOV     R1,-(SP)      ;SAVE R1
6869 036160 010146          MOV     #17,R0      ;SHIFT COUNTER
6870 036162 012700 000021          CLR     R1      ;COUNT # OF 1'S IN TEMP1
6871 036166 005001          CLC          ;CLEAR CARRY
6872 036170 000241
6873
6874 036172 006137 007412 1$:      ROL     TEMP1
6875 036176 103001          BCC     2$          ;BR IF CARRY CLEAR
6876 036200 005201          INC     R1      ;COUNT # OF 1'S
6877 036202 005300 2$:      DEC     R0      ;SHIFT COUNTER
6878 036204 001372          BNE      1$
6879
6880 036206 032701 000001          BIT     #BIT0,R1
6881 036212 001003          BNE      3$          ;BR IF ODD # IN R0
6882 036214 052737 100000 007412 BIS     #M.PAR,TEMP1      ;SET PARITY BIT
6883 036222 012601 3$:      MOV     (SP)+,R1      ;RESTORE R1
6884 036224 012600          MOV     (SP)+,R0      ;RESTORE R0
6885 036226 000207          RTS      PC
6886

```

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 134
GET VALUE FOR SOFTWARE SWITCH REGISTER

SEQ 0134

```

6887
6888
6889
6890
6891
6892 036230 032777 001000 142702 SCOP1$: BIT      #SW9,2SWR      ;LOOP ON ERROR?
6893 036236 001406          BEQ      1$          ;BR IF NO
6894 036240 105737 001103          TSTB     $ERFLG      ;HAD ERROR?
6895 036244 001403          BEQ      1$          ;BR IF NO
6896 036246 013716 001110          MOV      $LPERR,(SP)
6897 036252 000002          RTI
6898
6899 036254 011637 001110          1$:      MOV      (SP),$LPERR      ;SET LOOP ADDR FOR TIGHT SCOPE LOOP
6900 036260 000002          RTI
6901
6902
6903
6904
6905
6906
6907 036262 022626          STOP:    CMP      (SP)+,(SP)+      ;RESTORE STACK FROM INTERRUPT
6908
6909 036264 004737 034214          JSR      PC,SUBCLR      ;CERR AFTER
6910 036270 104024          ERROR    24
6911
6912 036272 005737 0017336          TST      UNLD      ;SEE IF HEADS UNLOADED
6913 036276 001431          BEQ      3$          ;BR IF NO
6914 036300 005737 000042          TST      42      ;SEE IF MANUAL OR AUTO MODE
6915 036304 001403          BEQ      1$          ;BR IF MANUAL MODE
6916 036306 104401 045775          TYPE     MSG74      ;PGM ABORT PENDING
6917 036312 000402          BR       2$
6918 036314 104401 046043          1$:      TYPE     ,MSG75      ;HALT PENDING
6919 036320
6920
6921 036320 004737 034214          JSR      PC,SUBCLR      ;CERR AFTER SCLR
6922 036324 104024          ERROR    24
6923
6924 036326 012737 000011 007354          MOV      #SRTSPL,HCS1
6925 036334 004737 032224          JSR      PC,DOCMD      ;DO START SPINDLE CMD & GET CONTR RDY
6926 036340 104121          ERROR    121      ;RDY NOT SET AFTER ST SPIN CMD.
6927
6928 036342 013737 001414 007414          MOV      T100,TEMP2      ;SETUP TIMEOUT
6929 036350 004737 032634          JSR      PC,FATT1      ;FIND ATTN
6930 036354 104074          ERROR    74      ;NO ATTN AFTER ST SPIN CMD.
6931
6932 036356 005037 007336          CLR      UNLD
6933
6934
6935 036362 005737 007340          3$:      TST      BADHDR      ;SEE IF HEADERS VALID
6936 036366 001460          BEQ      4$          ;BR IF YES
6937 036370 005237 007342          INC      HPEND
6938
6939 036374 012765 100000 000000          MOV      #CCLR,RKCS1(R5)
6940 036402 013765 001222 000010          MOV      $UNIT,RKCS2(R5)
6941 036410 012737 000013 007354          MOV      #RECAL,HCS1
6942 036416 004737 032224          JSR      PC,DOCMD      ;DO RECAL CMD & GET CONTR RDY

```

Address	Op Code	Op 1	Op 2	Op 3	Op 4	Op 5	Op 6	Op 7	Op 8	Op 9	Op 10	Op 11	Op 12	Op 13	Op 14	Op 15	Op 16	Op 17	Op 18	Op 19	Op 20	Op 21	Op 22	Op 23	Op 24	Op 25	Op 26	Op 27	Op 28	Op 29	Op 30	Op 31	Op 32	Op 33	Op 34	Op 35	Op 36	Op 37	Op 38	Op 39	Op 40	Op 41	Op 42	Op 43	Op 44	Op 45	Op 46	Op 47	Op 48	Op 49	Op 50	Op 51	Op 52	Op 53	Op 54	Op 55	Op 56	Op 57	Op 58	Op 59	Op 60	Op 61	Op 62	Op 63	Op 64	Op 65	Op 66	Op 67	Op 68	Op 69	Op 70	Op 71	Op 72	Op 73	Op 74	Op 75	Op 76	Op 77	Op 78	Op 79	Op 80	Op 81	Op 82	Op 83	Op 84	Op 85	Op 86	Op 87	Op 88	Op 89	Op 90	Op 91	Op 92	Op 93	Op 94	Op 95	Op 96	Op 97	Op 98	Op 99	Op 100	Op 101	Op 102	Op 103	Op 104	Op 105	Op 106	Op 107	Op 108	Op 109	Op 110	Op 111	Op 112	Op 113	Op 114	Op 115	Op 116	Op 117	Op 118	Op 119	Op 120	Op 121	Op 122	Op 123	Op 124	Op 125	Op 126	Op 127	Op 128	Op 129	Op 130	Op 131	Op 132	Op 133	Op 134	Op 135	Op 136	Op 137	Op 138	Op 139	Op 140	Op 141	Op 142	Op 143	Op 144	Op 145	Op 146	Op 147	Op 148	Op 149	Op 150	Op 151	Op 152	Op 153	Op 154	Op 155	Op 156	Op 157	Op 158	Op 159	Op 160	Op 161	Op 162	Op 163	Op 164	Op 165	Op 166	Op 167	Op 168	Op 169	Op 170	Op 171	Op 172	Op 173	Op 174	Op 175	Op 176	Op 177	Op 178	Op 179	Op 180	Op 181	Op 182	Op 183	Op 184	Op 185	Op 186	Op 187	Op 188	Op 189	Op 190	Op 191	Op 192	Op 193	Op 194	Op 195	Op 196	Op 197	Op 198	Op 199	Op 200	Op 201	Op 202	Op 203	Op 204	Op 205	Op 206	Op 207	Op 208	Op 209	Op 210	Op 211	Op 212	Op 213	Op 214	Op 215	Op 216	Op 217	Op 218	Op 219	Op 220	Op 221	Op 222	Op 223	Op 224	Op 225	Op 226	Op 227	Op 228	Op 229	Op 230	Op 231	Op 232	Op 233	Op 234	Op 235	Op 236	Op 237	Op 238	Op 239	Op 240	Op 241	Op 242	Op 243	Op 244	Op 245	Op 246	Op 247	Op 248	Op 249	Op 250	Op 251	Op 252	Op 253	Op 254	Op 255	Op 256	Op 257	Op 258	Op 259	Op 260	Op 261	Op 262	Op 263	Op 264	Op 265	Op 266	Op 267	Op 268	Op 269	Op 270	Op 271	Op 272	Op 273	Op 274	Op 275	Op 276	Op 277	Op 278	Op 279	Op 280	Op 281	Op 282	Op 283	Op 284	Op 285	Op 286	Op 287	Op 288	Op 289	Op 290	Op 291	Op 292	Op 293	Op 294	Op 295	Op 296	Op 297	Op 298	Op 299	Op 300	Op 301	Op 302	Op 303	Op 304	Op 305	Op 306	Op 307	Op 308	Op 309	Op 310	Op 311	Op 312	Op 313	Op 314	Op 315	Op 316	Op 317	Op 318	Op 319	Op 320	Op 321	Op 322	Op 323	Op 324	Op 325	Op 326	Op 327	Op 328	Op 329	Op 330	Op 331	Op 332	Op 333	Op 334	Op 335	Op 336	Op 337	Op 338	Op 339	Op 340	Op 341	Op 342	Op 343	Op 344	Op 345	Op 346	Op 347	Op 348	Op 349	Op 350	Op 351	Op 352	Op 353	Op 354	Op 355	Op 356	Op 357	Op 358	Op 359	Op 360	Op 361	Op 362	Op 363	Op 364	Op 365	Op 366	Op 367	Op 368	Op 369	Op 370	Op 371	Op 372	Op 373	Op 374	Op 375	Op 376	Op 377	Op 378	Op 379	Op 380	Op 381	Op 382	Op 383	Op 384	Op 385	Op 386	Op 387	Op 388	Op 389	Op 390	Op 391	Op 392	Op 393	Op 394	Op 395	Op 396	Op 397	Op 398	Op 399	Op 400	Op 401	Op 402	Op 403	Op 404	Op 405	Op 406	Op 407	Op 408	Op 409	Op 410	Op 411	Op 412	Op 413	Op 414	Op 415	Op 416	Op 417	Op 418
---------	---------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

```

6999 036616 032777 001000 142314 1$: BIT #SW9,2SWR ;LOOP ON ERROR?
7000 036624 001403 BEQ 2$ ;NO BRANCH
7001 036626 022626 CMP (SP)+,(SP)+ ;YES RESTORE STACK
7002 036630 000177 142252 JMP 2$LPADR ;GO TO STARTING ADDR OF TEST
7003 ;THAT GAVE BAD TIMEOUT
7004 036634 032777 040000 142276 2$: BIT #SW14,2SWR ;LOOP ON TEST?
7005 036642 001401 BEQ 3$ ;NO BRANCH
7006 036644 000002 RTI ;YES
7007
7008 036646 000000 3$: HALT ;UNEXPECTED TIME OUT OCCURRED
7009 ;AS INDICATED. YOU CAN LOOP ON
7010 ;ERROR, LOOP ON TEST OR INHIBIT
7011 ;ERROR TIMEOUT BY SETTING THOSE
7012 ;SWITCHES.
7013
7014 036650 022626 CMP (SP)+,(SP)+ ;RESTORE STACK
7015 036652 000137 031512 JMP $EOP1 ;ABORT TESTS
7016
7017 .SBTTL MEMORY CHECK ENABLE TRAP
7018
7019 036656 012737 036672 001176 MEMERR: MOV #15,$ESCAPE
7020 036664 011637 001334 MOV (SP),TRAPPC ;STORE PC
7021 036670 104041 ERROR 41 ;UNEXP MEM PARITY TRAP
7022 036672 005037 001176 1$: CLR $ESCAPE
7023 036676 032777 001000 142234 BIT #SW9,2SWR ;CHECK IF LOOP ON ERROR
7024 036704 001001 BNE 2$ ;YES, FORCE STACK AND TRY AGAIN
7025 036706 000002 RTI ;ELSE RETURN
7026
7027 036710 012706 001100 2$: MOV #STACK,SP ;INIT STACK
7028 036714 000177 142170 JMP 2$LPERR ;LOOP ON ERROR
7029
7030 .SBTTL RK06 INTERRUPT HANDLER
7031
7032 INTER: NOP
7033 036720 000240 NOP
7034 036722 000240 NOP
7035 036724 000240 NOP
7036 036726 011600 MOV (SP),RO ;SAVE PC WHERE INT OCCURRED.
7037 036730 005740 TST (RO) ;GET PC BEFORE UPDATE.
7038 036732 104401 044550 TYPE MSG6 ;INT AT PC=
7039 036736 010046 MOV RO,-(SP) ;SAVE RO FOR TYPEOUT
7040 ;TYPE PC
7041 036740 104403 TYPOS ;GO TYPE--OCTAL ASCII
7042 036742 006 .BYTE 6 ;TYPE 6 DIGIT(S)
7043 036743 000 .BYTE 0 ;SUPPRESS LEADING ZEROS
7044 036744 000000 HALT
7045 036746 000240 NOP
7046 036750 000240 NOP
7047 036752 000002 RTI
7048
7049 .SBTTL POWER DOWN AND UP ROUTINES
7050
7051 ;POWER DOWN ROUTINE
7052
7053 036754 012737 036766 000024 $PWRDN: MOV #PWRUP,PWRVEC ;SET UP VECTOR
7054 036762 000000 HALT

```

```

7055 036764 000776          BR      .-2          ;HANG UP.
7056
7057          ;POWER UP ROUTINE
7058
7059 036766 005037 037040    $PWRUP: CLR      $PWRCT      ;WAIT LOOP FOR TTY
7060 036772 005237 037040    1$: INC      $PWRCT      ;WAIT FOR THE INCR
7061 036776 001375          BNE      1$              ;OF WORD
7062 037000 012737 036754 000024    MOV      $PWRDN,PWRVEC ;SET POWER DOWN VECTOR
7063 037006 012737 000340 000026    MOV      #PR7,PWAVEC+2 ;PRIORITY 7
7064 037014 012737 000340 000036    MOV      #PR7,TRAPVEC+2 ;LOCKOUT ALL INTERRUPTS FOR TRAPS
7065 037022 012706 001100          MOV      #STACK,SP    ;INITIALIZE STACK
7066 037026 104401 044740          TYPE      ,MSG11      ;REPORT POWER FAIL
7067 037032 000005          RESET
7068 037034 000137 015446          JMP      PFSRT
7069
7070 037040 000000    $PWRCT: 0          ;WAIT COUNT FOR TTY
7071
7072          ;DIVISION UTILITY ROUTINE
7073
7074          ;R0-R1-R2-R3=DIVIDEND,
7075          ;R4-R5=DIVISOR
7076          ;R0-R1=REMAINDER AFTER DIVISION
7077          ;R2-R3=QUOTIENT AFTER DIVISION
7078          ;ENTER WITH JSR PC,M.DPID
7079
7080
7081 037042 012746 000040    M.DPID: MOV      #40,-(SP)      ;COUNTER FOR DIVISION CYCLES
7082 037046 010446          MOV      R4,-(SP)      ;HI ORDER
7083 037050 010546          MOV      R5,-(SP)      ;LO ORDER TO THE STACK
7084 037052 005466 000002          NEG      2(SP)      ;FORM NEGATIVE
7085 037056 005416          NEG      @SP          ;VERSION OF DIVISOR
7086 037060 005666 000002          SBC      2(SP)
7087 037064 061601          ADD      @SP,R1
7088 037066 005500          ADC      R0          ;PERFORM INIT SUBT.
7089 037070 066600 000002          ADD      2(SP),R0
7090 037074 103445          BCS      M.DP50      ;IF CARRY THEN OVERFLOW HAS OCCURRED
7091 037076 005046          CLR      -(SP)      ;THIS IS A LONGER LASTING CARRY BIT
7092 037100 006103          M.DP40: ROL      R3
7093 037102 006102          ROL      R2
7094 037104 006101          ROL      R1
7095 037106 006100          ROL      R0
7096 037110 005716          TST      @SP          ;TEST CARRY INDICATOR
7097 037112 001410          BEQ      M.DP41      ;IF TO CARRY THEN ADD, ELSE SUBT.
7098 037114 005016          CLR      @SP          ;CLEAR UP FOR NEXT TIME
7099 037116 066601 000002          ADD      2(SP),R1
7100 037122 005500          ADC      R0          ;ADD -(DIVISOR)
7101 037124 005516          ADC      @SP          ;SET CARRY
7102 037126 066600 000004          ADD      4(SP),R0
7103 037132 000404          BR      M.DP42
7104
7105 037134 060501          M.DP41: ADD      R5,R1          ;ADD +(DIVISOR)
7106 037136 005500          ADC      R0          ;SET CARRY
7107 037140 005516          ADC      @SP
7108 037142 060400          ADD      R4,R0
7109 037144 005516          M.DP42: ADC      @SP          ;SET CARRY
7110 037146 005716          TST      @SP          ;TEST THE UPDATE INDICATOR

```

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 138
POWER DOWN AND UP ROUTINES

I11

SEQ 0138

7111	037150	001401		BEG	.+4		; IF 0 FORGET IT
7112	037152	005203		INC	R3		; NO CARRY POSSIBLE HERE
7113	037154	005366	000006	DEC	6(SP)		; DECREMENT CTR
7114	037160	003347		BGT	M.DP40		; BR IF MORE TO DO
7115	037162	006003		ROR	R3		
7116	037164	103404		BCS	M.DP44		
7117	037166	060501		ADD	R5,R1		
7118	037170	005500		ADC	R0		
7119	037172	060400		ADD	R4,R0		
7120	037174	000241		CLC			
7121							
7122	037176	006103		M.DP44: ROL	R3		
7123	037200	062706	000010	ADD	#10,SP		; ADJUST STACK BY 4 WORDS
7124	037204	000242		CLV			
7125	037206	000207		RTS	PC		
7126							
7127	037210	062706	000006	M.DP50: ADD	#6,SP		
7128	037214	000262		SEV			
7129	037216	000207		RTS	PC		
7130							

```
7131 .SBTTL SCOPE HANDLER ROUTINE
7132
7133 ;*****
7134 ;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
7135 ;AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
7136 ;AND LOAD THE EROR FLAG ($ERFLG) INTO DISPLAY<15:08>
7137 ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7138 ;*SW14=1 LOOP ON TEST
7139 ;*SW11=1 INHIBIT ITERATIONS
7140 ;*SW09=1 LOOP ON ERROR
7141 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
7142 ;*CALL
7143 ;* SCOPE ;;SCOPE=IOT
7144
7145 $SCOPE:
7146 037220 104407 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
7147 037222 032777 040000 141710 1$: BIT #BIT14,$SWR ;;LOOP ON PRESENT TEST?
7148 037230 001114 $OVER ;;YES IF SW14=1
7149 ;*****START OF CODE FOR THE XOR TESTER*****
7150 037232 000416 $XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
7151 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
7152 037234 013746 000004 MOV $#ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
7153 037240 012737 037260 000004 MOV $S,$#ERRVEC ;;SET FOR TIMEOUT
7154 037246 005737 177060 TST $#177060 ;;TIME OUT ON XOR?
7155 037252 012637 000004 MOV (SP)+,$#ERRVEC ;;RESTORE THE ERROR VECTOR
7156 037256 000463 BR $SVLAD ;;GO TO THE NEXT TEST
7157 037260 022626 5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
7158 037262 012637 000004 MOV (SP)+,$#ERRVEC ;;RESTORE THE ERROR VECTOR
7159 037266 000423 BR 7$ ;;LOOP ON THE PRESENT TEST
7160 037270 6$: ;*****END OF CODE FOR THE XOR TESTER*****
7161 037270 032777 000400 141642 BIT #BIT08,$SWR ;;LOOP ON SPEC. TEST?
7162 037276 001404 BEQ 2$ ;;BR IF NO
7163 037300 127737 141634 001102 CMPB $SWR,$STNM ;;ON THE RIGHT TEST? SWR<7:0>
7164 037306 001465 BEQ $OVER ;;BR IF YES
7165 037310 105737 001103 2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
7166 037314 001421 BEQ 3$ ;;BR IF NO
7167 037316 123737 001115 001103 CMPB $ERMAX,$ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
7168 037324 101015 BHI 3$ ;;BR IF NO
7169 037326 032777 001000 141604 BIT #BIT09,$SWR ;;LOOP ON ERROR?
7170 037334 001404 BEQ 4$ ;;BR IF NO
7171 037336 013737 001110 001106 7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
7172 037344 000446 BR $OVER
7173 037346 105037 001103 4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG
7174 037352 005037 001174 CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
7175 037356 000415 BR 1$ ;;ESCAPE TO THE NEXT TEST
7176 037360 032777 004000 141552 3$: BIT #BIT11,$SWR ;;INHIBIT ITERATIONS?
7177 037366 001011 BNE 1$ ;;BR IF YES
7178 037370 005737 001216 TST $PASS ;;IF FIRST PASS OF PROGRAM
7179 037374 001406 BEQ 1$ ;;INHIBIT ITERATIONS
7180 037376 005237 001104 INC $ICNT ;;INCREMENT ITERATION COUNT
7181 037402 023737 001174 001104 CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
7182 037410 002024 BGE $OVER ;;BR IF MORE ITERATION REQUIRED
7183 037412 012737 000001 001104 1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
7184 037420 013737 037426 001174 MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
7185 037426 105237 001102 $SVLAD: INCB $STNM ;;COUNT TEST NUMBERS
7186 037432 113737 001102 001214 MOVB $STNM,$TESTN ;;SET TEST NUMBER IN APT MAILBOX
```

```

7187 037440 011637 001106      MOV      (SP), $LPADR      ;; SAVE SCOPE LOOP ADDRESS
7188 037444 011637 001110      MOV      (SP), $LPERR      ;; SAVE ERROR LOOP ADDRESS
7189 037450 005037 001176      CLR      $ESCAPE      ;; CLEAR THE ESCAPE FROM ERROR ADDRESS
7190 037454 112737 000001 001115  MOV      #1, $ERMAX      ;; ONLY ALLOW ONE(1) ERROR ON NEXT TEST
7191 037462 013777 001102 141452 $OVER:  MOV      $STNM, $DISPLAY  ;; DISPLAY TEST NUMBER
7192 037470 013716 001106      MOV      $LPADR, (SP)      ;; FUDGE RETURN ADDRESS
7193 037474 000002      RTI      ;; FIXES PS
7194 037476 003720      $MXCNT: 2000      ;; MAX. NUMBER OF ITERATIONS
7195      $SBTTL  ERROR HANDLER ROUTINE
7196
7197      ;*****
7198      ;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
7199      ;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
7200      ;AND GO TO TYPERR ON ERROR
7201      ;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
7202      ;$SW15=1      HALT ON ERROR
7203      ;$SW13=1      INHIBIT ERROR TYPEOUTS
7204      ;$SW10=1      BELL ON ERROR
7205      ;$SW09=1      LOOP ON ERROR
7206      ;CALL
7207      ;*      ERROR      N      ;; ERROR=EMT AND N=ERROR ITEM NUMBER
7208
7209 037500      $ERROR:
7210 037500 104407      CKSWR      ;; TEST FOR CHANGE IN SOFT-SWR
7211 037502 105237 001103 7$:      INCB      $ERFLG      ;; SET THE ERROR FLAG
7212 037506 001775      BEQ      7$      ;; DON'T LET THE FLAG GO TO ZERO
7213 037510 013777 001102 141424  MOV      $STNM, $DISPLAY  ;; DISPLAY TEST NUMBER AND ERROR FLAG
7214 037516 032777 002000 141414  BIT      #BIT10, $SWR      ;; BELL ON ERROR?
7215 037524 001402      BEQ      1$      ;; NO - SKIP
7216 037526 104401 001200      TYPE      $BELL      ;; RING BELL
7217 037532 005237 001112 1$:      INC      $ERTTL      ;; COUNT THE NUMBER OF ERRORS
7218 037536 011637 001116      MOV      (SP), $ERRPC      ;; GET ADDRESS OF ERROR INSTRUCTION
7219 037542 162737 000002 001116  SUB      #2, $ERRPC
7220 037550 117737 141342 001114  MOV      $ERRPC, $ITEMB      ;; STRIP AND SAVE THE ERROR ITEM CODE
7221 037556 032777 020000 141354  BIT      #BIT13, $SWR      ;; SKIP TYPEOUT IF SET
7222 037564 001004      BNE      20$      ;; SKIP TYPEOUTS
7223 037566 004737 055140      JSR      PC, TYPERR      ;; GO TO USER ERROR ROUTINE
7224 037572 104401 001205      TYPE      $CRLF
7225 037576
7226 037576 122737 000001 001230 20$:  CMPB      #APTENV, $ENV      ;; RUNNING IN APT MODE
7227 037604 001007      BNE      2$      ;; NO SKIP APT ERROR REPORT
7228 037606 113737 001114 037620  MOV      $ITEMB, 21$      ;; SET ITEM NUMBER AS ERROR NUMBER
7229 037614 004737 040424      JSR      PC, $ATY4      ;; REPORT FATAL ERROR TO APT
7230 037620 000      21$:      .BYTE      0
7231 037621 000      .BYTE      0
7232 037622 000777      BR      22$      ;; APT ERROR LOOP
7233 037624 005777 141310 2$:      TST      $SWR      ;; HALT ON ERROR
7234 037630 100002      BPL      3$      ;; SKIP IF CONTINUE
7235 037632 000000      HALT      ;; HALT ON ERROR!
7236 037634 104407      CKSWR      ;; TEST FOR CHANGE IN SOFT-SWR
7237 037636 032777 001000 141274 3$:  BIT      #BIT09, $SWR      ;; LOOP ON ERROR SWITCH SET?
7238 037644 001402      BEQ      4$      ;; BR IF NO
7239 037646 013716 001110      MOV      $LPERR, (SP)      ;; FUDGE RETURN FOR LOOPING
7240 037652 005737 001176 4$:      TST      $ESCAPE      ;; CHECK FOR AN ESCAPE ADDRESS
7241 037656 001402      BEQ      3$      ;; BR IF NONE
7242 037660 013716 001176      MOV      $ESCAPE, (SP)      ;; FUDGE RETURN ADDRESS FOR ESCAPE

```

```

7243 037664 022737 031600 000042 5$: CMP #SENDAD,2#42 ;;ACT-11 AUTO-ACCEPT?
7244 037664 001001 031600 000042 5$: BNE 6$ ;;BRANCH IF NO
7245 037672 001001 031600 000042 5$: HALT ;;YES
7246 037674 000000 031600 000042 5$:
7247 037676 000002 031600 000042 6$: RTI ;;RETURN
7248 037676 000002 031600 000042 6$:
7249 .SBTTL TYPE ROUTINE
7250
7251 *****
7252 ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
7253 ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
7254 ;NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
7255 ;NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
7256 ;NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
7257 ;
7258 ;CALL:
7259 ;1) USING A TRAP INSTRUCTION
7260 ; TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
7261 ;OR
7262 ; TYPE
7263 ; MESADR
7264 ;
7265
7266 037700 105737 001157 $TYPE: TSTB $TPFLG ;; IS THERE A TERMINAL?
7267 037704 100002 001157 1$ BPL 1$ ;; BR IF YES
7268 037706 000000 001157 1$ HALT ;; HALT HERE IF NO TERMINAL
7269 037710 000430 001157 1$ BR 3$ ;; LEAVE
7270 037712 010046 001157 1$ MOV RO, -(SP) ;; SAVE RO
7271 037714 017600 000002 001230 1$ MOV @2(SP), RO ;; GET ADDRESS OF ASCIZ STRING
7272 037720 122737 000001 001230 1$ CMPB #APTENV, $ENV ;; RUNNING IN APT MODE
7273 037726 001011 001230 1$ BNE 62$ ;; NO GO CHECK FOR APT CONSOLE
7274 037730 132737 000100 001231 1$ BITB #APTPOOL, $ENVM ;; SPOOL MESSAGE TO APT
7275 037736 001405 001231 1$ BEQ 62$ ;; NO GO CHECK FOR CONSOLE
7276 037740 010037 037750 1$ MOV RO, 61$ ;; SETUP MESSAGE ADDRESS FOR APT
7277 037744 004737 040414 1$ JSR PC, $ATY3 ;; SPOOL MESSAGE TO APT
7278 037750 000000 001231 61$: .WORD 0 ;; MESSAGE ADDRESS
7279 037752 132737 000040 001231 62$: BITB #APTCUP, $ENVM ;; APT CONSOLE SUPPRESSED
7280 037760 001003 001231 62$: BNE 60$ ;; YES, SKIP TYPE OUT
7281 037762 112046 001231 2$: MOVB (RO)+, -(SP) ;; PUSH CHARACTER TO BE TYPED ONTO STACK
7282 037764 001005 001231 2$: BNE 4$ ;; BR IF IT ISN'T THE TERMINATOR
7283 037766 005726 001231 2$: TST (SP)+ ;; IF TERMINATOR POP IT OFF THE STACK
7284 037770 012600 001231 2$: MOV (SP)+, RO ;; RESTORE RO
7285 037772 062716 000002 3$: ADD #2, (SP) ;; ADJUST RETURN PC
7286 037776 000002 001231 3$: RTI ;; RETURN
7287 040000 122716 000011 4$: CMPB #HT, (SP) ;; BRANCH IF <HT>
7288 040004 001430 000011 4$: BEQ 8$
7289 040006 122716 000200 4$: CMPB #CRLF, (SP) ;; BRANCH IF NOT <CRLF>
7290 040012 001006 000200 4$: BNE 5$
7291 040014 005726 000200 4$: TST (SP)+ ;; POP <CR><LF> EQUIV
7292 040016 104401 000200 4$: TYPE ;; TYPE A CR AND LF
7293 040020 001205 000200 4$: $CRLF
7294 040022 105037 040156 4$: CLRB $CHARCNT ;; CLEAR CHARACTER COUNT
7295 040026 000755 040156 4$: BR 2$ ;; GET NEXT CHARACTER
7296 040030 004737 040112 5$: JSR PC, $TYPEC ;; GO TYPE THIS CHARACTER
7297 040034 123726 001156 6$: CMPB $FILLC, (SP)+ ;; IS IT TIME FOR FILLER CHARS.?
7298 040040 001350 001156 6$: BNE 2$ ;; IF NO GO GET NEXT CHAR.

```

