

KDJ11-A

KDJ11 MEM MGT DIAG
CZKDKBO

COPYRIGHT (c) 1983-84
AH-T707B-MC
FICHE 01 OF 01

JUL 1984
digital
Made In USA

.REM &

IDENTIFICATION

PRODUCT CODE: AC-T706B-MC
PRODUCT NAME: CZKDKBO KDJ11 MEMORY MANAGEMENT DIAGNOSTIC
PRODUCT DATE: 15-MAR-84
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHORS: HENRY ENMAN, JIM PITTMAN, BARRY IRRGANG

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

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

COPYRIGHT (C) 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

&

39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

OCT-83 REV. A
FEB-84 REV. B

HISTORY

.REM 6

FIRST RELEASE

CORRECTIONS MADE TO:

1. CORRECT VECTOR AREA MAINTENANCE PROBLEM
2. PREVENT \$TESTN FROM GETTING OUT OF SYNC WHEN SKIPPING DESELECTED TESTS.
3. TURN CACHE MEMORY SYSTEM OFF DURING NON-CACHE TESTS.
4. ENSURE THAT CPU ERROR REGISTER IS CLEARED AFTER COMPLETION OF TEST THAT MIGHT CAUSE IT TO BE SET.
5. SAVE PC AND CONTENTS OF R6 ON UNEXPECTED INTERRUPTS

ADDITIONAL TESTS TO IMPROVE TEST COVERAGE INCLUDE:

1. NON-EXISTANT MEMORY TRAP TEST

6

56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78

.REM E

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	LOADING AND STARTING PROCEDURE
2.2	PROGRAM OPTIONS
2.3	OPERATION UNDER APT
3.0	ERROR INFORMATION
4.0	PROGRESS REPORT

E

.REM 6

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS IS AN APT COMPATIBLE VERSION OF THE KDJ11 MEMORY MANAGEMENT DIAGNOSTIC. IT FOCUSES ON TESTING THE FUNCTIONALITY OF THE MEMORY MANAGEMENT FEATURES. THE TEST REQUIRES 4 MEGABYTES OF QBUS MEMORY TO FULLY TEST THE MMU ADDER. A SUBSET OF THE ADDER IS TESTED IF LESS THAN 4 MEGABYTES OF MEMORY IS AVAILABLE (MINIMUM OF 28 KBYTES). IN ADDITION, FOR TESTING IN QBUS SYSTEMS WITH ONLY 18 ADDRESS BITS, A MEANS IS PROVIDED TO SKIP TESTS WHICH REQUIRE 22 BIT ADDRESSES. THIS FEATURE IS IMPLEMENTED BY SETTING BIT 08 IN THE SOFTWARE SWITCH REGISTER (LOCATION 176) TO A ONE. DEFAULT IS TO TEST 22 BIT ADDRESSES.

1.2 SYSTEM REQUIREMENTS

KDJ11-A PROCESSOR MODULE
ENSURE THAT HALT TRAP OPTION IS DISABLED (JUMPER W9 INSTALLED)
32KW MEMORY
Q-22 BACKPLANE (18 BIT QBUS MAY BE USED WITH REDUCED TEST COVERAGE)
SERIAL LINE UNIT AND CONSOLE TERMINAL (CONSOLE TERMINAL NOT REQUIRED FOR APT)

1.3 RELATED DOCUMENTS AND STANDARDS

KDJ11-A MODULE SPECIFICATION REV 2.2
PDP11 MAINDEC SYSMAC PACKAGE
J11 CONTROL CHIP SPECIFICATION 21-17679-00
J11 DATA CHIP SPECIFICATION 21-17677-00

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE KDJ11 CPU DIAGNOSTIC MUST RUN SUCCESSFULLY PRIOR TO RUNNING THE MEMORY MANAGEMENT TEST.

1.5 ASSUMPTIONS

IT IS ASSUMED THAT THE DIAGNOSTIC OPERATOR IS FAMILIAR WITH THE XXDP, OPERATING SYSTEM AND THE J11 MICRO-ODT.

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEEDURE

LOAD PROGRAM INTO MEMORY USING STANDARD XXDP, PROCEEDURES. THE PROGRAM IS STARTED BY LOADING ADDRESS 200 AND USING THE J11 MICRO-ODT G COMMAND TO START. THE PROGRAM IDENTIFICATION MESSAGE WILL BE TYPED AFTER THE FIRST PASS OF THE COMPLETE PROGRAM.

2.2 PROGRAM OPTIONS

THE FOLLOWING ASSIGNMENTS HAVE BEEN MADE FOR THE KDJ11-A
 DIAGNOSTIC SWITCH REGISTER BITS:

BIT#15	14	13	12	11	10	9	8
-----*					DON'T	18 BIT	EXTENDED
-----*					TEST	ADDRESS	CACHE
-----*					BEVENT	ONLY	TESTS
-----*					-----*		

DEFAULT SETTINGS ARE TO TEST 22 BIT ADDRESSES. THE OTHER BITS HAVE
 NO EFFECT ON THE OPERATION OF THE PROGRAM.

PRIOR TO EXECUTING THE FIRST PASS OF THE DIAGNOSTIC THE OPERATOR
 WILL BE DIRECTED TO SET THE SWITCH REGISTER TO INDICATE WHETHER
 THE KDJ11-A UNDER TEST IS IN A SYSTEM CONFIGURED FOR 18 OR 22 BIT
 ADDRESSING. AN 18 BIT ADDRESS CONFIGURATION SHOULD BE INDICATED IF
 ANY 18 ADDRESS BIT ONLY MEMORY BOARDS RESIDE IN THE SYSTEM OR IF
 THE SYSTEM BACKPLANE DOES NOT SUPPORT 22 ADDRESS BITS.

TO CHANGE THE SWITCH REGISTER; HALT THE PROGRAM, AND EITHER RESTART
 THE PROGRAM AT 200 ANSWERING THE INITIAL QUESTIONS, OR LOAD THE
 SOFTWARE SWITCH REGISTER (ADDRESS 176) WITH THE DESIRED OPTIONS AND
 RESTART THE PROGRAM USING THE J11 MICRO-ODT P COMMAND.

2.3 OPERATION UNDER APT

THERE ARE NO DIFFERENCES IN THE EXECUTION OF THIS DIAGNOSTIC
 WHEN OPERATING IN AN APT ENVIRONMENT. PROBLEMS CAUSED BY THE
 ASYNCHRONOUS HALTS OF THE DIAGNOSTIC BY THE APT MONITOR HAVE
 NOT BEEN NOTED.

3.0 ERROR INFORMATION

ERRORS WILL CAUSE THE FOLLOWING ERROR MESSAGE TO BE PRINTED:

ERROR DURING MMU TESTING
 ERROR # = (UNIQUE ERROR NUMBER)
 ERROR PC = (PC AT TIME OF ERROR)

THE ERROR WILL THEN BE REPORTED TO APT AND THE PROGRAM
 WILL HALT.

4.0 PROGRESS REPORT

AT THE END OF EACH PASS THE DIAGNOSTIC NAME AND PASS COUNT ARE PRINTED.

185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205

.TITLE PROGRAM HEADER AND TABLES
.SBTTL PROGRAM HEADER

.MCALL NEWTST,ERRDEF,.EQUAT,.KT11,.\$40CAT,.\$EOP,.\$APTBL,SETUP
.MCALL . \$TYPE,.\$TYPDEC,ERRDF,BGNTST,ENDTST,BGNMOD,ENDMOD,CKLOOP
.MCALL .HEADER,.SETUP,.\$TRAP,BGNSUB,ENDSUB,.\$ACT11,.\$APTHDR
.MCALL . \$ATYPE,.\$ERROR,.\$TYPOCT,.\$READ

.TITLE KDJ11-A MEMORY MANAGEMENT DIAGNOSTIC
;*COPYRIGHT (C) MARCH,1984
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*

;
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
;*

000001
160000

\$TN=1
\$SWR=160000 ;:HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT


```
206 .TITLE GLOBAL AREAS
207 .SBTTL GLOBAL EQUATES SECTION
208
209 ;**
210 ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
211 ; ARE USED IN MORE THAN ONE TEST.
212 ;**
213 .SBTTL BASIC DEFINITIONS
214
215 ;*INITIAL ADDRESS OF THE STACK POINTER *** 1000 ***
216 STACK= 1000
217
218 .EQUIV EMT.ERROR      ;;BASIC DEFINITION OF ERROR CALL
219 .EQUIV IOT.SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
220
221 ;*MISCELLANEOUS DEFINITIONS
222 HT= 11                ;;CODE FOR HORIZONTAL TAB
223 LF= 12                ;;CODE FOR LINE FEED
224 CR= 15                ;;CODE FOR CARRIAGE RETURN
225 CRLF= 200            ;;CODE FOR CARRIAGE RETURN-LINE FEED
226 PS= 177776           ;;PROCESSOR STATUS WORD
227 .EQUIV PS.PSW
228 STKLMT= 177774        ;;STACK LIMIT REGISTER
229 PIRQ= 177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
230 DSWR= 177570          ;;HARDWARE SWITCH REGISTER
231 DDISP= 177570         ;;HARDWARE DISPLAY REGISTER
232
233 ;*GENERAL PURPOSE REGISTER DEFINITIONS
234 R0= #0                ;;GENERAL REGISTER
235 R1= #1                ;;GENERAL REGISTER
236 R2= #2                ;;GENERAL REGISTER
237 R3= #3                ;;GENERAL REGISTER
238 R4= #4                ;;GENERAL REGISTER
239 R5= #5                ;;GENERAL REGISTER
240 R6= #6                ;;GENERAL REGISTER
241 R7= #7                ;;GENERAL REGISTER
242 SP= #6                ;;STACK POINTER
243 PC= #7                ;;PROGRAM COUNTER
244
245 ;*PRIORITY LEVEL DEFINITIONS
246 PR0= 0                ;;PRIORITY LEVEL 0
247 PR1= 40               ;;PRIORITY LEVEL 1
248 PR2= 100              ;;PRIORITY LEVEL 2
249 PR3= 140              ;;PRIORITY LEVEL 3
250 PR4= 200              ;;PRIORITY LEVEL 4
251 PR5= 240              ;;PRIORITY LEVEL 5
252 PR6= 300              ;;PRIORITY LEVEL 6
253 PR7= 340              ;;PRIORITY LEVEL 7
254
255 ;*"SWITCH REGISTER" SWITCH DEFINITIONS
256 SW15= 100000
257 SW14= 40000
258 SW13= 20000
259 SW12= 10000
260 SW11= 4000
261 SW10= 2000
    SW09= 1000
```

```
262      000400      SW08= 400
263      000200      SW07= 200
264      000100      SW06= 100
265      000040      SW05= 40
266      000020      SW04= 20
267      000010      SW03= 10
268      000004      SW02= 4
269      000002      SW01= 2
270      000001      SW00= 1
271      .EQUIV      SW09,SW9
272      .EQUIV      SW08,SW8
273      .EQUIV      SW07,SW7
274      .EQUIV      SW06,SW6
275      .EQUIV      SW05,SW5
276      .EQUIV      SW04,SW4
277      .EQUIV      SW03,SW3
278      .EQUIV      SW02,SW2
279      .EQUIV      SW01,SW1
280      .EQUIV      SW00,SW0
281
282      ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
283      100000      BIT15= 100000
284      040000      BIT14= 40000
285      020000      BIT13= 20000
286      010000      BIT12= 10000
287      004000      BIT11= 4000
288      002000      BIT10= 2000
289      001000      BIT09= 1000
290      000400      BIT08= 400
291      000200      BIT07= 200
292      000100      BIT06= 100
293      000040      BIT05= 40
294      000020      BIT04= 20
295      000010      BIT03= 10
296      000004      BIT02= 4
297      000002      BIT01= 2
298      000001      BIT00= 1
299      .EQUIV      BIT09,BIT9
300      .EQUIV      BIT08,BIT8
301      .EQUIV      BIT07,BIT7
302      .EQUIV      BIT06,BIT6
303      .EQUIV      BIT05,BIT5
304      .EQUIV      BIT04,BIT4
305      .EQUIV      BIT03,BIT3
306      .EQUIV      BIT02,BIT2
307      .EQUIV      BIT01,BIT1
308      .EQUIV      BIT00,BIT0
309
310      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
311      000004      ERRVEC= 4      ;; TIME OUT AND OTHER ERRORS
312      000010      RESVEC= 10     ;; RESERVED AND ILLEGAL INSTRUCTIONS
313      000014      TBITVEC= 14    ;; "T" BIT
314      000014      TRTVEC= 14     ;; TRACE TRAP
315      000014      BPTVEC= 14     ;; BREAKPOINT TRAP (BPT)
316      000020      IOTVEC= 20     ;; INPUT/OUTPUT TRAP (IOT) **SCOPE**
317      000024      PWRVEC= 24     ;; POWER FAIL
```



```
318      000030      EMTVEC= 30      ;;EMULATOR TRAP (EMT) **ERROR**
319      000034      TRAPVEC=34      ;; "TRAP" TRAP
320      000060      TKVEC= 60      ;;TTY KEYBOARD VECTOR
321      000064      TPVEC= 64      ;;TTY PRINTER VECTOR
322      000240      PIRQVEC=240      ;;PROGRAM INTERRUPT REQUEST VECTOR
323      .SBTTL      MEMORY MANAGEMENT DEFINITIONS
324
325      ;*KT11 VECTOR ADDRESS
326
327      000250      MMVEC= 250
328
329      ;*KT11 STATUS REGISTER ADDRESSES
330
331      177572      SR0= 177572
332      177574      SR1= 177574
333      177576      SR2= 177576
334      172516      SR3= 172516
335
336      ;*USER "I" PAGE DESCRIPTOR REGISTERS
337
338      177600      UIPDR0= 177600
339      177602      UIPDR1= 177602
340      177604      UIPDR2= 177604
341      177606      UIPDR3= 177606
342      177610      UIPDR4= 177610
343      177612      UIPDR5= 177612
344      177614      UIPDR6= 177614
345      177616      UIPDR7= 177616
346
347      ;*USER "D" PAGE DESCRIPTOR REGISTORS
348
349      177620      UDPDR0= 177620
350      177622      UDPDR1= 177622
351      177624      UDPDR2= 177624
352      177626      UDPDR3= 177626
353      177630      UDPDR4= 177630
354      177632      UDPDR5= 177632
355      177634      UDPDR6= 177634
356      177636      UDPDR7= 177636
357
358      ;*USER "I" PAGE ADDRESS REGISTERS
359
360      177640      UIPAR0= 177640
361      177642      UIPAR1= 177642
362      177644      UIPAR2= 177644
363      177646      UIPAR3= 177646
364      177650      UIPAR4= 177650
365      177652      UIPAR5= 177652
366      177654      UIPAR6= 177654
367      177656      UIPAR7= 177656
368
369      ;*USER "D" PAGE ADDRESS REGISTERS
370
371      177660      UDPAR0= 177660
372      177662      UDPAR1= 177662
373      177664      UDPAR2= 177664
```

374	177666	UDPAR3= 177666
375	177670	UDPAR4= 177670
376	177672	UDPAR5= 177672
377	177674	UDPAR6= 177674
378	177676	UDPAR7= 177676
379		
380		:*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS
381		
382	172200	SIPDR0= 172200
383	172202	SIPDR1= 172202
384	172204	SIPDR2= 172204
385	172206	SIPDR3= 172206
386	172210	SIPDR4= 172210
387	172212	SIPDR5= 172212
388	172214	SIPDR6= 172214
389	172216	SIPDR7= 172216
390		
391		:*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS
392		
393	172220	SDPDR0= 172220
394	172222	SDPDR1= 172222
395	172224	SDPDR2= 172224
396	172226	SDPDR3= 172226
397	172230	SDPDR4= 172230
398	172232	SDPDR5= 172232
399	172234	SDPDR6= 172234
400	172236	SDPDR7= 172236
401		
402		:*SUPERVISOR "I" PAGE ADDRESS REGISTERS
403		
404	172240	SIPAR0= 172240
405	172242	SIPAR1= 172242
406	172244	SIPAR2= 172244
407	172246	SIPAR3= 172246
408	172250	SIPAR4= 172250
409	172252	SIPAR5= 172252
410	172254	SIPAR6= 172254
411	172256	SIPAR7= 172256
412		
413		:*SUPERVISOR "D" PAGE ADDRESS REGISTERS
414		
415	172260	SDPAR0= 172260
416	172262	SDPAR1= 172262
417	172264	SDPAR2= 172264
418	172266	SDPAR3= 172266
419	172270	SDPAR4= 172270
420	172272	SDPAR5= 172272
421	172274	SDPAR6= 172274
422	172276	SDPAR7= 172276
423		
424		:*KERNEL "I" PAGE DESCRIPTOR REGISTERS
425		
426	172300	KIPDR0= 172300
427	172302	KIPDR1= 172302
428	172304	KIPDR2= 172304
429	172306	KIPDR3= 172306