```
7299 040042 013746 001154      MOV      $NULL,-(SP)      ;; GET # OF FILLER CHARS. NEEDED
7300                                ;; AND THE NULL CHAR.
7301 040046 105366 000001      7$:  DECB      1(SP)      ;; DOES A NULL NEED TO BE TYPED?
7302 040052 002770 040112      BLT       6$      ;; BR IF NO--GO POP THE NULL OFF OF STACK
7303 040054 004737 040112      JSR       PC,$TYPEC      ;; GO TYPE A NULL
7304 040060 105337 040156      DECB      $CHARCNT      ;; DO NOT COUNT AS A COUNT
7305 040064 000770              BR       7$      ;; LOOP
7306
7307      ;HORIZONTAL TAB PROCESSOR
7308
7309 040066 112716 000040      8$:  MOVB      #' ,(SP)      ;; REPLACE TAB WITH SPACE
7310 040072 004737 040112      9$:  JSR       PC,$TYPEC      ;; TYPE A SPACE
7311 040076 132737 000007 040156  BITB      #',$CHARCNT      ;; BRANCH IF NOT AT
7312 040104 001372              BNE      9$      ;; TAB STOP
7313 040106 005726              TST      (SP)+      ;; POP SPACE OFF STACK
7314 040110 000724              BR       2$      ;; GET NEXT CHARACTER
7315 040112 105777 141032      $TYPEC: TSTB      2$TPS      ;; WAIT UNTIL PRINTER IS READY
7316 040116 100375              BPL      $TYPEC      ;;
7317 040120 116677 000002 141024  MOVB      2(SP),2$TPB      ;; LOAD CHAR TO BE TYPED INTO DATA REG.
7318 040126 122766 000015 000002  CMPB      #CR,2(SP)      ;; IS CHARACTER A CARRIAGE RETURN?
7319 040134 001003              BNE      1$      ;; BRANCH IF NO
7320 040136 105037 040156      CLAB      $CHARCNT      ;; YES--CLEAR CHARACTER COUNT
7321 040142 000406              BR       $TYPEX      ;; EXIT
7322 040144 122766 000012 000002  1$:  CMPB      #LF,2(SP)      ;; IS CHARACTER A LINE FEED?
7323 040152 001402              BEQ      $TYPEX      ;; BRANCH IF YES
7324 040154 105227              INCB      (PC)+      ;; COUNT THE CHARACTER
7325 040156 000000      $CHARCNT: WORD 0      ;; CHARACTER COUNT STORAGE
7326 040160 000207      $TYPEX: RTS      PC
7327
7328      .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
7329
7330      ;*****
7331      ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
7332      ;SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
7333      ;NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
7334      ;BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
7335      ;REPLACED WITH SPACES.
7336      ;CALL:
7337      ;*      MOV      NUM,-(SP)      ;; PUT THE BINARY NUMBER ON THE STACK
7338      ;*      TYPDS      ;; GO TO THE ROUTINE
7339
7340      $TYPDS:
7341 040162 010046      MOV      R0,-(SP)      ;; PUSH R0 ON STACK
7342 040164 010146      MOV      R1,-(SP)      ;; PUSH R1 ON STACK
7343 040166 010246      MOV      R2,-(SP)      ;; PUSH R2 ON STACK
7344 040170 010346      MOV      R3,-(SP)      ;; PUSH R3 ON STACK
7345 040172 010546      MOV      R5,-(SP)      ;; PUSH R5 ON STACK
7346 040174 012746 020200      MOV      #20200,-(SP)      ;; SET BLANK SWITCH AND SIGN
7347 040200 016605 000020      MOV      20(SP),R5      ;; GET THE INPUT NUMBER
7348 040204 100004      BPL      1$      ;; BR IF INPUT IS POS.
7349 040206 005405      NEG      R5      ;; MAKE THE BINARY NUMBER POS.
7350 040210 112766 000055 000001  MOVB      #'-,1(SP)      ;; MAKE THE ASCII NUMBER NEG.
7351 040216 005000      CLR      R0      ;; ZERO THE CONSTANTS INDEX
7352 040220 012703 040376      MOV      #0$BLK,R3      ;; SETUP THE OUTPUT POINTER
7353 040224 112723 000040      MOVB      #' ,(R3)+      ;; SET THE FIRST CHARACTER TO A BLANK
7354 040230 005002      2$:  CLR      R2      ;; CLEAR THE BCD NUMBER
```

```

7355 040232 016001 040366      MOV      $DTBL(R0),R1      ;; GET THE CONSTANT
7356 040236 160105      3$: SUB      R1,R5      ;; FORM THIS BCD DIGIT
7357 040240 002402      BLT      4$      ;; BR IF DONE
7358 040242 005202      INC      R2      ;; INCREASE THE BCD DIGIT BY 1
7359 040244 000774      BR      3$
7360 040246 060105      4$: ADD      R1,R5      ;; ADD BACK THE CONSTANT
7361 040250 005702      TST      R2      ;; CHECK IF BCD DIGIT=0
7362 040252 001002      BNE      5$      ;; FALL THROUGH IF 0
7363 040254 105716      TSTB     (SP)      ;; STILL DOING LEADING 0'S?
7364 040256 100407      BMI      7$      ;; BR IF YES
7365 040260 106316      5$: ASLB     (SP)      ;; MSD?
7366 040262 103003      BCC      6$      ;; BR IF NO
7367 040264 116663 000001 177777 MOVB     1(SP),-1(R3)  ;; YES--SET THE SIGN
7368 040272 052702 000060      6$: BIS      #'0,R2      ;; MAKE THE BCD DIGIT ASCII
7369 040276 052702 000040      7$: BIS      #' ,R2      ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
7370 040302 110223      MOVB     R2,R3)+      ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
7371 040304 005720      TST      (R0)+      ;; JUST INCREMENTING
7372 040306 020027 000010      CMP      R0,'0      ;; CHECK THE TABLE INDEX
7373 040312 002746      BLT      2$      ;; GO DO THE NEXT DIGIT
7374 040314 003002      BGT      8$      ;; GO TO EXIT
7375 040316 010502      MOV      R5,R2      ;; GET THE LSD
7376 040320 000764      BR      6$      ;; GO CHANGE TO ASCII
7377 040322 105726      8$: TSTB     (SP)+      ;; WAS THE LSD THE FIRST NON-ZERO?
7378 040324 100003      BPL      9$      ;; BR IF NO
7379 040326 116663 177777 177776 9$: MOVB     -1(SP),-2(R3)  ;; YES--SET THE SIGN FOR TYPING
7380 040334 105013      CLRB     (R3)      ;; SET THE TERMINATOR
7381 040336 012605      MOV      (SP)+,R5      ;; POP STACK INTO R5
7382 040340 012603      MOV      (SP)+,R3      ;; POP STACK INTO R3
7383 040342 012602      MOV      (SP)+,R2      ;; POP STACK INTO R2
7384 040344 012601      MOV      (SP)+,R1      ;; POP STACK INTO R1
7385 040346 012600      MOV      (SP)+,R0      ;; POP STACK INTO R0
7386 040350 104401 040376 000004 TYPE     $DBLK      ;; NOW TYPE THE NUMBER
7387 040354 016666      MOV      2(SP),4(SP)      ;; ADJUST THE STACK
7388 040362 012616      MOV      (SP)+,(SP)
7389 040364 000002      RTI      ;; RETURN TO USER
7390 040366 023420      $DTBL: 10000.
7391 040370 001750      1000.
7392 040372 000144      100.
7393 040374 000012      10.
7394 040376 000004      $DBLK: .BLKW 4
7395      .SBTTL APT COMMUNICATIONS ROUTINE
7396
7397 *****
7398 040406 112737 000001 040652 $ATY1: MOVB     #1,$FFLG      ;; TO REPORT FATAL ERROR
7399 040414 112737 000001 040650 $ATY3: MOVB     #1,$MFLG      ;; TO TYPE A MESSAGE
7400 040422 000403      BR      $ATYC
7401 040424 112737 000001 040652 $ATY4: MOVB     #1,$FFLG      ;; TO ONLY REPORT FATAL ERROR
7402 040432      $ATYC:
7403 040432 010046      MOV      R0,-(SP)      ;; PUSH R0 ON STACK
7404 040434 010146      MOV      R1,-(SP)      ;; PUSH R1 ON STACK
7405 040436 105737 040650      TSTB     $MFLG      ;; SHOULD TYPE A MESSAGE?
7406 040442 001450      BEQ      5$      ;; IF NOT: BR
7407 040444 122737 000001 001230      CMPB     #APTENV,$ENV      ;; OPERATING UNDER APT?
7408 040452 001031      BNE      3$      ;; IF NOT: BR
7409 040454 132737 000100 001231      BITB     #APTPOOL,$ENVM      ;; SHOULD SPOOL MESSAGES?
7410 040462 001425      BEQ      3$      ;; IF NOT: BR

```

```

7411 040464 017600 000004      MOV    24(SP),R0      ;;GET MESSAGE ADDR.
7412 040470 062766 000002 000004  ADD    #2,4(SP)      ;;BUMP RETURN ADDR.
7413 040476 005737 001210      1$:    TST    $MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
7414 040502 001375              BNE     1$              ;;IF NOT: WAIT
7415 040504 010037 001224      MOV    R0,$MSGAD      ;;PUT ADDR IN MAILBOX
7416 040510 105720              2$:    TSTB   (R0)+      ;;FIND END OF MESSAGE
7417 040512 001376              BNE     2$              ;;
7418 040514 163700 001224      SUB     $MSGAD,R0      ;;SUB START OF MESSAGE
7419 040520 006200              ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
7420 040522 010037 001226      MOV    R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
7421 040526 012737 000004 001210  MOV    #4,$MSGTYPE      ;;TELL APT TO TAKE MSG.
7422 040534 000413              BR      5$              ;;
7423 040536 017637 000004 040562 3$:    MOV    24(SP),4$      ;;PUT MSG ADDR IN JSR LINKAGE
7424 040544 062766 000002 000004  ADD    #2,4(SP)      ;;BUMP RETURN ADDRESS
7425 040552 013746 177776      MOV    177776,-(SP)      ;;PUSH 177776 ON STACK
7426 040556 001737 037700      JSR     PC,$TYPE      ;;CALL TYPE MACRO
7427 040562 000000              4$:    .WORD  0
7428 040564              5$:
7429 040564 105737 040652      10$:   TSTB   $FFLG      ;;SHOULD REPORT FATAL ERROR?
7430 040570 001416              BEQ     12$      ;;IF NOT: BR
7431 040572 005737 001230      TST     $ENV      ;;RUNNING UNDER APT?
7432 040576 001413              BEQ     12$      ;;IF NOT: BR
7433 040600 005737 001210      11$:   TST    $MSGTYPE      ;;FINISHED LAST MESSAGE?
7434 040604 001375              BNE     11$      ;;IF NOT: WAIT
7435 040606 017637 000004 001212  MOV    24(SP),$FATAL      ;;GET ERROR #
7436 040614 062766 000002 000004  ADD    #2,4(SP)      ;;BUMP RETURN ADDR.
7437 040622 005237 001210      INC     $MSGTYPE      ;;TELL APT TO TAKE ERROR
7438 040626 105037 040652      12$:   CLRB   $FFLG      ;;CLEAR FATAL FLAG
7439 040632 105037 040651      CLRB   $LFLG      ;;CLEAR LOG FLAG
7440 040636 105037 040650      CLRB   $MFLG      ;;CLEAR MESSAGE FLAG
7441 040642 012601              MOV    (SP)+,R1      ;;POP STACK INTO R1
7442 040644 012600              MOV    (SP)+,R0      ;;POP STACK INTO R0
7443 040646 000207              RTS     PC      ;;RETURN
7444 040650 000              $MFLG: .BYTE  0      ;;MESSG. FLAG
7445 040651 000              $LFLG: .BYTE  0      ;;LOG FLAG
7446 040652 000              $FFLG: .BYTE  0      ;;FATAL FLAG
7447 040654 040654              .EVEN
7448 000200      APTSIZE=200
7449 000001      APTENV=001
7450 000100      APTSPool=100
7451 000040      APTCSUP=040
7452      .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
7453
7454      ;*****
7455      ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
7456      ;OCTAL (ASCII) NUMBER AND TYPE IT.
7457      ;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
7458      ;CALL:
7459      ;    MOV    NUM,-(SP)      ;;NUMBER TO BE TYPED
7460      ;    TYPOS      ;;CALL FOR TYPEOUT
7461      ;    .BYTE  N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
7462      ;    .BYTE  M      ;;M=1 OR 0
7463      ;    ;;1=TYPE LEADING ZEROS
7464      ;    ;;0=SUPPRESS LEADING ZEROS
7465      ;
7466      ;$STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST

```

```

7467      ;*STYPOS OR STYPOC
7468      ;*CALL:
7469      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
7470      ;*      TYPON      ;;CALL FOR TYPEOUT
7471      ;*
7472      ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
7473      ;*CALL:
7474      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
7475      ;*      TYPOC      ;;CALL FOR TYPEOUT
7476      ;*
7477      040654 017646 000000      STYPOS: MOV      2(SP),-(SP)      ;;PICKUP THE MODE
7478      040660 116637 000001      041077      MOV      1(SP),SOFILL      ;;LOAD ZERO FILL SWITCH
7479      040666 112637 041101      MOV      (SP)+,SOMODE+1      ;;NUMBER OF DIGITS TO TYPE
7480      040672 062716 000002      ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
7481      040676 000406      STYPOC: MOV      #1,SOFILL      ;;SET THE ZERO FILL SWITCH
7482      040700 112737 000001      041077      MOV      #6,SOMODE+1      ;;SET FOR SIX(6) DIGITS
7483      040706 112737 000006      041101      MOV      #5,SOCNT      ;;SET THE ITERATION COUNT
7484      040714 112737 000005      041076      MOV      R3,-(SP)      ;;SAVE R3
7485      040722 010346      MOV      R4,-(SP)      ;;SAVE R4
7486      040724 010446      MOV      R5,-(SP)      ;;SAVE R5
7487      040726 010546      MOV      SOMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
7488      040730 113704 04110      NEG      R4
7489      040734 005404      ADD      #6,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
7490      040736 062704 000006      MOV      R4,SOMODE      ;;SAVE IT FOR USE
7491      040742 110437 041100      MOV      SOFILL,R4      ;;GET THE ZERO FILL SWITCH
7492      040746 113704 041077      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
7493      040752 016605 000012      CLR      R3      ;;CLEAR THE OUTPUT WORD
7494      040756 005003      1$:      ROL      R5      ;;ROTATE MSB INTO "C"
7495      040760 006105      BR      3$      ;;GO DO MSB
7496      040762 000404      2$:      ROL      R5      ;;FORM THIS DIGIT
7497      040764 006105      ROL      R5
7498      040766 006105      ROL      R5
7499      040770 006105      ROL      R5
7500      040772 010503      MOV      R5,R3
7501      040774 006103      3$:      ROL      R3      ;;GET LSB OF THIS DIGIT
7502      040776 105337 041100      DECB      SOMODE      ;;TYPE THIS DIGIT?
7503      041002 100016      BPL      7$      ;;BR IF NO
7504      041004 042703 177770      BIC      #177770,R3      ;;GET RID OF JUNK
7505      041010 001002      BNE      4$      ;;TEST FOR 0
7506      041012 005704      TST      R4      ;;SUPPRESS THIS 0?
7507      041014 001403      BEQ      5$      ;;BR IF YES
7508      041016 005204      4$:      INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
7509      041020 052703 000060      BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
7510      041024 052703 000040      5$:      BIS      #'',R3      ;;MAKE ASCII IF NOT ALREADY
7511      041030 110337 041074      MOV      R3,8$      ;;SAVE FOR TYPING
7512      041034 104401 041074      TYPE      8$      ;;GO TYPE THIS DIGIT
7513      041040 105337 041076      7$:      DECB      SOCNT      ;;COUNT BY 1
7514      041044 003347      BGT      2$      ;;BR IF MORE TO DO
7515      041046 002402      BLT      6$      ;;BR IF DONE
7516      041050 005204      INC      R4      ;;INSURE LAST DIGIT ISN'T A BLANK
7517      041052 000744      BR      2$      ;;GO DO THE LAST DIGIT
7518      041054 012605      6$:      MOV      (SP)+,R5      ;;RESTORE R5
7519      041056 012604      MOV      (SP)+,R4      ;;RESTORE R4
7520      041060 012603      MOV      (SP)+,R3      ;;RESTORE R3
7521      041062 016666 000002 000004      MOV      2(SP),4(SP)      ;;SET THE STACK FOR RETURNING
7522      041070 012616      MOV      (SP)+,(SP)

```

```

7523 041072 000002      RTI                      ;; RETURN
7524 041074 000          8$: .BYTE 0              ;; STORAGE FOR ASCII DIGIT
7525 041075 000          .BYTE 0              ;; TERMINATOR FOR TYPE ROUTINE
7526 041076 000          SOCNT: .BYTE 0          ;; OCTAL DIGIT COUNTER
7527 041077 000          SOFILL: .BYTE 0         ;; ZERO FILL SWITCH
7528 041100 000000      SOMODE: .WORD 0          ;; NUMBER OF DIGITS TO TYPE
7529                      .SBTTL TTY INPUT ROUTINE
7530
7531                      ;*****
7532                      .ENABL LSB
7533 041102 000000      STKCNT: .WORD 0          ;; NUMBER OF ITEMS IN QUEUE
7534 041104 000000      STKQIN: .WORD 0          ;; INPUT POINTER
7535 041106 000000      STKQOUT: .WORD 0         ;; OUTPUT POINTER
7536 041110 000001      STKQSR: .BLKB 1          ;; TTY KEYBOARD QUEUE
7537                      STKQEND=.
7538                      .EVEN
7539
7540                      ;*TK INITIALIZE ROUTINE
7541                      ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
7542                      ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
7543
7544                      ;*CALL:
7545                      ;*      JSR      PC,STKINT
7546                      ;*      RETURN
7547
7548 041112 005037 041102 STKINT: CLR      STKCNT      ;; CLEAR COUNT OF ITEMS IN QUEUE
7549 041116 012737 041110 041104 MOV      #STKQSR,STKQIN ;; MOVE THE STARTING ADDRESS OF THE
7550 041124 013737 041110 041106 MOV      STKQIN,STKQOUT ;; QUEUE INTO THE INPUT & OUTPUT POINTERS.
7551 041132 012737 041162 000060 MOV      #STKSRV,#TKVEC ;; INITIALIZE THE KEYBOARD VECTOR
7552 041140 012737 000200 000062 MOV      #200,#TKVEC+2 ;; "BR" LEVEL 4
7553 041146 005777 137774 TST      #STKB        ;; CLEAR DONE FLAG
7554 041152 012777 000100 137764 MOV      #100,#STKS ;; ENABLE TTY KEYBOARD INTERRUPT
7555 041160 000207      RTS      PC              ;; RETURN TO CALLER
7556
7557                      ;*TK SERVICE ROUTINE
7558                      ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
7559                      ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
7560                      ;*IT IN THE QUEUE.
7561                      ;*IF THE CHARACTER IS A "CONTROL-C" (↑C) STKINT IS CALLED AND
7562                      ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (STOP)
7563
7564 041162 117746 137760 STKSRV: MOVB     #STKB,-(SP) ;; PICKUP THE CHARACTER
7565 041166 042716 177600 BIC      #↑C177,(SP) ;; STRIP THE JUNK
7566 041172 021627 000003 CMP      (SP),#3 ;; IS IT A CONTROL C?
7567 041176 001007 BNE      1$ ;; BRANCH IF NO
7568 041200 104401 042310 TYPE     #CNTLC ;; TYPE A CONTROL-C (↑C)
7569 041204 004737 041112 JSR      PC,STKINT ;; INIT THE KEYBOARD
7570 041210 005726 TST      (SP)+ ;; CLEAN UP STACK
7571 041212 000137 036262 JMP      STOP ;; CONTROL C RESTART
7572 041216 021627 000007 1$: CMP      (SP),#7 ;; IS IT A CONTROL G?
7573 041222 001004 BNE      2$ ;; BRANCH IF NO
7574 041224 022737 000176 001140 CMP      #SWREG,SWR ;; IS SOFT-SWR SELECTED?
7575 041232 001500 BEQ      6$ ;; GO TO SWR CHANGE
7576
7577 041234 2$: CMP      #1,STKCNT ;; IS THE QUEUE FULL?
7578 041234 022737 000001 041102

```

```

7579 041242 001004      BNE      3$      ;;BRANCH IF NO
7580 041244 104401 001200 TYPE      $BELL      ;;RING THE TTY BELL
7581 041250 005726      TST      (SP)+      ;;CLEAN CHARACTER OFF OF STACK
7582 041252 000451      BR       5$      ;;EXIT
7583 041254 021627 000023 3$:      CMP      (SP),#23      ;;IS IT A CONTROL-S?
7584 041260 001021      BNE      32$      ;;BRANCH IF NO
7585 041262 005077 137656      CLR      2$TKS      ;;DISABLE TTY KEYBOARD INTERRUPTS
7586 041266 005726      TST      (SP)+      ;;CLEAN CHAR OFF STACK
7587 041270 105777 137650 31$:      TSTB     2$TKS      ;;WAIT FOR A CHAR
7588 041274 100375      BPL      31$      ;;LOOP UNTIL ITS THERE
7589 041276 117746 137644      MOVB     2$TKB,-(SP)      ;;GET THE CHARACTER
7590 041302 042716 177600      BIC      #1C177,(SP)      ;;MAKE IT 7-BIT ASCII
7591 041306 022627 000021      CMP      (SP)+,#21      ;;IS IT A CONTROL-Q?
7592 041312 001366      BNE      31$      ;;BRANCH IF NO
7593 041314 012777 000100 137622      MOV      #100,2$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
7594 041322 000002      RTI      ;;RETURN
7595 041324 001237 041102 32$:      INC      $TKCNT      ;;COUNT THIS CHARACTER
7596 041330 021627 000140      CMP      (SP),#140      ;;IS IT UPPER CASE?
7597 041334 002405      BLT      4$      ;;BRANCH IF YES
7598 041336 021627 000175      CMP      (SP),#175      ;;IS IT A SPECIAL CHAR?
7599 041342 003002      BGT      4$      ;;BRANCH IF YES
7600 041344 042716 000040      BIC      #40,(SP)      ;;MAKE IT UPPER CASE
7601 041350 112677 177530 4$:      MOVB     (SP)+,2$TKQIN      ;;AND PUT IT IN QUEUE
7602 041354 005237 041104      INC      $TKQIN      ;;UPDATE THE POINTER
7603 041360 023727 041104 041111      CMP      $TKQIN,$$TKQEND      ;;GO OFF THE END?
7604 041366 001003      BNE      5$      ;;BRANCH IF NO
7605 041370 012737 041110 041104      MOV      $$TKQSRT,$$TKQIN      ;;RESET THE POINTER
7606 041376 000002 5$:      RTI      ;;RETURN
7607
7608      ;;*****
7609      ;;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
7610      ;;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
7611      ;;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
7612      ;;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
7613 041400 022737 000176 001140 $CKSWR: CMP      $SWREG,$SWR      ;;IS THE SOFT-SWR SELECTED
7614 041406 001124      BNE      15$      ;;EXIT IF NOT
7615 041410 105777 137530      TSTB     2$TKS      ;;IS A CHAR WAITING?
7616 041414 100121      BPL      15$      ;;IF NOT, EXIT
7617 041416 117746 137524      MOVB     2$TKB,-(SP)      ;;YES
7618 041422 042716 177600      BIC      #1C177,(SP)      ;;MAKE IT 7-BIT ASCII
7619 041426 021627 000007      CMP      (SP),#7      ;;IS IT A CONTROL-G?
7620 041432 001300      BNE      2$      ;;IF NOT, PUT IT IN THE TTY QUEUE
7621      ;;AND EXIT
7622
7623      ;;*****
7624      ;;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
7625      ;;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
7626      ;;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
7627 041434 123727 001134 000001 6$:      CMPB     $AUTOB,#1      ;;ARE WE RUNNING IN AUTO-MODE?
7628 041442 001674      BEQ      2$      ;;BRANCH IF YES
7629 041444 005726      TST      (SP)+      ;;CLEAR CONTROL-G OFF STACK
7630 041446 004737 041112      JSR      PC,$$TKINT      ;;FLUSH THE TTY INPUT QUEUE
7631 041452 005077 137466      CLR      2$TKS      ;;DISABLE TTY KEYBOARD INTERRUPTS
7632 041456 112737 000001 001135      MOVB     #1,$$INTAG      ;;SET INTERRUPT MODE INDICATOR
7633
7634 041464 104401 042322      TYPE      ,$$CNTLG      ;;ECHO THE CONTROL-G (↑G)

```

CZR61EO UNIBUS RKE DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 148
TTY INPUT ROUTINE

SEQ 0148

7635	041470	104401	042327		\$GTSWR: TYPE	\$MSWR	:: TYPE CURRENT CONTENTS
7636	041474	013746	000176		MOV	\$WREG, -(SP)	:: SAVE SWREG FOR TYPEOUT
7637	041500	104402			TYPOC		:: GO TYPE--OCTAL ASCII(ALL DIGITS)
7638	041502	104401	042340		TYPE	, \$MNEW	:: PROMPT FOR NEW SWR
7639	041506	005046			19\$: CLR	-(SP)	:: CLEAR COUNTER
7640	041510	005046			CLP	-(SP)	:: THE NEW SWR
7641	041512	105777	137426		7\$ TSTB	\$STKS	:: CHAR THERE?
7642	041516	100375			BPL	7\$	:: IF NOT TRY AGAIN
7643							
7644	041520	117746	137422		MOVB	\$STKB, -(SP)	:: PICK UP CHAP
7645	041524	042716	177600		BIC	#1C177, (SP)	:: MAKE IT 7-BIT ASCII
7646							
7647	041530	021627	000003		CMP	(SP), #3	:: IS IT A CONTROL-C?
7648	041534	001015			BNE	9\$	:: BRANCH IF NOT
7649	041536	104401	042310		TYPE	, \$CNTLC	:: YES, ECHO CONTROL-C (↑C)
7650	041542	062706	000006		ADD	#6, SP	:: CLEAN UP STACK
7651	041546	123727	001135	000001	CMPB	\$INTAG, #1	:: REENABLE TTY KEYBOARD INTERRUPTS?
7652	041554	001003			BNE	8\$	:: BRANCH IF NO
7653	041556	012777	000100	137360	MOV	#100, \$STKS	:: ALLOW TTY KEYBOARD INTERRUPTS
7654	041564	000137	036262		8\$: JMP	STOP	:: CONTROL-C RESTART
7655							
7656							
7657	041570	021627	000025		9\$: CMP	(SP), #25	:: IS IT A CONTROL-U?
7658	041574	001005			BNE	10\$	:: BRANCH IF NOT
7659	041576	104401	042315		TYPE	, \$CNTLU	:: YES, ECHO CONTROL-U (↑U)
7660	041602	062706	000006		20\$: ADD	#6, SP	:: IGNORE PREVIOUS INPUT
7661	041606	000737			BR	19\$	:: LET'S TRY IT AGAIN
7662							
7663							
7664	041610	021627	000015		10\$: CMP	(SP), #15	:: IS IT A <CR>?
7665	041614	001027			BNE	16\$	:: BRANCH IF NO
7666	041616	005766	000004		TST	4()	:: YES, IS IT THE FIRST CHAR?
7667	041622	001403			BEQ	11\$	:: BRANCH IF YES
7668	041624	016677	000002	137306	MOV	2(SP), \$SWR	:: SAVE NEW SWR
7669	041632	062706	000006		11\$: ADD	#6, SP	:: CLEAN UP STACK
7670	041636	104401	001205		14\$: TYPE	, \$CRLF	:: ECHO <CR> AND <LF>
7671	041642	123727	001135	000001	CMPB	\$INTAG, #1	:: RE-ENABLE TTY KBD INTERRUPTS?
7672	041650	001003			BNE	15\$	:: BRANCH IF NOT
7673	041652	012777	000100	137264	MOV	#100, \$STKS	:: RE-ENABLE TTY KBD INTERRUPTS
7674	041660	000002			15\$: RTI		:: RETURN
7675	041662	004737	040112		16\$: JSR	PC, \$TYPEC	:: ECHO CHAR
7676	041666	021627	000060		CMP	(SP), #60	:: CHAR < 0?
7677	041672	002420			BLT	18\$	:: BRANCH IF YES
7678	041674	021627	000067		CMP	(SP), #67	:: CHAR > 7?
7679	041700	003015			BGT	18\$	:: BRANCH IF YES
7680	041702	042726	000060		BIC	#60, (SP)+	:: STRIP-OFF ASCII
7681	041706	005766	000002		TST	2(SP)	:: IS THIS THE FIRST CHAR
7682	041712	001403			BEQ	17\$	:: BRANCH IF YES
7683	041714	006316			ASL	(SP)	:: NO, SHIFT PRESENT
7684	041716	006316			ASL	(SP)	:: CHAR OVER TO MAKE
7685	041720	006316			ASL	(SP)	:: ROOM FOR NEW ONE.
7686	041722	005266	000002		17\$: INC	2(SP)	:: KEEP COUNT OF CHAR
7687	041726	056616	177776		BIS	-2(SP), (SP)	:: SET IN NEW CHAR
7688	041732	000667			BR	7\$	:: GET THE NEXT ONE
7689	041734	104401	001204		18\$: TYPE	, \$QUES	:: TYPE ?<CR><LF>
7690	041740	000720			BR	20\$	:: SIMULATE CONTROL-U

```

7691      .DSABL  LSB
7692
7693
7694      ;*****
7695      ;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
7696      ;CALL:
7697      ;      RDCHR          ; GET A CHARACTER FROM THE QUEUE
7698      ;      RETURN HERE   ; CHARACTER IS ON THE STACK
7699      ;                    ; WITH PARITY BIT STRIPPED OFF
7700
7701
7702      041742 011646          $RDCHR: MOV      (SP), -(SP)      ; PUSH DOWN THE PC AND
7703      041744 016666 000004 000002      MOV      4(SP), 2(SP)      ;   PS
7704      041752 005066 000004          CLR      4(SP), 2(SP)      ; GET READY FOR A CHARACTER
7705      041756 005046          CLR      -(SP)      ; PUT NEW PS ON STACK
7706      041760 012746 041766          MOV      #64$, -(SP)      ; PUT NEW PC ON STACK
7707      041764 000002          RTI          ; POP NEW PC AND PS
7708      041766
7709      041766 005737 041102      64$:   TST      $TKCNT          ; WAIT ON A CHARACTER
7710      041772 001775          1$:   BEQ      1$
7711      041774 005337 041102      DEC      $TKCNT          ; DECREMENT THE COUNTER
7712      042000 117766 177102 000004      MOVB    2$TKQOUT, 4(SP)      ; GET ONE CHARACTER
7713      042006 005237 041106      INC      $TKQOUT          ; UPDATE THE POINTER
7714      042012 023727 041106 041111      CMP      $TKQOUT, #STI QEND ; DID IT GO OFF OF THE END
7715      042020 001003          BNE      2$          ; BRANCH IF NO
7716      042022 012737 041110 041106      MOV      #STKQSR, $TKQOUT ; RESET THE POINTER
7717      042030 000002          2$:   RTI          ; RETURN
7718
7719      ;*****
7720      ;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
7721      ;CALL:
7722      ;      RDLIN          ; INPUT A STRING FROM THE TTY
7723      ;      RETURN HERE   ; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
7724      ;                    ; TERMINATOR WILL BE A BYTE OF ALL 0'S
7725
7726      042032 010346          $RDLIN: MOV      R3, -(SP)      ; SAVE R3
7727      042034 005046          CLR      -(SP)      ; CLEAR THE RUBOUT KEY
7728      042036 012703 042266          1$:   MOV      #STTYIN, R3      ; GET ADDRESS
7729      042042 022703 042310          2$:   CMP      #STTYIN+22, R3 ; BUFFER FULL?
7730      042046 101456          BLOS     4$          ; BR IF YES
7731      042050 104410          RDCHR          ; GO READ ONE CHARACTER FROM THE TTY
7732      042052 112613          MOVB    (SP)+, (R3)      ; GET CHARACTER
7733      042054 122713 000177          10$:  CMPB    #177, (R3)      ; IS IT A RUBOUT
7734      042060 001022          BNE      5$          ; BR IF NO
7735      042062 005716          TST      (SP)          ; IS THIS THE FIRST RUBOUT?
7736      042064 001007          BNE      6$          ; BR IF NO
7737      042066 112737 000134 042264      MOVB    #' \, 9$      ; TYPE A BACK SLASH
7738      042074 104401 042264          TYPE    9$
7739      042100 012716 177777          MOV      #-1, (SP)      ; SET THE RUBOUT KEY
7740      042104 005303          6$:   DEC      R3          ; BACKUP BY ONE
7741      042106 020327 042266          CMP      R3, #STTYIN      ; STACK EMPTY?
7742      042112 103434          BLO      4$          ; BR IF YES
7743      042114 111337 042264          MOVB    (R3), 9$      ; SETUP TO TYPEOUT THE DELETED CHAR.
7744      042120 104401 042264          TYPE    9$          ; GO TYPE
7745      042124 000746          BR      2$          ; GO READ ANOTHER CHAR.
7746      042126 005716          5$:   TST      (SP)          ; RUBOUT KEY SET?
7747      042130 001406          BEQ      7$          ; BR IF NO

```

```

7747 042132 112737 000134 042264      MOVB      #' \, 9$      ;; TYPE A BACK SLASH
7748 042140 104401 042264              TYPE      , 9$
7749 042144 005016              CLR      (SP)      ;; CLEAR THE RUBOUT KEY
7750 042146 122713 000025      7$:      CMPB      #25, (R3)      ;; IS CHARACTER A CTRL U?
7751 042152 001003              BNE      8$      ;; BR IF NO
7752 042154 104401 042315      TYPE      $CNTLU      ;; TYPE A CONTROL "U"
7753 042160 000726              BR      1$      ;; GO START OVER
7754 042162 122713 000022      8$:      CMPB      #22, (R3)      ;; IS CHARACTER A "R"?
7755 042166 001011              BNE      3$      ;; BRANCH IF NO
7756 042170 105013              CLRB      (R3)      ;; CLEAR THE CHARACTER
7757 042172 104401 001205      TYPE      , $CRLF      ;; TYPE A "CR" & "LF"
7758 042176 104401 042266      TYPE      $TTYIN      ;; TYPE THE INPUT STRING
7759 042202 000717              BR      2$      ;; GO PICKUP ANOTHER CHACTER
7760 042204 104401 001204      4$:      TYPE      $QUES      ;; TYPE A '?'
7761 042210 000712              BR      1$      ;; CLEAR THE BUFFER AND LOOP
7762 042212 111337 042264      3$:      MOVB      (R3), 9$      ;; ECHO THE CHARACTER
7763 042216 104401 042264              TYPE      , 9$
7764 042222 122723 000015      CMPB      #15, (R3)+      ;; HECK FOR RETURN
7765 042226 001305              BNE      2$      ;; LOOP IF NOT RETURN
7766 042230 105063              CLRB      -1(R3)      ;; CLEAR RETURN (THE 15)
7767 042234 104401 001206      TYPE      $LF      ;; TYPE A LINE FEED
7768 042240 005726              TST      (SP)+      ;; CLEAN RUBOUT KEY FROM THE STACK
7769 042242 012603              MOV      (SP)+, R3      ;; RESTORE R3
7770 042244 011646              MOV      (SP), -(SP)      ;; ADJUST THE STACK AND PUT ADDRESS OF THE
7771 042246 016666 000004 000002      MOV      4(SP), 2(SP)      ;; FIRST ASCII CHARACTER ON IT
7772 042254 012766 042266 000004      MOV      $TTYIN, 4(SP)
7773 042262 000002              RTI      ;; RETURN
7774 042264 000          9$:      .BYTE      0      ;; STORAGE FOR ASCII CHAR. TO TYPE
7775 042265 000          .BYTE      0      ;; TERMINATOR
7776 042266 000022      $TTYIN: .BLKB      22      ;; RESERVE 22 BYTES FOR TTY INPUT
7777 042310 041536 005015 000      $CNTLC: .ASCIZ  /tC<15><12>      ;; CONTROL "C"
7778 042315 0136 006525 000012      $CNTLU: .ASCIZ  /tU<15><12>      ;; CONTROL "U"
7779 042322 043536 005015 000      $CNTLG: .ASCIZ  /tG<15><12>      ;; CONTROL "G"
7780 042327 015 051412 051127      $MSWR:  .ASCIZ  <15><12>/SWR = /
7781 042334 036440 000040              $MNEW: .ASCIZ  / NEW = /
7782 042340 020040 042516 020127      .EVEN
7783 042346 020075 000      .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
7784 042352
7785
7786
7787      ;*****
7788      ;THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
7789      ;CHANGE IT TO BINARY.
7790      ;THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
7791      ;OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
7792      ;FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
7793      ;THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
7794      ;CALL:
7795      ;      RDOCT      ;; READ AN OCTAL NUMBER
7796      ;      RETURN HERE      ;; LOW ORDER BITS ARE ON TOP OF THE STACK
7797      ;      ;; HIGH ORDER BITS ARE IN $HIOCT
7798
7799 042352 011646 000004 000002      $RDOCT: MOV      (SP), -(SP)      ;; PROVIDE SPACE FOR THE
7800 042354 016666              MOV      4(SP), 2(SP)      ;; INPUT NUMBER
7801 042362 010046              MOV      R0, -(SP)      ;; PUSH R0 ON STACK
7802 042364 010146              MOV      R1, -(SP)      ;; PUSH R1 ON STACK

```

7803 042366 010246
7804 042370 104411
7805 042372 012600
7806 042374 010037 042500
7807 042400 005001
7808 042402 005002
7809 042404 112046
7810 042406 001420
7811 042410 122716 000060
7812 042414 003026
7813 042416 122716 000067
7814 042422 002423
7815 04242 006301
7816 042426 006102
7817 042430 006301
7818 042432 006102
7819 042434 006301
7820 042436 006102
7821 042440 042716 177770
7822 042444 062601
7823 042446 000756
7824 042450 005726
7825 042452 010166 000012
7826 042456 010237 042510
7827 042462 012602
7828 042464 012601
7829 042466 012600
7830 042470 000002
7831 042472 005726
7832 042474 105010
7833 042476 104401
7834 042500 000000
7835 042502 104401 001204
7836 042506 000730
7837 042510 000000
7838
7839
7840
7841
7842
7843
7844
7845
7846
7847
7848
7849 042512 104413
7850 042514 016601 000002
7851 042520 012705 042631
7852 042524 012704 000014
7853 042530 012703 177770
7854 042534 012100
7855 042536 012101
7856 042540 005002
7857 042542 110245
7858 042544 010002

1\$: MOV R2, -(SP) ;; PUSH R2 ON STACK
RDLIN ;; READ AN ASCII LINE
MOV (SP)+, R0 ;; GET ADDRESS OF 1ST CHARACTER
MOV R0, 5\$;; AND SAVE IT
CLR R1 ;; CLEAR DATA WORD
CLR R2
2\$: MOVB (R0)+, -(SP) ;; PICKUP THIS CHARACTER
BEQ 3\$;; IF ZERO GET OUT
CMPB #'0, (SP) ;; MAKE SURE THIS CHARACTER
BGT 4\$;; IS AN OCTAL DIGIT
CMPB #'7, (SP)
BLT 4\$
ASL R1 ;; *2
ROL R2
ASL R1 ;; *4
ROL R2
ASL R1 ;; *8
ROL R2
BIC #'C7, (SP) ;; STRIP THE ASCII JUNK
ADD (SP)+, R1 ;; ADD IN THIS DIGIT
BR 2\$;; LOOP
3\$: TST (SP)+ ;; CLEAN TERMINATOR FROM STACK
MOV R1, 12(SP) ;; SAVE THE RESULT
MOV R2, \$HIOCT
MOV (SP)+, R2 ;; POP STACK INTO R2
MOV (SP)+, R1 ;; POP STACK INTO R1
MOV (SP)+, R0 ;; POP STACK INTO R0
RTI ;; RETURN
4\$: TST (SP)+ ;; CLEAN PARTIAL FROM STACK
CLRB (R0) ;; SET A TERMINATOR
TYPE ;; TYPE UP THRU THE BAD CHAR.
5\$: .WORD 0
TYPE \$QUES ;; "?" "CR" & "LF"
BR 1\$;; TRY AGAIN
\$HIOCT: .WORD 0 ;; HIGH ORDER BITS GO HERE
\$BTTL DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE

; THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
; UNSIGNED OCTAL ASCII NUMBER.
; CALL
; * MOV #PNTR, -(SP) ;; POINTER TO LOW WORD OF BINARY NUMBER
; * JSR PC, @#\$DB20 ;; CALL THE ROUTINE
; * RETURN ;; THE ADDRESS OF THE FIRST ASCII CHAR. IS ON THE STACK
\$DB20: SAVREG
MOV 2(SP), R1 ;; SAVE ALL REGISTERS
MOV #SOCTVL+13., R5 ;; PICKUP THE POINTER TO LOW WORD
MOV #12., R4 ;; POINTER TO DATA TABLE
MOV #C7, R3 ;; DO ELEVEN CHARACTERS
MOV (R1)+, R0 ;; MASK
MOV (R1)+, R1 ;; LOWER WORD
CLR R2 ;; HIGH WORD
1\$: MOVB R2, -(R5) ;; TERMINATOR
MOV R0, R2 ;; PUT CHARACTER IN DATA TABLE
;; GET THIS DIGIT

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 152
DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE

SEQ 0152

```

7859 042546 005304          DEC      R4          ;; COUNT THIS CHARACTER
7860 042550 003007          BGT      3$          ;; BR IF NOT THE LAST DIGIT
7861 042552 001405          BEQ      2$          ;; BR IF IT IS THE LAST DIGIT
7862 042554 005205          INC      R5          ;; ALL DIGITS DONE-ADJUST POINTER FOR FIRST
7863 042556 010566 000002   MOV      R5,2(SP)    ;; ASCII CHAR. & PUT IT ON THE STACK
7864 042562 104414          RESREG          ;; RESTORE ALL REGISTERS
7865 042564 000207          RTS      PC          ;; RETURN TO USER
7866 042566 006203          2$:  ASR      R3          ;; POSITION THE MASK FOR THE LAST DIGIT
7867 042570 006001          3$:  ROR      R1          ;; POSITION THE BINARY NUMBER FOR
7868 042572 006000          ROR      R0          ;; THE NEXT OCTAL DIGIT
7869 042574 006001          ROR      R1
7870 042576 006000          ROR      R0
7871 042600 006001          ROR      R1
7872 042602 006000          ROR      R0
7873 042604 040302          BIC      R3,R2          ;; MASK OUT ALL JUNK
7874 042606 062702 000060   ADD      #0,R2          ;; MAKE THIS CHAR. ASCII
7875 042612 000753          BR       1$          ;; GO PUT IT IN THE DATA TABLE
7876 042614 000016          $OCTVL: .BLKB 14.        ;; RESERVE DATA TABLE
7877                                     .SBTTL DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
7878
7879 *****
7880 *THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
7881 *DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
7882 *POSITIVE.
7883 *CALL
7884 *      MOV      #PNTR, -(SP)          ;; POINTER TO LOW WORD OF BINARY NUMBER
7885 *      JSR      PC, 2*$DB2D          ;; THE FIRST ADDRESS OF ASCII
7886 *      RETURN                          ;; IS ON THE STACK
7887
7888
7889 $DB2D: SAVREG          ;; SAVE REGISTERS
7890 042632 104413          MOV      2(SP), R2          ;; PICKUP THE DATA POINTER
7891 042634 016602 000002   MOV      #$DECVL, R0        ;; GET ADDRESS OF "$DECVL" STRING
7892 042640 012700 043012   MOV      R0, 2(SP)          ;; PUT ADDRESS OF ASCII STRING ON STACK
7893 042644 010066 000002   MOV      (R2)+, R1          ;; PICKUP THE BINARY NUMBER
7894 042650 012201          MOV      (R2)+, R2
7895 042652 012202          MOV      (R2)+, R2
7896 042654 012737 000012 042730   MOV      #10, 4$          ;; SET UP TO DO 10 CONVERSIONS
7897 042662 012704 042742   MOV      #$TNPWR, R4        ;; ADDRESS OF TEN POWER
7898 042666 012705 042744   MOV      #$TNPWR+2, R5
7899 042672 005003          1$:  CLR      R3          ;; CLEAR PARTIAL
7900 042674 161401          2$:  SUB      (R4), R1          ;; SUBTRACT TEN POWER
7901 042676 005602          SBC      R2
7902 042700 161502          SUB      (R5), R2
7903 042702 002402          BLT      3$          ;; BR IF TEN POWER TO LARGE
7904 042704 005203          INC      R3          ;; ADD 1 TO PARTIAL
7905 042706 000772          BR       2$          ;; LOOP
7906 042710 062401          3$:  ADD      (R4)+, R1          ;; RESTORE SUBTRACTED VALUE
7907 042712 005502          ADC      R2
7908 042714 062402          ADD      (R4)+, R2
7909 042716 022525          CMP      (R5)+, (R5)+          ;; MOVE TO NEXT TEN POWER
7910 042720 052703 000060   BIS      #0, R3          ;; CHANGE PARTIAL TO ASCII
7911 042724 110320          MOV8     R3, (R0)+          ;; SAVE IT
7912 042726 005327          DEC      (PC)+          ;; DONE?
7913 042730 000000          4$:  .WORD 0
7914 042732 001357          BNE      1$          ;; BR IF NO

```

```

7915 042734 105020          CLR8      (R0)+      ;; TERMINATOR
7916 042736 104414          RESREG          ;; RESTORE REGISTERS
7917 042740 000207          RTS      PC          ;; RETURN
7918 042742 145000          $TNPWR: 145000      ;; 1.0E09
7919 042744 035632          35632
7920 042746 160400          160400      ;; 1.0E08
7921 042750 002765          2765
7922 042752 113200          113200      ;; 1.0E07
7923 042754 000230          230
7924 042756 041100          041100      ;; 1.0E06
7925 042760 000017          17
7926 042762 103240          103240      ;; 1.0E05
7927 042764 000001          1
7928 042766 023420          23420      ;; 1.0E04
7929 042770 000000          0
7930 042772 001750          1750      ;; 1.0E03
7931 042774 000000          0
7932 042776 000144          144      ;; 1.0E02
7933 043000 000000          0
7934 043002 000012          12      ;; 1.0E01
7935 043004 000000          0
7936 043006 000001          1      ;; 1.0E00
7937 043010 000000          0
7938 043012 000014          $DECVL: .BLKB 12.      ;; RESERVE STORAGE FOR ASCII STRING
7939          .SBTTL SINGLE LENGTH BINARY TO DECIMAL ASCII ROUTINE
7940
7941          ;; *****
7942          ;; THIS ROUTINE WILL CONVERT A 16-BIT UNSIGNED BINARY NUMBER TO AN
7943          ;; UNSIGNED DECIMAL ASCII NUMBER.
7944          ;; *CALL
7945          ;; *      MOV      NUMBER, -(SP)      ;; PUT BINARY NUMBER ON THE STACK
7946          ;; *      JSR      PC, @#$SB2D      ;; CALL
7947          ;; *      RETURN      ;; ADDRESS OF THE 1ST ASCII CHAR. IS ON THE STACK
7948
7949
7950 043026 016637 000002 043056 $SB2D: MOV      2(SP), 1$      ;; SAVE BINARY NUMBER
7951 043034 012746 043056      MOV      1$, -(SP)      ;; SET POINTER
7952 043040 004737 042632      JSR      PC, @#$SB2D      ;; CALL DOUBLE LENGTH CONVERT
7953 043044 062716 000005      ADD      #5, (SP)      ;; ONLY ALLOW FIVE CHARACTERS
7954 043050 012666 000002      MOV      (SP)+, 2(SP)      ;; PICKUP POINTER
7955 043054 000207          RTS      PC      ;; RETURN
7956 043056 000000 000000      1$:      .WORD 0, 0
7957          .SBTTL TYPE NUMERICAL ASCII STRING SUPPRESS LEADING ZEROS
7958
7959          ;; *****
7960          ;; THIS ROUTINE IS USED TO TYPE AN ASCII NUMBER SUPPRESSING THE
7961          ;; LEADING NUMBERS.
7962          ;; *CALL
7963          ;; *      MOV      #NUMADR, -(SP)      ;; FIRST ADDRESS OF ASCII STRING
7964          ;; *      JSR      PC, @#$SUPRS
7965
7966
7967 043062 010046 000004 $SUPRS: MOV      R0, -(SP)      ;; SAVE R0
7968 043064 016600          MOV      4(SP), R0      ;; PICKUP THE POINTER
7969 043070 105710          1$:      TSTB      (R0)      ;; TERMINATE OR?
7970 043072 001403          BEQ      2$      ;; BR IF YES

```

7971 043074 122720 000060
7972 043100 001773
7973 043102 005300
7974 043104 010037 043112
7975 043110 104401
7976 043112 000000
7977 043114 012600
7978 043116 012616
7979 043120 000207
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994 043122
7995 043122 010046
7996 043124 010146
7997 043126 010246
7998 043130 005046
7999 043132 016601 000012
8000 043136 100002
8001 043140 005216
8002 043142 005401
8003 043144 016602 000014
8004 043150 100002
8005 043152 005316
8006 043154 005402
8007 043156 012746 000021
8008 043162 005000
8009 043164 103001
8010 043166 060200
8011 043170 006000
8012 043172 006001
8013 043174 005316
8014 043176 001372
8015 043200 022616
8016 043202 001403
8017 043204 005400
8018 043206 005401
8019 043210 005600
8020 043212 005726
8021 043214 010066 000012
8022 043220 010166 000010
8023 043224 012602
8024 043226 012601
8025 043230 012600
8026 043232 000207

```

CMPB    #'0',(RO)+    ;; IS THIS AN ASCII "0" ?
BEQ     1$             ;; BR IF YES
2$:      DEC          RO    ;; BACKUP BY "1"
        MOV          RO,3$  ;; SAVE FOR TYPING
        TYPE         PC    ;; GO TYPE
3$:      .WORD        0     ;; ASCIZ POINTER GOES HERE
        MOV          (SP)+,RO  ;; RESTORE RO
        MOV          (SP)+,(SP) ;; RESTORE THE STACK
        RTS          PC    ;; RETURN
.SBTTL   INTEGER MULTIPLY ROUTINE

;*****
;CALL
;      MOV          MULTIPLIER,-(SP)
;      MOV          MULTIPLICAND,-(SP)
;      JSR          PC,$MULT
;      RETURN      ;; PRODUCT IS ON THE STACK
;
;      STACK      PRODUCT
;      -----
;      TOP        LSB'S
;      +2         MSB'S
;
$MULT:   MOV          RO,-(SP)    ;; PUSH RO ON STACK
        MOV          R1,-(SP)    ;; PUSH R1 ON STACK
        MOV          R2,-(SP)    ;; PUSH R2 ON STACK
        CLR          -(SP)       ;; CLEAR THE SIGN KEY
        MOV          12(SP),R1   ;; GET THE MULTIPLICAND
        BPL          1$         ;; BR IF PLUS
        INC          (SP)       ;; SET THE SIGN KEY
        NEG          R1         ;; MAKE THE MULTIPLICAND POSTIVE
1$:      MOV          14(SP),R2   ;; GET THE MULTIPLIER
        BPL          2$         ;; BR IF PLUS
        DEC          (SP)       ;; UPDATE THE SIGN KEY
        NEG          R2         ;; MAKE THE MULTIPLIER POSTIVE
2$:      MOV          #17,-(SP)   ;; SET THE LOOP COUNT
        CLR          RO         ;; SETUP FOR THE MULTIPLY LOOP
        BCC          4$         ;; DON'T ADD IF MULTIPLICAND = 0
3$:      ADD          R2,RO
        ROR          RO         ;; POSITION THE PARTIAL PRODUCT AND
        ROR          R1         ;; THE MULTIPLICAND
        DEC          (SP)       ;; HAS ALL BITS OF THE MULTIPLICAND BEEN DONE?
        BNE          3$        ;; BR IF NO
        CMP          (SP)+,(SP) ;; SHOULD PRODUCT BE NEGATIVE?
        BEQ          5$        ;; GO TO EXIT IF NO
        NEG          RO         ;; YES--SO MAKE IT SO
        NEG          R1
        SBC          RO
4$:      TST          (SP)+
        MOV          RO,12(SP)   ;; CLEAR SIGN INFO. OFF OF STACK
        MOV          R1,10(SP)   ;; PUT THE PRODUCT ON THE STACK (MSB'S)
        MOV          (SP)+,R2    ;; LSB'S
        MOV          (SP)+,R1    ;; POP STACK INTO R2
        MOV          (SP)+,R0    ;; POP STACK INTO R1
        MOV          (SP)+,RO    ;; POP STACK INTO RO
        RTS          PC

```

M12

8027
8028
8029
8030
8031
8032
8033
8034
8035
8036
8037
8038
8039
8040
8041
8042
8043
8044 043234
8045 043234 010046
8046 043236 010146
8047 043240 010246
8048 043242 010346
8049 043244 010446
8050 043246 010546
8051 043250 016646 000022
8052 043254 016646 000022
8053 043260 016646 000022
8054 043264 016646 000022
8055 043270 000002
8056
8057
8058
8059
8060 043272
8061 043272 012666 000022
8062 043276 012666 000022
8063 043302 012666 000022
8064 043306 012666 000022
8065 043312 012605
8066 043314 012604
8067 043316 012603
8068 043320 012602
8069 043322 012601
8070 043324 012600
8071 043326 000002
8072
8073
8074
8075
8076
8077
8078
8079
8080 043330 010046
8081 043332 016600 000002
8082 043336 005740

.SBTTL SAVE AND RESTORE RO-R5 ROUTINES

```

;*****
;SAVE RO-R5
;CALL:
; SAVREG
;UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
;
;TOP---(+16)
; +2---(+18)
; +4---R5
; +6---R4
; +8---R3
; +10---R2
; +12---R1
; +14---R0

```

\$SAVREG:

```

MOV RO,-(SP)      ;; PUSH RO ON STACK
MOV R1,-(SP)      ;; PUSH R1 ON STACK
MOV R2,-(SP)      ;; PUSH R2 ON STACK
MOV R3,-(SP)      ;; PUSH R3 ON STACK
MOV R4,-(SP)      ;; PUSH R4 ON STACK
MOV R5,-(SP)      ;; PUSH R5 ON STACK
MOV 22(SP),-(SP)  ;; SAVE PS OF MAIN FLOW
MOV 22(SP),-(SP)  ;; SAVE PC OF MAIN FLOW
MOV 22(SP),-(SP)  ;; SAVE PS OF CALL
MOV 22(SP),-(SP)  ;; SAVE PC OF CALL
RTI

```

;*RESTORE RO-R5

;*CALL:

;* RESREG

\$RESREG:

```

MOV (SP)+,22(SP)  ;; RESTORE PC OF CALL
MOV (SP)+,22(SP)  ;; RESTORE PS OF CALL
MOV (SP)+,22(SP)  ;; RESTORE PC OF MAIN FLOW
MOV (SP)+,22(SP)  ;; RESTORE PS OF MAIN FLOW
MOV (SP)+,R5      ;; POP STACK INTO R5
MOV (SP)+,R4      ;; POP STACK INTO R4
MOV (SP)+,R3      ;; POP STACK INTO R3
MOV (SP)+,R2      ;; POP STACK INTO R2
MOV (SP)+,R1      ;; POP STACK INTO R1
MOV (SP)+,R0      ;; POP STACK INTO R0
RTI

```

.SBTTL TRAP DECODER

```

;*****
;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
;GO TO THAT ROUTINE.

```

```

$TRAP: MOV RO,-(SP)      ;; SAVE RO
        MOV 2(SP),RO     ;; GET TRAP ADDRESS
        TST -(RO)        ;; BACKUP BY 2

```