430	172310	KIPDR4=	172310	
431	172312	KIPDR5=	172312	
432	172314	KIPDR6=	172314	
433	172316	KIPDR7=	172316	
434				
435		;*KERNEL "D" PAGE DESCRIPTOR REGISTERS		
436				
437	172320	KDPDR0=	172320	
438	172322	KDPDR1=	172322	
439	172324	KDPDR2=	172324	
440	172326	KDPDR3=	172326	
441	172330	KDPDR4=	172330	
442	172332	KDPDR5=	172332	
443	172334	KDPDR6=	172334	
444	172336	KDPDR7=	172336	
445				
446		;*KERNEL "I" PAGE ADDRESS REGISTERS		
447				
448	172340	KIPAR0=	172340	
449	172342	KIPAR1=	172342	
450	172344	KIPAR2=	172344	
451	172346	KIPAR3=	172346	
452	172350	KIPAR4=	172350	
453	172352	KIPAR5=	172352	
454	172354	KIPAR6=	172354	
455	172356	KIPAR7=	172356	
456				
457		;*KERNEL "D" PAGE ADDRESS REGISTERS		
458				
459	172360	KDPAR0=	172360	
460	172362	KDPAR1=	172362	
461	172364	KDPAR2=	172364	
462	172366	KDPAR3=	172366	
463	172370	KDPAR4=	172370	
464	172372	KDPAR5=	172372	
465	172374	KDPAR6=	172374	
466	172376	KDPAR7=	172376	
467				
468		;THESE ARE FLOATING POINT ACCUMULATOR EQUATES		
469	000000	AC0=	%0	
470	000001	AC1=	%1	
471	000002	AC2=	%2	
472	000003	AC3=	%3	
473	000004	AC4=	%4	
474	000005	AC5=	%5	
475	000006	AC6=	%6	
476	000007	AC7=	%7	
477				
478	000244	FPVEC=	244	
479				
480		;THESE ARE CACHE REGISTER EQUATES		
481	177746	CCR=	177746	;CACHE CONTROL REGISTER
482	177744	MSER=	177744	;MEMORY SYSTEM ERROR REGISTER
483	177752	HITMIS=	177752	;HIT/MISS REGISTER
484	177766	CPEREG=	177766	;CPU ERROR REGISTER
485				

```
486                                     ;MISCELLANEOUS DEFINITIONS
487                                     BEVENT= 177546                ;BEVENT CONTROL REGISTER
488                                     RCSR= 177560
489                                     RBUF= 177562
490                                     XCSR= 177564
491                                     XBUF= 177566
492                                     ERRTN= HALT
493                                     $TSTNU=1
494                                     ERRNUM= 1                    ;INITIALIZE ERROR NUMBER COUNTER
495                                     ASWREG= 2000                  ;SWR FOR APT--NO BEVENT TESTING
496
497
498                                     ;THIS EQUATE DEFINES THE BOTTOM OF THE PROGRAM STACK POINTER
499                                     STBOT= 1000
500                                     .ASECT
501                                     .SBTTL TRAP CATCHER
502
503                                     .=0
504                                     ;*ALL UNUSED LOCATIONS OF THE VECTOR AREA CONTAIN
505                                     ;*A ".+2, IOT" SEQUENCE TO CATCH AND PROCESS ILLEGAL
506                                     ;*TRAPS AND INTERRUPTS THAT MIGHT OCCUR.
507                                     ;*THE IOT TRAP WHICH IS TAKEN ON THE ILLEGAL TRAP/INT
508                                     ;*TRAPS TO THE $SCOPE ROUTINE WHICH (IF THE RETURN PC IS
509                                     ;*LESS THAN 1002) JUMPS TO THE $ERROR ROUTINE.
510                                     ;*THE $ERROR ROUTINE WILL REPORT THE ERROR AS FOLLOWS:
511                                     ;* PC=YYYYYY UNEXPECTED TRAP TO XXX
512                                     ;*AND RETURN TO THE PROGRAM AT PC=YYYYYY+2
513                                     ;*WHERE XXX=LOCATION OF ILLEGAL TRAP
514                                     ;* YYYYYY=PC AT TIME OF TRAP
515                                     ;*NOTE: IF THE PROCESSOR IS NOT AN 11/05 THE PROGRAM
516                                     ;* CAN BE STARTED AT ADDRESS 0 AS WELL AS ADDRESS 200.
517
518 000000 000000 $40CAT: HALT                                     ;;HALT
519 000002 000737 BR -.100                                         ;;BRANCH TO 177700 & TIME OUT (NOT ON
520                                     ;;11/05)
521 000004 001604 .WORD START                                       ;;VECTOR TO STARTING ADDRESS
522 000006 000340 .WORD 340                                         ;;WITH PRIORITY LEVEL 7
523                                     .=174
524 000174 000000 DISPREG: .WORD 0                                   ;;SOFTWARE DISPLAY REGISTER
525 000176 000000 SWREG: .WORD 0                                    ;;SOFTWARE SWITCH REGISTER
526                                     .SBTTL STARTING ADDRES(ES)
527 000200 000137 001604 JMP @START ;;GO TO START OF PROGRAM
528                                     .SBTTL ACT11 HOOKS
529
530                                     ;*****
531                                     ;HOOKS REQUIRED BY ACT11
532                                     $SVPC=.                        ;SAVE PC
533                                     .=46
534 000046 020026 .ENDAD                                           ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
535                                     .=52
536 000052 000000 .WORD 0                                           ;;2)SET LOC.52 TO ZERO
537                                     .=$SVPC                       ;;RESTORE PC
538                                     .SBTTL APT PARAMETER BLOCK
539
540                                     ;*****
541                                     ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
```



```

542
543      000204      ;*****
544      000024      . $X=      ;;SAVE CURRENT LOCATION
545      000024      . =24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
546      000044      200      ;;FOR APT START UP
547      000044      . =44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
548      000204      $APTHDR    ;;POINT TO APT HEADER BLOCK
549      000204      . = $X      ;;RESET LOCATION COUNTER
550      ;*****
551      ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
552      ;INTERFACE SPEC.
553      000204      $APTHD:
554      000204      000000      $HIBTS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
555      000206      001000      $MBADR: .WORD $MAIL    ;;ADDRESS OF APT MAILBOX (BITS 0-15)
556      000210      000001      $TSTM: .WORD 1      ;;RUN TIM OF LONGEST TEST
557      000212      000002      $PASTM: .WORD 2      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
558      000214      000000      $UNITM: .WORD 0      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
559      000216      000014      .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
560      000204      . = $X      ;;SAVE CURRENT LOCATION COUNT
561      000002      . =2
562      000002      000000      0
563      000004      000006      6
564      000006      000004      4
565      000204      . = $X      ;SET UP SOME VECTORS
566      001000      . =1000    ;RESTORE LOCATION COUNT
  
```

```
567 .SBTTL GLOBAL DATA SECTION
568
569
570 ;**
571 ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
572 ; IN MORE THAN ONE TEST.
573 ;**
574 .SBTTL APT MAILBOX-ETABLE
575
576 ;*****
577 .EVEN
578 001000 $MAIL: ;:APT MAILBOX
579 001002 000000 $MSGTY: .WORD AMSTGY ;:MESSAGE TYPE CODE
580 001004 000000 $FATAL: .WORD AFATAL ;:FATAL ERROR NUMBER
581 001006 000000 $TESTN: .WORD ATESTN ;:TEST NUMBER
582 001010 000000 $PASS: .WORD APASS ;:PASS COUNT
583 001012 000000 $DEVCT: .WORD ADEVCT ;:DEVICE COUNT
584 001014 000000 $UNIT: .WORD AUNIT ;:I/O UNIT NUMBER
585 001016 000000 $MSGAD: .WORD AMSGAD ;:MESSAGE ADDRESS
586 001020 000000 $MSGLG: .WORD AMSGLG ;:MESSAGE LENGTH
587 001020 000 $ETABLE: ;:APT ENVIRONMENT TABLE
588 001021 000 $ENV: .BYTE AENV ;:ENVIRONMENT BYTE
589 001022 002000 $ENVM: .BYTE AENVM ;:ENVIRONMENT MODE BITS
590 001024 000000 $SWREG: .WORD ASWREG ;:APT SWITCH REGISTER
591 001026 000000 $USWR: .WORD AUSWR ;:USER SWITCHES
592 $CPUOP: .WORD ACPUOP ;:CPU TYPE,OPTIONS
593 ;* BITS 15-11-CPU TYPE
594 ;* 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
595 ;* 11/70=06,PDQ=07,Q=10
596 ;* BIT 10-REAL TIME CLOCK
597 ;* BIT 9-FLOATING POINT PROCESSOR
598 ;* BIT 8-MEMORY MANAGEMENT
598 001030 $ETEND:
599 .MEXIT
600
601 ;THESE LOCATIONS ARE USED IN MORE THAN ONE TEST TO STORE VECTOR DATA
602 ;WHEN THE TEST NEEDS TO HAVE AN ERROR CONDITION RESPOND DIFFERENTLY
603 ;FROM THE DEFAULT RESPONSE.
604 001030 000000 SLOC00: .WORD 0
605 001032 000000 SLOC01: .WORD 0
606
607 ;THESE LOCATIONS ARE USED IN MORE THAN ONE TEST TO STORE WORKING DATA.
608 001034 000000 EXPDAT: .WORD 0 ;:STORES EXPECTED (GOOD) DATA FOR COMPARISONS
609 001036 000000 RECDAT: .WORD 0 ;:STORES RECIEVED DATA TO BE VERIFIED
610 001040 000000 COUNT: .WORD 0 ;:ERROR INDICATOR FOR FLOATING POINT TESTS
611 001042 000000 FLAG: .WORD 0 ;:USED TO STORE "FLAG" CONDITIONS
612 001044 000000 ERRCNT: .WORD 0 ;:STORAGE FOR ERROR COUNT
613 001046 177570 SWR: .WORD DSWR ;:STORAGE FOR SWITCH REGISTER ADDRESS
614 001050 177570 DISPLAY: .WORD DDISP ;:STORAGE FOR DISPLAY REGISTER ADDRESS
615 001052 000000 $ERFLG: .WORD 0 ;:ERROR FLAG
616
617 ;THESE LOCATIONS ARE USED BY MORE THAN ONE TEST AS LOOP COUNTERS
617 001054 000000 DCOUNT: .WORD 0
618 001056 000000 ALLCTR: .WORD 0
619 001060 000000 LOOPIN: .WORD 0
620 001062 000000 SAVSP1: .WORD 0 ;:STORAGE FOR UNEXPECTED TRAP DATA
621 001064 000000 SAVSP2: .WORD 0 ;
622 ;
```


623					
624	001066	000000	SAVSUP: .WORD	0	:USED TO STORE SUPERVISOR STACK VALUE
625	001070	000000	SAVUSE: .WORD	0	:USED TO STORE USER STACK VALUE
626	001072	000000	SAVMRO: .WORD	0	:USED TO STORE MMU STATUS REGISTER 0 DATA
627	001074	000000	SAVMR1: .WORD	0	:USED TO STORE MMU STATUS REGISTER 1 DATA
628	001076	000000	SAVMR2: .WORD	0	:USED TO STORE MMU STATUS REGISTER 2 DATA
629	001100	000004	FLOAT: .BLKW	4	:USED TO STORE VALUES FOR MMU TESTS
630	001110	000004	FLO: .BLKW	4	:USED TO STORE VALUES FOR MMU TESTS
631					
632					
633					
634					
635					
636					
637					
638					
639	001120				
640	001120	000002			

!!!!!!THIS IS IT. THE PROGRAM TEST LOCATION AND WRITE BUFFER!!!!!!!!!!!!!!!!!!!!!!
 TSTLOC: .BLKW 2

641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675

.SBTTL GLOBAL TEXT SECTION

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

OPMSG2: .ASCIZ <CR><LF>/SET BIT 8 = 1 FOR 18 BIT SYSTEM/

ERRMSG: .ASCIZ <CR><LF>/CACHE SYSTEM ERROR/

MMUERR: .ASCIZ <CR><LF>/ERROR DURING MMU TESTING/

ERR1: .ASCIZ <CR><LF>/ERROR # =/

ERR2: .ASCIZ <CR><LF>/ERROR PC =/

\$CRLF: .ASCIZ <CR><LF>/ /

.EVEN

GLOBAL AREAS
KDJ11A.MAC

MACY11 30A(1052)
20-MAR-84 11:19

20-MAR-84 11:31 PAGE 17

GLOBAL ERROR REPORT SECTION

E2

SEQ 0017

676
677
678
679
680
681
682

.SBTTL GLOBAL ERROR REPORT SECTION

: THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
: USED BY MORE THAN TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.
:--

683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738001306 010046
001310 010146
001312 010246
001314 012700 177600
001320 004767 000062
001324 004767 000100
001330 004767 000074
001334 012700 172200
001340 004767 000042
001344 004767 000060
001350 004767 000054
001354 004767 000026
001360 004767 000044
001364 004767 000040
001370 012737 000027 172516
001376 012602
001400 012601
001402 012600
001404 000207

001406 005002
001410 012720 077406
001414 062702 000001
001420 022702 000020
001424 001371
001426 000207

001430 005001
001432 010120
001434 062701 000200
001440 022701 001600
001444 001372
001446 012720 177600
001452 000207

001454 005205
001456 000002

.SBTTL GLOBAL SUBROUTINES SECTION

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

;MMU GLOBAL SUBROUTINES

;ROUTINE TO INITIALIZE MEMORY MANAGEMENT

MMU: MOV R0,-(SP) ;SAVE CONTENTS OF REGISTERS
MOV R1,-(SP) ;
MOV R2,-(SP) ;
MOV #177600,R0 ;
JSR PC,PDR ;INIT I AND D USER PDR'S
JSR PC,PAR ;INIT I USER PAR'S
JSR PC,PAR ;INIT D USER PAR'S
MOV #172200,R0 ;
JSR PC,PDR ;INIT I AND D SUP PDR'S
JSR PC,PAR ;INIT I SUP PAR'S
JSR PC,PAR ;INIT D SUP PAR'S
JSR PC,PDR ;INIT I AND D KER PDR'S
JSR PC,PAR ;INIT I KER PAR'S
JSR PC,PAR ;INIT D KER PAR'S
MOV #27,0#172516 ;INIT MMR3
MOV (SP)+,R2 ;RESTORE REGISTERS
MOV (SP)+,R1 ;
MOV (SP)+,R0 ;
RTS PC ;RETURN

;ROUTINE TO INITIALIZE PDR'S

PDR: CLR R2 ;INIT CNTR
PDR1: MOV #77406,(R0)+ ;INIT PDR
ADD #1,R2 ;INCREMENT CNTR
CMP #16,,R2 ;ARE WE DONE?
BNE PDR1 ;BRANCH IF NOT
RTS PC ;RETURN

;ROUTINE TO INITIALIZE PAR'S

PAR: CLR R1 ;SETUP TO INIT PAR
PAR1: MOV R1,(R0)+ ;INIT PAR
ADD #200,R1 ;GET READY FOR NEXT PAR
CMP #1600,R1 ;REACHED A PAR?
BNE PAR1 ;BRANCH IF NOT
MOV #177600,(R0)+ ;INIT PAR?
RTS PC ;RETURN

;TIME OUT ROUTINE

ADDTRP: INC R5 ;INCREMENT TIME OUT FLAG
RTI ;RETURN


```

739      ;MMU TRAP ROUTINE
740      ;
741      001460 026727 177356 000001 MMUTRP: CMP     FLAG,#1
742      001466 001403          BEQ     1$
743      001470 104000          ERROR
744      001472 000001          .WORD   1
745      001474 001213          .WORD   MMUERR
746      001476 010046          1$:     MOV     R0,-(SP)
747      001500 013700 177776      MOV     @#177776,R0
748      001504 072027 177764      ASH     #-14,R0
749      001510 020027 000002      CMP     R0,#2
750      001514 001001          BNE     OK
751      001516 000411          BR      NOTOK
752      001520 013700 177776      OK:     MOV     @#177776,R0
753      001524 072027 000002      ASH     #2,R0
754      001530 072027 177764      ASH     #-14,R0
755      001534 020027 000002      CMP     R0,#2
756      001540 001002          BNE     OK1
757      001542 005066 000004      NOTOK:  CLR     4(SP)
758      001546 013767 177572 177316  OK1:   MOV     @#177572,SAVMR0
759      001554 013767 177574 177312  MOV     @#177574,SAVMR1
760      001562 013767 177576 177306  MOV     @#177576,SAVMR2
761      001570 005037 177572      CLR     @#177572
762      001574 005067 177242      CLR     FLAG
763      001600 012600          MOV     (SP)+,R0
764      001602 000002          RTI

```

;ARE WE EXPECTING AN ABORT
 ;YES GO ON
 ;ALL ERRORS TO TRAP TO EMT VECTOR
 ;UNIQUE ERROR NUMBER
 ;ADDRESS OF ERROR MESSAGE
 ;SAVE CONTENTS OF REG 0
 ;SAVE A COPY OF PSW
 ;LOOK AT BITS<15:14>
 ;WAS PS<15:14>=10
 ;NO GO ON
 ;YES CHANGE BITS TO 00
 ;SAVE A COPY OF PSW
 ;LOOK AT BITS<13:12>
 ;
 ;WAS PS<13:12>=10
 ;NO GO ON
 ;CLEAR ILLEGAL MODE FROM OLD PSW
 ;SAVE A COPY OF MMRO
 ;SAVE A COPY OF MMR1
 ;SAVE A COPY OF MMR2
 ;CLEAR ABORT BITS AND TURN MMU OFF
 ;CLEAR MMU ABORT FLAG
 ;RESTORE ORIGINAL CONTENTS OF REG 0
 ;RETURN

```
765 001604          START:
766 001604 012737 000014 177746      MOV    #14, @CCR          ;SET CACHE TO FORCE MISS
767                                     .SBTTL  INITIALIZE THE COMMON TAGS
768 001612 012706 001000      MOV    @STACK, SP      ;;SETUP THE STACK POINTER
769                                     ;;INITIALIZE A FEW VECTORS
770 001616 012737 022304 000030      MOV    @ERROR, @EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
771 001624 012737 000340 000032      MOV    #340, @EMTVEC+2 ;;LEVEL 7
772 001632 012737 021756 000034      MOV    @TRAP, @TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
773 001640 012737 000340 000036      MOV    #340, @TRAPVEC+2;LEVEL 7
774 001646 005067 177134          CLR    $PASS          ;;CLEAR THE PASS COUNT
775 001652 016767 016116 016106      MOV    $ENDCT, $EOPCT ;;SETUP END-OF-PROGRAM COUNTER
776 001660 105067 177166          CLR    $ERFLG        ;;CLEAR THE ERROR FLAG
777                                     ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
778                                     ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
779 001664 013746 000004          MOV    @ERRVEC, -(SP) ;;SAVE ERROR VECTOR
780 001670 012737 001724 000004      MOV    #64, @ERRVEC ;;SET UP ERROR VECTOR
781 001676 012767 177570 177142      MOV    @DSWR, SWR    ;;SETUP FOR A HARDWARE SWICH REGISTER
782 001704 012767 177570 177136      MOV    @DISP, DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
783 001712 022777 177777 177126      CMP    #-1, @SWR      ;;TRY TO REFERENCE HARDWARE SWR
784 001720 001012          BNE    66$                ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
785                                     ;;AND THE HARDWARE SWR IS NOT = -1
786 001722 000403          BR      65$                ;;BRANCH IF NO TIMEOUT
787 001724 012716 001732          64$: MOV    #65, (SP)    ;;SET UP FOR TRAP RETURN
788 001730 000002          RTI
789 001732 012767 000176 177106      65$: MOV    @SWREG, SWR    ;;POINT TO SOFTWARE SWR
790 001740 012767 000174 177102      MOV    @DISPREG, DISPLAY
791 001746 012637 000004          66$: MOV    (SP)+, @ERRVEC ;;RESTORE ERROR VECTOR
792
793 .MACRO    $$SETMAIL    ?$ARG1
794          CLR    $PASS          ;;CLEAR PASS COUNT
795          BITB    @APTSIZE, $ENVM ;;TEST USER SIZE UNDER APT
796          BEQ     $ARG1          ;;YES, USE NON-APT SWITCH
797          MOV     @SWREG, SWR    ;;NO, USE APT SWITCH REGISTER
798
799          $ARG1:
800 .ENDM
801 001752 005067 177030          $$SETMAIL
802 001756 132767 000200 177035      CLR    $PASS          ;;CLEAR PASS COUNT
803 001764 001403          BITB    @APTSIZE, $ENVM    ;;TEST USER SIZE UNDER APT
804 001766 012767 001022 177052      BEQ     67$          ;;YES, USE NON-APT SWITCH
805 001774          MOV     @SWREG, SWR    ;;NO, USE APT SWITCH REGISTER
806 001774 012737 022304 000020      67$: MOV    @ERROR, @IOTVEC    ;;SET UP IOT VECTORS
807 002002 012737 000340 000022      MOV    #340, @IOTVEC+2    ;;TO GO TO ERROR ROUTINE
808 002010 005037 177766          CLR    @177766        ;;CLEAR CPU ERROR REGISTER
809 002014 012767 001460 176226      MOV    @MMUTRP, MMVEC
810 002022 104401 001124          TYPE    ,OPMSG2        ;OPERATOR MESSAGE 2
811 002026 005737 000042          .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
812 002032 001012          TST     @42                ;;ARE WE RUNNING UNDER XXDP/ACT?
813 002034 126727 176760 000001      BNE    68$          ;;BRANCH IF YES
814 002042 001406          CMPB    $ENV, #1            ;;ARE WE RUNNING UNDER APT?
815 002044 026727 176776 000176      BEQ     68$          ;;BRANCH IF YES
816 002052 001005          CMP     SWR, @SWREG          ;;SOFTWARE SWITCH REG SELECTED?
817 002054 104406          BNE    69$          ;;BRANCH IF NO
818 002056 000403          GTSWR                    ;;GET SOFT-SWR SETTINGS
819 002060 112767 000001 017666      68$: BR      69$
820 002066          69$: MOVB    #1, $AUTOB        ;;SET AUTO-MODE INDICATOR
```