```

8083 043340 111000      MOV      (R0),R0      ;;GET RIGHT BYTE OF TRAP
8084 043342 006300      ASL      R0          ;;POSITION FOR INDEXING
8085 043344 016000 043364  MOV      $TRPAD(P0),R0  ;;INDEX TO TABLE
8086 043350 000200      RTS      R0          ;;GO TO ROUTINE
8087
8088
8089
8090      ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
8091 043352 011646 000004 000002 $TRAP2: MOV      (SP),-(SP)      ;;MOVE THE PC DOWN
8092 043354 016666      MOV      4(SP),2(SP)      ;;MOVE THE PSW DOWN
8093 043362 000002      RTI          ;;RESTORE THE PSW
8094
8095      .SBTTL  TRAP TABLE
8096
8097      ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
8098      ;*BY THE "TRAP" INSTRUCTION
8099
8100      ; ROUTINE
8101      ;-----
8102 043364 043352 $TRPAD: .WORD  $TRAP2      TRAP+1(104401) TTY TYPEOUT ROUTINE
8103 043366 037700      $TYPE      ;;CALL=TYPE      TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
8104 043370 040700      $TYPOC     ;;CALL=TYPOC     TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
8105 043372 040654      $TYP0S     ;;CALL=TYP0S     TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
8106 043374 040714      $TYP0N     ;;CALL=TYP0N     TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
8107 043376 040162      $TYPDS     ;;CALL=TYPDS
8108
8109 043400 041470      $GTSWR      ;;CALL=GTSWR      TRAP+6(104406) GET SOFT-SWR SETTING
8110
8111 043402 041400      $CKSWR      ;;CALL=CKSWR      TRAP+7(104407) TEST FOR CHANGE IN SOFT-SWR
8112 043404 041742      $RDCHR      ;;CALL=RDCHR      TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
8113 043406 042032      $RDLIN      ;;CALL=RDLIN      TRAP+11(104411) TTY TYPEIN STRING ROUTINE
8114 043410 042352      $RDOCT      ;;CALL=RDOCT      TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
8115 043412 043234      $SAVREG     ;;CALL=SAVREG     TRAP+13(104413) SAVE R0-R5 ROUTINE
8116 043414 043272      $RESREG     ;;CALL=RESREG     TRAP+14(104414) RESTORE R0-R5 ROUTINE
8117 043416 036230      $SCOP1$     ;;CALL=SCOP1      TRAP+15(104415) INTERNAL LOOP ON ERROR
8118

```

.SBTTL SERVICE MESSAGES

8119					
8120					
8121					
8122					
8123	043420	005015	047125	041111	MSG1: .ASCII <CR><LF>/UNIBUS RK06-RK07 DRIVE DIAGNOSTIC--PART 2/
8124	043426	051525	051040	030113	
8125	043434	026466	045522	033460	
8126	043442	042040	044522	042526	
8127	043450	042040	040511	047107	
8128	043456	051517	044524	026503	
8129	043464	050055	051101	020124	
8130	043472	062			
8131	043473	015	041412	051132	.ASCII <CR><LF>/CZR6IED/<CR><LF>
8132	043500	044466	030105	005015	
8133	043506	005015	025011	025052	.ASCII <CR><LF>/ ***** CAUTION *****/<CR><LF>
8134	043514	025052	041440	052501	
8135	043522	044524	047117	025040	
8136	043530	025052	025052	005015	
8137	043536	005015	044124	051511	.ASCII <CR><LF>/THIS PROGRAM SHOULD BE HALTED ONLY BY TYPING CONTROL-C/
8138	043544	050040	047522	051107	
8139	043552	046501	051440	047510	
8140	043560	046125	020104	041040	
8141	043566	020105	040510	052114	
8142	043574	042105	047440	046116	
8143	043602	020131	054502	052040	
8144	043610	051131	047111	020107	
8145	043616	04.503	052116	047522	
8146	043624	026514	103		
8147	043627	015	047412	044124	.ASCII <CR><LF>/OTHERWISE, CARTRIDGE FORMATTING AND, OR THE DRIVE/
8148	043634	051105	044527	042523	
8149	043642	020054	040503	052122	
8150	043650	044522	043504	020105	
8151	043656	047506	046522	052101	
8152	043664	044524	043516	040440	
8153	043672	042116	020054	051117	
8154	043700	052040	042510	042040	
8155	043706	044522	042526		
8156	043712	005015	040515	020131	.ASCII <CR><LF>/MAY BE LEFT IN AN UNDETERMINED STATE/
8157	043720	042502	046040	043105	
8158	043726	020124	047111	040440	
8159	043734	020116	047125	042504	
8160	043742	042524	046522	047111	
8161	043750	042105	051440	040524	
8162	043756	042524			
8163	043760	005015	047111	052111	.ASCII <CR><LF>/INITIALLY, DRIVES TO BE TESTED SHOULD HAVE:/<CR><LF>
8164	043766	040511	046114	026131	
8165	043774	042040	044522	042526	
8166	044002	020123	047524	041040	
8167	044010	020105	042524	052123	
8168	044016	042105	051440	047510	
8169	044024	046125	020104	040510	
8170	044032	042526	006472	012	
8171	044037	015	040412	020056	.ASCII <CR><LF>/A. HEADS MANUALLY LOADED/
8172	044044	042510	042101	020123	
8173	044052	040515	052516	046101	
8174	044060	054514	046040	040517	

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 158
SERVICE MESSAGES

SEQ 0158

8175	044066	042504	104		
8176	044071	015	041012	020056	.ASCII <CR><LF>/B. CORRECT PORT SELECTED/
8177	044076	047503	051122	041505	
8178	044104	020124	047520	052122	
8179	044112	051440	046105	041505	
8180	044120	042524	104		
8181	044123	015	041412	020056	.ASCII <CR><LF>/C. WRITE LOCK DISABLED/
8182	044130	051127	052111	020105	
8183	044136	047514	045503	042040	
8184	044144	051511	041101	042514	
8185	044152	104			
8186	044153	015	042012	020056	.ASCII <CR><LF>/D. DRIVE READY INDICATOR ON/<CR><LF>
8187	044160	051104	053111	020105	
8188	044166	042522	042101	020131	
8189	044174	047111	044504	040503	
8190	044202	047524	020122	047117	
8191	044210	005015			
8192	044212	005015	051104	053111	.ASCII <CR><LF>/DRIVES NOT TO BE TESTED MUST HAVE/
8193	044220	051505	047040	052117	
8194	044226	052040	020117	042502	
8195	044234	052040	051505	042524	
8196	044242	020104	052515	052123	
8197	044250	044040	053101	105	
8198	044255	015	041012	052117	.ASCIZ <CR><LF>/BOTH PORTS DESELECTED/<CR><LF>
8199	044262	020110	047520	052122	
8200	044270	020123	042504	042523	
8201	044276	042514	052103	042105	
8202	044304	005015	000		
8203					
8204	044307	015	041412	040510	MSG2: .ASCII <CR><LF>/CHANGE XXDP PACK/
8205	044314	043516	020105	054130	
8206	044322	050104	050040	041501	
8207	044330	113			
8208	044331	015	041412	042514	.ASCIZ <CR><LF>/CLEAR LOC 40,RESTART PROGRAM/
8209	044336	051101	046040	041517	
8210	044344	032040	026060	042522	
8211	044352	052123	051101	020124	
8212	044360	051120	043517	040522	
8213	044366	000115			
8214	044370	005015	051104	053111	MSG3: .ASCIZ <CR><LF>/DRIVE(S) TO BE TESTED: /
8215	044376	024105	024523	052040	
8216	044404	020117	042502	052040	
8217	044412	051505	042524	035104	
8218	044420	000040			
8219	044422	005015	054524	042520	MSG4: .ASCIZ <CR><LF>/TYPE BUSS ADDRESS IF NOT 177440: /
8220	044430	041040	051525	020123	
8221	044436	042101	051104	051505	
8222	044444	020123	043111	047040	
8223	044452	052117	030440	033467	
8224	044460	032064	035060	020040	
8225	044466	000			
8226	044467	015	052012	050131	MSG5: .ASCIZ <CR><LF>/TYPE CONTROLLER INTERRUPT VECTOR IF NOT 210: /
8227	044474	020105	047503	052116	
8228	044502	047522	046114	051105	
8229	044510	044440	052116	051105	
8230	044516	052522	052120	053040	

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 159
SERVICE MESSAGES

SEQ 0159

8231	044524	041505	047524	020122	
8232	044532	043111	047040	052117	
8233	044540	031040	030061	020072	
8234	044546	000040			
8235	044550	005015	047111	042524	MSG6: .ASCIZ <CR><LF>/INTERRUPT OCCURRED AT PC=
8236	044556	051122	050125	020124	
8237	044564	041517	052503	051122	
8238	044572	042105	040440	020124	
8239	044600	041520	000075		
8240	044604	005015	051104	053111	MSG7: .ASCIZ <CR><LF>/DRIVE 0 WILL NOT BE TESTED/
8241	044612	020105	020060	044527	
8242	044620	046114	047040	052117	
8243	044628	041040	020105	042524	
8244	044634	052123	042105	000	
8245	044641	015	051012	040505	MSG8: .ASCIZ <CR><LF>/READ DATA WITH OFFSET TEST/<CR><LF>
8246	044646	020104	040504	040524	
8247	044654	053440	052111	020110	
8248	044662	043117	051506	052105	
8249	044670	052040	051505	006524	
8250	044676	000012			
8251	044700	005015	042510	042101	MSG9: .ASCIZ <CR><LF>/HEAD NO./
8252	044706	047040	027117	000	
8253	044713	015	005012	044527	MSG10: .ASCIZ <CR><LF><LF>/WILL TEST DRIVES:/
8254	044720	046114	052040	051505	
8255	044726	020124	051104	053111	
8256	044734	051505	000072		
8257	044740	005015	050012	053517	MSG11: .ASCIZ <CR><LF><LF>/POWER UP RESTART TO TEST 1/<CR><LF>
8258	044746	051105	052440	020120	
8259	044754	042522	052123	051101	
8260	044762	020124	047524	052040	
8261	044770	051505	020124	006461	
8262	044776	000012			
8263	045000	005015	052012	042510	MSG12: .ASCII <CR><LF><LF>/THE ABOVE OFFSET FAILURES ARE NOT ERRORS/
8264	045006	040440	047502	042526	
8265	045014	047440	043106	042523	
8266	045022	020124	040506	046111	
8267	045030	051125	051505	040440	
8268	045036	042522	047040	052117	
8269	045044	042440	051122	051117	
8270	045052	123			
8271	045053	015	041012	052125	.ASCIZ <CR><LF>/BUT INDICATORS OF SURFACE, HEAD, & ELECTRONICS QUALITY/<CR><LF>
8272	045060	044440	042116	041511	
8273	045066	052101	051117	020123	
8274	045074	043117	051440	051125	
8275	045102	040506	042503	044054	
8276	045110	040505	026104	023040	
8277	045116	042440	042514	052103	
8278	045124	047522	044516	051503	
8279	045132	050440	040525	044514	
8280	045140	054524	005015	000	
8281	045145	015	047012	020117	MSG13: .ASCII <CR><LF>/NO L OR P CLOCKS PRESENT/
8282	045152	020114	051117	050040	
8283	045160	041440	047514	045503	
8284	045166	020123	051120	051505	
8285	045174	047105	124		
8286	045177	015	044012	040505	.ASCIZ <CR><LF>/HEAD SWITCHING TIME TEST BYPASSED/

E13

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.PJ1 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 160
SERVICE MESSAGES

SEQ 0160

8287	045204	020104	053523	052111	
8288	045212	044103	047111	020107	
8289	045220	044524	042515	052040	
8290	045226	051505	020124	054502	
8291	045234	040520	051523	042105	
8292	045242	000			
8293	045243	015	041012	050131	MSG14: .ASCIZ <CR><LF>/BYPASSING DRIVE /
8294	045250	051501	044523	043516	
8295	045256	042040	044522	042526	
8296	045264	000040			
8297	045266	005015	042012	044522	MSG15: .ASCIZ <CR><LF><LF>/DRIVE /
8298	045274	042526	000040		
8299	045300	005015	051104	053111	MSG16: .ASCIZ <CR><LF>/DRIVE SERIAL NO. /
8300	045306	020105	042523	044522	
8301	045314	046101	047040	027117	
8302	045322	000040			
8303	045324	005015	040503	052122	MSG17: .ASCIZ <CR><LF>/CARTRIDGE SERIAL NO. /
8304	045332	044522	043504	020105	
8305	045340	042523	044522	046101	
8306	045346	047040	027117	000040	
8307	045354	005015	040412	047502	MSG26: .ASCIZ <CR><LF><LF>/ABORTING BALANCE OF TESTS ON THIS DRIVE/<CR><LF><LF>
8308	045362	052122	047111	020107	
8309	045370	040502	040514	041516	
8310	045376	020105	043117	052040	
8311	045404	051505	051524	047440	
8312	045412	047116	044124	051511	
8313	045420	047040	044522	042526	
8314	045426	005115	000012		
8315	045432	005115	040412	046114	MSG27: .ASCIZ <CR><LF><LF>/ALL DRIVES TESTED/<CR><LF><LF>
8316	045440	042040	044522	042526	
8317	045446	020113	042524	052123	
8318	045454	042105	005015	000012	
8319	045462	005015	047516	053440	MSG37: .ASCIZ <CR><LF> NO WRITE CHECK ERROR AT MAX POSITIVE OFFSET/
8320	045470	044522	042524	041440	
8321	045476	042510	045503	042440	
8322	045504	051122	051117	040440	
8323	045512	020124	040515	020130	
8324	045520	047520	044523	044524	
8325	045526	042526	047440	043106	
8326	045534	042523	000124		
8327	045540	005015	047516	053440	MSG38: .ASCIZ <CR><LF>/NO WRITE CHECK ERROR AT MAX NEGATIVE OFFSET/<CR><LF>
8328	045546	044522	042524	041440	
8329	045554	042510	045503	042440	
8330	045562	051122	051117	040440	
8331	045570	020124	040515	020130	
8332	045576	042516	040507	044524	
8333	045604	042526	047440	043106	
8334	045612	042523	006524	000012	
8335	045620	005015	051127	052111	MSG39: .ASCIZ <CR><LF>/WRITE CHECK FAILURE AT OFFSET =/
8336	045626	020105	044103	041505	
8337	045634	020113	040506	046111	
8338	045642	051125	020105	052101	
8339	045650	047440	043106	042523	
8340	045656	020124	000075		
8341	045662	005015	047503	046125	MSG40: .ASCII <CR><LF> /COULD NOT READ BAD SECTOR INFO ON CYL 410/
8342	045670	020104	047516	020124	

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 161
SERVICE MESSAGES

SEQ 0161

8343	045676	042522	042101	041040
8344	045704	042101	051440	041505
8345	045712	047524	020122	047111
8346	045720	047506	047440	020116
8347	045726	054503	020114	030464
8348	045734	060		
8349	045735	015	047412	020122
8350	045742	046101	043511	046516
8351	045750	047105	020124	040503
8352	045756	052122	044522	043504
8353	045764	020105	051525	042105
8354	045772	005015	000	
8355	045775	015	050012	047522
8356	046002	051107	046501	040440
8357	046010	047502	052122	050040
8358	046016	047105	044504	043516
8359	046024	027056	050056	042514
8360	046032	051501	020105	040527
8361	046040	052111	000	
8362	046043	015	044012	046101
8363	046050	020124	042520	042116
8364	046056	047111	027107	027056
8365	046064	046120	040505	042523
8366	046072	053440	044501	000124
8367	046100	005015	051120	043517
8368	046106	040522	020115	041101
8369	046114	051117	042524	000104
8370	046122	005015	050103	020125
8371	046130	040510	052114	042105
8372	046136	000		
8373				
8374				
8375				
8376	046137	015	042412	051122
8377	046144	051117	020054	047117
8378	046152	054514	030040	052040
8379	046160	051110	020125	020067
8380	046166	046101	047514	042527
8381	046174	026104	052040	054522
8382	046202	040440	040507	047111
8383	046210	005015	000	
8384	046213	104	044522	042526
8385	046220	021440	044440	020116
8386	046226	045522	051503	020062
8387	046234	040503	047116	052117
8388	046242	041040	020105	042522
8389	046250	042101	041040	041501
8390	046256	020113	047503	051122
8391	046264	041505	046124	020131
8392	046272	047111	051040	046513
8393	046300	031122	000	
8394	046303	015	040412	047502
8395	046310	052122	052040	051505
8396	046316	051524	027056	052456
8397	046324	042516	050130	041505
8398	046332	042524	020104	044524

.ASCIZ <CR><LF>/OR ALIGNMENT CARTRIDGE USED/<CR><LF>

MSG74: .ASCIZ <CR><LF>/PROGRAM ABORT PENDING...PLEASE WAIT/

MSG75: .ASCIZ <CR><LF>/HALT PENDING...PLEASE WAIT/

MSG76: .ASCIZ <CR><LF>/PROGRAM ABORTED/

MSG77: .ASCIZ <CR><LF>/CPU HALTED/

.SBTTL ERROR MESSAGES

EM1: .ASCIZ <CR><LF>/ERROR, ONLY 0 THRU 7 ALLOWED, TRY AGAIN/<CR><LF>

EM2: .ASCIZ /DRIVE # IN RKCS2 CANNOT BE READ BACK CORRECTLY IN RKMR2/

EM3: .ASCIZ <CR><LF>/ABORT TESTS...UNEXPECTED TIME OUT AT PC=/

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 162
ERROR MESSAGES

SEQ 0.62

8399	046340	042515	047440	052125	
8400	046346	040440	020124	041520	
8401	046354	000075			
8402	046356	005015	041101	051117	EM4: .ASCIZ <CR><LF>/ABORT TESTS...UNEXPECTED INTERRUPT AT PC=
8403	046364	020127	042524	052123	
8404	046372	027123	027056	047125	
8405	046400	054105	042520	052103	
8406	046406	042105	044440	052116	
8407	046414	051105	052522	052120	
8408	046422	040440	020124	041520	
8409	046430	000075			
8410	046432	042115	020123	042523	EM5: .ASCIZ /MDS SET IN RKCS2/
8411	046440	020124	047111	051040	
8412	046446	041513	031123	000	
8413	046453	125	042506	051440	EM6: .ASCIZ /UFE SET IN RKCS2/
8414	046460	052105	044440	020116	
8415	046466	045522	051503	000062	
8416	046474	051104	020101	047111	EM7: .ASCIZ /DRA IN RKDS & NED IN RKCS2 RESET; WRONG PORT SELECTED?/
8417	046502	051040	042113	020123	
8418	046510	020046	042516	020104	
8419	046516	047111	051040	041513	
8420	046524	031123	051040	051505	
8421	046532	052105	020073	051127	
8422	046540	047117	020107	047520	
8423	046546	052122	051440	046105	
8424	046554	041505	042524	037504	
8425	046562	000			
8426	046563	104	044522	042526	EM8: .ASCIZ /DRIVE PRESENT BUT NOT SPECIFIED BY OPERATOR/
8427	046570	050040	042522	042523	
8428	046576	052116	041040	052125	
8429	046604	047040	052117	051440	
8430	046612	042520	044503	044506	
8431	046620	042105	041040	020131	
8432	046626	050117	051105	052101	
8433	046634	051117	000		
8434	046637	104	044522	042526	EM9: .ASCIZ /DRIVE NOT PRESENT BUT SPECIFIED BY OPERATOR/
8435	046644	047040	052117	050040	
8436	046652	042522	042523	052116	
8437	046660	041040	052125	051440	
8438	046666	042520	044503	044506	
8439	046674	042105	041040	020131	
8440	046702	050117	051105	052101	
8441	046710	051117	000		
8442	046713	101	047502	052122	EM10: .ASCIZ /ABORT TESTS...CANNOT REFERENCE CONTROLLER REGISTER/
8443	046720	052040	051505	051524	
8444	046726	027056	041456	047101	
8445	046734	057516	020124	042522	
8446	046742	042506	042522	041516	
8447	046750	020105	047503	052116	
8448	046756	047522	046114	051105	
8449	046764	051040	043505	051511	
8450	046772	042524	000122		
8451	046776	051104	020101	047111	EM11: .ASCIZ /DRA IN RKDS & NED IN RKCS2 BOTH SET/
8452	047004	051040	042113	020123	
8453	047012	020046	042516	020104	
8454	047020	047111	051040	041513	

8455	047026	031123	041040	052117	
8456	047034	020110	042523	000124	
8457	047042	047503	052116	047522	EM12: .ASCIZ /CONTROLLER NOT READY IN RKCS1
8458	047050	046114	051105	047040	
8459	047056	052117	051040	040505	
8460	047064	054504	044440	020116	
8461	047072	045522	051503	000061	
8462	047100	047516	040440	052124	EM13: .ASCIZ /NO ATTN IN RKASOF/
8463	047106	020116	047111	051040	
8464	047114	040513	047523	000106	
8465	047122	047123	054105	042520	EM14: .ASCIZ /UNEXPECTED MEMORY PARITY TRAP/
8466	047130	052103	042105	046440	
8467	047136	046505	051117	020131	
8468	047144	040520	044522	054524	
8469	047152	052040	040522	000120	
8470	047160	045522	041504	023040	EM15: .ASCII /RKDC & RKDA INDICATE THAT WCE OCCURRED AT/
8471	047166	051040	042113	020101	
8472	047174	047111	044504	040503	
8473	047202	042524	052040	040510	
8474	047210	020124	041527	020105	
8475	047216	041517	052503	051122	
8476	047224	042105	040440	124	
8477	047231	015	041412	046131	.ASCIZ <CR><LF>/CYL 411, TRACK 2, SECTOR 21/
8478	047236	032040	030461	020054	
8479	047244	051124	041501	020113	
8480	047252	026062	051440	041505	
8481	047260	047524	020122	030462	
8482	047266	000			
8483	047267	103	047101	047516	EM16: .ASCIZ /CANNOT READ BAD SECTOR INFORMATION/
8484	047274	020124	042522	042101	
8485	047302	041040	042101	051440	
8486	047310	041505	047524	020122	
8487	047316	047111	047506	046522	
8488	047324	052101	047511	000116	
8489	047332	042515	051523	043501	EM17: .ASCIZ /MESSAGE A0 ERROR/
8490	047340	020105	030101	042440	
8491	047346	051122	051117	000	
8492	047353	115	051505	040523	EM18: .ASCIZ /MESSAGE B0 ERROR/
8493	047360	042507	041040	020060	
8494	047366	051105	047522	000122	
8495	047374	042515	051523	043501	EM19: .ASCIZ /MESSAGE A1 ERROR/
8496	047402	020105	030501	042440	
8497	047410	051122	051117	000	
8498	047415	115	051505	040523	EM20: .ASCIZ /MESSAGE B1 ERROR/
8499	047422	042507	041040	020061	
8500	047430	051105	047522	000122	
8501	047436	042503	051122	051440	EM21: .ASCIZ /CERR SET IN RKCS1/
8502	047444	052105	044440	020116	
8503	047452	045522	051503	000061	
8504	047460	047516	042040	044522	EM22: .ASCII /NO DRIVES FOUND IN DEVICE MAP (\$DEVN)/<CR> LF/
8505	047466	042526	020123	047506	
8506	047474	047125	020104	047111	
8507	047502	042040	053105	041511	
8508	047510	020105	040515	020120	
8509	047516	022050	042504	046526	
8510	047524	006451	012		

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 164
ERROR MESSAGES

SEQ 0164

8511	047527	123	052105	050125		.ASCIZ /SETUP CORRECTLY AND RESTART/(<CR><LF>)
8512	047534	041440	051117	042522		
8513	047542	052103	054514	040440		
8514	047550	042116	051040	051505		
8515	047556	040524	052122	005015		
8516	047564	000				
8517	047565	116	020117	051104	EM23:	.ASCII /NO DRIVES FOUND ON BUSS/(<CR><LF>)
8518	047572	053111	051505	043040		
8519	047600	052517	042116	047440		
8520	047606	020116	052502	051523		
8521	047614	005015				
8522	047616	042523	052524	020120		.ASCIZ /SETUP CORRECTLY AND PRESS 'CONTINUE'/(<CR><LF>)
8523	047624	047503	051122	041505		
8524	047632	046124	020131	047101		
8525	047640	020104	051120	051505		
8526	047646	020123	041447	047117		
8527	047654	044524	052516	023505		
8528	047662	005015	000			
8529	047665	126	046117	053040	EM24:	.ASCIZ /VOL VALID NOT SET IN RKMR2/
8530	047672	046101	042111	047040		
8531	047700	052117	051440	052105		
8532	047706	044440	020116	045522		
8533	047714	051115	000062			
8534	047720	005015	042504	042524	EM25:	.ASCIZ (<CR><LF>)/DETECTED 10 BAD SECTORS...ABORTING TEST/
8535	047726	052103	042105	030440		
8536	047734	020060	040502	020104		
8537	047742	042523	052103	051117		
8538	047750	027123	027056	041101		
8539	047756	051117	044524	043516		
8540	047764	052040	051505	000124		
8541	047772	042504	042524	052103	EM26:	.ASCIZ /DETECTED BSE BUT NOT LISTED IN BAD SECTOR FILE/
8542	050000	042105	041040	042523		
8543	050006	041040	052125	047040		
8544	050014	052117	046040	051511		
8545	050022	042524	020104	047111		
8546	050030	041040	042101	051440		
8547	050036	041505	047524	020122		
8548	050044	044506	042514	000		
8549	050051	04	052105	041505	EM27:	.ASCII /DETECTED BSE IN READ COMMAND/
8550	050056	042524	020104	051502		
8551	050064	020105	047111	051040		
8552	050072	040505	020104	047503		
8553	050100	046515	047101	104		
8554	050105	015	041012	052125		.ASCIZ (<CR><LF>)/BUT NOT IN PREVIOUS WRITE COMMAND TO SAME SECTOR/
8555	050112	047040	052117	044440		
8556	050120	020116	051120	053105		
8557	050126	047511	051525	053440		
8558	050134	044522	042524	041440		
8559	050142	04517	040515	042116		
8560	050150	052040	020117	040523		
8561	050156	042515	051440	041505		
8562	050164	047524	000122			
8563	050170	054503	020114	042101	EM36:	.ASCIZ /CYL ADDR IN RKMR3 NOT SAME AS RKDC/
8564	050176	051104	044440	020116		
8565	050204	045522	051115	020063		
8566	050212	047516	020124	040523		

J13

CZREIEO UNIBUS RK6 DR PRT2
CZREIE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 165
ERROR MESSAGES

SEQ 0165

8567	050220	042515	040440	020123	
8568	050226	045522	041504	000	
8569	050233	103	046131	042040	EM39: .ASCIZ /CYL DIFF & OFFSET IN RKMR2 NOT CLEARED/
8570	050240	043111	020106	020046	
8571	050246	043117	051506	052105	
8572	050254	044440	020116	045522	
8573	050262	051115	020062	047516	
8574	050270	020124	046103	040505	
8575	050276	042522	000104		
8576	050302	054503	020114	042101	EM40: .ASCIZ /CYL ADDR IN RKMR3 NOT CLEARED/
8577	050310	051104	044440	020116	
8578	050316	045522	051115	020063	
8579	050324	047516	020124	046103	
8580	050332	040505	042522	000104	
8581	050340	054503	020114	042101	EM41: .ASCIZ /CYL ADDR IN B2 DID NOT REMAIN CLEARED/
8582	050346	051104	044440	020116	
8583	050354	031102	042040	042111	
8584	050362	047040	052117	051040	
8585	050370	046505	044501	020116	
8586	050376	046103	040505	042522	
8587	050404	000104			
8588	050406	052101	047124	047040	EM55: .ASCIZ /ATTN NOT CLEARED IN RKASOF/
8589	050414	052117	041440	042514	
8590	050422	051101	042105	044440	
8591	050430	020116	045522	051501	
8592	050436	043117	000		
8593	050441	104	052114	051440	EM63: .ASCIZ /DLT SET IN RKCS2/
8594	050446	052105	044440	020116	
8595	050454	045522	051503	000062	
8596	050462	042522	042101	044040	EM65: .ASCIZ /READ HEADER ERROR/
8597	050470	040505	042504	020122	
8598	050476	051105	047522	000122	
8599	050504	046101	043511	046516	EM69: .ASCIZ /ALIGNMENT CARTRIDGE USED/
8600	050512	047105	020124	040503	
8601	050520	052122	044522	043504	
8602	050526	020105	051525	042105	
8603	050534	000			
8604	050535	103	047524	051440	EM73: .ASCIZ /CTO SET IN RKCS1/
8605	050542	052105	044440	020116	
8606	050550	045522	051503	000061	
8607	050556	052122	020132	047516	EM74: .ASCIZ /RTZ NOT SET IN RKMR2/
8608	050564	020124	042523	020124	
8609	050572	047111	051040	046513	
8610	050600	031122	000		
8611	050603	116	042105	051440	EM79: .ASCIZ /NED SET IN RKCS2/
8612	050610	052105	044440	020116	
8613	050616	045522	051503	000062	
8614	050624	051127	052111	020105	EM80: .ASCIZ /WRITE CHECK ERROR SET IN RKCS2/
8615	050632	044103	041505	020113	
8616	050640	051105	047522	020122	
8617	050646	042523	020124	047111	
8618	050654	051040	041513	031123	
8619	050662	000			
8620	050663	127	044522	042524	EM81: .ASCIZ /WRITE CHECK COMMAND NOT FUNCTIONING/
8621	050670	041440	042510	045503	
8622	050676	041440	046517	040515	

K13

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 166
ERROR MESSAGES

SEQ 0166

8623	050704	042116	047040	052117	
8624	050712	043040	047125	052103	
8625	050720	047511	044516	043516	
8626	050726	000			
8627	050727	122	040505	020104	EM82: .ASCIZ /READ DATA DID NOT COMPARE WITH WRITE DATA/
8628	050734	040504	040524	042040	
8629	050742	042111	047040	052117	
8630	050750	041440	046517	040520	
8631	050756	042522	053440	052111	
8632	050764	020110	051127	052111	
8633	050772	020105	040504	040524	
8634	051000	000			
8635	051001	104	052101	020101	EM83: .ASCIZ /DATA CHECK ERROR SET IN RKER/
8636	051006	044103	041505	020113	
8637	051014	051105	047522	020122	
8638	051022	042523	020124	047111	
8639	051030	051040	042513	000122	
8640	051036	044127	046111	020105	EM84: .ASCIZ /WHILE WAITING FOR CONTR READY OR AFTER CONTR READY REC'D/
8641	051044	040527	052111	047111	
8642	051052	020107	047506	020122	
8643	051060	047503	052116	020122	
8644	051066	042522	042101	020131	
8645	051074	051117	040440	052106	
8646	051102	051105	041440	047117	
8647	051110	051124	051040	040505	
8648	051116	054504	051040	041505	
8649	051124	042047	000		
8650	051127	117	043106	042523	EM85: .ASCIZ /OFFSET STATUS BIT IN RKMR2 CLEARED/
8651	051134	020124	052123	052101	
8652	051142	051525	041040	052111	
8653	051150	044440	020116	045522	
8654	051156	051115	020062	046103	
8655	051164	040505	042522	000104	
8656	051172	043117	051506	022105	EM86: .ASCIZ /OFFSET REG IN A2 NOT = RKASOF/
8657	051200	051040	043505	044440	
8658	051206	020116	031101	047040	
8659	051214	052117	036440	051040	
8660	051222	040513	047523	000106	
8661	051230	044504	020104	047516	EM88: .ASCIZ /DID NOT FIND SECTOR 0 FROM INDEX/
8662	051236	020124	044506	042116	
8663	051244	051440	041505	047524	
8664	051252	020122	020060	051106	
8665	051260	046517	044440	042116	
8666	051266	054105	000		
8667	051271	122	040505	044504	EM93: .ASCIZ READING WRONG CYLINDER # IN HEADER...MISPOSITION/
8668	051276	043516	053440	047522	
8669	051304	043516	041440	046131	
8670	051312	047111	042504	020122	
8671	051320	020043	047111	044040	
8672	051326	040505	042504	027122	
8673	051334	027056	044515	050123	
8674	051342	051517	052111	047511	
8675	051350	000116			
8676	051352	043117	051506	052105	EM94: .ASCIZ /OFFSET IT IN A2 NOT CLEARED.
8677	051360	044440	020124	047111	
8678	051366	040440	020062	047516	

CZREIED UNIBUS RK6 DR PRT2
CZREIE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 167
ERROR MESSAGES

SEQ 0167

8679	051374	020124	046103	040505	
8680	051402	042522	000104		
8681	051406	047506	046522	052101	EM95: .ASCIZ /FORMAT BIT NOT SET IN RKMR2/
8682	051414	041040	052111	047040	
8683	051422	052117	051440	052105	
8684	051430	044440	020116	045522	
8685	051436	051115	000062		
8686	051442	040503	047116	052117	EM96: .ASCIZ /CANNOT FIND SECTOR 23(8)/
8687	051450	043040	047111	020104	
8688	051456	042523	052103	051117	
8689	051464	031040	024063	024470	
8690	051472	000			
8691	051473	110	040505	020104	EM97: .ASCIZ /HEAD SWITCHING REQ'D ANOTHER FULL REVOLUTION OF DISK/
8692	051500	053523	052111	044103	
8693	051506	047111	020107	042522	
8694	051514	023521	020104	047101	
8695	051522	052117	042510	020122	
8696	051530	052506	046114	051040	
8697	051536	053105	046117	052125	
8698	051544	047511	020116	043117	
8699	051552	042040	051511	000113	
8700	051560	051104	053111	020105	EM100: .ASCIZ /DRIVE OFF TRACK SET IN RKMR3/
8701	051566	043117	020106	051124	
8702	051574	041501	020113	042523	
8703	051602	020124	047111	051040	
8704	051610	046513	031522	000	
8705	051615	104	042111	047040	EM101: .ASCIZ /DID NOT GO TO CYLINDER 10/
8706	051622	052117	043440	020117	
8707	051630	047524	041440	046131	
8708	051636	047111	042504	020122	
8709	051644	030061	000		
8710					
8711					.SBTTL DATA HEADERS
8712					
8713	051647	124	051505	020124	DH1: .ASCIZ /TEST NO. PC/
8714	051654	047516	020056	050040	
8715	051662	000103			
8716	051664	045522	051115	004461	DH2: .ASCIZ /RKMR1 RKMR2 RKMR3 RKER RKDS RKCS1 RKCS2/
8717	051672	045522	051115	004462	
8718	051700	045522	051115	004463	
8719	051706	045522	051105	051011	
8720	051714	042113	004523	045522	
8721	051722	051503	004461	045522	
8722	051730	051503	000062		
8723	051734	045522	041527	051011	DH3: .ASCIZ /RKWC RKBA RKDA RKASOF RKDC RKECPS RKECPT/
8724	051742	041113	004501	045522	
8725	051750	040504	051011	040513	
8726	051756	047523	004506	045522	
8727	051764	041504	051011	042513	
8728	051772	050103	004523	045522	
8729	052000	041505	052120	000	
8730	052005	106	047522	020115	DH6: .ASCIZ /FROM CYL TO CYL CYL DIFF/
8731	052012	054503	020114	052040	
8732	052020	020117	054503	020114	
8733	052026	041440	046131	042040	
8734	052034	043111	000106		

M13

CZR61EO UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 168
DATA HEADERS

SEQ 0168

8735	052040	042524	052123	047040	DH8:	.ASCIZ	/TEST NO.	TRAP PC/
8736	052046	027117	052011	040522				
8737	052054	020120	041520	000				
8738	052061	101	052106	051105	DH9:	.ASCIZ	/AFTER START SPINDLE COMMAND REC'D BY DRIVE/	
8739	052066	051440	040524	052122				
8740	052074	051440	044520	042116				
9741	052102	042514	041440	046517				
8742	052110	040515	042116	051040				
8743	052116	041505	042047	041040				
8744	052124	020131	051104	053111				
8745	052132	000105						
8746	052134	052101	042440	042116	DH10:	.ASCIZ	/AT END OF HEAD LOADING/	
8747	052142	047440	020106	042510				
8748	052150	042101	046040	040517				
8749	052156	044504	043516	000				
8750	052163	105	050130	041505	DH11:	.ASCIZ	/EXPECTED	WAS/
8751	052170	042524	004504	040527				
8752	052176	000123						
8753	052200	047117	051440	041505	DH13:	.ASCIZ	/ON SECTORS 10, 12, 14, 16, 18 OR 20 CYL 410 TRACK 2/	
8754	052206	047524	051522	030440				
8755	052214	026060	030440	026062				
8756	052222	030440	026064	030440				
8757	052230	026066	030440	020070				
8758	052236	051117	031040	020060				
8759	052244	054503	020114	030464				
8760	052252	020060	051124	041501				
8761	052260	020113	000062					
8762	052264	047117	051440	041505	DH14:	.ASCIZ	/ON SECTORS 11, 13, 15, 17, 19 OR 21 CYL 410 TRACK 2/	
8763	052272	047524	051522	030440				
8764	052300	026061	030440	02 063				
8765	052306	030440	026065	030440				
8766	052314	026067	030440	020071				
8767	052322	051117	031040	020061				
8768	052330	054503	020114	030464				
8769	052336	020060	051124	041501				
8770	052344	020113	000062					
8771	052350	043101	042524	020122	DH17:	.ASCIZ	/AFTER RECAL COMMAND/	
8772	052356	042522	040503	020114				
8773	052364	047503	046515	047101				
8774	052372	000104						
8775	052374	043101	042524	020122	DH19:	.ASCIZ	/AFTER PACK COMMAND/	
8776	052402	040520	045503	041440				
8777	052410	046517	040515	042116				
8778	052416	000						
8779	052417	101	052106	051105	DH20:	.ASCIZ	/AFTER SELECT DRIVE COMMAND/	
8780	052424	051440	046105	041505				
8781	052432	020124	051104	053111				
8782	052440	020105	047503	046515				
8783	052446	047101	000104					
8784	052452	043101	042524	020122	DH21:	.ASCIZ	/AFTER SUBSYSTEM CLEAR/	
8785	052460	052523	051502	051531				
8786	052466	042524	020115	046103				
8787	052474	040505	000122					
8788	052500	043101	042524	020122	DH22:	.ASCIZ	/AFTER DRIVE CLEAR COMMAND/	
8789	052506	051104	053111	020105				
8790	052514	046103	040505	020122				

SEQ 0169

Address	Offset	Value	Label	Comment
8791	052522	047503	046515	047101
8792	052530	000104		
8793	052532	043101	042524	020122
8794	052540	043117	051506	052105
8795	052546	041440	046517	040515
8796	052554	042116	000	
8797	052557	101	052106	051105
8798	052564	051440	042505	020113
8799	052572	047503	046515	047101
8800	052600	000104		
8801	052602	043101	042524	020122
8802	052610	042522	042101	042040
8803	052616	052101	020101	047503
8804	052624	046515	047101	000104
8805	052632	043101	042524	020122
8806	052640	051127	052111	020105
8807	052646	040504	040524	041440
8808	052654	046517	040515	042116
8809	052662	000		
8810	052663	011	054105	042520
8811	052670	052103	000	
8812	052673	011	041501	052524
8813	052700	046101	000	
8814	052703	101	052106	051105
8815	052710	051040	040505	020104
8816	052716	042510	042101	051105
8817	052724	041440	046517	040515
8818	052732	042116	000	
8819	052735	101	004460	030102
8820	052742	040411	004461	030502
8821	052750	040411	004462	031102
8822	052756	041011	000063	
8823	052762	043101	042524	020122
8824	052770	051127	052111	020105
8825	052776	044103	041505	020113
8826	053004	047503	046515	047101
8827	053012	000104		
8828	053014	043101	042524	020122
8829	053022	051127	052111	020105
8830	053030	042510	042101	051105
8831	053036	041440	046517	040515
8832	053044	042116	000	
8833	053047	127	051117	021504
8834	053054	044011	040505	042504
8835	053062	020122	040527	020123
8836	053070	051440	047510	046125
8837	053076	020104	042502	000
8838	053103	104	051125	047111
8839	053110	020107	042522	040503
8840	053116	020114	047503	046515
8841	053124	047101	000104	
8842	053130	047117	051440	041505
8843	053136	047524	051522	030040
8844	053144	031054	032054	033054
8845	053152	047444	020122	020070
8846	053160	041440	046131	032040

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 170
DATA HEADERS

SEQ 0170

8847	053166	030061	052040	040522	
8848	053174	045503	031040	000	
8849	053201	117	020116	042523	DH43: .ASCIZ /ON SECTORS 1,3,5,7 OR 9 CYL 410 TRACK 2/
8850	053206	052103	051117	020123	
8851	053214	026061	026063	026065	
8852	053222	020067	051117	034440	
8853	053230	020040	054503	020114	
8854	053236	030464	020060	051124	
8855	053244	041501	020113	000062	
8856	053252	047506	046522	052101	DH44: .ASCIZ /FORMAT & ALL READ-WRITE TESTS WILL BE BYPASSED/
8857	053260	023040	040440	046114	
8858	053266	051040	040505	026504	
8859	053274	051127	052111	020105	
8860	053302	042524	052123	020123	
8861	053310	044527	046114	041040	
8862	053316	020105	054502	040520	
8863	053324	051523	042105	000	
8864	053331	101	052106	051105	DH47: .ASCIZ /AFTER READ HEADER COMMAND WITH MOVEMENT/
8865	053336	051040	040505	020104	
8866	053344	042510	042101	051105	
8867	053352	041440	046517	040515	
8868	053360	042116	053440	052111	
8869	053366	020110	047515	042526	
8870	053374	042515	052116	000	
8871	053401	115	043523	040440	DH49: .ASCIZ /MSG A & B IN RKMR2 & RKMR3 RESP. ARE INVALID/
8872	053406	023040	041040	044440	
8873	053414	020116	045522	051115	
8874	053422	020062	020046	045522	
8875	053430	051115	020063	042522	
8876	053436	050123	020056	051101	
8877	053444	020105	047111	040526	
8878	053452	044514	000104		
8879	053456	043101	042524	020122	DH51: .ASCIZ /AFTER SEEK TO SELF COMMAND/
8880	053464	042523	045505	052040	
8881	053472	020117	042523	043114	
8882	053500	041440	046517	040515	
8883	053506	042116	000		
8884	053511	127	052111	020110	DH52: .ASCIZ /WITH INTENTIONAL MISCOMPARE/
8885	053516	047111	042524	052116	
8886	053524	047511	040516	020114	
8887	053532	044515	041523	046517	
8888	053540	040520	042522	000	
8889	053545	104	051125	047111	DH53: .ASCIZ /DURING OFFSET COMMAND/
8890	053552	020107	043117	051506	
8891	053560	052105	041440	046517	
8892	053566	040515	042116	000	
8893	053573	101	052106	051105	DH54: .ASCIZ /AFTER FORMAT CHANGE AND CONTR READY REC'D/
8894	053600	043040	051117	040515	
8895	053606	020124	044103	047101	
8896	053614	042507	040440	042116	
8897	053622	041440	047117	051124	
8898	053630	051040	040505	054504	
8899	053636	051040	041505	042047	
8900	053644	000			
8901	053645	103	046131	021440	DH56: .ASCIZ /CYL # HEADER WORD 0/
8902	053652	044011	040505	042504	

8903	053660	020122	047527	042122	
8904	053666	030040	000		
8905	053671	101	052106	051105	DMS7: .ASCIZ /AFTER WRITE COMMAND WITH OFFSET/
8906	053676	053440	044522	042524	
8907	053704	041440	046517	040515	
8908	053712	042116	053440	052111	
8909	053720	020110	043117	051506	
8910	053726	052105	000		
8911					
8912					.SBTTL ERROR OUTPUT DATA
8913					
8914		053732			.EVEN
8915	053732	001214	001116		DT1: \$TESTN,\$ERRPC
8916	053736	007400	007402	007404	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8917	053744	007370	007366	007354	
8918	053752	007356			
8919	053754	007360	007362	007364	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8920	053762	007372	007374	007406	
8921	053770	007410			
8922	053772	001214	001334		DT3: \$TESTN,TRAPPC
8923	053776	001214	001116	001344	DT4: \$TESTN,\$ERRPC,FRCYL,TOCYL,CALDIF
8924	054004	001346	001354		
8925	054010	007400	007402	007404	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8926	054016	007370	007366	007354	
8927	054024	007356			
8928	054026	007360	007362	007364	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8929	054034	007372	007374	007406	
8930	054042	007410			
8931	054044	001214	001116	001454	DT6: \$TESTN,\$ERRPC,WD2,WD1
8932	054052	001452			
8933	054054	007400	007402	007404	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8934	054062	007370	007366	007354	
8935	054070	007356			
8936	054072	007360	007362	007364	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8937	054100	007372	007374	007406	
8938	054106	007410			
8939	054110	001214	001116	001472	DT7: \$TESTN,\$ERRPC,WD0CNT,HDWD,TEMP1
8940	054116	001510	007412		
8941	054122	007400	007402	007404	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8942	054130	007370	007366	007354	
8943	054136	007356			
8944	054140	007360	007362	007364	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8945	054146	007372	007374	007406	
8946	054154	007410			
8947	054156	001214	001116	001346	DT8: \$TESTN,\$ERRPC,TOCYL,FRCYL,CALDIF
8948	054164	001344	001354		
8949	054170	007400	007402	007404	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8950	054176	007370	007366	007354	
8951	054204	007356			
8952	054206	007360	007362	007364	HWC,HBA,HDA,HASOF,HDC,HPOS,HPAT
8953	054214	007372	007374	007406	
8954	054222	007410			
8955	054224	001214	001116	001346	DT9: \$TESTN,\$ERRPC,TOCYL,RHTAB
8956	054232	001726			
8957	054234	007400	007402	007404	HMR1,HMR2,HMR3,HER,HDS,HCS1,HCS2
8958	054242	007370	007366	007354	

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 172
ERROR OUTPUT DATA

SEQ 0172

8959	054250	007356				
8960	054252	007360	007362	007364		HWC, HBA, HDA, HASOF, HDC, HPOS, HPAT
8961	054260	007372	007374	007406		
8962	054266	007410				
8963	054270	001214	001116	007444	DT13:	\$TESTN, \$ERRPC, E.A0, E.B0, E.A1, E.B1, H.A0, H.B0, H.A1, H.B1
8964	054276	007446	007450	007452		
8965	054304	007424	007426	007430		
8966	054312	007432				
8967	054314	007400	007402	007404		HMR1, HMR2, HMR3, HER, HDS, HCS1, HCS2
8968	054322	007370	007366	007354		
8969	054330	007356				
8970	054332	007360	007362	007364		HWC, HBA, HDA, HASOF, HDC, HPOS, HPAT
8971	054340	007372	007374	007406		
8972	054346	007410				
8973	054350	001214	001116	007444	DT14:	\$TESTN, \$ERRPC, E.A0, E.B0, E.A1, E.B1, E.A2, E.B2
8974	054356	007446	007450	007452		
8975	054364	007454	007456			
8976	054370	007424	007426	007430		H.A0, H.B0, H.A1, H.B1, H.A2, H.B2
8977	054376	007432	007434	007436		
8978	054404	007400	007402	007404		HMR1, HMR2, HMR3, HER, HDS, HCS1, HCS2
8979	054412	007370	007366	007354		
8980	054420	007356				
8981	054422	007360	007362	007364		HWC, HBA, HDA, HASOF, HDC, HPOS, HPAT
8982	054430	007372	007374	007406		
8983	054436	007410				
8984	054440	001214	001116	007444	DT15:	\$TESTN, \$ERRPC, E.A0, E.B0, E.A1, E.B1, E.A2, E.B2, E.B3
8985	054446	007446	007450	007452		
8986	054454	007454	007456	007462		
8987	054462	007424	007426	007430		H.A0, H.B0, H.A1, H.B1, H.A2, H.B2, H.B3
8988	054470	007432	007434	007436		
8989	054476	007442				
8990	054500	007400	007402	007404		HMR1, HMR2, HMR3, HER, HDS, HCS1, HCS2
8991	054506	007370	007366	007354		
8992	054514	007356				
8993	054516	007360	007362	007364		HWC, HBA, HDA, HASOF, HDC, HPOS, HPAT
8994	054524	007372	007374	007406		
8995	054532	007410				
8996						
8997						
8998						
8999	054534	000003				
9000	054536	002	000		DF1:	3
9001	054540	051664				.BYTE 2,0
9002	054542	007	000			DH2
9003	054544	051734				.BYTE 7,0
9004	054546	007	000			DH3
9005						.BYTE 7,0
9006	054550	000001			DF2:	1
9007	054552	002	000			.BYTE 2,0
9008						
9009	054554	000005			DF3:	5
9010	054556	000	000			.BYTE 0,0
9011	054560	051647				DH1
9012	054562	002	000			.BYTE 2,0
9013	054564	052163				DH11
9014	054566	002	000			.BYTE 2,0

.SBTTL ERROR DATA FORMATS

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 173
ERROR DATA FORMATS

SEQ 0173

9015	054570	051664		DH2	
9016	054572	007	000	.BYTE	7,0
9017	054574	051734		DH3	
9018	054576	007	000	.BYTE	7,0
9019					
9020	05460C	000003		DF4:	3
9021	051602	002	000	.BYTE	2,0
9022	054604	051664		DH2	
9023	054606	007	000	.BYTE	7,0
9024	054610	051734		DH3	
9025	054612	007	000	.BYTE	7,0
9026					
9027	054614	000005		DF5:	5
9028	054616	000	000	.BYTE	0,0
9029	054620	053401		DH49	
9030	054622	000	000	.BYTE	0,0
9031	054624	051647		DH1	
9032	054626	002	000	.BYTE	2,0
9033	054630	051664		DH2	
9034	054632	007	000	.BYTE	7,0
9035	054634	051734		DH3	
9036	054636	007	000	.BYTE	7,0
9037					
9038	054640	000005		DF6:	5
9039	054642	000	000	.BYTE	0,0
9040	054644	051647		DH1	
9041	054646	002	000	.BYTE	2,0
9042	054650	052005		DH6	
9043	054652	003	000	.BYTE	3,0
9044	054654	051664		DH2	
9045	054656	007	000	.BYTE	7,0
9046	054660	051734		DH3	
9047	054662	007	000	.BYTE	7,0
9048					
9049					
9050	054664	000004		DF10:	4
9051	054666	000	000	.BYTE	0,0
9052	054670	051647		DH1	
9053	054672	002	000	.BYTE	2,0
9054	054674	051664		DH2	
9055	054676	007	000	.BYTE	7,0
9056	054700	051734		DH3	
9057	054702	007	000	.BYTE	7,0
9058					
9059	054704	000004		DF14:	4
9060	054706	002	000	.BYTE	2,0
9061	054710	053047		DH40	
9062	054712	003	000	.BYTE	3,0
9063	054714	051664		DH2	
9064	054716	007	000	.BYTE	7,0
9065	054720	051734		DH3	
9066	054722	007	000	.BYTE	7,0
9067					
9068					
9069	054724	000004		DF15:	4
9070	054726	000	000	.BYTE	0,0

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 174
ERROR DATA FORMATS

SEQ 0174

9071	054730	051647		DH1	
9072	054732	002	000	.BYTE	2,0
9073	054734	051664		DH2	
9074	054736	007	000	.BYTE	7,0
9075	054740	051734		DH3	
9076	054742	007	000	.BYTE	7,0
9077					
9078	054744	000005		DF17: 5	
9079	054746	000	000	.BYTE	0,0
9080	054750	053252		DH44	
9081	054752	000	000	.BYTE	0,0
9082	054754	051647		DH1	
9083	054756	002	000	.BYTE	2,0
9084	054760	051664		DH2	
9085	054762	007	000	.BYTE	7,0
9086	054764	051734		DH3	
9087	054766	007	000	.BYTE	7,0
9088	054770	000005		DF20: 5	
9089	054772	000	000	.BYTE	0,0
9090	054774	051647		DH1	
9091	054776	002	000	.BYTE	2,0
9092	055000	053645		DH56	
9093	055002	002	000	.BYTE	2,0
9094	055004	051664		DH2	
9095	055006	007	000	.BYTE	7,0
9096	055010	051734		DH3	
9097	055012	007	000	.BYTE	7,0
9098					
9099	055014	000007		DF21: 7	
9100	055016	000	000	.BYTE	0,0
9101	055020	051647		DH1	
9102	055022	002	000	.BYTE	2,0
9103	055024	052663		DH28	
9104	055026	000	000	.BYTE	0,0
9105	055030	052735		DH31	
9106	055032	004	000	.BYTE	4,0
9107	055034	052673		DH29	
9108	055036	004	000	.BYTE	4,0
9109	055040	051664		DH2	
9110	055042	007	000	.BYTE	7,0
9111	055044	051734		DH3	
9112	055046	007	000	.BYTE	7,0
9113					
9114	055050	000007		DF22: 7	
9115	055052	000	000	.BYTE	0,0
9116	055054	051647		DH1	
9117	055056	002	000	.BYTE	2,0
9118	055060	052663		DH28	
9119	055062	000	000	.BYTE	0,0
9120	055064	052735		DH31	
9121	055066	006	000	.BYTE	6,0
9122	055070	052673		DH29	
9123	055072	006	000	.BYTE	6,0
9124	055074	051664		DH2	
9125	055076	007	000	.BYTE	7,0
9126	055100	051734		DH3	

```

9127 055102 007 000
9128
9129 055104 000007
9130 055106 000 000
9131 055110 051647
9132 055112 002 000
9133 055114 052663
9134 055116 000 000
9135 055120 052735
9136 055122 007 000
9137 055124 052673
9138 055126 007 000
9139 055130 051664
9140 055132 007 000
9141 055134 051734
9142 055136 007 000
9143
9144
9145
9146
9147
9148
9149
9150
9151
9152
9153
9154 055140 104413
9155 055142 113700 001114
9156 055146 042700 177400
9157 055152 005300
9158 055154 006300
9159 055156 006300
9160 055160 006300
9161 055162 062700 007526
9162 055166 012037 055202
9163 055172 001404
9164 055174 104401 001205
9165 055200 104401
9166 055202 000000
9167 055204 012037 055220
9168 055210 001404
9169 055212 104401 001205
9170 055216 104401
9171 055220 000000
9172 055222 012001
9173 055224 001455
9174 055226 005004
9175 055230 012000
9176 055232 012002
9177 055234 001446
9178 055236 005104
9179 055240 104401 001205
9180 055244 112003
9181 055246 105720
9182 055250 005703

```

```

.BYTE 7,0
DF23: 7
.BYTE 0,0
DH1
.BYTE 2,0
DH28
.BYTE 0,0
DH31
.BYTE 7,0
DH29
.BYTE 7,0
DH2
.BYTE 7,0
DH3
.BYTE 7,0

```

```

.EVEN
;*****
;SBTTL TYPE ERROR ROUTINE
;ENTRY JSR PC,TYP ERR
;RETURN RTS PC
;
;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
;ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
;THE ERROR.
;*****
TYPERR: SAVREG
        MOVR    $ITEMB,R0      ;ENTER ERROR NUMBER
        BIC     #177400,R0     ;CLEAR SIGN EXTENSION
        DEC     R0             ;FORM INDEX FOR ERROR TABLE
        ASL     R0
        ASL     R0
        ASL     R0
1$:      ADD     # $ERRTB,R0     ;FORM ADDRESS OF ERROR ENTRY
        MOV     (R0)+,2$       ;GET EM POINTER
        BEQ     3$             ;BRANCH IF THERE ISN'T ONE
        TYPE    , $CRLF        ;TYPE CARRIAGE RETURN LINE FEED
        TYPE    , $CRLF        ;TYPE ERROR MESSAGE (EM)
2$:      .WORD   0              ;EM POINTER GOES HERE
3$:      MOV     (R0)+,4$       ;GET DH POINTER
        BEQ     5$             ;BRANCH IF THERE ISN'T ONE
        TYPE    , $CRLF        ;TYPE CR-LF
        TYPE    , $CRLF        ;TYPE DATA HEADER
4$:      .WORD   0              ;DH POINTER GOES HERE
5$:      MOV     (R0)+,R1       ;GET DT POINTER
        BEQ     20$            ;BRANCH IF THERE ARE NONE
        CLR     R4             ;SET INDENT SWITCH
        MOV     (R0)+,R0       ;GET OF POINTER
        MOV     (R0)+,R2       ;STORE NUMBER OF DH'S
        BEQ     17$            ;DH NUM IS 0-BRANCH
        COM     R4             ;NO INDENT
        TYPE    , $CRLF
10$:     MOV     (R0)+,R3       ;GET & STORE NUMBER OF DATA WORDS
        TST     (R0)+         ;BUMP PAST FORMAT WORD
        TST     R3            ;TEST IF ANY DATA FOR THIS HEADER

```

TYPE ERROR ROUTINE

```

9183 055252 001407
9184 055254 013146
9185 055256 104402
9186 055260 005303
9187 055262 001403
9188 055264 104401 055414
9189 055270 000771
9190 055272 005302
9191 055274 003431
9192 055276 104401 001205
9193 055302 005760 000002
9194 055306 001404
9195 055310 005104
9196 055312 001002
9197 055314 104401 055414
9198 055320 012037 055326
9199 055324 104401
9200 055326 000000
9201 055330 105710
9202 055332 001003
9203 055334 062700 000002
9204 055340 000754
9205 055342 104401 001205
9206 055346 005704
9207 055350 001335
9208 055352 104401 055414
9209 055356 000732
9210 055360 104414
9211
9212 055362 032777 010000 123550
9213 055370 001410
9214 055372 023727 001103 000024
9215 055400 001004
9216 055402 012706 001100
9217 055406 000137 031452
9218
9219 055412 000207
9220 055414 020C40 000
9221
9222
9223
9224
9225
9226
9227
9228
9229 055420
9230 055500
9231 000000
9232 000001
9233 000002
9234 000003
9235 000004
9236 000005
9237 000006
9238 000007

11$: BEQ 14$ :NO - SKIP DATA PRINT
      MOV 2(R1)+, -(SP) :PUT FIRST DATA WORD ON STACK
      TYP0C :TYPE IT
      DEC R3 :MORE DATA WORDS
      BEQ 14$ :NO-BRANCH
      TYPE SPACE2 :TYPE SEPARATORS
      BR 11$ :LOOP
14$: DEC R2 :MORE DH'S?
      BLE 20$ :NO-BRANCH
      TYPE $CRLF
      TST 2(R0) :ONLY A DH IN THIS REQUEST?
      BEQ 15$ :YES-BRANCH BYPASS INDENT
      COM R4 :INDENT?
      BNE 15$ :NO-BRANCH
      TYPE SPACE2 :YES-TYPE SPACES
15$: MOV (R0)+, 16$ :GET NEXT DH POINTER
      TYPE :TYPE DH
16$: .WORD 0 :DH POINTER GOES HERE
      TSTB (R0) :TYPE A DT?
      BNE 21$ :YES-BRANCH
      ADD #2, R0 :INCREMENT OF POINTER
      BR 14$ :SEE IF END OF DF BLOCK
21$: TYPE $CRLF
      TST R4 :INDENT?
      BNE 10$ :NO-BRANCH
17$: TYPE SPACE2 :YES-TYPE SPACES
      BR 10$ :LOOP
20$: RESREG

      BIT #SW12, 2SWR :SEE IF ABORT DRV AFTER 20 ERRORS
      BEQ 25$ :BR IF NO
      CMP $ERFLG, #20. :ELSE SEE IF HAVE 20 ERRORS
      BNE 25$ :BR IF NO
      MOV #STACK, SP :ELSE RESTORE STACK PTR
      JMP $EOP :AND GO TO NEXT DRV

25$: RTS PC
SPACE2: .ASCIZ/ / :2 SPACES
; ODT-11 -- V005A
; DEC-11-U00PA-A-LA
; COPYRIGHT 1969, 1970, 1972
; DIGITAL EQUIPMENT CORPORATION
; MAYNARD, MASSACHUSETTS 01754
; .ENABL ABS,AMA
; .EVEN
; .=. +60