821 002066 005067 176712
822 002072 012737 000014 177746
823
824
825
826
827
828
829
830 002100
831
832
833
834 002100
835 002100 005267 176700
836 002104 005067 175656
837 002110 005037 177572
838 002114 005037 001042
839 002120 013746 000004
840 002124 012737 001454 000004
841 002132 005005
842 002134 013701 177572
843 002140 013701 177574
844 002144 013701 177576
845 002150 013701 172516
846 002154 012637 000004
847 002160 020527 000000
848 002164 001403
849 002166 104000
850 002170 000002
851 002172 001213
852
853 002174
854
855 002174
856
857
858
859 002174
860 002174 005267 176604
861 002200 005067 175562
862 002204 005037 177572
863 002210 005037 001042
864 002214 013746 000244
865 002220 013746 000246
866 002224 013746 000004
867 002230 012737 000246 000244
868 002236 012737 000002 000246
869 002244 012737 001454 000004
870 002252 005005
871 002254 012700 172200
872 002260 005020
873 002262 020027 172400
874 002266 001374
875 002270 012700 177600
876 002274 005020

RESTART: CLR \$TESTN
MOV #14,\$CCR

;RESET \$TESTN TO ZERO
;SET CACHE TO FORCE MISS

.SBTTL MEMORY MANAGEMENT TESTS

BEGIN MMU TESTING

TSMMU1:

; *TEST 1 STATUS REGISTER TEST

TST1:

INC \$TESTN ;INCREMENT TEST NUMBER
CLR CPEREG ;CLEAR CPU ERROR REGISTER
CLR #177572 ;TURN MMU OFF
CLR #FLAG ;CLEAR MMU TRAP FLAG
MOV #4,-(SP) ;SAVE OLD VECTOR
MOV #ADDRP,#4 ;SETUP NEW VECTOR
CLR R5 ;CLEAR FLAG
MOV #177572,R1 ;TEST MMR0
MOV #177574,R1 ;TEST MMR1
MOV #177576,R1 ;TEST MMR2
MOV #172516,R1 ;TEST MMR3
MOV (SP),#4 ;RESTORE VECTOR
CMP R5,#0 ;DID WE TRAP
BEQ 1\$;NO, THEN BRANCH
ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
.WORD 2 ;UNIQUE ERROR NUMBER
.WORD MMUERR ;ADDRESS OF ERROR MESSAGE

;YES, GO TO ERROR

1\$:

TSMMU2:

; *TEST 2 ADDRESS TEST OF PARS,PDRS, AND FP REGS

TST2:

INC \$TESTN ;INCREMENT TEST NUMBER
CLR CPEREG ;CLEAR CPU ERROR REGISTER
CLR #177572 ;MMU OFF
CLR #FLAG ;CLEAR MMU TRAP FLAG
MOV #244,-(SP) ;SAVE FP VECTOR
MOV #246,-(SP)
MOV #4,-(SP) ;SAVE TIME OUT VECTOR
MOV #246,#244 ;SETUP NEW FP VECTOR
MOV #2,#246
MOV #ADDRP,#4 ;SETUP NEW TIME OUT VECTOR
CLR R5 ;CLEAR TIMEOUT FLAG
MOV #172200,R0 ;LOAD ALL PARS AND PDRS WITH ZERO
CLR (R0)
CMP R0,#172400
BNE 1\$
MOV #177600,R0
CLR (R0)

1\$:

2\$:

877 002276 020027 177700 CMP R0,#177700 ;
878 002302 001374 BNE 2\$;
879 002304 170127 000200 LDFPS #200 ;
880 002310 012700 001100 MOV #FLOAT,R0 ;LOAD ACO-AC5 WITH 0
881 002314 005020 CLR (R0)+ ;
882 002316 005020 CLR (R0)+ ;
883 002320 005020 CLR (R0)+ ;
884 002322 005020 CLR (R0)+ ;
885 002324 012700 001100 MOV #FLOAT,R0 ;
886 002330 172410 LDD (R0),AC0 ;
887 002332 172510 LDD (R0),AC1 ;
888 002334 172610 LDD (R0),AC2 ;
889 002336 172710 LDD (R0),AC3 ;
890 002340 174004 STD ACO,AC4 ;
891 002342 174005 STD ACO,AC5 ;
892 002344 174500 3\$: DIVD ACO,AC1 ;LOAD FEC WITH 4 AND FEA WITH #3\$
893 002346 170337 001110 STST #FLO ;CHECK FEC FOR 4 AND FEA FOR #3\$
894 002352 012704 001110 MOV #FLO,R4 ;
895 002356 022427 000004 CMP (R4)+,#4 ;
896 002362 001403 BEQ 21\$;
897 002364 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
898 002366 000003 .WORD 3 ;UNIQUE ERROR NUMBER
899 002370 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
900 ;
901 002372 021427 002344 21\$: CMP (R4),#3\$;
902 002376 001403 BEQ 22\$;
903 002400 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
904 002402 000004 .WORD 4 ;UNIQUE ERROR NUMBER
905 002404 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
906 ;
907 002406 012704 172200 22\$: MOV #172200,R4 ;
908 002412 012701 000001 MOV #1,R1 ;CHECK EACH PAR, PDR FOR 0 THEN
909 002416 010102 4\$: MOV R1,R2 ;WRITE A UNIQUE NUMBER TO IT
910 002420 072227 000010 ASH #10,R2 ;
911 002424 021427 000000 CMP (R4),#0 ;
912 002430 001403 BEQ 5\$;
913 002432 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
914 002434 000005 .WORD 5 ;UNIQUE ERROR NUMBER
915 002436 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
916 ;
917 002440 010224 5\$: MOV R2,(R4)+ ;
918 002442 005201 INC R1 ;
919 002444 020427 172400 CMP R4,#172400 ;
920 002450 001362 BNE 4\$;
921 002452 012704 177600 MOV #177600,R4 ;
922 002456 010102 6\$: MOV R1,R2 ;
923 002460 072227 000010 ASH #10,R2 ;
924 002464 021427 000000 CMP (R4),#0 ;
925 002470 001403 BEQ 7\$;
926 002472 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
927 002474 000006 .WORD 6 ;UNIQUE ERROR NUMBER
928 002476 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
929 ;
930 002500 010224 7\$: MOV R2,(R4)+ ;
931 002502 005201 INC R1 ;
932 002504 020427 177700 CMP R4,#177700 ;

933 002510 001362 BNE 6\$;
934 002512 012704 001110 MOV #FLO,R4 ;CHECK AC5 FOR ALL ZEROES THEN LOAD A 6
935 002516 012703 001100 MOV #FLOAT,R3 ;
936 002522 174014 STD AC0,(R4) ;
937 002524 172405 LDD AC5,AC0 ;
938 002526 174013 STD AC0,(R3) ;
939 002530 012702 000004 MOV #4,R2 ;
940 002534 022327 000000 8\$: CMP (R3)+,#0 ;
941 002540 001403 BEQ 9\$;
942 002542 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
943 002544 000007 .WORD 7 ;UNIQUE ERROR NUMBER
944 002546 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
945 ;
946 002550 005302 9\$: DEC R2 ;
947 002552 001370 BNE 8\$;
948 002554 012703 001100 MOV #FLOAT,R3 ;
949 002560 012713 000006 MOV #6,(R3) ;
950 002564 172413 LDD (R3),AC0 ;
951 002566 174005 STD AC0,AC5 ;
952 002570 172404 LDD AC4,AC0 ;CHECK AC4 FOR ALL ZEROES THEN LOAD A 5
953 002572 174013 STD AC0,(R3) ;
954 002574 012702 000004 MOV #4,R2 ;
955 002600 022327 000000 10\$: CMP (R3)+,#0 ;
956 002604 001403 BEQ 11\$;
957 002606 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
958 002610 000010 .WORD 10 ;UNIQUE ERROR NUMBER
959 002612 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
960 ;
961 002614 005302 11\$: DEC R2 ;
962 002616 001370 BNE 10\$;
963 002620 012703 001100 MOV #FLOAT,R3 ;
964 002624 012713 000005 MOV #5,(R3) ;
965 002630 172413 LDD (R3),AC0 ;
966 002632 174004 STD AC0,AC4 ;
967 002634 012702 000004 MOV #4,R2 ;CHECK AC0 FOR ALL ZEROES THEN LOAD A 1
968 002640 022427 000000 12\$: CMP (R4)+,#0 ;
969 002644 001403 BEQ 13\$;
970 002646 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
971 002650 000011 .WORD 11 ;UNIQUE ERROR NUMBER
972 002652 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
973 ;
974 002654 005302 13\$: DEC R2 ;
975 002656 001370 BNE 12\$;
976 002660 012713 000001 MOV #1,(R3) ;
977 002664 172413 LDD (R3),AC0 ;
978 002666 012704 001110 MOV #FLO,R4 ;CHECK AC1 FOR ALL ZEROES THEN LOAD A 2
979 002672 012702 000004 MOV #4,R2 ;
980 002676 174114 STD AC1,(R4) ;
981 002700 022427 000000 14\$: CMP (R4)+,#0 ;
982 002704 001403 BEQ 15\$;
983 002706 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
984 002710 000012 .WORD 12 ;UNIQUE ERROR NUMBER
985 002712 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
986 ;
987 002714 005302 15\$: DEC R2 ;
988 002716 001370 BNE 14\$;

989	002720	012713	000002		MOV	#2,(R3)	:
990	002724	172513			LDD	(R3),AC1	:
991	002726	012704	001110		MOV	#FLO,R4	:
992	002732	012702	000004		MOV	#4,R2	;CHECK AC2 FOR ALL ZEROES THEN LOAD A 3
993	002736	174214			STD	AC2,(R4)	:
994	002740	022427	000000	16\$:	CMP	(R4)+,#0	:
995	002744	001403			BEQ	17\$	:
996	002746	104000			ERROR		:
997	002750	000013			.WORD	13	;ALL ERRORS TO TRAP TO EMT VECTOR
998	002752	001213			.WORD	MMUERR	;UNIQUE ERROR NUMBER
999							;ADDRESS OF ERROR MESSAGE
1000	002754	005302		17\$:	DEC	R2	:
1001	002756	001370			BNE	16\$	:
1002	002760	012713	000003		MOV	#3,(R3)	:
1003	002764	172613			LDD	(R3),AC2	:
1004	002766	012704	001110		MOV	#FLO,R4	:
1005	002772	012702	000004		MOV	#4,R2	;CHECK AC3 FOR ALL ZEROES THEN LOAD A 4
1006	002776	174314			STD	AC3,(R4)	:
1007	003000	022427	000000	18\$:	CMP	(R4)+,#0	:
1008	003004	001403			BEQ	19\$	:
1009	003006	104000			ERROR		:
1010	003010	000014			.WORD	14	;ALL ERRORS TO TRAP TO EMT VECTOR
1011	003012	001213			.WORD	MMUERR	;UNIQUE ERROR NUMBER
1012							;ADDRESS OF ERROR MESSAGE
1013	003014	005302		19\$:	DEC	R2	:
1014	003016	001370			BNE	18\$	:
1015	003020	012713	000004		MOV	#4,(R3)	:
1016	003024	172713			LDD	(R3),AC3	:
1017	003026	012704	001110		MOV	#FLO,R4	:
1018	003032	170214			STFPS	(R4)	;CHECK FPS FOR 100204 THEN LOAD IT WITH 200
1019	003034	022714	100204		CMP	#100204,(R4)	:
1020	003040	001403			BEQ	20\$	:
1021	003042	104000			ERROR		:
1022	003044	000015			.WORD	15	;ALL ERRORS TO TRAP TO EMT VECTOR
1023	003046	001213			.WORD	MMUERR	;UNIQUE ERROR NUMBER
1024							;ADDRESS OF ERROR MESSAGE
1025	003050	170127	000200	20\$:	LDFPS	#200	:
1026	003054	012704	172200		MOV	#172200,R4	:
1027	003060	012701	000001		MOV	#1,R1	;CHECK PDR, PAR FOR UNIQUE NUMBERS
1028	003064	010102		23\$:	MOV	R1,R2	:
1029	003066	072227	000010		ASH	#10,R2	:
1030	003072	022402			CMP	(R4)+,R2	:
1031	003074	001403			BEQ	24\$	:
1032	003076	104000			ERROR		:
1033	003100	000016			.WORD	16	;ALL ERRORS TO TRAP TO EMT VECTOR
1034	003102	001213			.WORD	MMUERR	;UNIQUE ERROR NUMBER
1035							;ADDRESS OF ERROR MESSAGE
1036	003104	005201		24\$:	INC	R1	:
1037	003106	020427	172400		CMP	R4,#172400	:
1038	003112	001364			BNE	23\$	:
1039	003114	012704	177600		MOV	#177600,R4	:
1040	003120	010102		25\$:	MOV	R1,R2	:
1041	003122	072227	000010		ASH	#10,R2	:
1042	003126	022402			CMP	(R4)+,R2	:
1043	003130	001403			BEQ	26\$	:
1044	003132	104000			ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR

Line	Address	Offset	Value	Instruction	Register	Comment
1045	003134	000017		.WORD	17	;UNIQUE ERROR NUMBER
1046	003136	001213		.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1047						
1048	003140	005201		26\$: INC	R1	
1049	003142	020427	177700	CMP	R4,#177700	
1050	003146	001364		BNE	25\$	
1051	003150	012701	001100	MOV	#FLOAT,R1	;CHECK AC5 FOR #6
1052	003154	012704	001110	MOV	#FLO,R4	
1053	003160	174014		STD	AC0,(R4)	
1054	003162	172405		LDD	AC5,AC0	
1055	003164	174011		STD	AC0,(R1)	
1056	003166	022127	000006	CMP	(R1)+,#6	
1057	003172	001403		BEQ	27\$	
1058	003174	104000		ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1059	003176	000020		.WORD	20	;UNIQUE ERROR NUMBER
1060	003200	001213		.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1061						
1062	003202	012703	000003	27\$: MOV	#3,R3	
1063	003206	022127	000000	28\$: CMP	(R1)+,#0	
1064	003212	001403		BEQ	29\$	
1065	003214	104000		ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1066	003216	000021		.WORD	21	;UNIQUE ERROR NUMBER
1067	003220	001213		.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1068						
1069	003222	005303		29\$: DEC	R3	
1070	003224	001370		BNE	28\$	
1071	003226	012701	001100	MOV	#FLOAT,R1	;CHECK AC4 FOR #5
1072	003232	172404		LDD	AC4,AC0	
1073	003234	174011		STD	AC0,(R1)	
1074	003236	022127	000005	CMP	(R1)+,#5	
1075	003242	001403		BEQ	30\$	
1076	003244	104000		ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1077	003246	000022		.WORD	22	;UNIQUE ERROR NUMBER
1078	003250	001213		.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1079						
1080	003252	012703	000003	30\$: MOV	#3,R3	
1081	003256	022127	000000	31\$: CMP	(R1)+,#0	
1082	003262	001403		BEQ	32\$	
1083	003264	104000		ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1084	003266	000023		.WORD	23	;UNIQUE ERROR NUMBER
1085	003270	001213		.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1086						
1087	003272	005303		32\$: DEC	R3	
1088	003274	001370		BNE	31\$	
1089	003276	022427	000001	CMP	(R4)+,#1	;CHECK AC0 FOR #1
1090	003302	001403		BEQ	33\$	
1091	003304	104000		ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1092	003306	000024		.WORD	24	;UNIQUE ERROR NUMBER
1093	003310	001213		.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1094						
1095	003312	012703	000003	33\$: MOV	#3,R3	
1096	003316	022427	000000	34\$: CMP	(R4)+,#0	
1097	003322	001403		BEQ	35\$	
1098	003324	104000		ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1099	003326	000025		.WORD	25	;UNIQUE ERROR NUMBER
1100	003330	001213		.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE

1101							
1102	003332	005303		35\$:	DEC R3	:	:
1103	003334	001370			BNE 34\$	:	:
1104	003336	012701	001100		MOV #FLOAT,R1	:	:CHECK AC1 FOR #2
1105	003342	174111			STD AC1,(R1)	:	:
1106	003344	022127	000002		CMP (R1)+,#2	:	:
1107	003350	001403			BEQ 36\$	:	:
1108	003352	104000			ERROR	:	:ALL ERRORS TO TRAP TO EMT VECTOR
1109	003354	000026			.WORD 26	:	:UNIQUE ERROR NUMBER
1110	003356	001213			.WORD MMUERR	:	:ADDRESS OF ERROR MESSAGE
1111						:	:
1112	003360	012703	000003	36\$:	MOV #3,R3	:	:
1113	003364	022127	000000	37\$:	CMP (R1)+,#0	:	:
1114	003370	001403			BEQ 38\$	:	:
1115	003372	104000			ERROR	:	:ALL ERRORS TO TRAP TO EMT VECTOR
1116	003374	000027			.WORD 27	:	:UNIQUE ERROR NUMBER
1117	003376	001213			.WORD MMUERR	:	:ADDRESS OF ERROR MESSAGE
1118						:	:
1119	003400	005303		38\$:	DEC R3	:	:
1120	003402	001370			BNE 37\$	:	:
1121	003404	012701	001100		MOV #FLOAT,R1	:	:CHECK AC2 FOR #3
1122	003410	174211			STD AC2,(R1)	:	:
1123	003412	022127	000003		CMP (R1)+,#3	:	:
1124	003416	001403			BEQ 39\$	:	:
1125	003420	104000			ERROR	:	:ALL ERRORS TO TRAP TO EMT VECTOR
1126	003422	000030			.WORD 30	:	:UNIQUE ERROR NUMBER
1127	003424	001213			.WORD MMUERR	:	:ADDRESS OF ERROR MESSAGE
1128						:	:
1129	003426	012703	000003	39\$:	MOV #3,R3	:	:
1130	003432	022127	000000	40\$:	CMP (R1)+,#0	:	:
1131	003436	001403			BEQ 41\$	:	:
1132	003440	104000			ERROR	:	:ALL ERRORS TO TRAP TO EMT VECTOR
1133	003442	000031			.WORD 31	:	:UNIQUE ERROR NUMBER
1134	003444	001213			.WORD MMUERR	:	:ADDRESS OF ERROR MESSAGE
1135						:	:
1136	003446	005303		41\$:	DEC R3	:	:
1137	003450	001370			BNE 40\$	:	:
1138	003452	012701	001100		MOV #FLOAT,R1	:	:CHECK AC3 FOR #4
1139	003456	174311			STD AC3,(R1)	:	:
1140	003460	022127	000004		CMP (R1)+,#4	:	:
1141	003464	001403			BEQ 42\$	:	:
1142	003466	104000			ERROR	:	:ALL ERRORS TO TRAP TO EMT VECTOR
1143	003470	000032			.WORD 32	:	:UNIQUE ERROR NUMBER
1144	003472	001213			.WORD MMUERR	:	:ADDRESS OF ERROR MESSAGE
1145						:	:
1146	003474	012703	000003	42\$:	MOV #3,R3	:	:
1147	003500	022127	000000	43\$:	CMP (R1)+,#0	:	:
1148	003504	001403			BEQ 44\$	:	:
1149	003506	104000			ERROR	:	:ALL ERRORS TO TRAP TO EMT VECTOR
1150	003510	000033			.WORD 33	:	:UNIQUE ERROR NUMBER
1151	003512	001213			.WORD MMUERR	:	:ADDRESS OF ERROR MESSAGE
1152						:	:
1153	003514	005303		44\$:	DEC R3	:	:
1154	003516	001370			BNE 43\$	:	:
1155	003520	020527	000000		CMP R5,#0	:	:IS TIME OUT FLAG 0
1156	003524	001403			BEQ 45\$	:	:YES GO ON