R0 = %0 : REGISTER
R1 = %1 : NAMING
R2 = %2 : CONVENTIONS
R3 = %3
R4 = %4
R5 = %5
SP = %6
PC = %7

```

```

9239      177776      ST      =      177776      ;STATUS REGISTER
9240
9241      000014      0.TVEC =      14      ;TRT VECTOR LOCATION
9242      000340      0.STM  =      340      ;PRIORITY MASK - STATUS REGISTER
9243      000020      0.TBT  =      20      ;T-BIT MASK - STATUS REGISTER
9244      000003      TRT    =      000003      ;TRT INSTRUCTION
9245      000006      RTT    =      000006      ;RTT INSTRUCTION
9246
9247      ;
9248      ; R5 IS USUALLY CONSIDERED SAFE. THE CURRENT ADDRESS WORD
9249      ; RESIDES IN IT. AFTER A BREAKPOINT, IT IS SET TO ZERO, AND SEARCH
9250      ; OPERATIONS LEAVE IT RANDOMLY FILLED. OTHERWISE, IT SHOULD NOT
9251      ; BE USED EXCEPT FOR JSR'S AND THE CURRENT ADDRESS POINTER (CAP).
9252
9253      177562      0.RDB   =      177562      ;R DATA BUFFER
9254      177560      0.RCSR  =      177560      ;R C/SR
9255      177566      0.TDB   =      177566      ;T DATA BUFFER
9256      177564      0.TCSR  =      177564      ;T C/SR
9257
9258      ;
9259      ; INITIALIZE ODT
9260      ; USE 0.ODT FOR A NORMAL ENTRY
9261      ; USE 0.ODT+2 TO RESTART ODT - WIPING OUT ALL BREAKPOINTS
9262      ; USE 0.ODT+4 TO RE-ENTER (I.E. - FAKE A BREAKPOINT)
9263
9264      055500      000413      0.ODT: BR      0.STRT      ;NORMAL ENTRY
9265      055502      000417      BR      0.RST      ;RESTART
9266      055504      013737      177776      055460      0.ENTR: MOV      ST,0.UST      ;RE-ENTER -- SAVE STATUS
9267      055512      013737      000016      177776      MOV      0.TVEC+2,ST      ;SET UP LOCAL STATUS
9268      055520      010737      055456      MOV      PC,0.UPC      ;FAKE THE PC
9269      055524      000137      056656      JMP      0.BK1
9270
9271      055530      012706      055440      0.STRT: MOV      #0,URO,SP      ;SET UP STACK
9272      055534      010637      055454      MOV      SP,0.USR      ;FAKE THE SAVED STACK
9273      055540      000414      BR      0.RST1      ;CLEAR BREAKPOINT TABLES
9274      055542      004037      057064      0.RST: JSR      0,0.SVR      ;SAVE REGISTERS
9275      055546      013777      055476      177716      MOV      0.UIN,00.ADR1      ;REMOVE THE BREAKPOINT
9276      055554      113704      055462      MOV      0.PRI,R4      ;GET ODT PRIORITY
9277      055560      106004      RORB      R4      ;SHIFT
9278      055562      106004      RORB      R4      ;INTO
9279      055564      106004      RORB      R4      ;POSITION
9280      055566      110437      177776      MOV      R4,ST      ;STORE IN STATUS
9281      055572      000127      0.RST1: JMP      (PC)+
9282      055574      000403      BR      0.45
9283      055576      012737      000002      056566      0.45: MOV      #RTI,0.RTIT      ;SET TO RTI IF 11'20 OR '05
9284      055604      105037      057505      CLR      0.P      ;DISALLOW PROCEED
9285      055610      012737      000340      000016      MOV      #0.STM,0.TVEC+2      ;STATUS WORD TO TRT VECTOR + 2
9286      055616      012737      056646      000014      MOV      #0.BRK,0.TVEC      ;PC TO TRT VECTOR
9287      055624      000447      BR      0.RALL      ;CLEAR BREAKPOINT TABLES
9288
9289      ;
9290      ; SPECIAL NAME HANDLER
9291      ; DEPENDS UPON THE EXPLICIT ORDER OF THE TWO TABLES 0.TL AND 0.URD
9292
9293      055626      004537      057306      0.REGT: JSR      5,0.GET      ;SPECIAL NAME, GET ONE MORE CHARACTER
9294      055632      012704      057531      MOV      #0.TL,R4      ;TABLE START ADDRESS
9295      055636      120024      CMP      R0,(R4)+      ;IS THIS THE CORRECT CHARACTER?
9296      055640      001413      BEQ      0.SP      ;JUMP IF YES

```

9295	055642	022704	057537		CMP	#0.TL+0.LG,R4	: IS THE SEARCH DONE?
9296	055646	101373			BHI	0.RSP	: BRANCH IF NOT
9297	055650	042700	177770		BIC	#177770,R0	: MASK OFF OCTAL
9298	055654	010004			MOV	R0,R4	
9299	055656	006304		0.SP1:	ASL	R4	
9300	055660	062704	055440		ADD	#0.URD,R4	: GENERATE ADDRESS
9301	055664	005202			INC	R2	: SET FOUND FLAG
9302	055666	000444			BR	0.SCAN	: GO FIND NEXT CHARACTER
9303	055670	162704	057522	0.SP:	SUB	#0.TL-7,R4	: CORRECT CONSTANT
9304	055674	000770			BR	0.SP1	
9305							
9306							
9307							
9308	055676	004737	057432				
9309	055702	010502		0.0RPC:	JSR	PC,0.TCLS	
9310	055704	061202			MOV	R5,R2	: CURRENT ADDRESS IN R2
9311	055706	006202			ADD	#R2,R2	: COMPUTE
9312	055710	103421			ASR	R2	: MOVE ONE BIT TO CARRY
9313	055712	006302			BCS	0.ERR	: ERROR IF ODD NUMBER
9314	055714	005722			ASL	R2	: RESTORE WORD
9315	055716	010205			TST	(R2)+	: AND INCREMENT BY TWO
9316	055720	000137	056172		MOV	R2,R5	: UPDATE CAD
9317					JMP	0.0P2	: GO FINISH UP
9318							
9319							
9320	055724	005702					
9321	055726	001406					
9322	055730	006204					
9323	055732	103410					
9324	055734	006304					
9325	055736	010437	055472				
9326	055742	000412					
9327	055744	012737	057546	055472	0.RALL:	MOV	#0.TRTC,0.ADR1
9328	055752	000406			BR	0.DCD	: CLEAR BREAKPOINT
9329							
9330							
9331							
9332							
9333							
9334							
9335	055754	052705	000001		0.ERR:	BIS	#1,R5
9336	055760	012700	000077		MOV	#1,R0	: CLOSE EVERYTHING
9337	055764	004537	057364		JSR	5,0.FTYP	: ? TO BE TYPED
9338	055770	004537	057464		JSR	5,0.CRLS	: OUTPUT ?
9339	055774	005004			CLR	R4	: TYPE <CR><LF>*
9340	055776	005002			CLR	R2	: R4 CONTAINS THE CONVERTED OCTAL
9341	056000	004537	057306		JSR	5,0.GET	: R2 IS THE NUMBER FOUND FLAG
9342	056004	022700	000060		CMP	#0,R0	: GET A CHAR, RETURN IN R0
9343	056010	101013			BHI	0.CLGL	: COMPARE WITH ASCII 0
9344	056012	022700	000067		CMP	#7,R0	: CHECK LEGALITY IF NON-NUMERIC
9345	056016	103410			BLO	0.CLGL	: COMPARE WITH ASCII 7
9346	056020	042700	177770		BIC	#177770,R0	: CHECK LEGALITY IF NOT OCTAL
9347	056024	006304			ASL	R4	: CONVERT TO BCD
9348	056026	006304			ASL	R4	: MAKE ROOM
9349	056030	006304			ASL	R4	: IN
9350	056032	060004			ADD	R0,R4	: PACK THREE BITS IN R4

: * HANDLER - OPEN INDEXED ON THE PC

: B HANDLER - SET AND REMOVE BREAKPOINTS

: COMMAND DECODER - ODT11

: REGISTERS R0-R4 MAY BE USED
: REGISTER R5 WILL BE CONSIDERED SAFE

9351 056034 005202
9352 056036 000760
9353 056040 005001
9354 056042 120061 057515
9355 056046 001405
9356 056050 005201
9357 056052 020127 000014
9358 056056 103336
9359 056060 000770
9360 056062 006301
9361 056064 000171 056070
9362
9363 056070 056120
9364 056072 056152
9365 056074 055626
9366 056076 056462
9367 056100 056164
9368 056102 055676
9369 056104 056216
9370 056106 056226
9371 056110 056304
9372 056112 056300
9373 056114 055724
9374 056116 056570
9375 000030
9376
9377
9378
9379 056120 005702
9380 056122 001410
9381 056124 010405
9382 056126 006205
9383 056130 103711
9384 056132 006305
9385 056134 011500
9386 056136 004537 057222
9387 056142 000714
9388 056144 042705 000001
9389 056150 000766
9390
9391
9392
9393 056152 004737 057432
9394 056156 052705 000001
9395 056162 000702
9396
9397
9398
9399 056164 004737 057432
9400 056170 005725
9401 056172 004537 057456
9402 056176 010500
9403 056200 004537 057222
9404 056204 012700 000057
9405 056210 004537 057364
9406 056214 000744

```

      INC      R2      ; R2 HAS NUMERIC FLAG
      BR       0.SCAN  ; AND TRY AGAIN
0.CLGL: CLR     R1      ; CLEAR INDEX
0.LGL1: CMPB    R0,0.LGCH(R1) ; DO THE CODES MATCH?
      BEQ      0.LGL2  ; JUMP IF YES
      INC      R1      ; SET INDEX FOR NEXT SEARCH
      CMP      R1,0.CLGT ; IS THE SEARCH DONE?
      BHS      0.ERR    ; OOPS!
      BR       0.LGL1  ; RE-LOOP
0.LGL2: ASL     R1      ; MULTIPLY BY TWO
      JMP      20.LGDR(R1) ; GO TO PROPER ROUTINE

0.LGDR: 0.WRD    ; / OPEN WORD
      0.CRET    ; CARRIAGE RETURN CLOSE
      0.REGT    ; $ REGISTER OPS
      0.GO      ; G GO TO ADDRESS K
      0.OP1     ; <LF> MODIFY CLOSE OPEN NEXT
      0.ORPC    ; + OPEN RELATED INDEX - PC
      0.BACK    ; - OPEN PREVIOUS
      0.OFST    ; 0 OFFSET
      0.WSCH    ; W SEARCH WORD
      0.EFF     ; E SEARCH EFFECTIVE ADDRESS
      0.BKPT    ; B BREAKPOINTS
      0.PROC    ; P PROCEED
0.LGL = -0.LGDR ; LGL MUST EQUAL 2X CHLGT ALWAYS

; PROCESS / - OPEN WORD
0.WRD: TST      R2      ; GET VALUE IF R2 IS NON-ZERO
      BEQ      0.WRDA   ; SKIP OTHERWISE
      MOV      R4,R5    ; PUT VALUE IN CAD
0.WRD1: ASR      R5      ; MOVE ONE BIT TO CARRY
0.ERR2: BCS      0.ERR   ; JUMP IF ODD ADDRESS
      ASL      R5      ; RESTORE THE CARRY BIT
      MOV      2R5,R0   ; GET CONTENTS OF WORD
      JSR      5,0.CADV ; GO GET AND TYPE OUT 2CAD
      BR       0.DCD1   ; GO BACK TO DECODER
0.WRDA: BIC      #1,R5   ; CLEAR CLOSED BIT
      BR       0.WRD1   ; GO BACK TO MAIN-LINE

; PROCESS CARRIAGE RETURN
0.CRET: JSR      PC,0.TCLS ; CLOSE LOCATION
      BIS      #1,R5    ; CLOSE EVERYTHING
      BR       0.DCD    ; RETURN TO DECODER

; PROCESS <LF>, OPEN NEXT WORD
0.OP1: JSR      PC,0.TCLS ; CLOSE PRESENT CELL
      TST      (R5)+    ; GENERATE NEW ADDRESS
0.OP2: JSR      5,0.CRLF ; <CR><LF>
      MOV      R5,R0    ; NUMBER TO TYPE
      JSR      5,0.CADV ; TYPE OUT ADDRESS
      MOV      #1/R0    ; TYPE A /
      JSR      5,0.FTYP ;
      BR       0.WRD1   ; GO PROCESS IT

```

9407
9408
9409
9410 056216 004737 057432
9411 056222 005745
9412 056224 000762
9413
9414
9415
9416 056226 006205
9417 056230 103737
9418 056232 006305
9419 056234 012700 000040
9420 056240 004537 057364
9421 056244 160504
9422 056246 005304
9423 056250 005304
9424 056252 010400
9425 056254 010402
9426 056256 004537 057222
9427 056262 010200
9428 056264 006200
9429 056266 103402
9430 056270 004537 057222
9431 056274 000137 055774
9432
9433
9434
9435

```

: PROCESS *, OPEN PREVIOUS WORD
0.BACK: JSR PC,0.TCLS
: TST -(R5, : GENERATE NEW ADDRESS
: BR 0.OP2 : GO DO THE REST

: PROCESS 0, COMPUTE OFFSET
0.OFST: ASR R5 : GET LOW ORDER BIT
: BCS C,FRR2 : ERROR IF CLOSED
: ASL R5 : RESTORE WORD
: MOV R5,R0 : TYPE ONE BLANK
: JSR S,0.FTYP : AS A SEPARATOR
: SUB R5,R4 : COMPUTE
: DEC R4 : 16 BIT OFFSET
: DEC R4 : TYPE A
: MOV R4,R0 : SAVE R4
: MOV R4,R2 : NUMBER IN R0 - WORD MODE
: JSR S,0.CADV : DIVIDE BY TWO
: MOV R2,R0 : BRANCH IF ODD
: ASR R0 : NUMBER IN R0 - BYTE MODE
: BCS 0.OF1 : ALL DONE
: JSR S,0.CADV
0.OF1: JMP 0.DCD1

: SEARCHES - SMSK HAS THE MASK
: SMSK+2 HAS THE FWA
: SMSK+4 HAS THE LWA

```

```

9436
9437 056300 005201      0.EFF: INC R1 ;SET EFFECTIVE SEARCH
9438 056302 000401      BR 0.WDS ;
9439 056304 005001      0.WSCH: CLR R1 ;SET WORD SEARCH
9440 056306 005702      0.WDS: TST R2 ;CHECK FOR OBJECT FOUND
9441 056310 001621      0.ERR1: BEQ 0.ERR ;ERROR IF NO OBJECT
9442 056312 013702 055466 MOV 0.MSK+2,R2 ;SET ORIGIN
9443 056316 013705 055464 MOV 0.MSK,R5 ;SET MASK
9444 056322 005105      COM R5 ;AND COMPLEMENT IT
9445 056324 020237 055470 0.WDS2: CMP R2,0.MSK+4 ;IS THE SEARCH ALL DONE?
9446 056330 101217      BHI 0.DCD ;YES
9447 056332 011200      MOV R2,R0 ;GET OBJECT
9448 056334 005701      TST R1 ;NO
9449 056336 001027      BNE 0.EFF1 ;BRANCH IF EFFECTIVE SEARCH
9450 056340 010046      MOV R0,-(SP) ;
9451 056342 010403      MOV R4,R3 ;EXCLUSIVE OR
9452 056344 040400      BIC R4,R0 ;IS DONE
9453 056346 042603      BIC (SP)+,R3 ;
9454 056350 050003      BIS R0,R3 ;IN A VERY
9455 056352 040503      BIC R5,R3 ;FANCY MANNER HERE
9456 056354 001016      0.WDS3: BNE 0.WDS4 ;AND RESULT WITH MASK
9457 056356 010446      MOV R4,-(SP) ;RE-LOOP IF NO MATCH
9458 056360 004537 057456 JSR 5,0.CRLF ;REGISTERS R2,R4, AND R5 ARE SAFE
9459 056364 010200      MOV R2,R0 ;TYPE (CR,LF)
9460 056366 004537 057222 JSR 5,0.CADV ;GET READY TO TYPE
9461 056372 012700 000057 MOV R1,R0 ;TYPE ADDRESS
9462 056376 004537 057364 JSR 5,0.FTYP ;SLASH TO R0
9463 056402 011200      MOV R2,R0 ;TYPE IT
9464 056404 004537 057222 JSR 5,0.CADV ;GET CONTENTS
9465 056410 012604      MOV (SP)+,R4 ;TYPE CONTENTS
9466 056412 005722      0.WDS4: TST (R2)+ ;RESTORE R4
9467 056414 000743      BR 0.WDS2 ;INCREMENT TO NEXT CELL AND
9468 056416 020004      0.EFF1: CMP R0,R4 ;RETURN
9469 056420 001755      BEQ 0.WDS3 ;IS (X)=K?
9470 056422 010003      MOV R0,R3 ;TYPE IF EQUAL
9471 056424 060203      ADD R2,R3 ;(X) TO R3
9472 056426 005203      INC R3 ;(X)+X
9473 056430 005203      INC R3 ;(X)+X+2
9474 056432 020304      CMP R3,R4 ;IS (X)+X+2=K?
9475 056434 001747      BEQ 0.WDS3 ;BRANCH IF EQUAL
9476 056436 042700 177400 BIC #177400,R0 ;WIPE OUT EXTRANEIOUS BITS
9477 056442 110000      MOVB R0,R0 ;EXTEND SIGN
9478 056444 000257      CCC ;
9479 056446 006300      ASL R0 ;MULTIPLY BY TWO
9480 056450 005200      INC R0 ;ADD TWO
9481 056452 005200      INC R0 ;
9482 056454 060700      ADD R2,R0 ;ADD PC
9483 056456 020004      CMP R0,R4 ;IS THE RESULT A PROPER REL. BRANCH?
9484 056460 000735      BR 0.WDS3 ;
9485
9486 ; PROCESS G - GO
9487
9488 056462 105037 057505 0.GO: CLRB 0.P ;DISALLOW PROCEED
9489 056466 006204      ASR R4 ;CHECK LOW ORDER BIT
9490 056470 103617      BCS 0.ERR2 ;ERROR IF ODD NUMBER
9491 056472 006304      ASL R4 ;RESTORE WORD

```

```

9492 056474 010437 055456      MOV      R4,0.UPC      ;SET UP NEW PC
9493 056500 112737 000340 177776  MOVB     #0,STM,ST    ;SET HIGH PRIORITY
9494 056506 004537 057154      JSR      5,0.RSTT    ;RESTORE TELETYPE
9495 056512 105037 057504      O.TBIT: CLRB     0,T      ;CLEAR BOTH
9496 056516 042737 000020 055460  BIC      #0,TBT,0.UST ;T-BIT FLAGS
9497 056524 017737 176742 055476  MOV      #0,ADR1,0.UIN ;SAVE INSTRUCTION
9498 056532 013777 057546 176732  MOV      0,TRTC, #0,ADR1 ;REPLACE WITH TRAP
9499 056540 012600      O.G02: MOV      (SP)+,R0    ;RESTORE
9500 056542 012601      MOV      (SP)+,R1    ;R0
9501 056544 012602      MOV      (SP)+,R2    ;THRU
9502 056546 012603      MOV      (SP)+,R3
9503 056550 012604      MOV      (SP)+,R4
9504 056552 012605      MOV      (SP)+,R5
9505 056554 012606      MOV      (SP)+,SP
9506 056556 013746 055460      MOV      0,UST,-(SP) ;AND SP
9507 056562 013746 055456      MOV      0,UPC,-(SP) ;AND STATUS
9508 056566 000006      O.RTIT: RTT      ;AND PC
9509                                     ;CHANGED TO RTI FOR 11/20 AND /05
9510                                     ;
9511                                     ;PROCESS P - PROCEED
9512                                     ;ONLY ALLOWED AFTER A BREAKPOINT
9513 056570 105737 057505      O.PROC: TSTB     0,P      ;CHECK LEGALITY OF PROCEED
9514 056574 001645      BEQ      0,ERR1    ;NOT LEGAL
9515 056576 105037 057505      CLRB     0,P      ;CLEAR PROCEED FLAG
9516 056602 005702      TST      R2      ;WAS COUNT SPECIFIED?
9517 056604 001402      BEQ      0,PR1    ;NO
9518 056606 010437 055474      MOV      R4,0.CT    ;YES, PUT AWAY COUNT
9519 056612 112737 000340 177776  O.PR1:  MOVB     #0,STM,ST    ;FORCE HIGH PRIORITY
9520 056620 004537 057154      JSR      5,0.RSTT    ;RESTORE TTY
9521 056624 112737 000340 177776  O.C1:   MOVB     #0,STM,ST    ;SET HIGH PRIORITY
9522 056632 105237 057504      INCB     0,T      ;SET T-BIT FLAG
9523 056636 052737 000020 055460  BIS      #0,TBT,0.UST ;SET T-BIT
9524 056644 000735      BR        0,G02
9525                                     ;
9526                                     ;BREAKPOINT HANDLER
9527                                     ;A TRT BREAKPOINT CAUSES O.BRK TO BE ENTERED, WHICH SAVES
9528                                     ;VARIOUS ODDS AND ENDS, FINDS OUT IF THE BREAKPOINT WAS LEGAL,
9529                                     ;AND GIVES CONTROL TO THE COMMAND DECODER
9530                                     ;
9531 056646 012637 055456      O.BRK:  MOV      (SP)+,0.UPC    ;PRIORITY IS 7 UPON ENTRY
9532 056652 012637 055460      MOV      (SP)+,0.UST    ;SAVE STATUS AND PC
9533 056656 004037 057064      O.BK1:  JSR      0,0.SVR    ;SAVE VARIOUS REGISTERS
9534 056662 105737 057504      TSTB     0,T      ;CHECK FOR T-BIT SET
9535 056666 001311      BNE      0,TBIT    ;JUMP IF SET
9536 056670 013777 055476 176574  MOV      0,UIN, #0,ADR1 ;REMOVE BREAKPOINTS
9537 056676 105737 055462      TSTB     0,PR1    ;CHECK IF PRIORITY
9538 056702 100003      BPL      0,BK2    ;IS AS SAME AS USER PGM
9539 056704 113705 055460      MOVB     0,UST,R5    ;PICK UP USER UST IF SO
9540 056710 000407      BR        0,BK3    ;AND DON'T COMPUTE THE PRIORITY
9541 056712 113705 055462      O.BK2:  MOVB     0,PR1,R5    ;OTHERWISE PICK UP ACTUAL PRIORITY
9542 056716 000257      CCC      ;CLEAR CARRY
9543 056720 106005      RORB      R5      ;SHIFT LOW ORDER BITS
9544 056722 106005      RORB      R5      ;INTO
9545 056724 106005      RORB      R5      ;HIGH ORDER
9546 056726 106005      RORB      R5      ;POSITION
9547 056730 110537 177776      O.BK3:  MOVB     R5,ST    ;PUT THE STATUS AWAY WHERE IT BELONGS

```

```

9548 056734 013705 055456      MOV      0,UPC,R5      ;GET PC, IT POINTS TO THE TRT
9549 056740 005745      TST      -(R5)      ;SUBTRACT TWO
9550 056742 010537      MOV      R5,0,UPC      ;FROM THE USER'S PC
9551 056746 020537      CMP      R5,0,ADR1      ;COMPARE WITH LIST
9552 056752 001417      BEQ      0,B2      ;JUMP IF FOUND
9553 056754 004537      JSR      5,0,SVTT      ;SAVE TELETYPE STATUS
9554 056760 004537      JSR      5,0,CRLF      ;
9555 056764 012704      MOV      #0,B0,R4      ;ERROR, NOTHING FOUND
9556 056770 012703      MOV      #0,B0+1,R3      ;
9557 056774 004537      JSR      5,0,TYPE      ;OUTPUT "BE" FOR BAD ENTRY
9558 057000 010500      MOV      R5,R0      ;
9559 057002 042737      BIC      #0,TBT,0,UST      ;CLEAR OUT ANY POSSIBLE FAKE T-BIT
9560 057010 000420      BR      0,B3      ; AND CONTINUE
9561 057012 005337      0.B2: DEC      0,CT      ;
9562 057016 003302      BGT      0,C1      ;JUMP IF REPEAT
9563 057020 012737      MOV      #1,0,CT      ;RESET COUNT TO 1
9564 057026 105237      INCB     0,P      ;ALLOW PROCEED
9565 057032 004537      JSR      5,0,SVTT      ;SAVE TELETYPE STATUS, R4 IS SAFE
9566 057036 012700      MOV      #1,B,R0      ;
9567 057042 004537      JSR      5,0,FTYP      ;TYPE "B"
9568 057046 013700      MOV      0,ADR1,R0      ;GET ADDRESS OF BREAK
9569 057052 004537      0.B3: JSR      5,0,CADV      ;TYPE ADDRESS
9570 057056 005005      CLR      R5      ;CLEAR CAD
9571 057060 000137      JMP      0,DCD      ;GO TO DECODER
9572      ;
9573      ; SAVE REGISTERS R0-R6 IN INTERNAL STACK
9574      ;
9575 057064 012637      0.SVR: MOV      (SP)+,0,XXX      ;PICK REGISTER FROM STACK AND SAVE
9576 057070 010637      MOV      SP,0,USP      ;SAVE USER STACK ADDRESS
9577 057074 012706      MOV      #0,USP,SP      ;SET TO INTERNAL STACK
9578 057100 010546      MOV      R5,-(SP)      ;SAVE
9579 057102 010446      MOV      R4,-(SP)      ;REGISTERS
9580 057104 010346      MOV      R3,-(SP)      ;1
9581 057106 010246      MOV      R2,-(SP)      ;THRU
9582 057110 010146      MOV      R1,-(SP)      ;5
9583 057112 013746      MOV      0,XXX,-(SP)      ;PUT SAVED REGISTER ON STACK
9584 057116 005746      TST      -(SP)      ;
9585 057120 000200      RTS      R0      ;
9586      ;
9587      ; SAVE TELETYPE STATUS
9588      ;
9589 057122 113737      0.SVTT: MOVB     0,RCR,0,CSR1      ;SAVE R C/SR
9590 057130 113737      MOVB     0,TCSR,0,CSR2      ;SAVE T C/SR
9591 057136 105037      CLRB     0,RCR      ;CLEAR ENABLE AND MAINTENANCE
9592 057142 105037      CLRB     0,TCSR      ;BITS IN BOTH C/SR
9593 057146 004537      JSR      5,0,CRLF      ;TYPE <CR,LF>
9594 057152 000205      RTS      R5      ;
9595      ;
9596      ; RESTORE TELETYPE STATUS
9597      ;
9598 057154 004537      0.RSTT: JSR      5,0,CRLF      ;<CR,LF> BEFORE RESTORING
9599 057160 105737      TSTB     0,TCSR      ;WAIT READY ON PRINTER
9600 057164 100375      BPL      -4      ;
9601 057166 032737      BIT      #4000,0,RCR      ;CHECK BUSY FLAG ON READER
9602 057174 001403      BEQ      0,RSE1      ;SKIP READY LOOP IF NOT BUSY
9603 057176 105737      TSTB     0,RCR      ;WAIT READY

```

```

9604 057202 100375      BPL      -4      ; ON READER
9605 057204 113737 057506 177560 0.RSE1: MOV 0.CSR1,0.RCSR ;RESTORE
9606 057212 113737 057507 177564 MOV 0.CSR2,0.TCSR ; THE STATUS REGISTERS
9607 057220 000205      RTS      R5
9608
9609 ; TYPE OUT CONTENTS OF WORD OR BYTE WITH ONE TRAILING SPACE
9610 ; WORD IS IN R0
9611
9612 057222 010246      0.CADV: MOV R2,-(SP) ;SAVE R2
9613 057224 012704 057545      MOV #0.BUF+6,R4 ;BUFFER START ADDRESS
9614 057230 012746 000060      MOV #0,-(SP) ;CONSTANT ASCII 0
9615 057234 010002      0.SPC: MOV R0,R2 ; GET
9616 057236 042702 177770      BIC #177770,R2 ; OCTAL CHARACTER
9617 057242 061602      ADD #SP,R2 ; CONVERT TO ASCII
9618 057244 110244      MOV R2,-(R4) ; STORE IN BUFFER
9619 057246 006200      ASR R0 ; SHIFT THIS MESS
9620 057250 006200      ASR R0 ; RIGHT
9621 057252 006200      ASR R0 ; THREE WHOLE PLACES
9622 057254 020427 057540      CMP R4,#0.BUF+1 ; DONE?
9623 057260 101365      BHI 0.SPC ; NO
9624 057262 042700 177776      BIC #177776,R0 ; GET LAST BIT
9625 057266 062600      ADD (SP)+,R0 ; CONVERT TO ASCII
9626 057270 110044      MOV R0,-(R4) ; AND PUT IT AWAY
9627 057272 012703 057545      MOV #0.BUF+6,R3 ; LWA
9628 057276 004537 057350      JSR 5,0.FTYP ; TYPE WHOLE STRING OF CHARACTERS
9629 057302 012602      MOV (SP)+,R2 ; RESTORE R2
9630 057304 000205      RTS      R5
9631
9632 ; GENERAL CHARACTER INPUT ROUTINE
9633 ; CHARACTER INPUT GOES TO R0
9634
9635 057306 105737 177560      0.GET: TSTB 0.RCSR ;WAIT FOR
9636 057312 100375      BPL -4 ; INPUT FROM KEYBOARD
9637 057314 113700 177562      MOV 0.RDB,R0 ; GET A CHARACTER
9638 057320 004537 057364      JSR 5,0.FTYP ; ECHO CHARACTER
9639 057324 042700 177600      BIC #177600,R0 ; STRIP OFF PARITY FROM CHARACTER
9640 057330 001766      BEQ 0.GET ; IGNORE NULLS
9641 057332 122700 000040      CMPB #40,R0 ; CHECK FOR SPACES
9642 057336 001763      BEQ 0.GET ; IGNORE NULLS
9643 057340 122700 000073      CMPB #'',R0 ; CHECK FOR SEMI-COLON
9644 057344 001760      BEQ 0.GET ; IGNORE THEM IF FOUND
9645 057346 000205      RTS      R5
9646
9647 ; GENERAL CHARACTER OUTPUT ROUTINE
9648 ; ADDRESS OF FIRST BYTE IN R4,
9649 ; ADDRESS OF LAST BYTE IN R3, (R3)>(R4)
9650
9651 057350 020304      0.TYPE: CMP R3,R4 ;CHECK FOR COMPLETION
9652 057352 103426      BLO 0.TYP1 ; EXIT WHEN DONE
9653 057354 112400      MOV (R4)+,R0 ; GET A CHARACTER
9654 057356 004537 057364      JSR 5,0.FTYP ; TYPE ONE CHARACTER
9655 057362 000772      BR 0.TYPE ; LOOP UNTIL DONE
9656
9657 ; TYPE ONLY ONE CHARACTER (CONTAINED IN R0)
9658
9659 057364 105737 177564      0.FTYP: TSTB 0.TCSR ;CHECK STATUS

```

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 185
TYPE ERROR ROUTINE

SEQ 0185

9660 057370 100375
9661 057372 110037 177566
9662 057376 120037 000045
9663 057402 001012
9664 057404 113746 000044
9665 057410 105737 177564
9666 057414 100375
9667 057416 105037 177566
9668 057422 105316
9669 057424 003371
9670 057426 005726
9671 057430 000205
9672
9673
9674
9675
9676 057432 006205
9677 057434 103405
9678 057436 006305
9679 057440 005702
9680 057442 001401
9681 057444 010415
9682 057446 000207
9683 057450 005746
9684 057452 000137 055754
9685
9686
9687
9688
9689 057456 012703 057513
9690 057462 000402
9691 057464 012703 057514
9692 057470 012704 057512
9693 057474 004537 057350
9694 057500 000205
9695
9696 057502 000000
9697 057504 000
9698 057505 000
9699
9700 057506 000
9701 057507 000
9702
9703
9704 057510 042502
9705
9706 057512 015
9707 057513 012
9708 057514 052
9709
9710 057515 057
9711 057516 015
9712 057517 044
9713 057520 107
9714 057521 012
9715 057522 137

```

BPL      -4      ;WAIT UNTIL READY
MOV      RO,0.TDB ;TYPE ONE CHARACTER
CMP      RO,0#45  ;IS CHAR TO BE FILLED?
BNE      0.TYP1   ;NO
MOV      0#44,-(SP) ;YES, INIT THE COUNT
0.TYP2:  TST      0.TCSR
BPL      0.TYP2   ;GENERATE NULL FILLER
CLRB     0.TDB
DECB     0SP
BGT      0.TYP2
TST      (SP)+    ;POP STACK
0.TYP1:  RTS      R5

; CLOSE WORD OR BYTE AND EXIT
; UPON ENTERING, R2 HAS NUMERIC FLAG, R4 HAS CONTENTS
0.TCLS:  ASR      R5      ;GET LOW ORDER BIT
BCS      0.TC      ;JUMP IF ALREADY CLOSED
ASL      R5
TST      R2      ;IF NO NUMBER WAS TYPED THERE IS
BEQ      0.CLS1   ;NO CHANGE TO THE OPEN CELL
MOV      R4,R5    ;STORE WORD
0.CLS1:  RTS      PC
0.TC:    TST      -(SP)  ;POP EXTRA CELL FROM STACK
JMP      0.ERR    ;AND SCREAM BLOODY MURDER

; 0.CRLF - TYPE <CR,LF>
; 0.CRLS - TYPE <CR,LF>*
0.CRLF:  MOV      #0.CR+1,R3 ;LWA <CR,LF>
BR       0.CRS
0.CRLS:  MOV      #0.CR+2,R3 ;LWA <CR,LF>*
MOV      #0.CR,R4 ;FWA
JSR      5.0.TYPE ;TYPE SOMETHING
RTS      R5

; .XXX: .WORD 0 ;TEMPORARY STORAGE
; .T: .BYTE 0 ;T-BIT FLAG
; .P: .BYTE 0 ;PROCEED FLAG = 0 IF PROCEED NOT ALLOWED
; ; = 1 IF PROCEED ALLOWED
0.CSR1:  .BYTE 0 ;SAVE CELL - R C/SR
0.CSR2:  .BYTE 0 ;SAVE CELL - T C/SR

; .BD: .WORD "BE
; .CR: .BYTE 015 ;<CR>
; .BYTE 012 ;<LF>
; .BYTE '*' ;*

; .LGCH: .BYTE '/' ;/
; .BYTE 015 ;CARRIAGE RETURN
; .BYTE '$' ;$
; .BYTE 'G' ;G
; .BYTE 012 ;<LF>
; .BYTE '*' ;*

```

CZR6IED UNIBUS PK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(052) 10-JAN-78 11:25 PAGE 186
TYP. ERROR ROUTINE

SEQ 0186

9716 057523 136
9717 057524 117
9718 057525 127
9719 057526 105
9720 057527 102
9721 057530 120
9722 000014
9723
9724 057531 123
9725 057532 120
9726 057533 115
9727 057534 000
9728 057535 000
9729 057536 102
9730 000006
9731
9732 057537
9733 057545
9734 057545 040
9735
9736
9737 057546 000003
9738
9739
9740
9741 055440
9742 055440 000000
9743 055442 000000
9744 055444 000000
9745 055446 000000
9746 055450 000070
9747 055452 000000
9748 055454 000000
9749 055456 000000
9750 055460 000000
9751 055462 000007
9752 055464 000000
9753 055466 000000
9754 055470 000000
9755
9756
9757
9758
9759 055472 000000
9760 055474 000000
9761 055476 000000
9762 000001

```

      .BYTE  '↑      :↑
      .BYTE  'O      :O
      .BYTE  'W      :W
      .BYTE  'E      :E
      .BYTE  'B      :B
      .BYTE  'P      :P
      0.CLGT  =      -0.LGCH      ;TABLE LENGTH
      0.TL:   .BYTE  'S      :DO      1
      .BYTE  'P      :NOT      2
      .BYTE  'M      :CHANGE    3
      .BYTE  'O      :THE      4
      .BYTE  'O      :ORDER    5
      .BYTE  'B      :HERE     6
      0.LG    =      -0.TL
      0.BUF:   =      ;+6      ;6 CHAR. BUFFER WITH
      .BYTE   ;+6      ;TRAILING BLANK
      .EVEN
      0.TRTC: TRT      ;TRACE TRAP PROTOTYPE
      ;THE ORDER OF THE FOLLOWING ENTRIES IS CRITICAL
      0.URD:   =      0.ODT-40
      000000  ;USER R0
      000000  ;R1
      000000  ;R2
      000000  ;R3
      000000  ;R4
      000000  ;R5
      0.USP:   000000  ;USER SP
      0.JPC:   000000  ;USER PC
      0.UST:   000000  ;USER ST
      0.PRI:   7      ;ODT PRIORITY
      0.MSK:   0      ;MASK
      000000  ;LOW LIMIT
      000000  ;HIGH LIMIT
      ;BREAK POINT LISTS, ADR1 = ADDRESS OF BREAKPOINT, CT = COUNT,
      ;UIN = CONTENTS
      0.ADR1: 0
      0.CT:   0
      0.UIN:   0
      .END

```

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 188
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0187

ABASE = 177440	1889	1930	1944*							
ACDW1 = 000000	1889	1932								
ACDW2 = 000000	1889	1933								
ACLO = 000010	1245*									
ACPUOP = 000000	1889	1904								
ACT11 = 007470	2122*	3298*								
ADDW0 = 000000	1889	1934								
ADDW1 = 000000	1889	1935								
ADDW10 = 000000	1889									
ADDW11 = 000000	1889									
ADDW12 = 000000	1889									
ADDW13 = 000000	1889									
ADDW14 = 000000	1889									
ADDW15 = 000000	1889									
ADDW2 = 000000	1889	1936								
ADDW3 = 000000	1889	1937								
ADDW4 = 000000	1889	1938								
ADDW5 = 000000	1889	1939								
ADDW6 = 000000	1889	1940								
ADDW7 = 000000	1889	1941								
ADDW8 = 000000	1889									
ADDW9 = 000000	1889									
ADEVCT = 000000	1889	1895								
ADEVN = 000000	1889	1931								
AENV = 000000	1889	1900								
AENVN = 000000	1889	1901								
AFATAL = 000000	1889	1892								
AMADR1 = 000000	1889	1917								
AMADR2 = 000000	1889	1921								
AMADR3 = 000000	1889	1924								
AMADR4 = 000000	1889	1927								
AMAMS1 = 000000	1889	1911								
AMAMS2 = 000000	1889	1919								
AMAMS3 = 000000	1889	1922								
AMAMS4 = 000000	1889	1925								
AMSGAD = 000000	1889	1897								
AMSGLG = 000000	1889	1898								
AMSGTY = 000000	1889	1891								
AMTYP1 = 000000	1889	1912								
AMTYP2 = 000000	1889	1920								
AMTYP3 = 000000	1889	1923								
AMTYP4 = 000000	1889	1926								
APASS = 000000	1889	1894								
APRIOR = 000000	1889									
APTCU = 000040	7279	7451*								
APTENV = 000001	5976	7226	7272	7407	7449*					
APTSIZ = 000200	3226	7448*								
APTSPO = 000100	7274	7409	7450*							
ASWREG = 000000	1889	1902								
ATESTN = 000000	1889	1893								
ATTN = 007344	2055*	6183	6202	6228						
AUNIT = 000000	1889	1896								
AUSWR = 000000	1889	1903								
AVECT1 = 000000	1889	1928								
AVECT2 = 000000	1889	1929								
BADHDR = 007340	2045*	3306*	4020*	4151*	4418*	4608*	5708*	5923*	5926*	6935

G15

CZR6IE0 UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 189
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0188

BADTMO = 036570	3380	6989*												
BAI = 000020	1208*	4162	4311	4449	4519	4921	4981	5106	5268	5344	5469	5522	5642	
	5776	5821												
BA16 = 000400	1194*													
BA17 = 001000	1195*													
BIT0 = 000001	1137*	1191	1223	1242	2112	3421	6880							
BIT00 = 000001	1127*	1137												
BIT01 = 000002	1126*	1136												
BIT02 = 000004	1125*	1135												
BIT03 = 000010	1124*	1134												
BIT04 = 000020	1123*	1133												
BIT05 = 000040	1122*	1132												
BIT06 = 000100	1121*	1131												
BIT07 = 000200	1120*	1130												
BIT08 = 000400	1119*	1129	7161											
BIT09 = 001000	1118*	1128	7169	7237										
BIT1 = 000002	1136*	1224	2113											
BIT10 = 002000	1117*	1196	1214	1233	1265	1279	1293	1306	1320	7214				
BIT11 = 004000	1116*	1197	1215	1234	1251	1266	1280	1294	1307	1321	7176			
BIT12 = 010000	1115*	1198	1216	1235	1267	1281	1295	1308	1322					
BIT13 = 020000	1114*	1199	1217	1236	1252	1268	1282	1296	1309	1323	7221			
BIT14 = 040000	1113*	1200	1218	1237	1253	1269	1283	1297	1310	1324	1334	6682	6693	
	7147													
BIT15 = 100000	1112*	1201	1202	1219	1238	1254	1270	1337	5710	5870	6678	6689		
BIT2 = 000004	1135*	1225	1244	2114										
BIT3 = 000010	1134*	1207	1226	1245										
BIT4 = 000020	1133*	1208	1227	1246	1259	1287	1314							
BIT5 = 000040	1132*	1209	1228	1247	1260	1274	1288	1301	1315					
BIT6 = 000100	1131*	1192	1210	1229	1248	1261	1275	1289	1302	1316				
BIT7 = 000200	1130*	1193	1211	1230	1249	1262	1276	1290	1303	1317	3411	4673	4723	
	4775	4872	5130											
BIT8 = 000400	1129*	1194	1212	1231	1250	1263	1277	1291	1304	1318				
BIT9 = 001000	1128*	1195	1213	1232	1264	1278	1292	1306	1319					
BPTVEC = 000014	1144*													
BSE = 000200	1230*	4065	4172	4236	4462	4933	5278	5476						
BSERR = 001512	2028*	3929*	3934*	4011										
BSE20H = 002336	2037*	3950	6687	6792										
BSE20S = 004336	2039*	3959	6691	6796										
BSE22H = 003336	2038*	3870	3931	3968	6676	6781								
BSE22S = 005336	2040*	3941	6680	6785										
BYP = 033026	3497	3525	3529	544	3603	3630	3634	3638	3642	6255*				
BYP CER = 001516	2030*	3408*	3655*	6370										
CALADD = 001362	1971*	4134*	4421*	4591*	5879*	6649	6655	6657	6711					
CALDIF = 001354	1968*	5768*	8923	8947										
CCLP = 100000	1202*	3835	3850	4075	4182	4472	4693	4703	4943	5095	5163	5178	5254	
	5288	5404	5419	5579	5594	6410	6418	6426	6434	6939	6954			
CCYL = 001350	1966*													
CDT = 002000	1196*	3512	3613	3694	3696									
CERR = 100000	1201*	3476	3587	3881	3986	4038	4062	4144	4169	4234	4320	4433	4459	
	4528	4601	4745	4802	4865	4930	4989	5275	5352	5476	5529	5661	5733	
	5755	5788	5833	5889	5913	6372	6467							
CFMT = 010000	1198*	4427	4430	4453	4456	4494	4522	4525	4557	6778				
CHKFLG = 001520	2031*	6265*	6348	6351	6355									
CHKMSG = 033042	3893	4087	4106	4194	4213	4260	4277	4339	4358	4392	4484	4511	4547	
	4574	4644	4665	4714	4766	4829	4955	4974	5008	5027	5066	5087	5225	
	5246	5300	5319	5371	5390	5494	5513	5548	5567	5673	5802	5847	6265*	

5E0 0189

6133#	6137	6148	6153#	6370#								
7146#	7210	7236	8111#									
1179#	3852	4695	5097	5180	5256	5421	5596	6956				
6851#												
6829#												
3329	6840#											
3247	5988#											
1232#												
1975#	6841*	6843*										
1052#	7318	7328										
8186	8192	8198	8123	8131	8133	8137	8147	8156	8163	8171	8176	8181
8257	8263	8271	8204	8208	8214	8219	8226	8235	8240	8245	8251	8253
8335	8341	8349	8281	8286	8293	8297	8299	8303	8307	8315	8319	8327
8517	8522	8534	8355	8362	8367	8370	8376	8394	8402	8477	8504	8511
1053#	7289	7328	8554									
1197#	6376											
1992#	5462											
1970#	4672	4846	5331	5808	5853	6525*	6526*	6527*	6528*	6529*	6530*	6548*
6549*	6550*	6551*	6552*	6553*								
1969#	4676*	4682	4687	4720*	4726	4731	4772*	4778	4783	4850	5328	6490*
6491*	6492*	6493*	6494*	6495*	6496	6498*	6516*	6517*	6535*	6536*	6537*	6538*
6539*	6540*	6541	6543*									
1999#	5460											
2019#	4160	4161	4295	4371	5267	5343	5357					
2020#	5641											
2021#	4054	4299	4300	4312	4325	4448	4518	4533	4920	4980	4994	5105
5775	5820											
3878	4035	4059	4141	4166	4231	4317	4375	4428	4454	4523	4598	4800
4927	4986	5110	5272	5349	5473	5526	5730	5780	5825	5886	6113#	
1238#	4243											
1246#												
1199#												
1059#	1864	3214										
2121#	3294*	3484	3687									
1250#												
2120#	3260*	5988										
2162	2168	2174	2180	2191	2197	2203	2429	2469	2474	2514	8999#	
2208	2213	2218	2223	2228	2238	2248	2253	2263	2279	2299	2304	2339
2349	2354	2389	2419	2424	2464	2494	2499	2554	2559	2564	2569	2584
2609	2614	2639	2689	2704	2790	2810	2815	2865	2935	2970	3006	9050#
2335	9059#											
2519	2800	2805	2825	2830	2940	9069#						
2945	2950	2960	2965	9078#								
2329	9006#											
2369	9088#											
2258	2268	2273	2284	2289	2294	2309	2314	2319	2324	2374	2379	2384
2394	2399	2404	2409	2								

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 191
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0190

		8713#	9011	9031	9040	9052	9071	9082	9090	9101	9116	9131		
DM10	052134	2462	2808	2813	8746#									
DM11	052163	8750#	9013											
DM13	052200	2943	8753#											
DM14	052264	2948	8762#											
DM17	052350	2357	2362	2387	2582	2908	2913	3129	3134	8771#				
DM19	052374	2277	2552	8775#										
DM2	051664	8716#	9001	3015	9022	9033	9044	9054	9063	9073	9084	9094	9109	9124
		9139												
DM20	052417	2557	8779#											
DM21	052452	2261	2562	8784#										
DM22	052500	2337	2547	2687	2702	3089	3094	3119	3124	8788#				
DM24	052532	2297	2302	2317	2322	2332	2542	3064	3069	8793#				
DM25	052557	2367	2607	2612	2637	2858	2863	2999	8797#					
DM26	052602	2216	2221	2241	2246	2271	2282	2382	2392	2933	2938	8801#	2537	8805#
DM27	052632	2206	2211	2256	2266	2352	2372	2377	2417	2492	2517	2532		
DM28	052663	8810#	9103	9118	9133									
DM29	052673	8812#	9107	9122	9137									
DM3	051734	8723#	9003	9017	9024	9035	9046	9056	9065	9075	9086	9096	9111	9126
		9141												
DM30	052703	2572	2788	2798	2803	3109	3114	3149	3154	8814#				
DM31	052735	8819#	9105	9120	9135									
DM32	052762	2226	2231	2251	2287	2292	2397	2402	2497	8823#				
DM39	053014	2823	2828	3099	3104	3139	3144	8828#						
DM40	053047	8833#	9061											
DM41	053103	3004	8838#											
DM42	053130	2958	8842#											
DM43	053201	2963	8849#											
DM44	053252	2968	8856#	9080										
DM47	053331	2487	8864#											
DM49	053401	8871#	9029											
DM51	053456	2347	2577	2642	2647	2652	2657	8879#						
DM52	053511	2236	8884#											
DM53	053545	2307	2312	2407	2412	8889#								
DM54	053573	2422	8893#											
DM56	053645	8901#	9092											
DM57	053671	2502	2507	8905#										
DM6	052005	8730#	9042											
DM8	052040	2327	8735#											
DM9	052061	2567	8738#											
DI	= 040000	1200#												
DISPLA	= 001142	1864#	3214#	3222#	7191#	7213#								
DISPRE	= 000174	1346#	3222											
DLT	= 100000	1219#	4813											
DLY	= 033010	3472	3583	5652	6246#	6249								
DMD	= 000040	1260#												
DOCMD	032224	3795	3838	3853	3979	4431	4457	4495	4526	4558	4636	4696	4738	4858
		5058	5098	5166	5181	5217	5257	5407	5422	5582	5597	5748	5906	6102#
		6400	6925	6942	6957									
DOTIM	007522	2140#	3328#	3337#	5629									
DPAT1	001502	2022#	5468	5521	5534									
DPAT2	001504	2023#												
DRA	= 000001	1242#	3519	3620										
DRDY	= 000200	1249#												
DRIVS	007474	2124#	3416#	3423#	3440	3456#	3486#	3505	3607#	3674	5943	6043#		
DRIVO	007476	2129#	3310	3418	3458	3569	6026							

SEO 0191

[illegible]

SEQ 0192

[illegible]

CZR61ED UNIBUS RK6 DR PRT2
CZR61E.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 194
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0193

EM21	047436	2210	2220	2250	2260	2802	2827	2862	2937	8501*							
EM22	047460	2466	8504*														
EM23	047565	2471	8517*														
EM24	047665	2276	8529*														
EM25	047720	2351	8534*														
EM26	047772	2416	8541*														
EM27	050051	2426	8549*														
EM3	046303	6993	8394*														
EM36	050170	2536	2857	2998	8563*												
EM39	050233	2356	2501	2636	2807	8569*											
EM4	046356	8402*															
EM40	050302	2361	2506	2812	8576*												
EM41	050340	2331	8581*														
EM5	046432	2165	2596	8410*													
EM55	050406	2701	8588*														
EM6	046453	2171	8413*														
EM63	050441	2797	8593*														
EM65	050462	2832	8596*														
EM69	050504	2967	8599*														
EM7	046474	2177	8416*														
EM73	050535	2586	8604*														
EM74	050556	3003	8607*														
EM79	050603	2591	8611*														
EM8	046563	8426*															
EM80	050624	2230	8614*														
EM81	050663	2235	8620*														
EM82	050727	2240	8627*														
EM83	051001	2245	8635*														
EM84	051036	2587	2592	2597	8640*												
EM85	051127	2336	2346	8650*													
EM86	051172	2541	2546	2576	8656*												
EM88	051230	2421	8661*														
EM9	046637	2188	8434*														
EM93	051271	2366	8667*														
EM94	051552	2486	8676*														
EM95	051406	2491	2496	8681*													
EM96	051442	2511	8686*														
EM97	051473	2516	8691*														
ERRVEC=	000004	1140*	3211	3212*	3223*	3230*	3231*	3244*	3245*	3313*	3317*	3324*	3362*	3380*			
		3381*	7152	7153*	7155*	7158*											
ESEC	001400	1981*															
E.A0	007444	2101*	3885*	4079*	4098*	4186*	4205*	4252*	4269*	4331*	4350*	4384*	4476*	4503*			
		4539*	4566*	4639*	4657*	4709*	4758*	4821*	4947*	4966*	5000*	5019*	5061*	5079*			
		5220*	5238*	5292*	5311*	5363*	5382*	5486*	5505*	5540*	5559*	5665*	5794*	5839*			
		6269*	6273*	6277	6279*	6308	8963	8973	8984								
E.A1	007450	2103*	3887*	4081*	4100*	4188*	4207*	4254*	4271*	4333*	4352*	4386*	4478*	4505*			
		4541*	4568*	4641*	4659*	4711*	4760*	4823*	4949*	4968*	5002*	5021*	5063*	5081*			
		5222*	5240*	5294*	5313*	5365*	5384*	5488*	5507*	5542*	5561*	5667*	5796*	5841*			
		6270*	6281	6283*	6328	8963	8973	8984									
E.A2	007454	2105*	3889*	4083*	4102*	4190*	4209*	4256*	4273*	4335*	4354*	4388*	4480*	4507*			
		4543*	4570*	4661*	4762*	4825*	4951*	4970*	5004*	5023*	5083*	5242*	5296*	5315*			
		5367*	5386*	5490*	5509*	5544*	5563*	5669*	5798*	5843*	6271*	6285	6287*	8973			
		8984															
E.A3	007460	2107*	6272*														
E.B0	007446	2102*	3886*	4080*	4099*	4187*	4206*	4253*	4270*	4332*	4351*	4385*	4477*	4504*			
		4540*	4567*	4640*	4658*	4710*	4759*	4822*	4948*	4967*	5001*	5020*	5062*	5080*			

M15

CZP6IEO UNIBUS RK6 DR PRT2
CZP6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 195
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0194

		5221*	5239*	5293*	5312*	5364*	5383*	5487*	5506*	5541*	5560*	5666*	5795*	5840*
		6289	6291*	6318	8963	8973	8984							
E.B1	007452	2104*	3888*	4082*	4101*	4189*	4208*	4255*	4272*	4334*	4353*	4387*	4479*	4506*
		4542*	4569*	4642*	4660*	4712*	4761*	4824*	4950*	4969*	5003*	5022*	5064*	5082*
		5223*	5241*	5295*	5314*	5366*	5385*	5489*	5508*	5543*	5562*	5668*	5797*	5842*
		6293	6295*	6338	8963	8973	8984							
E.B2	007456	2106*	3890*	4084*	4103*	4191*	4210*	4257*	4274*	4336*	4355*	4389*	4481*	4508*
		4544*	4571*	4662*	4763*	4826*	4952*	4971*	5005*	5024*	5084*	5243*	5297*	5316*
		5368*	5387*	5491*	5510*	5545*	5564*	5670*	5799*	5844*	6297	6299*	8973	8984
E.B3	007462	2108*	3891*	4085*	4104*	4192*	4211*	4258*	4275*	4337*	4356*	4390*	4482*	4509*
		4545*	4572*	4663*	4764*	4827*	4953*	4972*	5006*	5025*	5085*	5244*	5298*	5317*
		5369*	5388*	5492*	5511*	5546*	5565*	5671*	5800*	5845*	6301	6303*	8984	
E.DOT	015444	3711*	3718*	3728*	6273									
FATT1	032634	3847	5175	5416	5591	6199*	6929	6951						
FATT2	032730	3983	4653	4742	4862	5075	5234	5752	5910	6226*				
FHDHM	035070	6602*	6610											
FHD TAB	035212	4137	4424	4594	5882	6635*								
FLGTST	035472	6677	6681	6688	6692	6704*								
FLOAD	035144	6619*												
FMT =	000020	1227*												
FMT1	001470	2016*	6645*	6646*	6647*	6652								
FORM	031174	5773	5865	5868*	6965									
FORMAT	001466	2015*	4136*	4423*	4593*	5881*	6645	6674						
FRCYL	001344	1964*	5763*	8923	8947									
FRDY	032320	3451	3465	3474	3576	3585	5656	6105	6116	6128*	6131	6462		
FRDY1	032366	6142*	6145	6448										
FSEC23	035000	5637	6578*											
FTITLE	001340	1960*	5999	6001*										
GBA	032126	3280	6066*											
GDRVS	031766	3278	6021*											
GINT	032154	3282	6079*											
GNS =	***** U	1345	8103	8104	8105	8106	8107	8109	8111	8112	8113	8114	8115	8116
		8117												
GO =	000001	1191*												
GSTAT	033664	3749	3842	3880	4037	4061	4143	4168	4233	4319	4377	4600	4704	4815
		4929	4988	5112	5170	5274	5327	5351	5411	5475	5528	5586	5658	5732
		5783	5828	5888	6209	613	6233	6237	6397*	6466	6477	6489	6524	6604
		6621	6946											
GSTAT1	033720	6267	6408*											
GSTAT2	034150	6413	6421	6429	6436	6444*								
GTSWR =	104406	6010	8109*											
HASOF	007372	2072*	6167*	6183	8919	8928	8936	8944	8952	8960	8970	8981	8993	
HBA	007362	2068*	6163*	8919	8928	8936	8944	8952	8960	8970	8981	8993		
HCS1	007354	2065*	3468*	3469*	3470	3476	3579*	3580*	3581	3587	3794*	3837*	3852*	3877*
		3881	3978*	3986	4034*	4038	4058*	4062	4140*	4144	4165*	4169	4230*	4234
		4316*	4320	4374*	4427*	4430*	4433	4453*	4456*	4459	4494*	4522*	4525*	4528
		4557*	4597*	4601	4635*	4695*	4737*	4745	4799*	4802	4857*	4865	4926*	4930
		4985*	4989	5057*	5097*	5109*	5165*	5180*	5216*	5256*	5271*	5275	5348*	5352
		5406*	5421*	5472*	5476	5525*	5529	5581*	5596*	5645*	5646*	5647	5661	5729*
		5733	5747*	5755	5779*	5788	5824*	5833	5885*	5889	5905*	5913	6102*	6103
		6113*	6114	6160*	6372	6376	6399*	6444*	6445*	6446	6467	6506*	6507*	6508
		6778	6924*	6941*	6956*	8916	8925	8933	8941	8949	8957	8967	8978	8990
HCS2	007356	2066*	3515	3517	3521	3533	3616	3618	3622	3625	4322	4378	4530	4991
		5113	5354	5531	6161*	6381	6385	8916	8925	8933	8941	8949	8957	8967
		8978	8990											
HDA	007364	2069*	6164*	6821	8919	8928	8936	8944	8952	8960	8970	8981	8993	

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 196
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0195

HDB	007376	2074#												
HDC	007374	2073#	6168*	6813	8919	9928	8936	8514	8952	8960	8970	8981	8993	
HDS	007366	2070#	3519	3620	6165*	8916	8925	8933	8941	8949	8957	8967	8978	8990
HDTAB	001522	2033#	4022	4028	4031	4130	4415	4586	5715	5722	5725	5875	6637	
HDWD	001510	2026#	8939											
HD1	001464	2014#	6639*	6640*	6641*	6642*	6643*	6644*	6651					
HEAD	001460	2012#	4135*	4422*	4592*	5860*	6639	6716						
HEADR	001462	2013#	6568*	6569*	6570*	6571*								
HER	007370	2071#	3510	3611	4065	4172	4236	4243	4462	4933	5278	5479	6166*	8916
		8925	8933	8941	8949	8957	8967	8978	8990					
HMR1	007400	2075#	6169*	8916	8925	8933	8941	8949	8957	8967	8978	8990		
HMR2	007402	2076#	3478	3589	3750	3785	3798	3843	4497	4560	4705	4752	4842	5171
		5412	5587	6146*	6151*	6170*	6415	6423	6431	6438	6490	6605	6622	6947
		8916	8925	8933	8941	8949	8957	8967	8978	8990				
HMR3	007404	2077#	5784	5829	6147*	6152*	6171*	6416	6424	6432	6439	6478	6525	8916
		8925	8933	8941	8949	8957	8967	8978	8990					
HOLD	032454	6132	6136	6160#	6409									
HPAT	007410	2079#	6173*	8919	8928	8936	8944	8952	8960	8970	8981	8993		
HPEND	007342	2047#	3307*	5920	5922*	6937*								
HPOS	007406	2078#	6172*	8919	8928	8936	8944	8952	8960	8970	8981	8993		
HT	= 000011	1050#	7287	7328										
HVRC	= 000400	1231#												
HWC	= 007360	2067#	6162*	8919	8928	8936	8944	8952	8960	8970	8981	8993		
HZ	001364	1973#	6843											
H.A0	007424	2090#	6308	6438*	8963	8976	8987							
H.A1	007430	2092#	6328	6431*	8963	8976	8987							
H.A2	007434	2094#	6423*	6535	8976	8987								
H.A3	007440	2096#	6415*											
H.B0	007426	2091#	6318	6439*	8963	8976	8987							
H.B1	007432	2093#	6338	6432*	8963	8976	8987							
H.B2	007436	2095#	6424*	6548	8976	8987								
H.B3	007442	2097#	6416*	6558	6568	8987								
IDAE	= 002000	1233#												
IE	= 000100	1192#												
ILF	= 000001	1223#												
INTER	036720	6094	7033#											
IOTVEC	= 000020	1145#	3195*	3196*										
IP	= 000100	1210#												
LC	015434	3710#	3717*	3724#	3875									

[illegible]

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 3C (1052) 10-JAN-78 11:25 PAGE 198
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0197

OP =	000200	1211#									
O.ADR1	055472	9274*	9325*	9327*	9497	9498*	9536*	9551	9568	9759#	
O.BACK	056216	9369	9410#								
O.BD	057510	9555	9556	9704#							
O.BKPT	055724	9320#	9373								
O.BK1	056656	9268	9533#								
O.BK2	056712	9538	9541#								
O.BK3	056730	9540	9547#								
O.BRK	056646	9285	9531#								
O.BUF	057537	9613	9622	9627	9732#						
O.B2	057012	9552	9561#								
O.B3	057052	9560	9569#								
O.CADV	057222	9386	9403	9426	9430	9460	9464	9569	9612#		
O.CLGL	056040	9343	9345	9353#							
O.CLGT =	000014	9357	9722#								
O.CLS1	057446	9680	9682#								
O.CR	057512	9689	9691	9692	9706#						
O.CRET	056152	9364	9393#								
O.CRLF	057456	9401	9458	9554	9593	9598	9689#				
O.CRLS	057464	9338	9691#								
O.CRS	057470	9690	9692#								
O.CSR1	057506	9589#	9605	9700#							
O.CSR2	057507	9590#	9606	9701#							
O.CT	055474	9518#	9561*	9563*	9760#						
O.C1	056624	9521#	9562								
O.DCD	055770	9326	9328	9338#	9395	9446	9571				
O.DCD1	055774	9339#	9387	9431							
O.EFF	056300	9372	9437#								
O.EFF1	056416	9449	9468#								
O.ENTR	055504	9265#									
O.ERR	055754	9312	9323	9335#	9358	9383	9441	9684			
O.ERR1	056310	9441#	9514								
O.ERR2	056130	9383#	9417	9490							
O.FTYP	057364	9337	9405	9420	9462	9567	9638	9654	9659#		
O.GET	057306	9291	9341	9635#	9640	9642	9644				
O.GO	056462	9366	9488#								
O.G02	056540	9499#	9524								
O.LG =	000006	9295	9730#								
O.LGCH	057515	9354	9710#	9722							
O.LGDR	056070	9361	9363#	9375							
O.LGL =	000030	9375#									
O.LGL1	056042	9354#	9359								
O.LGL2	056062	9355	9360#								
O.MSK	055464	9442	9443	9445	9752#						
O.ODT	055500	1354	9263#	9741							
O.OFST	056226	9370	9416#								
O.OF1	056274	9429	9431#								
O.OP1	056164	9367	9399#								
O.OP2	056172	9316	9401#	9412							
O.ORPC	055676	9308#	9368								
O.P	057505	9283#	9488#	9513	9515*	9564*	9698#				
O.PRI	055462	9275	9537	9541	9751#						
O.PROC	056570	9374	9513#								
O.PR1	056612	9517	9519#								
O.RALL	055744	9286	9321	9327#							
O.RCSR =	177560	9253#	9589	9591*	9601	9603	9605*	9635			

CZR6IED UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43

MACY11 30A(1052) 10-JAN-78 11:25 PAGE 199
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0198

O.RDB =	177562	9252#	9637						
O.REGT	055626	9291#	9365						
O.RSE1	057204	9602	9605#						
O.RSP	055636	9293#	9296						
O.RST	055542	9264	9273#						
O.RSTT	057154	9494	9520	9598#					
O.RST1	055572	9272	9280#						
O.RTIT	056566	9282#	9508#						
O.SCAN	056000	9302	9341#	9352					
O.SP	055670	9294	9303#						
O.SPC	057234	9615#	9623						
O.SPI	055656	9299#	9304						
O.STM =	000340	9242#	9284	9493	9519	9521			
O.STRT	055530	9263	9270#						
O.SVR	057064	9273	9533	9575#					
O.SVTT	057122	9553	9565	9589#					
O.T	057504	9495*	9522*	9534	9697#				
O.TBIT	056512	9495#	9535						
O.TBT =	000020	9243#	9496	9523	9559				
O.TC	057450	9677	9683#						
O.TCLS	057432	9308	9393	9399	9410	9676#			
O.TCSR=	177564	9255#	9590	9592*	9599	9606*	9659	9665	
O.TDB =	177566	9254#	9661*	9667*					
O.TL	057531	9292	9295	9303	9724#	9730			
O.TRTC	057546	9327	9498	9737#					
O.TVEC=	000014	9241#	9266	9284*	9285*				
O.TYPE	057350	9557	9628	9651#	9655	9693			
O.TYP1	057430	9652	9663	9671#					
O.TYP2	057410	9665#	9666	9669					
O.UIN	055476	9274	9497*	9536	9761#				
O.UPC	055456	9267*	9492*	9507	9531*	9548	9550*	9749#	
O.URD	055440	9270	9300	9742#					
O.USP	055454	9271*	9576*	9577	9748#				
O.UST	055460	9265*	9496*	9506	9523*	9532*	9539	9559*	9750#
O.WDS	056306	9438	9440#						
O.WDS2	056324	9445#	9467						
O.WDS3	056354	9456#	9469	9475	9484				
O.WDS4	056412	9456	9466#						
O.WRD	056120	9363	9379#						
O.WRDA	056144	9380	9388#						
O.WRD1	056126	9382#	9389	9406					
O.WSCH	056304	9371	9439#						
O.XXX	057502	9575*	9583	9696#					
O.45	055604	9281	9283#						
PACK =	000003	1178#	3794						
PARAM	001336	1959#	3161*	3164*	3274				
PARSRT	012546	1351	3161#						
PAT =	000020	1259#							
PCA =	004000	1266#							
PCD =	010000	1267#							
PCLKF	007520	2139#	3319*	3326*	6830	6851			
PCVEC	001332	1953#	3320	3327					
PCYL	001352	1967#							
PFSRT	015446	3730#	7068						
PGE =	002000	1214#							
PIP =	020000	1252#							

[illegible]