1157	003526	104000		ERROR					
1158	003530	000034		.WORD	34				: ALL ERRORS TO TRAP TO EMT VECTOR
1159	003532	001213		.WORD	MMUERR				: UNIQUE ERROR NUMBER
1160									: ADDRESS OF ERROR MESSAGE
1161	003534	012637	000004						: NO GO TO ERROR
1162	003540	012637	000246	45%: MOV	(SP)+,004				: RESTORE TIME OUT VECTOR
1163	003544	012637	000244	MOV	(SP)+,00246				: RESTORE FP VECTOR
1164				MOV	(SP)+,00244				:
1165	003550								
1166				TSMU3:					
1167				::*****					
1168				: TEST 3	WRITE ALL PARS/PDRS WITH ONES THEN ZEROS				
1169	003550			::*****					
1170	003550	005267	175230	TST3:					
1171	003554	005037	177572	INC	\$TESTN				: INCREMENT TEST NUMBER
1172	003560	005037	001042	CLR	00177572				: MMU OFF
1173	003564	012703	172200	CLR	00FLAG				: CLEAR MMU ABORT FLAG
1174	003570	012723	177777	MOV	0172200,R3				: LOAD ALL PARS AND PDRS WITH ONES
1175	003574	020327	172400	1%: MOV	0177777,(R3).				:
1176	003600	001373		CMP	R3,0172400				:
1177	003602	012703	177600	BNE	1%				:
1178	003606	012723	177777	MOV	0177600,R3				:
1179	003612	020327	177700	2%: MOV	0177777,(R3).				:
1180	003616	001373		CMP	R3,0177700				:
1181	003620	012703	172200	BNE	2%				:
1182	003624	022327	177416	MOV	0172200,R3				: CHECK SPDRS FOR ONES
1183	003630	001403		3%: CMP	(R3)+,0177416				:
1184	003632	104000		BEQ	4%				:
1185	003634	000035		ERROR					: ALL ERRORS TO TRAP TO EMT VECTOR
1186	003636	001213		.WORD	35				: UNIQUE ERROR NUMBER
1187				.WORD	MMUERR				: ADDRESS OF ERROR MESSAGE
1188	003640	020327	172240	4%: CMP	R3,0172240				:
1189	003644	001367		BNE	3%				:
1190	003646	022327	177777	5%: CMP	(R3)+,0177777				: CHECK SPARS FOR ONES
1191	003652	001403		BEQ	6%				:
1192	003654	104000		ERROR					: ALL ERRORS TO TRAP TO EMT VECTOR
1193	003656	000036		.WORD	36				: UNIQUE ERROR NUMBER
1194	003660	001213		.WORD	MMUERR				: ADDRESS OF ERROR MESSAGE
1195									
1196	003662	020327	172300	6%: CMP	R3,0172300				:
1197	003666	001367		BNE	5%				:
1198	003670	022327	177416	7%: CMP	(R3)+,0177416				: CHECK KPDRS FOR ONES
1199	003674	001403		BEQ	8%				:
1200	003676	104000		ERROR					: ALL ERRORS TO TRAP TO EMT VECTOR
1201	003700	000037		.WORD	37				: UNIQUE ERROR NUMBER
1202	003702	001213		.WORD	MMUERR				: ADDRESS OF ERROR MESSAGE
1203									
1204	003704	020327	172340	8%: CMP	R3,0172340				:
1205	003710	001367		BNE	7%				:
1206	003712	022327	177777	9%: CMP	(R3)+,0177777				: CHECK KPARS FOR ONES
1207	003716	001403		BEQ	10%				:
1208	003720	104000		ERROR					: ALL ERRORS TO TRAP TO EMT VECTOR
1209	003722	000040		.WORD	40				: UNIQUE ERROR NUMBER
1210	003724	001213		.WORD	MMUERR				: ADDRESS OF ERROR MESSAGE
1211									
1212	003726	020327	172400	10%: CMP	R3,0172400				:

```
1213 003732 001367      BNE      9%
1214 003734 012703 177600 MOV      @177600,R3
1215 003740 022327 177416 11$: CMP      (R3)+,@177416      ;CHECK UPDRS FOR ONES
1216 003744 001403      BEQ      12%      ;
1217 003746 104000      ERROR      ;ALL ERRORS TO TRAP TO EMT VECTOR
1218 003750 000041      .WORD      41      ;UNIQUE ERROR NUMBER
1219 003752 001213      .WORD      MMUERR      ;ADDRESS OF ERROR MESSAGE
1220
1221 003754 020327 177640 12$: CMP      R3,@177640      ;
1222 003760 001367      BNE      11%      ;
1223 003762 022327 177777 13$: CMP      (R3)+,@177777      ;CHECK UPARS FOR ONES
1224 003766 001403      BEQ      14%      ;
1225 003770 104000      ERROR      ;ALL ERRORS TO TRAP TO EMT VECTOR
1226 003772 000042      .WORD      42      ;UNIQUE ERROR NUMBER
1227 003774 001213      .WORD      MMUERR      ;ADDRESS OF ERROR MESSAGE
1228
1229 003776 020327 177700 14$: CMP      R3,@177700      ;
1230 004002 001367      BNE      13%      ;
1231 004004 012703 172200 MOV      @172200,R3      ;LOAD ALL PARS AND PDRS WITH ZEROES
1232 004010 012723 000000 15$: MOV      @0,(R3)+      ;
1233 004014 020327 172400      CMP      R3,@172400      ;
1234 004020 001373      BNE      15%      ;
1235 004022 012703 177600      MOV      @177600,R3      ;
1236 004026 012723 000000      MOV      @0,(R3)+      ;
1237 004032 020327 177700      CMP      R3,@177700      ;
1238 004036 001373      BNE      16%      ;
1239 004040 012703 172200      MOV      @172200,R3      ;CHECK ALL PARS AND PDRS FOR ZEROES
1240 004044 022327 000000 17$: CMP      (R3)+,@0      ;
1241 004050 001403      BEQ      18%      ;
1242 004052 104000      ERROR      ;ALL ERRORS TO TRAP TO EMT VECTOR
1243 004054 000043      .WORD      43      ;UNIQUE ERROR NUMBER
1244 004056 001213      .WORD      MMUERR      ;ADDRESS OF ERROR MESSAGE
1245
1246 004060 020327 172400 18$: CMP      R3,@172400      ;
1247 004064 001367      BNE      17%      ;
1248 004066 012703 177600      MOV      @177600,R3      ;
1249 004072 022327 000000 19$: CMP      (R3)+,@0      ;
1250 004076 001403      BEQ      20%      ;
1251 004100 104000      ERROR      ;ALL ERRORS TO TRAP TO EMT VECTOR
1252 004102 000044      .WORD      44      ;UNIQUE ERROR NUMBER
1253 004104 001213      .WORD      MMUERR      ;ADDRESS OF ERROR MESSAGE
1254
1255 004106 020327 177700 20$: CMP      R3,@177700      ;
1256 004112 001367      BNE      19%      ;
1257
1258 004114      TSMMU4:
1259      ;*****
1260      ;*TEST 4      TEST FOR ADJACENT SHORTS IN PARS/PDRS
1261      ;*****
1262      TST4:
1263 004114 005267 174664      INC      $TESTN      ;INCREMENT TEST NUMBER
1264 004120 005037 177572      CLR      @177572      ;MMU OFF
1265 004124 005067 174712      CLR      FLAG      ;CLEAR MMU ABORT FLAG
1266 004130 012700 172200      MOV      @172200,R0      ;LOAD SPDRS WITH ALTERNATING PATTERN
1267 004134 012720 052404 1$: MOV      @52404,(R0)+      ;
1268 004140 012720 125012      MOV      @125012,(R0)+      ;
```


1269	004144	020027	172240		CMP	R0,#172240	:
1270	004150	001371			BNE	1:	:
1271	004152	012720	125252	2:	MOV	#125252,(R0).	:LOAD SPARS WITH ALTERNATING PATTERN
1272	004156	012720	052525		MOV	#52525,(R0).	:
1273	004162	020027	172300		CMP	R0,#172300	:
1274	004166	001371			BNE	2:	:
1275	004170	012720	052404	3:	MOV	#52404,(R0).	:LOAD KPDRS WITH ALTERNATING PATTERN
1276	004174	012720	125012		MOV	#125012,(R0).	:
1277	004200	020027	172340		CMP	R0,#172340	:
1278	004204	001371			BNE	3:	:
1279	004206	012720	125252	4:	MOV	#125252,(R0).	:LOAD KPARS WITH ALTERNATING PATTERN
1280	004212	012720	052525		MOV	#52525,(R0).	:
1281	004216	020027	172400		CMP	R0,#172400	:
1282	004222	001371			BNE	4:	:
1283	004224	012700	177600		MOV	#177600,R0	:LOAD UPDRS WITH ALTERNATING PATTERN
1284	004230	012720	052404	5:	MOV	#52404,(R0).	:
1285	004234	012720	125012		MOV	#125012,(R0).	:
1286	004240	020027	177640		CMP	R0,#177640	:
1287	004244	001371			BNE	5:	:
1288	004246	012720	125252	6:	MOV	#125252,(R0).	:LOAD UPARS WITH ALTERNATING PATTERN
1289	004252	012720	052525		MOV	#52525,(R0).	:
1290	004256	020027	177700		CMP	R0,#177700	:
1291	004262	001371			BNE	6:	:
1292				:			:
1293	004264	012703	172200		MOV	#172200,R3	:CHECK SPDRS
1294	004270	022327	052404	7:	CMP	(R3),#52404	:
1295	004274	001403			BEQ	8:	:
1296	004276	104000			ERROR		:ALL ERRORS TO TRAP TO EMT VECTOR
1297	004300	000045			.WORD	45	:UNIQUE ERROR NUMBER
1298	004302	001213			.WORD	MMUERR	:ADDRESS OF ERROR MESSAGE
1299							:
1300	004304	022327	125012	8:	CMP	(R3),#125012	:
1301	004310	001403			BEQ	9:	:
1302	004312	104000			ERROR		:ALL ERRORS TO TRAP TO EMT VECTOR
1303	004314	000046			.WORD	46	:UNIQUE ERROR NUMBER
1304	004316	001213			.WORD	MMUERR	:ADDRESS OF ERROR MESSAGE
1305							:
1306	004320	020327	172240	9:	CMP	R3,#172240	:
1307	004324	001361			BNE	7:	:
1308	004326	022327	125252	10:	CMP	(R3),#125252	:CHECK SPARS
1309	004332	001403			BEQ	11:	:
1310	004334	104000			ERROR		:ALL ERRORS TO TRAP TO EMT VECTOR
1311	004336	000047			.WORD	47	:UNIQUE ERROR NUMBER
1312	004340	001213			.WORD	MMUERR	:ADDRESS OF ERROR MESSAGE
1313							:
1314	004342	022327	052525	11:	CMP	(R3),#52525	:
1315	004346	001403			BEQ	12:	:
1316	004350	104000			ERROR		:ALL ERRORS TO TRAP TO EMT VECTOR
1317	004352	000050			.WORD	50	:UNIQUE ERROR NUMBER
1318	004354	001213			.WORD	MMUERR	:ADDRESS OF ERROR MESSAGE
1319							:
1320	004356	020327	172300	12:	CMP	R3,#172300	:
1321	004362	001361			BNE	10:	:
1322	004364	022327	052404	13:	CMP	(R3),#52404	:CHECK KPDRS
1323	004370	001403			BEQ	14:	:
1324	004372	104000			ERROR		:ALL ERRORS TO TRAP TO EMT VECTOR

1325	004374	000051			.WORD	51			:UNIQUE ERROR NUMBER
1326	004376	001213			.WORD	MMUERR			:ADDRESS OF ERROR MESSAGE
1327									
1328	004400	022327	125012	14:	CMP	(R3)+, #125012			
1329	004404	001403			BEQ	15:			
1330	004406	104000			ERROR				:ALL ERRORS TO TRAP TO EMT VECTOR
1331	004410	000052			.WORD	52			:UNIQUE ERROR NUMBER
1332	004412	001213			.WORD	MMUERR			:ADDRESS OF ERROR MESSAGE
1333									
1334	004414	020327	172340	15:	CMP	R3, #172340			
1335	004420	001361			BNE	13:			
1336	004422	022327	125252	16:	CMP	(R3)+, #125252			:CHECK KPARS
1337	004426	001403			BEQ	17:			
1338	004430	104000			ERROR				:ALL ERRORS TO TRAP TO EMT VECTOR
1339	004432	000053			.WORD	53			:UNIQUE ERROR NUMBER
1340	004434	001213			.WORD	MMUERR			:ADDRESS OF ERROR MESSAGE
1341									
1342	004436	022327	052525	17:	CMP	(R3)+, #52525			
1343	004442	001403			BEQ	18:			
1344	004444	104000			ERROR				:ALL ERRORS TO TRAP TO EMT VECTOR
1345	004446	000054			.WORD	54			:UNIQUE ERROR NUMBER
1346	004450	001213			.WORD	MMUERR			:ADDRESS OF ERROR MESSAGE
1347									
1348	004452	020327	172400	18:	CMP	R3, #172400			
1349	004456	001361			BNE	16:			
1350	004460	012703	177600		MOV	#177600, R3			:CHECK UPDRS
1351	004464	022327	052404	19:	CMP	(R3)+, #52404			
1352	004470	001403			BEQ	20:			
1353	004472	104000			ERROR				:ALL ERRORS TO TRAP TO EMT VECTOR
1354	004474	000055			.WORD	55			:UNIQUE ERROR NUMBER
1355	004476	001213			.WORD	MMUERR			:ADDRESS OF ERROR MESSAGE
1356									
1357	004500	022327	125012	20:	CMP	(R3)+, #125012			
1358	004504	001403			BEQ	21:			
1359	004506	104000			ERROR				:ALL ERRORS TO TRAP TO EMT VECTOR
1360	004510	000056			.WORD	56			:UNIQUE ERROR NUMBER
1361	004512	001213			.WORD	MMUERR			:ADDRESS OF ERROR MESSAGE
1362									
1363	004514	020327	177640	21:	CMP	R3, #177640			
1364	004520	001361			BNE	19:			
1365	004522	022327	125252	22:	CMP	(R3)+, #125252			:CHECK UPARS
1366	004526	001403			BEQ	23:			
1367	004530	104000			ERROR				:ALL ERRORS TO TRAP TO EMT VECTOR
1368	004532	000057			.WORD	57			:UNIQUE ERROR NUMBER
1369	004534	001213			.WORD	MMUERR			:ADDRESS OF ERROR MESSAGE
1370									
1371	004536	022327	052525	23:	CMP	(R3)+, #52525			
1372	004542	001403			BEQ	24:			
1373	004544	104000			ERROR				:ALL ERRORS TO TRAP TO EMT VECTOR
1374	004546	000060			.WORD	60			:UNIQUE ERROR NUMBER
1375	004550	001213			.WORD	MMUERR			:ADDRESS OF ERROR MESSAGE
1376									
1377	004552	020327	177700	24:	CMP	R3, #177700			
1378	004556	001361			BNE	22:			
1379									
1380									:REVERSE ALTERNATING PATTERN

1381				:			
1382	004560	012700	172200		MOV	#172200,R0	:LOAD SPDRS WITH REVERSE PATTERN
1383	004564	012720	125012	25\$:	MOV	#125012,(R0)+	:
1384	004570	012720	052404		MOV	#52404,(R0)+	:
1385	004574	020027	172240		CMP	R0,#172240	:
1386	004600	001371			BNE	25\$	:
1387	004602	012720	052525	26\$:	MOV	#52525,(R0)+	:LOAD SPARS WITH REVERSE PATTERN
1388	004606	012720	125252		MOV	#125252,(R0)+	:
1389	004612	020027	172300		CMP	R0,#172300	:
1390	004616	001371			BNE	26\$	:
1391	004620	012720	125012	27\$:	MOV	#125012,(R0)+	:LOAD KPDRS WITH REVERSE PATTERN
1392	004624	012720	052404		MOV	#52404,(R0)+	:
1393	004630	020027	172340		CMP	R0,#172340	:
1394	004634	001371			BNE	27\$	:
1395	004636	012720	052525	28\$:	MOV	#52525,(R0)+	:LOAD KPARS WITH REVERSE PATTERN
1396	004642	012720	125252		MOV	#125252,(R0)+	:
1397	004646	020027	172400		CMP	R0,#172400	:
1398	004652	001371			BNE	28\$	:
1399	004654	012700	177600		MOV	#177600,R0	:LOAD UPDRS WITH REVERSE PATTERN
1400	004660	012720	125012	29\$:	MOV	#125012,(R0)+	:
1401	004664	012720	052404		MOV	#52404,(R0)+	:
1402	004670	020027	177640		CMP	R0,#177640	:
1403	004674	001371			BNE	29\$	:
1404	004676	012720	052525	30\$:	MOV	#52525,(R0)+	:LOAD UPARS WITH REVERSE PATTERN
1405	004702	012720	125252		MOV	#125252,(R0)+	:
1406	004706	020027	177700		CMP	R0,#177700	:
1407	004712	001371			BNE	30\$	:
1408				:			
1409	004714	012703	172200		MOV	#172200,R3	:CHECK SPDRS
1410	004720	022327	125012	31\$:	CMP	(R3)+,#125012	:
1411	004724	001403			BEQ	32\$	:
1412	004726	104000			ERROR		:ALL ERRORS TO TRAP TO EMT VECTOR
1413	004730	000061			.WORD	61	:UNIQUE ERROR NUMBER
1414	004732	001213			.WORD	MMUERR	:ADDRESS OF ERROR MESSAGE
1415							
1416	004734	022327	052404	32\$:	CMP	(R3)+,#52404	:
1417	004740	001403			BEQ	33\$	:
1418	004742	104000			ERROR		:ALL ERRORS TO TRAP TO EMT VECTOR
1419	004744	000062			.WORD	62	:UNIQUE ERROR NUMBER
1420	004746	001213			.WORD	MMUERR	:ADDRESS OF ERROR MESSAGE
1421							
1422	004750	020327	172240	33\$:	CMP	R3,#172240	:
1423	004754	001361			BNE	31\$	:
1424	004756	022327	052525	34\$:	CMP	(R3)+,#52525	:CHECK SPARS
1425	004762	001403			BEQ	35\$	:
1426	004764	104000			ERROR		:ALL ERRORS TO TRAP TO EMT VECTOR
1427	004766	000063			.WORD	63	:UNIQUE ERROR NUMBER
1428	004770	001213			.WORD	MMUERR	:ADDRESS OF ERROR MESSAGE
1429							
1430	004772	022327	125252	35\$:	CMP	(R3)+,#125252	:
1431	004776	001403			BEQ	36\$	:
1432	005000	104000			ERROR		:ALL ERRORS TO TRAP TO EMT VECTOR
1433	005002	000064			.WORD	64	:UNIQUE ERROR NUMBER
1434	005004	001213			.WORD	MMUERR	:ADDRESS OF ERROR MESSAGE
1435							
1436	005006	020327	172300	36\$:	CMP	R3,#172300	:

Address	OpCode	Op1	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418</
---------	--------	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	---------


```

1493 005202 020327 177700 48$: CMP R3,#177700 ;
1494 005206 001361 46$: BNE 46$ ;
1495
1496 005210
1497 TSMMU5:
1498 ;*****
1499 ;*TEST 5 TEST MMRO ABORT BITS
1500 ;*****
1501 005210 005267 173570 TST5: INC $TESTN ;INCREMENT TEST NUMBER
1502 005214 012737 160000 177572 MOV #160000,#177572 ;LOAD MMRO<15:13>=111
1503 005222 005067 173614 CLR FLAG ;CLEAR MMU ABORT FLAG
1504 005226 013700 177572 MOV @SRO,R0 ;SAVE SRO IN R0
1505 005232 042700 000176 BIC #176,R0 ;CLEAR UNDEFINED BITS FROM SRO
1506 005236 020027 160000 CMP R0,#160000 ;CHECK MMRO
1507 005242 001403 BEQ 1$ ;
1508 005244 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
1509 005246 000075 .WORD 75 ;UNIQUE ERROR NUMBER
1510 005250 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
1511
1512 005252 005037 177572 1$: CLR @#177572 ;LOAD MMRO=0
1513 005256 013700 177572 MOV @SRO,R0 ;SAVE SRO IN R0
1514 005262 042700 000176 BIC #176,R0 ;CLEAR UNDEFINED BITS FROM SRO
1515 005266 020027 000000 CMP R0,#0 ;CHECK MMRO
1516 005272 001403 BEQ 2$ ;
1517 005274 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
1518 005276 000076 .WORD 76 ;UNIQUE ERROR NUMBER
1519 005300 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
1520
1521 005302 012737 120000 177572 2$: MOV #120000,#177572 ;LOAD MMRO<15:13>=101
1522 005310 013700 177572 MOV @SRO,R0 ;SAVE SRO IN R0
1523 005314 042700 000176 BIC #176,R0 ;CLEAR UNDEFINED BITS FROM SRO.
1524 005320 020027 120000 CMP R0,#120000 ;CHECK MMRO
1525 005324 001403 BEQ 3$ ;
1526 005326 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
1527 005330 000077 .WORD 77 ;UNIQUE ERROR NUMBER
1528 005332 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
1529
1530 005334 012737 040000 177572 3$: MOV #40000,#177572 ;LOAD MMRO<15:13>=010
1531 005342 013700 177572 MOV @SRO,R0 ;SAVE SRO IN R0
1532 005346 042700 000176 BIC #176,R0 ;CLEAR UNDEFINED BITS FROM SRO.
1533 005352 020027 040000 CMP R0,#40000 ;CHECK MMRO
1534 005356 001403 BEQ 4$ ;
1535 005360 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
1536 005362 000100 .WORD 100 ;UNIQUE ERROR NUMBER
1537 005364 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
1538 005366
1539 4$:
1540 005366
1541 TSMMU6:
1542 ;*****
1543 ;*TEST 6 TEST MMR3 BITS 5-0
1544 ;*****
1545 005366 005267 173412 TST6: INC $TESTN ;INCREMENT TEST NUMBER
1546 005372 005037 177572 CLR @#177572 ;MMU OFF
1547 005376 005067 173440 CLR FLAG ;CLEAR MMU ABORT FLAG
1548 005402 012737 000077 172516 MOV #77,#172516 ;LOAD MMR3<5:0>=77

```

```
1549 005410 023727 172516 000077      CMP      @#172516,@#77      ;CHECK MMR3
1550 005416 001403                      BEQ      1$                      ;
1551 005420 104000                      ERROR                      ;ALL ERRORS TO TRAP TO EMT VECTOR
1552 005422 000101                      .WORD      101                      ;UNIQUE ERROR NUMBER
1553 005424 001213                      .WORD      MMUERR                      ;ADDRESS OF ERROR MESSAGE
1554 005426 005037 172516 000000 1$:    CLR      @#172516      ;LOAD MMR3<5:0>=0
1555 005432 023727 172516 000000      CMP      @#172516,@#0      ;CHECK MMR3
1556 005440 001403                      BEQ      2$                      ;
1557 005442 104000                      ERROR                      ;ALL ERRORS TO TRAP TO EMT VECTOR
1558 005444 000102                      .WORD      102                      ;UNIQUE ERROR NUMBER
1559 005446 001213                      .WORD      MMUERR                      ;ADDRESS OF ERROR MESSAGE
1560 005450 012737 000052 172516 2$:    MOV      @52,@#172516      ;LOAD MMR3<5:0>=52
1561 005456 023727 172516 000052      CMP      @#172516,@52      ;CHECK MMR3
1562 005464 001403                      BEQ      3$                      ;
1563 005466 104000                      ERROR                      ;ALL ERRORS TO TRAP TO EMT VECTOR
1564 005470 000103                      .WORD      103                      ;UNIQUE ERROR NUMBER
1565 005472 001213                      .WORD      MMUERR                      ;ADDRESS OF ERROR MESSAGE
1566 005474 012737 000025 172516 3$:    MOV      @25,@#172516      ;LOAD MMR3<5:0>=25
1567 005502 023727 172516 000025      CMP      @#172516,@25      ;CHECK MMR3
1568 005510 001403                      BEQ      4$                      ;
1569 005512 104000                      ERROR                      ;ALL ERRORS TO TRAP TO EMT VECTOR
1570 005514 000104                      .WORD      104                      ;UNIQUE ERROR NUMBER
1571 005516 001213                      .WORD      MMUERR                      ;ADDRESS OF ERROR MESSAGE
1572 005520 4$:
1573
1574 005520      TSMM6A:
1575      ;*****
1576      ;*TEST 7      TEST MFPI (MOVE FROM PREVIOUS INST SPACE)
1577      ;*****
1578 005520      TST7:
1579 005520 005267 173260      INC      $TESTN      ;INCREMENT TEST NUMBER
1580 005524 005037 177572      CLR      @#177572      ;MMU OFF
1581 005530 005037 001042      CLR      @#FLAG      ;CLEAR MMU ABORT FLAG
1582 005534 012737 140000 177776      MOV      @140000,@#177776      ;POINT TO USER SPACE
1583 005542 012706 001000      MOV      @STBOT,SP      ;INIT THE USER STACK POINTER
1584 005546 010637 001070      MOV      R6,@#SAVUSE      ;SAVE USER SP
1585 005552 012737 040000 177776      MOV      @40000,@#177776      ;POINT TO SUPERVISOR SPACE
1586 005560 012706 001000      MOV      @STBOT,SP      ;INIT THE SUPERVISOR STACK POINTER
1587 005564 010637 001066      MOV      R6,@#SAVSUP      ;SAVE SUPERVISOR SP
1588 005570 012737 030000 177776      MOV      @30000,@#177776      ;SETUP PSW
1589 005576 004767 173504      JSR      PC,MMU      ;INIT MMU
1590 005602 012737 000027 172516      MOV      @27,@#172516      ;SETUP MMR3
1591 005610 013746 000244      MOV      @#244,-(SP)      ;SAVE DATA AT TEST LOCATION
1592 005614 012746 177777      MOV      @177777,-(SP)      ;PUT KNOWN DATA ON TOP OF STACK
1593 005620 012737 135072 000244      MOV      @135072,@#244      ;SETUP DATA AT TEST LOCATION
1594 005626 012767 077400 171764      MOV      @77400,UDPDR0      ;SETUP UDPDR0 TO ABORT
1595 005634 012703 000244      MOV      @244,R3      ;SETUP POINTER TO TEST LOCATION
1596 005640 005237 177572      INC      @#177572      ;TURN MMU ON
1597 005644 006523      MFPI      (R3),      ;TEST INSTRUCTION
1598 005646 022737 030010 177776      CMP      @30010,@#177776      ;IS PSW CORRECT
1599 005654 001403                      BEQ      1$                      ;ES GO ON
1600 005656 104000                      ERROR                      ;ALL ERRORS TO TRAP TO EMT VECTOR
1601 005660 000105                      .WORD      105                      ;UNIQUE ERROR NUMBER
1602 005662 001213                      .WORD      MMUERR                      ;ADDRESS OF ERROR MESSAGE
1603
1604 005664 005037 177572 1$:    CLR      @#177572      ;NO GO TO ERROR
                                ;TURN MMU OFF
```


1605	005670	012737	140000	177776		MOV	#140000, #177776		:POINT TO USER SPACE
1606	005676	020637	001070			CMP	R6, #SAVUSE		:IS USER SP CORRECT
1607	005702	001403				BEQ	100\$		:YES GO ON
1608	005704	104000				ERROR			:ALL ERRORS TO TRAP TO EMT VECTOR
1609	005706	000106				.WORD	106		:UNIQUE ERROR NUMBER
1610	005710	001213				.WORD	MMUERR		:ADDRESS OF ERROR MESSAGE
1611									
1612	005712	012737	040000	177776	100\$:	MOV	#40000, #177776	:NO GO TO ERROR	:POINT TO SUPERVISOR SPACE
1613	005720	020637	001066			CMP	R6, #SAVSUP		:IS SUPERVISOR SP CORRECT
1614	005724	001403				BEQ	200\$		:YES GO ON
1615	005726	104000				ERROR			:ALL ERRORS TO TRAP TO EMT VECTOR
1616	005730	000107				.WORD	107		:UNIQUE ERROR NUMBER
1617	005732	001213				.WORD	MMUERR		:ADDRESS OF ERROR MESSAGE
1618									
1619	005734	023727	000244	135072	200\$:	CMP	#244, #135072	:NO GO TO ERROR	:IS TEST DATA OK
1620	005742	001403				BEQ	2\$		:YES GO ON
1621	005744	104000				ERROR			:ALL ERRORS TO TRAP TO EMT VECTOR
1622	005746	000110				.WORD	110		:UNIQUE ERROR NUMBER
1623	005750	001213				.WORD	MMUERR		:ADDRESS OF ERROR MESSAGE
1624									
1625	005752	020327	000246		2\$:	CMP	R3, #246	:NO GO TO ERROR	:IS R3 CORRECT
1626	005756	001403				BEQ	3\$		:YES GO ON
1627	005760	104000				ERROR			:ALL ERRORS TO TRAP TO EMT VECTOR
1628	005762	000111				.WORD	111		:UNIQUE ERROR NUMBER
1629	005764	001213				.WORD	MMUERR		:ADDRESS OF ERROR MESSAGE
1630									
1631	005766	005037	177776		3\$:	CLR	#177776	:NO GO TO ERROR	:SET PSW TO KERNEL MODE
1632	005772	022627	135072			CMP	(SP), #135072		:IS KERNEL STACK CORRECT
1633	005776	001403				BEQ	4\$		:YES GO ON
1634	006000	104000				ERROR			:ALL ERRORS TO TRAP TO EMT VECTOR
1635	006002	000112				.WORD	112		:UNIQUE ERROR NUMBER
1636	006004	001213				.WORD	MMUERR		:ADDRESS OF ERROR MESSAGE
1637									
1638	006006	021627	177777		4\$:	CMP	(SP), #177777	:NO GO TO ERROR	:IS STACK CORRECT
1639	006012	001403				BEQ	5\$		:YES GO ON
1640	006014	104000				ERROR			:ALL ERRORS TO TRAP TO EMT VECTOR
1641	006016	000113				.WORD	113		:UNIQUE ERROR NUMBER
1642	006020	001213				.WORD	MMUERR		:ADDRESS OF ERROR MESSAGE
1643									
1644	006022	012737	030017	177776	5\$:	MOV	#30017, #177776	:NO GO TO ERROR	:SETUP PSW
1645	006030	012737	173621	000244		MOV	#173621, #244		:SETUP TEST LOCATION
1646	006036	012701	000244			MOV	#244, R1		:SETUP R1
1647	006042	005237	177572			INC	#177572		:TURN MMU ON
1648	006046	006511				MFPI	(R1)		:TEST INSTRUCTION
1649	006050	022737	030011	177776		CMP	#30011, #177776		:IS PSW CORRECT
1650	006056	001403				BEQ	300\$		:YES GO ON
1651	006060	104000				ERROR			:ALL ERRORS TO TRAP TO EMT VECTOR
1652	006062	000114				.WORD	114		:UNIQUE ERROR NUMBER
1653	006064	001213				.WORD	MMUERR		:ADDRESS OF ERROR MESSAGE
1654									
1655	006066	005037	177572		300\$:	CLR	#177572	:NO GO TO ERROR	:TURN MMU OFF
1656	006072	023727	000244	173621		CMP	#244, #173621		:IS TEST LOCATION CORRECT
1657	006100	001403				BEQ	301\$		:YES GO ON
1658	006102	104000				ERROR			:ALL ERRORS TO TRAP TO EMT VECTOR
1659	006104	000115				.WORD	115		:UNIQUE ERROR NUMBER
1660	006106	001213				.WORD	MMUERR		:ADDRESS OF ERROR MESSAGE

```
1661
1662 006110 020127 000244      301$: CMP      R1,#244      ;NO GO TO ERROR
1663 006114 001403              BEQ      302$      ;IS R1 CORRECT
1664 006116 104000              ERROR              ;YES GO ON
1665 006120 000116              .WORD 116          ;ALL ERRORS TO TRAP TO EMT VECTOR
1666 006122 001213              .WORD MMUERR       ;UNIQUE ERROR NUMBER
1667                                     ;ADDRESS OF ERROR MESSAGE
1668 006124 005037 177776      302$: CLR      @#177776    ;NO GO TO ERROR
1669 006130 022627 173621      CMP      (SP)+,#173621 ;SET PSW TO KERNEL MODE
1670 006134 001403              BEQ      303$      ;IS STACK CORRECT
1671 006136 104000              ERROR              ;YES GO ON
1672 006140 000117              .WORD 117          ;ALL ERRORS TO TRAP TO EMT VECTOR
1673 006142 001213              .WORD MMUERR       ;UNIQUE ERROR NUMBER
1674                                     ;ADDRESS OF ERROR MESSAGE
1675 006144 021627 177777      303$: CMP      (SP),#177777 ;NO GO TO ERROR
1676 006150 001403              BEQ      304$      ;IS STACK CORRECT
1677 006152 104000              ERROR              ;YES GO ON
1678 006154 000120              .WORD 120          ;ALL ERRORS TO TRAP TO EMT VECTOR
1679 006156 001213              .WORD MMUERR       ;UNIQUE ERROR NUMBER
1680                                     ;ADDRESS OF ERROR MESSAGE
1681 006160 005003              304$: CLR      R3      ;NO GO TO ERROR
1682 006162 005237 177572      INC      @#177572    ;SETUP SOURCE FOR NEXT TEST
1683 006166 006503              MFPI      R3      ;TURN MMU ON
1684 006170 022737 000004 177776 CMP      @4,@#177776 ;TEST INSTRUCTION
1685 006176 001403              BEQ      6$      ;IS PSW CORRECT
1686 006200 104000              ERROR              ;YES GO ON
1687 006202 000121              .WORD 121          ;ALL ERRORS TO TRAP TO EMT VECTOR
1688 006204 001213              .WORD MMUERR       ;UNIQUE ERROR NUMBER
1689                                     ;ADDRESS OF ERROR MESSAGE
1690 006206 005037 177572      6$: CLR      @#177572    ;NO GO TO ERROR
1691 006212 020327 000000      CMP      R3,#0      ;TURN MMU OFF
1692 006216 001403              BEQ      7$      ;IS R3 CORRECT
1693 006220 104000              ERROR              ;YES GO ON
1694 006222 000122              .WORD 122          ;ALL ERRORS TO TRAP TO EMT VECTOR
1695 006224 001213              .WORD MMUERR       ;UNIQUE ERROR NUMBER
1696                                     ;ADDRESS OF ERROR MESSAGE
1697 006226 022627 000000      7$: CMP      (SP)+,#0    ;NO GO TO ERROR
1698 006232 001403              BEQ      8$      ;IS STACK CORRECT
1699 006234 104000              ERROR              ;YES GO ON
1700 006236 000123              .WORD 123          ;ALL ERRORS TO TRAP TO EMT VECTOR
1701 006240 001213              .WORD MMUERR       ;UNIQUE ERROR NUMBER
1702                                     ;ADDRESS OF ERROR MESSAGE
1703 006242 022627 177777      8$: CMP      (SP)+,#177777 ;NO GO TO ERROR
1704 006246 001403              BEQ      9$      ;IS STACK CORRECT
1705 006250 104000              ERROR              ;YES GO ON
1706 006252 000124              .WORD 124          ;ALL ERRORS TO TRAP TO EMT VECTOR
1707 006254 001213              .WORD MMUERR       ;UNIQUE ERROR NUMBER
1708                                     ;ADDRESS OF ERROR MESSAGE
1709 006256 012637 000244      9$: MOV      (SP)+,@#244  ;NO GO TO ERROR
1710                                     ;RESTORE TEST LOCATION
1711
1712 006262      ;
1713      TSMM6B:
1714      ;*****
1715      ;*TEST 10      TEST MFPI (MOVE FROM PREVIOUS DATA SPACE)
1716      ;*****
1716 006262      TST10:
```


1717	006262	005267	172516			INC	\$TESTN		; INCREMENT TEST NUMBER
1718	006266	005037	177572			CLR	\$#177572		; MMU OFF
1719	006272	005037	001042			CLR	\$#FLAG		; CLEAR MMU ABORT FLAG
1720	006276	012737	140000	177776		MOV	\$140000,\$#177776		; POINT TO USER SPACE
1721	006304	010637	001070			MOV	R6,\$#SAVUSE		; SAVE USER SP
1722	006310	012737	040000	177776		MOV	\$40000,\$#177776		; POINT TO SUPERVISOR SPACE
1723	006316	010637	001066			MOV	R6,\$#SAVSUP		; SAVE SUPERVISOR SP
1724	006322	012737	030000	177776		MOV	\$30000,\$#177776		; SETUP PSW
1725	006330	004767	172752			JSR	PC,MMU		; INIT MMU
1726	006334	012737	000027	172516		MOV	\$27,\$#172516		; SETUP MMR3
1727	006342	013746	000244			MOV	\$#244,-(SP)		; SAVE DATA AT TEST LOCATION
1728	006346	012746	177777			MOV	\$177777,-(SP)		; PUT KNOWN DATA ON TOP OF STACK
1729	006352	012737	157002	000244		MOV	\$157002,\$#244		; SETUP DATA AT TEST LOCATION
1730	006360	012767	077400	171212		MOV	\$77400,UIPDRO		; SETUP UIPDRO TO ABORT
1731	006366	012703	000244			MOV	\$244,R3		; SETUP POINTER TO TEST LOCATION
1732	006372	005237	177572			INC	\$#177572		; TURN MMU ON
1733	006376	106523				MFPD	(R3)+		; TEST INSTRUCTION
1734	006400	022737	030010	177776		CMP	\$30010,\$#177776		; IS PSW CORRECT
1735	006406	001403				BEQ	1\$		; YES GO ON
1736	006410	104000				ERROR			; ALL ERRORS TO TRAP TO EMT VECTOR
1737	006412	000125				.WORD	125		; UNIQUE ERROR NUMBER
1738	006414	001213				.WORD	MMUERR		; ADDRESS OF ERROR MESSAGE
1739									; NO GO TO ERROR
1740	006416	005037	177572		1\$:	CLR	\$#177572		; TURN MMU OFF
1741	006422	012737	140000	177776		MOV	\$140000,\$#177776		; POINT TO USER SPACE
1742	006430	020637	001070			CMP	R6,\$#SAVUSE		; IS USER SP CORRECT
1743	006434	001403				BEQ	100\$		; YES GO ON
1744	006436	104000				ERROR			; ALL ERRORS TO TRAP TO EMT VECTOR
1745	006440	000126				.WORD	126		; UNIQUE ERROR NUMBER
1746	006442	001213				.WORD	MMUERR		; ADDRESS OF ERROR MESSAGE
1747									; NO GO TO ERROR
1748	006444	012737	040000	177776	100\$:	MOV	\$40000,\$#177776		; POINT TO SUPERVISOR SPACE
1749	006452	020637	001066			CMP	R6,\$#SAVSUP		; IS SUPERVISOR SP CORRECT
1750	006456	001403				BEQ	200\$		; YES GO ON
1751	006460	104000				ERROR			; ALL ERRORS TO TRAP TO EMT VECTOR
1752	006462	000127				.WORD	127		; UNIQUE ERROR NUMBER
1753	006464	001213				.WORD	MMUERR		; ADDRESS OF ERROR MESSAGE
1754									; NO GO TO ERROR
1755	006466	023727	000244	157002	200\$:	CMP	\$#244,\$157002		; IS TEST DATA OK
1756	006474	001403				BEQ	2\$		; YES GO ON
1757	006476	104000				ERROR			; ALL ERRORS TO TRAP TO EMT VECTOR
1758	006500	000130				.WORD	130		; UNIQUE ERROR NUMBER
1759	006502	001213				.WORD	MMUERR		; ADDRESS OF ERROR MESSAGE
1760									; NO GO TO ERROR
1761	006504	020327	000246		2\$:	CMP	R3,\$#246		; IS R3 CORRECT
1762	006510	001403				BEQ	3\$		; YES GO ON
1763	006512	104000				ERROR			; ALL ERRORS TO TRAP TO EMT VECTOR
1764	006514	000131				.WORD	131		; UNIQUE ERROR NUMBER
1765	006516	001213				.WORD	MMUERR		; ADDRESS OF ERROR MESSAGE
1766									; NO GO TO ERROR
1767	006520	005037	177776		3\$:	CLR	\$#177776		; SET PSW TO KERNEL MODE
1768	006524	022627	157002			CMP	(SP)+,\$157002		; IS KERNEL STACK CORRECT
1769	006530	001403				BEQ	4\$		; YES GO ON
1770	006532	104000				ERROR			; ALL ERRORS TO TRAP TO EMT VECTOR
1771	006534	000132				.WORD	132		; UNIQUE ERROR NUMBER
1772	006536	001213				.WORD	MMUERR		; ADDRESS OF ERROR MESSAGE

```

1773
1774 006540 021627 177777 4$: CMP (SP),#177777 ;NO GO TO ERROR
1775 006544 001403 BEQ 5$ ;IS STACK CORRECT
1776 006546 104000 ERROR ;YES GO ON
1777 006550 000133 .WORD 133 ;ALL ERRORS TO TRAP TO EMT VECTOR
1778 006552 001213 .WORD MMUERR ;UNIQUE ERROR NUMBER
1779 ;ADDRESS OF ERROR MESSAGE
1780 006554 012737 030017 177776 5$: MOV #30017,#177776 ;NO GO TO ERROR
1781 006562 012737 103456 000244 MOV #103456,#244 ;SETUP PSW
1782 006570 012701 000244 MOV #244,R1 ;SETUP TEST LOCATION
1783 006574 005237 177572 INC #177572 ;SETUP R1
1784 006600 106511 MFPD (R1) ;TURN MMU ON
1785 006602 022737 030011 177776 CMP #30011,#177776 ;TEST INSTRUCTION
1786 006610 001403 BEQ 300$ ;IS PSW CORRECT
1787 006612 104000 ERROR ;YES GO ON
1788 006614 000134 .WORD 134 ;ALL ERRORS TO TRAP TO EMT VECTOR
1789 006616 001213 .WORD MMUERR ;UNIQUE ERROR NUMBER
1790 ;ADDRESS OF ERROR MESSAGE
1791 006620 005037 177572 300$: CLR #177572 ;NO GO TO ERROR
1792 006624 023727 000244 103456 CMP #244,#103456 ;TURN MMU OFF
1793 006632 001403 BEQ 301$ ;IS TEST LOCATION CORRECT
1794 006634 104000 ERROR ;YES GO ON
1795 006636 000135 .WORD 135 ;ALL ERRORS TO TRAP TO EMT VECTOR
1796 006640 001213 .WORD MMUERR ;UNIQUE ERROR NUMBER
1797 ;ADDRESS OF ERROR MESSAGE
1798 006642 020127 000244 301$: CMP R1,#244 ;NO GO TO ERROR
1799 006646 001403 BEQ 302$ ;IS R1 CORRECT
1800 006650 104000 ERROR ;YES GO ON
1801 006652 000136 .WORD 136 ;ALL ERRORS TO TRAP TO EMT VECTOR
1802 006654 001213 .WORD MMUERR ;UNIQUE ERROR NUMBER
1803 ;ADDRESS OF ERROR MESSAGE
1804 006656 005037 177776 302$: CLR #177776 ;NO GO TO ERROR
1805 006662 022627 103456 CMP (SP),#103456 ;SET PSW TO KERNEL MODE
1806 006666 001403 BEQ 303$ ;IS STACK CORRECT
1807 006670 104000 ERROR ;YES GO ON
1808 006672 000137 .WORD 137 ;ALL ERRORS TO TRAP TO EMT VECTOR
1809 006674 001213 .WORD MMUERR ;UNIQUE ERROR NUMBER
1810 ;ADDRESS OF ERROR MESSAGE
1811 006676 021627 177777 303$: CMP (SP),#177777 ;NO GO TO ERROR
1812 006702 001403 BEQ 304$ ;IS STACK CORRECT
1813 006704 104000 ERROR ;YES GO ON
1814 006706 000140 .WORD 140 ;ALL ERRORS TO TRAP TO EMT VECTOR
1815 006710 001213 .WORD MMUERR ;UNIQUE ERROR NUMBER
1816 ;ADDRESS OF ERROR MESSAGE
1817 006712 012737 030017 177776 304$: MOV #30017,#177776 ;NO GO TO ERROR
1818 006720 012737 113672 000244 MOV #113672,#244 ;SETUP PSW
1819 006726 012701 000246 MOV #246,R1 ;SETUP TEST LOCATION
1820 006732 005237 177572 INC #177572 ;SETUP R1
1821 006736 106541 MFPD -(R1) ;TURN MMU ON
1822 006740 022737 030011 177776 CMP #30011,#177776 ;TEST INSTRUCTION
1823 006746 001403 BEQ 400$ ;IS PSW CORRECT
1824 006750 104000 ERROR ;YES GO ON
1825 006752 000141 .WORD 141 ;ALL ERRORS TO TRAP TO EMT VECTOR
1826 006754 001213 .WORD MMUERR ;UNIQUE ERROR NUMBER
1827 ;ADDRESS OF ERROR MESSAGE
1828 006756 005037 177572 400$: CLR #177572 ;NO GO TO ERROR
;TURN MMU OFF

```


1829	006762	023727	000244	113672		CMP	@#244, #113672	;IS TEST LOCATION CORRECT
1830	006770	001403				BEQ	401\$	;YES GO ON
1831	006772	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1832	006774	000142				.WORD	142	;UNIQUE ERROR NUMBER
1833	006776	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1834								;NO GO TO ERROR
1835	007000	020127	000244		401\$:	CMP	R1, #244	;IS R1 CORRECT
1836	007004	001403				BEQ	402\$	;YES GO ON
1837	007006	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1838	007010	000143				.WORD	143	;UNIQUE ERROR NUMBER
1839	007012	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1840								;NO GO TO ERROR
1841	007014	005037	177776		402\$:	CLR	@#177776	;SET PSW TO KERNEL MODE
1842	007020	022627	113672			CMP	(SP)+, #113672	;IS STACK CORRECT
1843	007024	001403				BEQ	403\$	;YES GO ON
1844	007026	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1845	007030	000144				.WORD	144	;UNIQUE ERROR NUMBER
1846	007032	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1847								;NO GO TO ERROR
1848	007034	021627	177777		403\$:	CMP	(SP), #177777	;IS STACK CORRECT
1849	007040	001403				BEQ	404\$	;YES GO ON
1850	007042	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1851	007044	000145				.WORD	145	;UNIQUE ERROR NUMBER
1852	007046	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1853								;NO GO TO ERROR
1854	007050	005003			404\$:	CLR	R3	;SETUP SOURCE FOR NEXT TEST
1855	007052	005237	177572			INC	@#177572	;TURN MMU ON
1856	007056	106503				MFPD	R3	;TEST INSTRUCTION
1857	007060	022737	000004	177776		CMP	#4, @#177776	;IS PSW CORRECT
1858	007066	001403				BEQ	6\$	;YES GO ON
1859	007070	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1860	007072	000146				.WORD	146	;UNIQUE ERROR NUMBER
1861	007074	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1862								;NO GO TO ERROR
1863	007076	005037	177572		6\$:	CLR	@#177572	;TURN MMU OFF
1864	007102	020327	000000			CMP	R3, #0	;IS R3 CORRECT
1865	007106	001403				BEQ	7\$	;YES GO ON
1866	007110	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1867	007112	000147				.WORD	147	;UNIQUE ERROR NUMBER
1868	007114	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1869								;NO GO TO ERROR
1870	007116	022627	000000		7\$:	CMP	(SP)+, #0	;IS STACK CORRECT
1871	007122	001403				BEQ	8\$	;YES GO ON
1872	007124	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1873	007126	000150				.WORD	150	;UNIQUE ERROR NUMBER
1874	007130	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1875								;NO GO TO ERROR
1876	007132	022627	177777		8\$:	CMP	(SP)+, #177777	;IS STACK CORRECT
1877	007136	001403				BEQ	9\$	;YES GO ON
1878	007140	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
1879	007142	000151				.WORD	151	;UNIQUE ERROR NUMBER
1880	007144	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
1881								;NO GO TO ERROR
1882	007146	012637	000244		9\$:	MOV	(SP)+, @#244	;RESTORE TEST LOCATION
1883								
1884								

```
1885 007152
1886
1887
1888
1889 007152
1890 007152 005267 171626
1891 007156 005037 177572
1892 007162 005037 001042
1893 007166 012737 140000 177776
1894 007174 010637 001070
1895 007200 012737 040000 177776
1896 007206 010637 001066
1897 007212 012737 030000 177776
1898 007220 004767 172062
1899 007224 012737 000027 172516
1900 007232 013746 000244
1901 007236 012746 177777
1902 007242 012746 120413
1903 007246 012737 177777 000244
1904 007254 012767 077400 170336
1905 007262 012703 000244
1906 007266 005237 177572
1907 007272 006623
1908 007274 022737 030010 177776
1909 007302 001403
1910 007304 104000
1911 007306 000152
1912 007310 001213
1913
1914 007312 005037 177572
1915 007316 012737 140000 177776
1916 007324 020637 001070
1917 007330 001403
1918 007332 104000
1919 007334 000153
1920 007336 001213
1921
1922 007340 012737 040000 177776 100:
1923 007346 020637 001066
1924 007352 001403
1925 007354 104000
1926 007356 000154
1927 007360 001213
1928
1929 007362 023727 000244 120413 200:
1930 007370 001403
1931 007372 104000
1932 007374 000155
1933 007376 001213
1934
1935 007400 020327 000246 2:
1936 007404 001403
1937 007406 104000
1938 007410 000156
1939 007412 001213
1940

TSM6C:
*****
; *TEST 11 TEST MTPI (MOVE TO PREVIOUS INSTRUCTION SPACE)
*****
TST11:
INC $TESTN ; INCREMENT TEST NUMBER
CLR $0177572 ; MMU OFF
CLR $0FLAG ; CLEAR MMU ABORT FLAG
MOV $140000,$0177776 ; POINT TO USER SPACE
MOV R6,$0SAVUSE ; SAVE USER SP
MOV $40000,$0177776 ; POINT TO SUPERVISOR SPACE
MOV R6,$0SAVSUP ; SAVE SUPERVISOR SP
MOV $30000,$0177776 ; SETUP PSW
JSR PC,MMU ; INIT MMU
MOV $27,$0172516 ; SETUP MMR3
MOV $0244,-(SP) ; SAVE DATA AT TEST LOCATION
MOV $177777,-(SP) ; PUT KNOWN DATA ON STACK
MOV $120413,-(SP) ; PUT TEST DATA ON STACK
MOV $177777,$0244 ; PUT KNOWN DATA AT TEST LOCATION
MOV $77400,UDPDR0 ; SETUP UDPDR0 TO ABORT
MOV $244,R3 ; SETUP POINTER TO TEST LOCATION
INC $0177572 ; TURN MMU ON
MTPI (R3). ; TEST INSTRUCTION
CMP $30010,$0177776 ; IS PSW CORRECT
BEQ 1: ; YES GO ON
ERROR ; ALL ERRORS TO TRAP TO EMT VECTOR
.WORD 152 ; UNIQUE ERROR NUMBER
.WORD MMUERR ; ADDRESS OF ERROR MESSAGE
;NO GO TO ERROR
CLR $0177572 ; TURN MMU OFF
MOV $140000,$0177776 ; POINT TO USER SPACE
CMP R6,$0SAVUSE ; IS USER SP CORRECT
BEQ 100: ; YES GO ON
ERROR ; ALL ERRORS TO TRAP TO EMT VECTOR
.WORD 153 ; UNIQUE ERROR NUMBER
.WORD MMUERR ; ADDRESS OF ERROR MESSAGE
;NO GO TO ERROR
MOV $40000,$0177776 ; POINT TO SUPERVISOR SPACE
CMP R6,$0SAVSUP ; IS SUPERVISOR SP CORRECT
BEQ 200: ; YES GO ON
ERROR ; ALL ERRORS TO TRAP TO EMT VECTOR
.WORD 154 ; UNIQUE ERROR NUMBER
.WORD MMUERR ; ADDRESS OF ERROR MESSAGE
;NO GO TO ERROR
CMP $0244,$120413 ; IS TEST LOCATION CORRECT
BEQ 2: ; YES GO ON
ERROR ; ALL ERRORS TO TRAP TO EMT VECTOR
.WORD 155 ; UNIQUE ERROR NUMBER
.WORD MMUERR ; ADDRESS OF ERROR MESSAGE
;NO GO TO ERROR
CMP R3,$246 ; IS R3 CORRECT
BEQ 3: ; YES GO ON
ERROR ; ALL ERRORS TO TRAP TO EMT VECTOR
.WORD 156 ; UNIQUE ERROR NUMBER
.WORD MMUERR ; ADDRESS OF ERROR MESSAGE
;NO GO TO ERROR
```


Line	Address	Offset	PC	PSW	Instruction	Comment
1941	007414	005037	177776	3\$:	CLR	\$0177776
1942	007420	021627	177777		CMP	(SP), \$0177777
1943	007424	001403			BEQ	4\$
1944	007426	104000			ERROR	
1945	007430	000157			.WORD	157
1946	007432	001213			.WORD	MMUERR
1947						
1948	007434	012737	030017	177776	4\$:	MOV
1949	007442	012746	145121		MOV	\$030017, \$0177776
1950	007446	012701	000244		MOV	\$0145121, -(SP)
1951	007452	005237	177572		MOV	\$0244, R1
1952	007456	006611			INC	\$0177572
1953	007460	022737	030011	177776	MTPI	(R1)
1954	007466	001403			CMP	\$030011, \$0177776
1955	007470	104000			BEQ	300\$
1956	007472	000160			ERROR	
1957	007474	001213			.WORD	160
1958					.WORD	MMUERR
1959	007476	005037	177572			
1960	007502	023727	000244	145121	300\$:	CLR
1961	007510	001403			CMP	\$0177572
1962	007512	104000			BEQ	\$0244, \$0145121
1963	007514	000161			ERROR	301\$
1964	007516	001213			.WORD	161
1965					.WORD	MMUERR
1966	007520	020127	000244			
1967	007524	001403			301\$:	CMP
1968	007526	104000			BEQ	R1, \$0244
1969	007530	000162			ERROR	302\$
1970	007532	001213			.WORD	162
1971					.WORD	MMUERR
1972	007534	005037	177776			
1973	007540	021627	177777		302\$:	CLR
1974	007544	001403			CMP	\$0177776
1975	007546	104000			BEQ	(SP), \$0177777
1976	007550	000163			ERROR	304\$
1977	007552	001213			.WORD	163
1978					.WORD	MMUERR
1979	007554	012737	030017	177776	304\$:	MOV
1980	007562	012746	122347		MOV	\$030017, \$0177776
1981	007566	012701	000246		MOV	\$0122347, -(SP)
1982	007572	005237	177572		MOV	\$0246, R1
1983	007576	006641			INC	\$0177572
1984	007600	022737	030011	177776	MTPI	-(R1)
1985	007606	001403			CMP	\$030011, \$0177776
1986	007610	104000			BEQ	400\$
1987	007612	000164			ERROR	
1988	007614	001213			.WORD	164
1989					.WORD	MMUERR
1990	007616	005037	177572			
1991	007622	023727	000244	122347	400\$:	CLR
1992	007630	001403			CMP	\$0177572
1993	007632	104000			BEQ	\$0244, \$0122347
1994	007634	000165			ERROR	401\$
1995	007636	001213			.WORD	165
1996					.WORD	MMUERR

```
1997 007640 020127 000244      401$:  CMP      R1,0244      ;IS R1 CORRECT
1998 007644 001403              BEQ      402$      ;YES GO ON
1999 007646 104000              ERROR              ;ALL ERRORS TO TRAP TO EMT VECTOR
2000 007650 000166              .WORD    166      ;UNIQUE ERROR NUMBER
2001 007652 001213              .WORD    MMUERR    ;ADDRESS OF ERROR MESSAGE
2002                                ;NO GO TO ERROR
2003 007654 005037 177776      402$:  CLR      00177776    ;SET PSW TO KERNEL MODE
2004 007660 021627 177777      CMP      (SP),0177777    ;IS STACK CORRECT
2005 007664 001403              BEQ      404$      ;YES GO ON
2006 007666 104000              ERROR              ;ALL ERRORS TO TRAP TO EMT VECTOR
2007 007670 000167              .WORD    167      ;UNIQUE ERROR NUMBER
2008 007672 001213              .WORD    MMUERR    ;ADDRESS OF ERROR MESSAGE
2009                                ;NO GO TO ERROR
2010 007674 005046              404$:  CLR      -(SP)      ;SETUP STACK FOR NEXT TEST
2011 007676 005237 177572      INC      00177572    ;TURN MMU ON
2012 007702 006603              MTPI      R3      ;TEST INSTRUCTION
2013 007704 022737 000004 177776  CMP      04,00177776    ;IS PSW CORRECT
2014 007712 001403              BEQ      5$      ;YES GO ON
2015 007714 104000              ERROR              ;ALL ERRORS TO TRAP TO EMT VECTOR
2016 007716 000170              .WORD    170      ;UNIQUE ERROR NUMBER
2017 007720 001213              .WORD    MMUERR    ;ADDRESS OF ERROR MESSAGE
2018                                ;NO GO TO ERROR
2019 007722 005037 177572      5$:   CLR      00177572    ;TURN MMU OFF
2020 007726 020327 000000      CMP      R3,00      ;IS R3 CORRECT
2021 007732 001403              BEQ      6$      ;YES GO ON
2022 007734 104000              ERROR              ;ALL ERRORS TO TRAP TO EMT VECTOR
2023 007736 000171              .WORD    171      ;UNIQUE ERROR NUMBER
2024 007740 001213              .WORD    MMUERR    ;ADDRESS OF ERROR MESSAGE
2025                                ;NO GO TO ERROR
2026 007742 022627 177777      6$:   CMP      (SP),,0177777    ;IS STACK CORRECT
2027 007746 001403              BEQ      7$      ;YES GO ON
2028 007750 104000              ERROR              ;ALL ERRORS TO TRAP TO EMT VECTOR
2029 007752 000172              .WORD    172      ;UNIQUE ERROR NUMBER
2030 007754 001213              .WORD    MMUERR    ;ADDRESS OF ERROR MESSAGE
2031                                ;NO GO TO ERROR
2032 007756 012637 000244      7$:   MOV      (SP),,00244    ;RESTORE TEST LOCATION
2033
2034
2035 007762      ;TSMM6D:
2036      ;*****
2037      ;*TEST 12      TEST MTPD (MOVE TO PREVIOUS DATA SPACE)
2038      ;*****
2039      TST12:
2040      INC      $TESTN      ;INCREMENT TEST NUMBER
2041      CLR      00177572    ;MMU OFF
2042      CLR      00FLAG      ;CLEAR MMU ABORT FLAG
2043      MOV      0140000,00177776    ;POINT TO USER SPACE
2044      MOV      R6,00SAVUSE    ;SAVE USER SP
2045      MOV      040000,00177776    ;POINT TO SUPERVISOR SPACE
2046      MOV      R6,00SAVSUP    ;SAVE SUPERVISOR SP
2047      MOV      030000,00177776    ;SETUP PSW
2048      JSR      PC,MMU      ;INIT MMU
2049      MOV      027,00172516    ;SETUP MMR3
2050      MOV      00244,-(SP)    ;SAVE DATA AT TEST LOCATION
2051      MOV      0177777,-(SP)    ;PUT KNOWN DATA ON STACK
2052      MOV      0100004,-(SP)    ;PUT TEST DATA ON STACK
```