```

RKER      = 000014
RKMR1     = 000026

RKMR2     = 000034
RKMR3     = 000036
RKPRI     = 001316
RKVEC     = 001314
RKWC      = 000002

```

1163#	3370	6166										
1167#	3374	3748*	3841*	5169*	5326*	5410*	5585*	6169	6208*	6212*	6232*	6236*
6412*	6420*	6428*	6465*	6476*	6488*	6505*	6523*	6603*	6620*	6945*		
1168#	3375	6146	6151	6170	6516							
1169#	3376	6147	6152	6171								
1946#	6095											
1945#	3301*	6083*	6086*	6093								
1158#	3366	3876*	4032*	4055*	4131*	4163*	4227*	4313*	4372*	4416*	4450*	4517*
4587*	4922*	4982*	5107*	5269*	5345*	5470*	5523*	5643*	5726*	5777*	5822*	5876*

```

RLS      = 000010
RSEC     034714
RTT      = 000006
SAVREG= 104413
SBPAR    036156
SCLR     = 000040
SCOP1    = 104415

```

[illegible]

```
SCOPIS 036230
SEC 001370
SECFLG 035372
SECNT 001374
SECTOR 001402
```

8117#	8117											
6892#	6844*											
1977#	6673#											
6653												
1979#												
1982#	4052*	4053	4070*	4071	4158*	4159	4177*	4178	4225	4301	4314	4373
4446#	4447	4467*	4468	4520	4918*	4919	4938*	4939	4983	5108	5265*	5266
5283*	5284	5346	6478*	6479*	6480*	6481*	6482*	6483*	6558*	6559*	6560*	6561*
6562*	6563*	6580	6584	6587	6735*	6739	6740*	6741*	6752	6762*	6763	6764*

```
SEEK      = 000017
SELDREV= 000001
SETINT    032206
SIZFLG    007524
SKI       = 000002
SORT      035550
SPACE2    055414
SRTSPL= 000011
SRTTAB    002132
ST        = 177776
STACK     = 001100
```

8783*	3978	4737	4857	5747	5905								
1184*	3468	3579	4430	4456	4494	4525	4557	6399	6444	6506			
1177*													
3302	6084	6093*											
2141*	3303*	3453	6039*	6059*									
1224*													
6732*													
9188	9197	9208	9220*										
1181*	6924												
2035*	6744												
9239*	9265	9266*	9279*	9493*	9519*	9521*	9547*						
1045*	3166	3193	3355	3406	3461	3567	3572	3670	3740	3780	3789	3829	
4009	4047	4125	4153	4220	4306	4366	4410	4441	4581	4625	4913	5206	
5337	5456	5627	5707	5766	5814	5941	6363	7027	7065	9216			

```

START      031634
START      012556
START1     013164
STKLMT=    177774
STOP       036262
ST2        013234
ST3        013300
ST4        013332
ST5        013356
SUBCLR     034214

```

[illegible]

```
SVAL = 100000
SWR  001140
```

[illegible]

SEQ 0201

[illegible]

SEO 0202

WRTCHK=	000031	1189#	4316	4374	4522	4985	5109	5348	5525										
WRTGAT=	040000	1269#																	
\$APTHO	001000	1376	1382#																
\$ASTAT=	*****	7429	7444																
\$ATYC	040432	7400	7402#																
\$ATY1	040406	7398#																	
\$ATY3	040414	7277	7399#																
\$ATY4	040424	7229	7401#																
\$AUTOB	001134	1860#	6012*	7627	7784														
\$BASE	001264	1930#	3300*	3363	6070*	6072*													
\$BDAOR	001122	1855#																	
\$BDDAT	001126	1857#																	
\$BELL	001200	1881#	7216	7249	7580	7777													
\$CDW1	001270	1932#																	
\$CDW2	001272	1933#																	
\$CHARC	040156	7294*	7304*	7311	7320*	7325#													
\$CKSWR	041400	7613#	8111																
\$CMTAG	001100	1843#	3188	3189	3197	3203	3204	3205											
\$CM3	= 000000	1873#																	
\$CM4	= 000006	1873#	1874#	1875#	1876#	1877#	1878#	1879#											
\$CNTLC	042310	7568	7649	7777#															
\$CNTLG	042322	7634	7779#																
\$CNTLU	042315	7659	7752	7778#															
\$CPUOP	001236	1904#																	
\$CRLF	001205	1883#	3425	3489	3971	3972	5044	5127	5128	7224	7249	7293	7328	7670					
		7757	7777	7838	9164	9169	9179	9192	9205										
\$DBLK	040376	7352	7386	7394#															
\$DB2D	042632	7890#	7952																
\$DB2D	042512	3969	7849#																
\$DDW0	001274	1934#																	
\$DDW1	001276	1935#																	
\$DDW2	001300	1936#																	
\$DDW3	001302	1937#																	
\$DDW4	001304	1938#																	
\$DDW5	001306	1939#																	
\$DDW6	001310	1940#																	
\$DDW7	001312	1941#																	
\$DECVL	043012	7892	7938#																

\$ERROR	037500	3197	7209*											
\$ERRPC	001116	1853#	7218*	7219*	7220	7249	8915	8923	8931	8939	8947	8955	8963	8373
		8984												
\$ERRTB	007526	2156#	9161											
\$ERTTL	001112	1850#	7217*	7249										
\$ESCAP	001176	1880#	3205*	3308*	4634*	4651*	5056*	5073*	5215*	5232*	5773*	5868*	5903*	5939*
		6306	6310*	6320*	6330*	6340*	6360*	6364*	6971*	7019*	7022*	7189*	7240	7242
		7249												
\$ETABL	001230	1899#												
\$ETEND	001314	1388	1942#											
\$FATAL	001212	1892#	7435*											
\$FFLG	040652	7398*	7401*	7429	7438*	7446#								
\$FILLC	001156	1871#	7297	7328										
\$FILLS	001155	1870#	7328											
\$GADDR	001120	1854#												
\$GOODAT	001124	1856#												
\$GET42	031570	5961#												
\$GTSWR	041470	7635#	8109											
\$HD =	000000	1016												
\$HIBTS	001000	1383#												
\$HIOCT	042510	7826*	7837#											
\$ICNT	001104	1847#	7180*	7181	7183*	7194								
\$INTAG	001135	1861#	7632*	7651	7671	7784								
\$ITEMB	001114	1851#	7220*	7228	7249	9155								
\$LF	001206	1884#	7249	7328	7757	7777	7838							
\$LFLG	040651	7439*	7445#											
\$LPADR	001106	1848#	3207*	7002	7171*	7187*	7192	7194	7188*	7194	7239			
\$LPERR	001110	1849#	3208*	6365	6896	6899*	7028	7171						
\$MADR1	001242	1917#												
\$MADR2	001246	1921#												
\$MADR3	001252	1924#												
\$MADR4	001256	1927#												
\$MAIL	001210	1384	1388	1890*	3225	6006	7186	7226	7272					
\$MAMS1	001240	1911#												
\$MAMS2	001244	1919#												
\$MAMS3	001250	1922#												
\$MAMS4	001254	1925#												
\$MBADR	001002	1384#												
\$MFLG	040650	7399*	7405	7440*	7444#									
\$MNEW	042340	7638	7782#											
\$MSGAD	001224	1897#	7415*	7418										
\$MSGLG	001226	1898#	7420*											
\$MSGTY	001210	1891#	7413	7421*	7433	7437*								
\$MSWR	042327	7635	7780#											
\$MTYP1	001241	1912#												
\$MTYP2	001245	1920#												
\$MTYP3	001251	1923#												
\$MTYP4	001255	1926#												
\$MULT	043122	7994#												
\$MXCNT	037476	7184	7194#											
\$NULL	001154	1869#	7299	7328										
\$NWTST=	000001	3343#	3345	3389#	3391	3550#	3552	3659#	3661	3731#	3733	3772#	3774	3804#
		3906	3997#	3999	4113#	4115	4400#	4402	4610#	4612	4890#	4892	5194#	5196
		5432#	5434	5607#	5609	5686#	5688							
\$OCNT	041076	7484*	7513*	7526#										
\$OCTLV	042614	3751	7851	7876#										

K16

CZR6IEO UNIBUS RK6 DR PRT2
CZR6IE.P11 10-JAN-78 09:43MACY11 30A(1052) 10-JAN-78 11:25 PAGE 206
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0205

\$OMODE	041100	7479*	7483*	7488	7491*	7502*	7528*							
\$OVER	037462	7148	7164	7172	7182	7191*								
\$PASS	001216	1894*	3225*	3742	3965	5949*	5950*	5958	5971	5978	7178	7195		
\$PASTM	001006	1386*												
\$PWACT	037040	7059*	7060*	7070*										
\$PWADM	036754	3201	7053*	7062										
\$PWUP	036766	7053	7059*											
\$QUES	001204	1882*	7249	7328	7689	7760	7777	7835	7838					
\$RDCHR	041742	7702*	8112											
\$RDDEC=	***** U	8115												
\$RDLIN	042032	7725*	8113											
\$RDOCT	042352	7799*	8114											
\$RDSZ =	000022	7718*												
\$RESRE	043272	8060*	8116											
\$RTNAD	031612	5970*												
\$R2A =	***** U	8117												
\$SAVRE	043234	8044*	8115											
\$SB20	043026	7950*												
\$SCOPE	037220	3195	7145*											
\$SETUP=	000137	3158*	3194	3195	3197	3199	3201	3203	3204	3205	3207	5947	6003	7146
		7210	7236	7244	7572	7577	7578	7608	7784					
\$STUP =	177777	3158*												
\$SUPRS	043062	3768	3970	7967*										
\$SVLAD	037426	7156	7185*											
\$SVPC =	000244	1360*	1365											
\$SWR =	167400	1001*	1016	1021	1022	1023	1024	1025	1026	1027	1028	1879	1880	1881
		3204	3205	3207	3208	3354	3405	3566	3669	3739	3779	3828	4008	4124
		4409	4624	4912	5205	5455	5626	5706	5933	5948	5963	5969	5971	7137
		7138	7139	7140	7141	7147	7159	7161	7162	7165	7166	7167	7174	7175
		7176	7188	7191	7194	7201	7202	7203	7204	7205	7214	7221	7233	7237
		7249												
\$SWREG	001232	1902*	3228											
\$SWRMK=	000000	1028	1029	7141	7142	7163								
\$TESTN	001214	1893*	3671*	7186*	8915	8922	8923	8931	8939	8947	8955	8963	8973	9984
\$TIMES	001174	1879*	3204*	3354*	3405*	3566*	3669*	3739*	3779*	3828*	4008*	4124*	4409*	4624*
		4912*	5205*	5455*	5626*	5706*	5940*	5948*	7174*	7181	7184*	7194		
		1866*	7532	7553	7564	7589	7617	7644						
\$TKB	001146	7533*	7548*	7578	7595*	7709	7711*							
\$TKCNT	041102	3171	7548*	7569	7630									
\$TKINT	041112	7537*	7603	7714										
\$TKQEN=	041111	7534*	7549*	7550	7601*	7602*	7603	7605*						
\$TKQIN	041104	7535*	7550*	7712	7713*	7714	7716*							
\$TKQOU	041106	7536*	7549	7605	7716									
\$TKQSR	041110	1865*	7532	7554*	7585*	7587	7593*	7615	7631*	7641	7653*	7673*		
\$TKS	001144	7551	7564*											
\$TKSRV	041162	1873*												
\$TMP0	001160	1874*												
\$TMP1	001162	1875*												
\$TMP2	001160	1876*												
\$TMP3	001160	1877*	3309*	3469	3488	3502*	3512*	3580	3595	3598*	3613*	3657*	3696*	3708
\$TMP4	001160	5458	5646	6102	6113	6445	6507							
\$TMP5	001172	1878*	6306*	6360	6364									
\$TN =	000021	1002*	1016	3331	3343	3354*	3382	3389	3405*	3550	3566*	3601	3659	3669*
		3671	3672	3715	3722	3731	3739*	3743	3772	3779*	3786	3799	3804	3828*
		3930	3966	3997	4008*	4113	4124*	4400	4409*	4610	4624*	4887	4890	4912*
		5153	5194	5205*	5432	5455*	5607	5626*	5631	5639	5681	5686	5706*	

\$TNPWR	042742		7897	7898	7918#											
\$TPB	001152		1868#	7317*	7328											
\$TPFLG	001157		1872#	7266	7328											
\$TPS	001150		1867#	7315	7328											
\$TRAP	043330		3199	8080#												
\$TRAP2	043352		8091#	8102												
\$TRP =	000016		8095#	8104#	8105#	8106#	8107#	8108#	8109	8110#	8111	8112#	8113#	8114#	8115#	
			8116#	8117#	8118#											
\$TRPAD	043364		8085	8102#												
\$TSTM	001004		1385#													
\$TSTNM	001102		1845#	3672*	5947*	5380*	7136	7163	7185*	7186	7191	7195	7213	7249		
\$TTYIN	042266		7727	7728	7740	7758	7772	7776#								
\$TYPNB=	***** U		8108													
\$TYPOS	040162		734^#	8107												
\$TYPE	037700		7266#	7426	8095	8103										
\$YPEC	040112		7296	7303	7310	7315#	7316	7675								
\$YPEX	040160		7321	7323	7326#											
\$YPOC	040700		7482#	8104												
\$YPON	040714		7481	7484#	8106											
\$YPOS	040654		7477#	8105												
\$UNIT	001222		1896#	3249*	3312*	3682*	3689	3699	3836	3851	4694	5096	5164	5179	5255	
			5405	5420	5580	5595	6182	6201	6227	6269	6270	6271	6272	6398	6411	
			6419	6427	6435	6464	6940	6955								
\$UNITM	001010		1387#													
\$USWR	001234		1903#													
\$VECT1	001260		1928#													
\$VECT2	001262		1929#													
\$XTSTR	037232		7150#													
\$SGET4=	000000		5963#													
\$OFILL	041077		7478*	7482*	7492	7527#										
\$4OCAT=	***** U		7147	7223												
.	= 055500		1341#	1345#	1350#	1353#	1360	1361#	1363#	1365#	1366#	1372	1373#	1375#	1377#	
			1842#	1885	2033#	2034#	2035#	2037#	2038#	2039#	2040#	2041#	3192	3207	3208	
			5971	5975	7055	7111	7194	7195	7249	7328	7394#	7447#	7532	7536#	7537	
			7538#	7776#	7777	7784#	7838	7876#	7938#	8914#	9229#	9230#	9375	9600	9604	
			9636	9660	9722	9730	9733#	9741#								
			7399	7402												
. \$ASTA= ***** U			1372#	1377												
. \$X = 001000																

SEQ 7207

CALIB	1479#	3834	5161	5402	5577	6938									
CHECK	1433#	3892	4086	4105	4193	4212	4259	4276	4338	4357	4391	4483	4510	4546	4573
	4643	4664	4713	4765	4828	4954	4973	5007	5026	5065	5086	5224	5245	5299	5318
	5370	5389	5493	5512	5547	5566	5672	5801	5846						
COMMEN	1152#														
CWD2	1448#	5325													
DRCLR	1461#	3849	4692	5094	5177	5253	5418	5593	6953						
ENDCOM	1152#														
EOPCM	1800#														
ERROR	1046#	5937													
	3385	3444													
	3631	3635	3452	3466	3475	3498	3507	3526	3530	3541	3545	3577	3586	3604	
	3863	3879	3643	3745	3783	3792	3796	3800	3832	3839	3845	3848	3854	3857	
	3984	3988	3883	3895	3897	3898	3901	3908	3913	3918	3921	3933	3975	3980	
	4109	4110	4018	4036	4040	4060	4073	4077	4089	4090	4091	4092	4095	4108	
	4215	4216	4111	4128	4143	4146	4156	4167	4180	4184	4196	4197	4198	4199	
	4281	4282	4217	4218	4223	4232	4238	4245	4248	4262	4263	4264	4265	4279	
	4369	4376	4290	4309	4318	4326	4329	4341	4342	4343	4344	4360	4361	4363	
	4474	4486	4380	4394	4395	4396	4397	4413	4429	4432	4435	4444	4455	4458	
	4537	4549	4487	4488	4492	4496	4499	4513	4514	4515	4515	4516	4524	4527	
	4637	4646	4550	4551	4552	4559	4562	4576	4577	4578	4579	4584	4599	4603	
	4707	4716	4647	4648	4649	4654	4667	4668	4669	4670	4674	4684	4689	4697	
	4780	4785	4717	4718	4719	4728	4733	4739	4743	4747	4754	4768	4769	4770	
	4859	4863	4790	4801	4804	4816	4831	4832	4833	4834	4838	4844	4848	4852	
	4979	4987	4867	4916	4928	4941	4945	4957	4958	4959	4960	4963	4976	4977	
	5069	5070	4995	4998	5010	5011	5012	5013	5029	5030	5031	5032	5049	5059	
	5185	5211	5071	5076	5089	5090	5091	5092	5099	5102	5111	5167	5173	5176	
	5286	5290	5218	5227	5228	5229	5230	5235	5248	5249	5250	5251	5258	5261	
	5358	5361	5302	5303	5304	5305	5308	5321	5322	5323	5324	5330	5333	5340	
	5465	5474	5373	5374	5375	5376	5392	5393	5394	5395	5408	5414	5417	5423	
	5550	5551	5484	5496	5497	5498	5499	5502	5515	5516	5517	5518	5527	5535	
	5657	5659	5552	5553	5569	5570	5571	5572	5583	5589	5592	5598	5601	5635	
	5786	5790	5663	5675	5676	5677	5678	5713	5731	5735	5749	5753	5757	5771	
	5855	5873	5804	5805	5806	5807	5810	5817	5826	5831	5835	5849	5850	5851	
	6463	6514	5887	5891	5907	5911	5915	6378	6383	6387	6401	6414	6422	6430	
ESCAPE	1152#														
F.EAB	1413#	3884	4078	4097	4185	4204	4250	4268	4330	4349	4382	4475	4501	4538	4564
	4656	4756	4820	4946	4965	4999	5018	5078	5237	5291	5310	5362	5381	5485	5504
	5539	5558	5664	5792	5837										
GETPRI	1152#														
GETSWR	1152#	6003													
HDCHK3	1590#	4796													
HDTBL	1603#	4133	4420	4590	5878										
LOOP	1397#	3459	3570	3788	4046	4152	4219	4304	4365	4440	4580	5336	5765	5812	
MSG	3343#	3345	3389#	3391	3550#	3552	3659#	3661	3731#	3733	3772#	3774	3804#	3806	3997#
	3999	4113#	4115	4400#	4402	4609#	4612	4890#	4892	5194#	5196	5432#	5434	5607#	5609
	5686#	5688													
MULT	1152#														
NEWST	1152#	3343	3389	3550	3659	3731	3772	3794	3997	4113	4400	4610	4890	5194	5432
	5607	5686													
OFFDIR	1816#	4677	4721	4773											
OFFSET	1776#	4633	5055	5214											
OWNTAG	1390#	1944													
POP	1152#	7381	7441	7442	7827	8023	8065								
PUSH	1152#	7340	7402	7404	7425	7801	7994	8045							
QKSEEK	1615#	3977	4736	4856	5746	5904									
QKSRT	1507#	6919													

RDAT	1694#	4229													
RDHOR	1558#	4797													
REPORT	1152#														
SCOPE	1047#	3353	3404	3565	3668	3738	3778	3827	4007	4123	4408	4623	4911	5204	5454
	5635	5705	5938	5946											
SETPR	1152#	7705													
SETTRA	8035#	8104	8105	8106	8107	8109	8111	8112	8113	8114	8115	8116	8117		
SETUP	1152#	3187													
SKIP	1152#	3331	3382	3601	3715	3722	3743	3786	3799	3930	3966	4887	5153	5631	5639
	5681														
SLASH	1152#														
SPACE	1152#														
STARS	1152#	1358	1369	1371	1378	1838	1885	1888	3343	3352	3389	3403	3550	3564	3659
	3667	3731	3737	3772	3777	3804	3826	3997	4006	4113	4122	4400	4407	4610	4622
	4890	4910	5194	5203	5432	5453	5607	5624	5686	5704	5930	7133	7197	7251	7330
	7397	7454	7531	7608	7623	7694	7718	7787	7840	7879	7941	7959	7982	8029	8074
	9144	9153													
SWRSU	1152#	3209#													
TRMTRP	8095#														
TYPBIN	1152#														
TYPDEC	1152#	5958													
TYPNAM	1152#														
TYPNUM	1152#														
TYPOCS	1152#	3426	3490	3536	3700	5039	5122	6256	6994	7039					
TYPOCT	1152#	7636													
TYPTXT	1152#														
WDATA	1642#	4057	4164	4452	4925	5270	5471								
WCHK	1740#	4315	4521	4984	5347	5524									
WHOR	1528#	4033	4139	4426	4596	5728	5884								
SSCMRE	1836#														
SSCMTM	1836#	1873	1874	1875	1876	1877	1878								
SSSCA	1152#														
SSSCUT	1152#	3343	3389	3550	3659	3731	3772	3804	3997	4113	4400	4610	4890	5194	5432
	5607	5686													
SSSET	8095#	8104	8105	8106	8107	8109	8111	8112	8113	8114	8115	8116	8117		
SSSETM	3225#														
SSSKIP	1152#	3331	3382	3601	3715	3722	3743	3786	3799	3930	3966	4887	5153	5631	5639
	5681														
.EQUAT	1001#	1042													
.HEADE	1001#	1006													
.SETUP	1001#	3158													
.SWRHI	1001#	1017													
.SWRLO	1001#	1029#													
.\$ACT1	1001#	1356													
.\$APT8	1886#														
.\$APTH	1001#	1367													
.\$APTY	1001#	7395													
.\$CATC	1001#	1339													
.\$CMTA	1001#	1836													
.\$DB20	1001#	7877													
.\$DB20	1001#	7838													
.\$EOP	1001#	5928													
.\$ERRO	1001#	7195													
.\$MULT	1001#	7980													
.\$RDOC	1001#	7785													
.\$READ	1001#	7529													

C01

CZR6IE UNIBUS BK6 DR PRT2 10-JAN-78 09:43 MACY11 300(1052) 10-JAN-78 11:25 PAGE 211
CZR6IE.P11 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0209

.\$SAVE	1001#	8027
.\$SB20	1001#	7939
.\$SCOP	1001#	7131
.\$SUPR	1001#	7957
.\$TRAP	1001#	8072
.\$TYPO	1001#	7328
.\$TYPE	1001#	7249
.\$TYPO	1001#	7452

. ABS. 057550 000

ERRORS DETECTED: 0

DSKZ:CZR6IE,DSKZ:CZR6IE.SEQ/SOL/CRF/NL:TOC/DOC=DSKM:CZR6IE.P11
RUN-TIME: 33 25 3 SECONDS
RUN-TIME RATIO: 273/62=4.3
CORE USED: 30K (59 PAGES)

DOCUMENT PAGES: 209