2053	010056	012737	177777	000244		MOV	#177777,#0244	;PUT KNOWN DATA AT TEST LOCATION
2054	010064	012767	077400	167506		MOV	#77400,UIPDR0	;SETUP UIPDR0 TO ABORT
2055	010072	012703	000244			MOV	#244,R3	;SETUP POINTER TO TEST LOCATION
2056	010076	005237	177572			INC	#0177572	;TURN MMU ON
2057	010102	106623				MTPD	(R3),	;TEST INSTRUCTION
2058	010104	022737	030010	177776		CMP	#30010,#0177776	;IS PSW CORRECT
2059	010112	001403				BEQ	1\$	;YES GO ON
2060	010114	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2061	010116	000173				.WORD	173	;UNIQUE ERROR NUMBER
2062	010120	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2063								;NO GO TO ERROR
2064	010122	005037	177572		1\$:	CLR	#0177572	;TURN MMU OFF
2065	010126	012737	140000	177776		MOV	#140000,#0177776	;POINT TO USER SPACE
2066	010134	020637	001070			CMP	R6,#0SAVUSE	;IS USER SP CORRECT
2067	010140	001403				BEQ	100\$	;YES GO ON
2068	010142	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2069	010144	000174				.WORD	174	;UNIQUE ERROR NUMBER
2070	010146	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2071								;NO GO TO ERROR
2072	010150	012737	040000	177776	100\$:	MOV	#40000,#0177776	;POINT TO SUPERVISOR SPACE
2073	010156	020637	001066			CMP	R6,#0SAVSUP	;IS SUPERVISOR SP CORRECT
2074	010162	001403				BEQ	200\$	;YES GO ON
2075	010164	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2076	010166	000175				.WORD	175	;UNIQUE ERROR NUMBER
2077	010170	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2078								;NO GO TO ERROR
2079	010172	023727	000244	100004	200\$:	CMP	#0244,#100004	;IS TEST LOCATION CORRECT
2080	010200	001403				BEQ	2\$	;YES GO ON
2081	010202	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2082	010204	000176				.WORD	176	;UNIQUE ERROR NUMBER
2083	010206	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2084								;NO GO TO ERROR
2085	010210	020327	000246		2\$:	CMP	R3,#0246	;IS R3 CORRECT
2086	010214	001403				BEQ	3\$	;YES GO ON
2087	010216	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2088	010220	000177				.WORD	177	;UNIQUE ERROR NUMBER
2089	010222	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2090								;NO GO TO ERROR
2091	010224	005037	177776		3\$:	CLR	#0177776	;SET PSW TO KERNEL MODE
2092	010230	021627	177777			CMP	(SP),#0177777	;IS KERNEL STACK CORRECT
2093	010234	001403				BEQ	4\$	;YES GO ON
2094	010236	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2095	010240	000200				.WORD	200	;UNIQUE ERROR NUMBER
2096	010242	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2097								;NO GO TO ERROR
2098	010244	012737	030017	177776	4\$:	MOV	#30017,#0177776	;SETUP PSW
2099	010252	012746	100737			MOV	#100737,-(SP)	;SETUP TEST DATA
2100	010256	012701	000244			MOV	#244,R1	;SETUP R1
2101	010262	005237	177572			INC	#0177572	;TURN MMU ON
2102	010266	106611				MTPD	(R1)	;TEST INSTRUCTION
2103	010270	022737	030011	177776		CMP	#30011,#0177776	;IS PSW CORRECT
2104	010276	001403				BEQ	300\$	;YES GO ON
2105	010300	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2106	010302	000201				.WORD	201	;UNIQUE ERROR NUMBER
2107	010							

2109	010306	005037	177572		300\$:	CLR	@#177572	;TURN MMU OFF
2110	010312	023727	000244	100737		CMP	@#244,@#100737	;IS TEST LOCATION CORRECT
2111	010320	001403				BEG	301\$	;YES GO ON
2112	010322	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2113	010324	000202				.WORD	202	;UNIQUE ERROR NUMBER
2114	010326	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2115								;NO GO TO ERROR
2116	010330	020127	000244		301\$:	CMP	R1,@#244	;IS R1 CORRECT
2117	010334	001403				BEG	302\$	;YES GO ON
2118	010336	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2119	010340	000203				.WORD	203	;UNIQUE ERROR NUMBER
2120	010342	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2121								;NO GO TO ERROR
2122	010344	005037	177776		302\$:	CLR	@#177776	;SET PSW TO KERNEL MODE
2123	010350	021627	177777			CMP	(SP),@#177777	;IS STACK CORRECT
2124	010354	001403				BEG	304\$	;YES GO ON
2125	010356	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2126	010360	000204				.WORD	204	;UNIQUE ERROR NUMBER
2127	010362	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2128								;NO GO TO ERROR
2129	010364	012737	030017	177776	304\$:	MOV	@#30017,@#177776	;SETUP PSW
2130	010372	012746	156711			MOV	@#156711,-(SP)	;SETUP TEST DATA
2131	010376	012701	000246			MOV	@#246,R1	;SETUP R1
2132	010402	005237	177572			INC	@#177572	;TURN MMU ON
2133	010406	106641				MTPD	-(R1)	;TEST INSTRUCTION
2134	010410	022737	030011	177776		CMP	@#30011,@#177776	;IS PSW CORRECT
2135	010416	001403				BEG	400\$	;YES GO ON
2136	010420	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2137	010422	000205				.WORD	205	;UNIQUE ERROR NUMBER
2138	010424	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2139								;NO GO TO ERROR
2140	010426	005037	177572		400\$:	CLR	@#177572	;TURN MMU OFF
2141	010432	023727	000244	156711		CMP	@#244,@#156711	;IS TEST LOCATION CORRECT
2142	010440	001403				BEG	401\$	;YES GO ON
2143	010442	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2144	010444	000206				.WORD	206	;UNIQUE ERROR NUMBER
2145	010446	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2146								;NO GO TO ERROR
2147	010450	020127	000244		401\$:	CMP	R1,@#244	;IS R1 CORRECT
2148	010454	001403				BEG	402\$	;YES GO ON
2149	010456	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2150	010460	000207				.WORD	207	;UNIQUE ERROR NUMBER
2151	010462	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2152								;NO GO TO ERROR
2153	010464	005037	177776		402\$:	CLR	@#177776	;SET PSW TO KERNEL MODE
2154	010470	021627	177777			CMP	(SP),@#177777	;IS STACK CORRECT
2155	010474	001403				BEG	404\$	;YES GO ON
2156	010476	104000				ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
2157	010500	000210				.WORD	210	;UNIQUE ERROR NUMBER
2158	010502	001213				.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
2159								;NO GO TO ERROR
2160	010504	005046			404\$:	CLR	-(SP)	;SETUP STACK FOR NEXT TEST
2161	010506	005237	177572			INC	@#177572	;TURN MMU ON
2162	010512	106603				MTPD	R3	;TEST INSTRUCTION
2163	010514	022737	000004	177776		CMP	@#4,@#177776	;IS PSW CORRECT


```
2165 010524 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
2166 010526 000211 .WORD 211 ;UNIQUE ERROR NUMBER
2167 010530 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2168 ;NO GO TO ERROR
2169 010532 005037 177572 5$: CLR @#177572 ;TURN MMU OFF
2170 010536 020327 000000 CMP R3,#0 ;IS R3 CORRECT
2171 010542 001403 BEQ 6$ ;YES GO ON
2172 010544 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
2173 010546 000212 .WORD 212 ;UNIQUE ERROR NUMBER
2174 010550 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2175 ;NO GO TO ERROR
2176 010552 022627 177777 6$: CMP (SP)+,#177777 ;IS STACK CORRECT
2177 010556 001403 BEQ 7$ ;YES GO ON
2178 010560 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
2179 010562 000213 .WORD 213 ;UNIQUE ERROR NUMBER
2180 010564 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2181 ;NO GO TO ERROR
2182 010566 012637 000244 7$: MOV (SP)+,@#244 ;RESTORE TEST LOCATION
2183
2184 ;
2185 010572 ;TSMMU7:
2186 ;*****
2187 ;*TEST 13 TEST NON-RESIDENT ABORT
2188 ;*****
2189 ;TST13:
2190 010572 005267 170206 INC $TESTN ;INCREMENT TEST NUMBER
2191 010576 005037 177572 CLR @#177572 ;MMU OFF
2192 010602 005067 170234 CLR FLAG ;CLEAR MMU ABORT FLAG
2193 010606 013746 000214 MOV @#214,-(SP) ;SAVE DATA AT TEST LOCATIONS
2194 010612 013746 000216 MOV @#216,-(SP) ;
2195 010616 005067 170250 CLR SAVMR0 ;CLEAR STATUS REGS SAVE AREAS
2196 010622 005067 170246 CLR SAVMR1 ;
2197 010626 005067 170244 CLR SAVMR2 ;
2198 010632 004767 170450 JSR PC,MMU ;INIT MMU
2199 010636 012737 030000 177776 MOV @#30000,@#177776 ;SETUP PSW
2200 010644 012702 000200 MOV @#200,R2 ;
2201 010650 012737 077400 177600 MOV @#77400,@#177600 ;SETUP FOR AN ABORT
2202 010656 004767 000164 JSR PC,TS7 ;CAUSE AN ABORT TO OCCUR AND
2203 ;THEN CHECK IF ABORT FLAG REGISTERED
2204 ;THIS EVENT AND CHECK IF STATUS REGS
2205 ;CONTAINED EXPECTED VALUES.
2206 ;IF NO ABORT OCCURRED THEN GO TO ERROR
2207 ;OTHERWISE CONTINUE.
2208 010662 012737 077404 177600 MOV @#77404,@#177600 ;SETUP FOR AN ABORT
2209 010670 004767 000152 JSR PC,TS7 ;CAUSE AN ABORT TO OCCUR AND
2210 ;THEN CHECK IF ABORT FLAG REGISTERED
2211 ;THIS EVENT AND CHECK IF STATUS REGS
2212 ;CONTAINED EXPECTED VALUES.
2213 ;IF NO ABORT OCCURRED THEN GO TO ERROR
2214 ;OTHERWISE CONTINUE.
2215 010674 012701 000220 MOV @#220,R1 ;
2216 010700 004767 170402 JSR PC,MMU ;INIT MMU
2217 010704 005003 CLR R3 ;SETUP MMR1 EXPECTED DATA
2218 010706 012767 000001 170126 MOV @#1,FLAG ;SETUP FLAG FOR AN ABORT
2219 010714 012737 000001 177572 MOV @#1,@#177572 ;TURN MMU ON
2220 010722 012737 100000 177776 MOV @#100000,@#177776 ;SETUP PSW FOR AN ABORT (ILLEGAL MODE)
```

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
---------	---------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------	--------


```

2277 011156 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2278 ;NOT OK THEN GO TO ERROR
2279 011160 005067 167706 OKAY7A: CLR SAVMRO ;CLEAR STATUS REGS SAVE AREAS
2280 011164 005067 167704 CLR SAVMR1 ;
2281 011170 005067 167702 CLR SAVMR2 ;
2282 011174 000207 RTS PC ;RETURN
2283 ;
2284 ;ROUTINE TO CHECK IF A NONRESIDENT ABORT OCCURRED
2285 ;
2286 011176 022767 000000 167636 TSM7: CMP #0,FLAG ;DID AN ABORT OCCUR
2287 011204 001403 BEQ TSMA ;IF YES GO ON
2288 011206 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
2289 011210 000220 .WORD 220 ;UNIQUE ERROR NUMBER
2290 011212 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2291 ;IF NO THEN GO TO ERROR
2292 011214 042737 040377 001072 TSMA: BIC #40377,#SAVMRO ;SETUP EXPECTED DATA
2293 011222 022767 100000 167642 CMP #100000,SAVMRO ;TEST MMRO FOR EXPECTED VALUE
2294 011230 001403 BEQ TSMB ;IF OK THEN CONTINUE
2295 011232 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
2296 011234 000221 .WORD 221 ;UNIQUE ERROR NUMBER
2297 011236 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2298 ;IF NO THEN GO TO ERROR
2299 011240 020367 167630 TSMB: CMP R3,SAVMR1 ;TEST MMR1 FOR EXPECTED VALUE
2300 011244 001403 BEQ TSMC ;IF OK THEN CONTINUE
2301 011246 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
2302 011250 000222 .WORD 222 ;UNIQUE ERROR NUMBER
2303 011252 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2304 ;IF NOT OK THEN GO TO ERROR
2305 011254 000207 TSMC: RTS PC ;RETURN
2306 ;
2307 011256 TS7FIN:
2308 011256 TSMMU8:
2309 ;*****
2310 ;*TEST 14 TEST READ ONLY ABORTS
2311 ;*****
2312 TST14:
2313 011256 005267 167522 INC #TESTN ;INCREMENT TEST NUMBER
2314 011262 005037 177572 CLR #177572 ;MMU OFF
2315 011266 005067 167550 CLR FLAG ;CLEAR MMU ABORT FLAG
2316 011272 013746 000244 MOV #244,-(SP) ;SAVE DATA AT TEST LOCATIONS
2317 011276 013746 000246 MOV #246,-(SP) ;
2318 011302 005067 167564 CLR SAVMRO ;CLEAR STATUS REGS SAVE AREAS
2319 011306 005067 167562 CLR SAVMR1 ;
2320 011312 005067 167560 CLR SAVMR2 ;
2321 011316 004767 167764 JSR PC,MMU ;INIT MMU
2322 011322 012737 030000 177776 MOV #30000,#177776 ;SETUP PSW
2323 011330 012702 000244 MOV #244,R2 ;
2324 011334 012737 077402 177600 MOV #77402,#177600 ;SETUP FOR AN ABORT
2325 011342 012746 000246 MOV #246,-(SP) ;PUSH DATA ONTO THE STACK
2326 011346 012767 000001 167466 MOV #1,FLAG ;SETUP FLAG FOR AN ABORT
2327 011354 012737 000001 177572 MOV #1,#177572 ;TURN MMU ON
2328 011362 010701 MOV R7,R1 ;SAVE PC
2329 011364 006622 MTPI (R2). ;CAUSE ABORT
2330 011366 022767 000000 167446 CMP #0,FLAG ;DID ABORT OCCUR
2331 011374 001403 BEQ 1 ;IF YES THEN GO ON
2332 ;IF NO THEN GO TO ERROR

```

```

ERROR
.WORD      223
.WORD      MMUERR
CLRB       SAVMRO
CMP        #20000,SAVMRO
BEQ        2$

ERROR
.WORD      224
.WORD      MMUERR
CMP        #11026,SAVMR1
BEQ        3$

ERROR
.WORD      225
.WORD      MMUERR
CMP        R1,SAVMR2
BEQ        4$

ERROR
.WORD      226
.WORD      MMUERR
CMP        #244,2(SP)
BEQ        5$

ERROR
.WORD      227
.WORD      MMUERR
MOV        #30000,#177776
MOV        #2,-(SP)
MTPI       (R2)+
MOV        (SP)+,#246
MOV        (SP)+,#244

```

```

;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;SETUP EXPECTED DATA
; TEST MMRO FOR EXPECTED VALUE
;IF OK THEN CONTINUE
;OTHERWISE GO TO ERROR
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
; TEST MMR1 FOR EXPECTED VALUE
;IF OK THEN CONTINUE
;OTHERWISE GO TO ERROR
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
; TEST MMR2 FOR EXPECTED VALUE
;IF OK THEN CONTINUE
;OTHERWISE GO TO ERROR
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;CHECK THAT ABORT BLOCKED WRITE
;BRANCH IF CURRENT CONTENTS= SAVED
;ELSE
;ERROR! ABORT DIDN'T BLOCK ACCESS
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;SETUP PSW
;PUSH DATA ONTO STACK
;TRY TO CAUSE ABORT
;RESTORE DATA AT TEST LOCATIONS
;

```

TSMMU9:

```

*****
;*TEST 15      TEST PAGE LENGTH ERROR ABORTS
*****

```

IST15:

```

INC      $TESTN
CLR      @0177572
CLR      FLAG
CLR      SAVMR0
CLR      SAVMR1
CLR      SAVMR2
MOV      @30000,@0177776
JSR      PC,MMU
MOV      @PLF0,R3
MOV      @BNO,R1
MOV      @ABORT0,R2
MOV      @26,@0172516
JSR      PC,TSM9

```

```

; INCREMENT TEST NUMBER
; MMU OFF
; CLEAR MMU ABORT FLAG
; CLEAR STATUS REGS SAVE AREAS
;
;
; SETUP PSW
; INIT MMU
; LET R3, R1, AND R2 POINT TO THE
; UPWARD EXPANSION TABLES
;
; DISABLE USER DATA SPACE
; TURN MMU ON
; DO RELOCATIONS FOR THE DIFFERENT
; VALUES OF THE PAGE LENGTH FIELD AND
; BLOCK NUMBER.  IF AN ABORT OCCURS

```



```
2389
2390
2391
2392 011612 012703 012264
2393 011616 012701 012334
2394 011622 012702 012402
2395 011626 004767 000030
2396
2397
2398
2399
2400
2401
2402 011632 005037 177572
2403 011636 012703 012274
2404 011642 012701 012344
2405 011646 011337 177600
2406 011652 006521
2407 011654 012605
2408
2409 011656 000167 000566
2410
2411
2412
2413 011662 012337 177600
2414 011666 010100
2415 011670 012767 000001 167144
2416 011676 012737 000001 177572
2417 011704 010704
2418 011706 006530
2419 011710 021227 000000
2420 011714 001011
2421 011716 012605
2422 011720 022767 000001 167114
2423 011726 001403
2424 011730 104000
2425 011732 000230
2426 011734 001213
2427
2428 011736 000435
2429 011740 022767 000000 167074
2430 011746 001403
2431 011750 104000
2432 011752 000231
2433 011754 001213
2434
2435 011756 105067 167110
2436 011762 022767 040000 167102
2437 011770 001403
2438 011772 104000
2439 011774 000232
2440 011776 001213
2441
2442 012000 022767 000020 167066
2443 012006 001403
2444 012010 104000

MOV #PLF1,R3
MOV #BN1,R1
MOV #ABORT1,R2
JSR PC,TSM9

CLR @#177572
MOV #PLF1+10,R3
MOV #BN1+10,R1
MOV (R3),@#177600
MFPI (R1)+
MOV (SP)+,R5

JMP TS9FIN

;ROUTINE TO CAUSE AND CHECK PAGE LENGTH ERROR ABORTS
TSM9: MOV (R3)+,@#177600
MOV R1,R0
MOV #1,FLAG
MOV #1,@#177572
MOV R7,R4
MFPI @R0+
CMP (R2),#0
BNE 2$
MOV (SP)+,R5
CMP #1,FLAG
BEQ 1$
ERROR
.WORD 230
.WORD MMUERR
;YES GO TO ERROR
BR 6$
CMP #0,FLAG
BEQ 3$
ERROR
.WORD 231
.WORD MMUERR
;NO GO TO ERROR
CLRB SAVMR0
CMP #40000,SAVMR0
BEQ 4$
ERROR
.WORD 232
.WORD MMUERR
;NOT OK THEN GO TO ERROR
CMP #20,SAVMR1
BEQ 5$
ERROR
```

;CHECK TO SEE IF IT WAS SUPPOSED TO,
;AND IF YES CHECK ABORT FLAG AND
;STATUS REGISTERS.
;LET R3, R1, AND R2 POINT TO THE
;DOWNWARD EXPANSION TABLES
;
;TURN MMU ON
;DO RELOCATIONS FOR THE DIFFERENT
;VALUES OF THE PAGE LENGTH FIELD AND
;BLOCK NUMBER. IF AN ABORT OCCURS
;CHECK TO SEE IF IT WAS SUPPOSED TO,
;AND IF YES CHECK ABORT FLAG AND
;STATUS REGISTERS.
;MMU OFF
;POINT TO A VALUE WHICH SHOULD CAUSE
;AN ABORT IF MMU IS ON.
;SETUP UIPDRO
;DO A RELOCATION
;POP THE STACK
;
;SETUP UIPDRO
;SAVE A COPY OF R1
;SETUP FOR AN ABORT
;TURN MMU ON
;SAVE PC
;DO A RELOCATION OPERATION
;WAS AN ABORT SUPPOSED TO OCCUR
;IF YES GO TO 2\$
;POP THE STACK
;DID AN ABORT OCCUR
;NO GO ON
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;
;DID AN ABORT OCCUR
;YES GO ON
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;
;SETUP EXPECTED DATA
;TEST MMRO FOR EXPECTED VALUE
;IF OK THEN CONTINUE
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;THEN GO TO ERROR
;TEST MMR1 FOR EXPECTED VALUE
;IF OK THEN CONTINUE
;ALL ERRORS TO TRAP TO EMT VECTOR

2445	012012	000233			.WORD	233				
2446	012014	001213			.WORD	MMUERR				
2447										
2448	012016	020467	167054	5\$:	CMP	R4,SAVMR2			:NOT OK	
2449	012022	001403			BEQ	6\$				
2450	012024	104000			ERROR					
2451	012026	000234			.WORD	234				
2452	012030	001213			.WORD	MMUERR				
2453										
2454	012032	005067	167004	6\$:	CLR	FLAG			:NOT OK	
2455	012036	005067	167030		CLR	SAVMR0				
2456	012042	005067	167026		CLR	SAVMR1				
2457	012046	005067	167024		CLR	SAVMR2				
2458	012052	005201			INC	R1				
2459	012054	005201			INC	R1				
2460	012056	005202			INC	R2				
2461	012060	005202			INC	R2				
2462	012062	021327	000777		CMP	(R3),#777				
2463	012066	001275			BNE	TSM9				
2464	012070	000207			RTS	PC				
2465										
2466										
2467										
2468	012072	070006								
2469	012074	070006								
2470	012076	070006								
2471	012100	013406								
2472	012102	020006								
2473	012104	004006								
2474	012106	040006								
2475	012110	070006								
2476	012112	024006								
2477	012114	004006								
2478	012116	014006								
2479	012120	012006								
2480	012122	002006								
2481	012124	001406								
2482	012126	004006								
2483	012130	002006								
2484	012132	000406								
2485	012134	007406								
2486	012136	001006								
2487	012140	003406								
2488	012142	000777								
2489	012144	013000								
2490	012146	016000								
2491	012150	017000								
2492	012152	002700								
2493	012154	014000								
2494	012156	002000								
2495	012160	004000								
2496	012162	007000								
2497	012164	002000								
2498	012166	000700								
2499	012170	004000								
2500	012172	001000								

:UPWARD EXPANSION TABLES									
:PLF0:									
		.WORD	70006						
		.WORD	70006						
		.WORD	70006						
		.WORD	13406						
		.WORD	20006						
		.WORD	04006						
		.WORD	40006						
		.WORD	70006						
		.WORD	24006						
		.WORD	04006						
		.WORD	14006						
		.WORD	12006						
		.WORD	02006						
		.WORD	01406						
		.WORD	04006						
		.WORD	02006						
		.WORD	00406						
		.WORD	07406						
		.WORD	01006						
		.WORD	03406						
		.WORD	777						
:BNO:									
		.WORD	013000						
		.WORD	016000						
		.WORD	017000						
		.WORD	002700						
		.WORD	014000						
		.WORD	002000						
		.WORD	004000						
		.WORD	007000						
		.WORD	002000						
		.WORD	000700						
		.WORD	004000						
		.WORD	001000						

```

:UNIQUE ERROR NUMBER
:ADDRESS OF ERROR MESSAGE
:NOT OK THEN GO TO ERROR
:TEST MMR2 FOR EXPECTED VALUE
:IF OK THEN CONTINUE
:ALL ERRORS TO TRAP TO EMT VECTOR
:UNIQUE ERROR NUMBER
:ADDRESS OF ERROR MESSAGE
:NOT OK THEN GO TO ERROR
:CLEAR MMU ABORT FLAG
:CLEAR STATUS REGS SAVE AREAS
:
:
:POINT TO NEXT ENTRY
:
:
:
:HAVE ALL ENTRIES BEEN TRIED
:NO REPEAT
:YES RETURN

```


2501	012174	000300	.WORD	000300
2502	012176	000400	.WORD	000400
2503	012200	001400	.WORD	001400
2504	012202	000600	.WORD	000600
2505	012204	000200	.WORD	000200
2506	012206	001700	.WORD	001700
2507	012210	000300	.WORD	000300
2508	012212	000700	.WORD	000700
2509	012214	000000	.WORD	0
2510	012216	000000	.WORD	0
2511	012220	000001	.WORD	1
2512	012222	000000	.WORD	0
2513	012224	000001	.WORD	1
2514	012226	000001	.WORD	1
2515	012230	000000	.WORD	0
2516	012232	000000	.WORD	0
2517	012234	000000	.WORD	0
2518	012236	000000	.WORD	0
2519	012240	000001	.WORD	1
2520	012242	000000	.WORD	0
2521	012244	000000	.WORD	0
2522	012246	000001	.WORD	1
2523	012250	000001	.WORD	1
2524	012252	000001	.WORD	1
2525	012254	000001	.WORD	1
2526	012256	000000	.WORD	0
2527	012260	000001	.WORD	1
2528	012262	000000	.WORD	0

ABORTO:

;DOWNWARD EXPANSION TABLES

2530				
2531				
2532	012264	000416	.WORD	00416
2533	012266	020016	.WORD	20016
2534	012270	024016	.WORD	24016
2535	012272	034016	.WORD	34016
2536	012274	074016	.WORD	74016
2537	012276	040016	.WORD	40016
2538	012300	020016	.WORD	20016
2539	012302	000016	.WORD	00016
2540	012304	030016	.WORD	30016
2541	012306	010016	.WORD	10016
2542	012310	014016	.WORD	14016
2543	012312	004016	.WORD	04016
2544	012314	002016	.WORD	02016
2545	012316	000416	.WORD	00416
2546	012320	000016	.WORD	00016
2547	012322	003416	.WORD	03416
2548	012324	001016	.WORD	01016
2549	012326	001416	.WORD	01416
2550	012330	000416	.WORD	00416
2551	012332	000777	.WORD	777
2552	012334	000100	.WORD	000100
2553	012336	010000	.WORD	010000
2554	012340	006000	.WORD	006000
2555	012342	016000	.WORD	016000
2556	012344	016000	.WORD	016000

BN1:

2557 012346 004000
 2558 012350 000000
 2559 012352 000000
 2560 012354 004000
 2561 012356 004000
 2562 012360 004000
 2563 012362 000000
 2564 012364 000300
 2565 012366 000000
 2566 012370 000400
 2567 012372 001000
 2568 012374 000100
 2569 012376 000400
 2570 012400 000200
 2571 012402 000000
 2572 012404 000000
 2573 012406 000000
 2574 012410 000000
 2575 012412 000001
 2576 012414 000001
 2577 012416 000001
 2578 012420 000000
 2579 012422 000001
 2580 012424 000000
 2581 012426 000000
 2582 012430 000001
 2583 012432 000001
 2584 012434 000001
 2585 012436 000000
 2586 012440 000000
 2587 012442 000001
 2588 012444 000000
 2589 012446 000000
 2590
 2591 012450
 2592 012450
 2593
 2594
 2595
 2596 012450
 2597 012450 005267 166330
 2598 012454 005037 177572
 2599 012460 005067 166356
 2600 012464 005067 166402
 2601 012470 005067 166400
 2602 012474 005067 166376
 2603 012500 004767 166602
 2604 012504 005037 177776
 2605 012510 012702 020200
 2606 012514 012737 077400 172302
 2607 012522 012767 000001 166312
 2608 012530 012737 000001 177572
 2609 012536 010701
 2610 012540 006522
 2611 012542 012704 100003
 2612 012546 004767 000210

ABORT1:

.WORD 004000
 .WORD 000000
 .WORD 000000
 .WORD 004000
 .WORD 004000
 .WORD 004000
 .WORD 000000
 .WORD 000300
 .WORD 000000
 .WORD 000400
 .WORD 001000
 .WORD 000100
 .WORD 000400
 .WORD 000200
 .WORD 0
 .WORD 0
 .WORD 0
 .WORD 0
 .WORD 1
 .WORD 1
 .WORD 1
 .WORD 0
 .WORD 1
 .WORD 0
 .WORD 0
 .WORD 1
 .WORD 1
 .WORD 1
 .WORD 0
 .WORD 0
 .WORD 1
 .WORD 0
 .WORD 0

;
 TS9FIN:
 TSMM10:

;*****
 ;*TEST 16 FUNCTIONAL TEST OF BITS <6:1> OF MMRO
 ;*****
 TST16:

INC \$TESTN ; INCREMENT TEST NUMBER
 CLR @#177572 ; MMU OFF
 CLR FLAG ; CLEAR MMU ABORT FLAG
 CLR SAVMR0 ; CLEAR STATUS REGS SAVE AREAS
 CLR SAVMR1 ;
 CLR SAVMR2 ;
 JSR PC,MMU ; INIT MMU
 CLR @#177776 ; INIT PSW: PREVIOUS MODE = KERNAL
 MOV @20200,R2 ;
 MOV @77400,@#172302 ; SETUP KIPDR1 TO ABORT
 MOV #1,FLAG ; SETUP FLAG FOR AN ABORT
 MOV #1,@#177572 ; TURN MMU ON
 MOV R7,R1 ; SAVE PC
 MFPI (R2)+ ; DO A RELOCATION VIA KIPAR1
 MOV @100003,R4 ; SETUP EXPECTED DATA
 JSR PC,TS10 ; CHECK IF AN ABORT OCCURRED AND


```
2613
2614 012552 012737 030000 177776      MOV    #30000,0#177776      ;IF YES CHECK BITS <6:1> OF MMRO.
2615 012560 004767 166522              JSR    PC,MMU              ;INIT PSW: PREVIOUS MODE = USER
2616 012564 012737 077400 177636      MOV    #77400,0#177636    ;INIT MMU
2617 012572 012737 001600 177676      MOV    #1600,0#177676    ;SETUP UDPDR7 TO ABORT
2618                                ;SET UDPAR7 TO RELOCATE TO PHYSICAL
2619 012600 012702 160000              MOV    #160000,R2          ;ADDRESSES 160000-177776
2620 012604 012767 000001 166230      MOV    #1,FLAG          ;
2621 012612 012737 000001 177572      MOV    #1,0#177572      ;SETUP FLAG FOR AN ABORT
2622 012620 010701                    MOV    R7,R1              ;TURN MMU ON
2623 012622 106522                    MFPD   (R2),              ;SAVE PC
2624 012624 012704 100177            MOV    #100177,R4          ;DO A RELOCATION VIA UDPAR7
2625 012630 004767 000126            JSR    PC,TS10          ;SETUP EXPECTED DATA
2626                                ;CHECK IF AN ABORT OCCURRED AND
2627 012634 012737 010000 177776      MOV    #10000,0#177776    ;IF YES CHECK BITS <6:1> OF MMRO.
2628 012642 004767 166440              JSR    PC,MMU              ;INIT PSW: PREVIOUS MODE= SUPERVISOR
2629 012646 012737 077400 172212      MOV    #77400,0#172212    ;INIT MMU
2630 012654 012702 120000            MOV    #120000,R2          ;SETUP SIPDR5 TO ABORT
2631 012660 012767 000001 166154      MOV    #1,FLAG          ;ACCESS PAGE 05
2632 012666 012737 000001 177572      MOV    #1,0#177572      ;SETUP FLAG FOR AN ABORT
2633 012674 010701                    MOV    R7,R1              ;TURN MMU ON
2634 012676 006522                    MFPD   (R2),              ;SAVE PC
2635 012700 012704 100053            MOV    #100053,R4          ;DO A RELOCATION VIA SIPAR5
2636 012704 004767 000052            JSR    PC,TS10          ;SETUP EXPECTED DATA:ABORT, PAGE 05
2637                                ;CHECK IF AN ABORT OCCURRED AND
2638                                ;IF YES CHECK BITS <6:1> OF MMRO.
2639
2640                                ;TEST THAT ILLEGAL MODE CAUSES MMU ABORT
2641                                ;
2641 012710 012737 020000 177776      MOV    #20000,0#177776    ;INIT PSW:SET ILLEGAL PREVIOUS MODE
2642 012716 004767 166364              JSR    PC,MMU              ;INIT MMU
2643 012722 012702 040000            MOV    #40000,R2          ;SET UP ACCESS TO PAGE 2
2644 012726 012767 000001 166106      MOV    #1,FLAG          ;SETUP FLAG FOR AN ABORT
2645 012734 012737 000001 177572      MOV    #1,0#177572      ;TURN MMU ON
2646 012742 010701                    MOV    R7,R1              ;SAVE PC
2647 012744 106522                    MFPD   (R2),              ;DO A RELOCATION
2648 012746 012704 100105            MOV    #100105,R4          ;SETUP EXPECTED DATA:ABORT, ILLEGAL
2649                                ;PROCESSOR MODE, PAGE 02
2650 012752 004767 000004              JSR    PC,TS10          ;CHECK IF AN ABORT OCCURRED AND
2651                                ;IF YES CHECK BITS <6:1> OF MMRO.
2652
2653 012756 000167 000102              JMP     T10FIN
2654
2655                                ;ROUTINE TO CHECK IF A MMU ABORT OCCURRED AND IF STATUS REG MMRO
2656                                ;CONTAINS EXPECTED DATA
2657                                ;
2658 012762 022767 000000 166052      TS10:  CMP    #0,FLAG          ;DID AN ABORT OCCUR
2659 012770 001403                    BEQ     1#              ;YES GO ON
2660 012772 104000                    ERROR   235             ;ALL ERRORS TO TRAP TO EMT VECTOR
2661 012774 000235                    .WORD   MMUERR          ;UNIQUE ERROR NUMBER
2662 012776 001213                    .WORD   MMUERR          ;ADDRESS OF ERROR MESSAGE
2663                                ;NO GO TO ERROR
2664 013000 020467 166066              1#:  CMP    R4,SAVMRO        ;TEST MMRO FOR EXPECTED DATA
2665 013004 001403                    BEQ     2#              ;OK GO ON
2666 013006 104000                    ERROR   236             ;ALL ERRORS TO TRAP TO EMT VECTOR
2667 013010 000236                    .WORD   MMUERR          ;UNIQUE ERROR NUMBER
2668 013012 001213                    .WORD   MMUERR          ;ADDRESS OF ERROR MESSAGE
```

```
2669
2670 013014 022767 000022 166052 2$: CMP      #22,SAVMR1      ;NO GO TO ERROR
2671 013022 001403          BEQ      3$          ; TEST MMR1 FOR EXPECTED DATA
2672 013024 104000          ERROR          ;OK GO ON
2673 013026 000237          .WORD      237      ;ALL ERRORS TO TRAP TO EMT VECTOR
2674 013030 001213          .WORD      MMUERR    ;UNIQUE ERROR NUMBER
2675                                     ;ADDRESS OF ERROR MESSAGE
2676 013032 020167 166040 3$: CMP      R1,SAVMR2      ;NO GO TO ERROR
2677 013036 001403          BEQ      4$          ; TEST MMR2 FOR EXPECTED DATA
2678 013040 104000          ERROR          ;OK GO ON
2679 013042 000240          .WORD      240      ;ALL ERRORS TO TRAP TO EMT VECTOR
2680 013044 001213          .WORD      MMUERR    ;UNIQUE ERROR NUMBER
2681                                     ;ADDRESS OF ERROR MESSAGE
2682 013046 005067 166020 4$: CLR      SAVMR0      ;NO GO TO ERROR
2683 013052 005067 166016          CLR      SAVMR1      ;CLEAR MMU STATUS REGS SAVE AREAS
2684 013056 005067 166014          CLR      SAVMR2      ;
2685 013062 000207          RTS      PC          ;RETURN
2686
2687 013064      ;
2688 013064      T10FIN:
2689      TSMM11:
2690      ;*****
2691      ;*TEST 17      TEST DATA SPACE BITS MMR3
2692      ;*****
2693      TST17:
2694      INC      $TESTN          ;INCREMENT TEST NUMBER
2695      CLR      #0177572        ;MMU OFF
2696      CLR      FLAG          ;CLEAR MMU ABORT FLAG
2697      MOV      #30000,#0177776 ;SETUP PSW
2698      MOV      #26,R1          ;SETUP FIRST MMR3 VALUE
2699      MOV      #177610,R3      ;POINT TO UIPDR4
2700      JSR      PC,TS11        ;SETUP SECOND MMR3 VALUE
2701      MOV      #0,#0177776    ; TEST ENABLE USER DATA SPACE BIT
2702      MOV      #23,R1          ;SETUP PSW
2703      MOV      #172310,R3      ;SETUP FIRST MMR3 VALUE
2704      MOV      #24,R4          ;POINT TO KIPDR4
2705      JSR      PC,TS11        ;SETUP SECOND MMR3 VALUE
2706      MOV      #10000,#0177776 ; TEST ENABLE KERNEL DATA SPACE BIT
2707      MOV      #25,R1          ;SETUP PSW
2708      MOV      #172210,R3      ;SETUP FIRST MMR3 VALUE
2709      MOV      #22,R4          ;POINT TO SIPDR4
2710      JSR      PC,TS11        ;SETUP SECOND MMR3 VALUE
2711                                     ; TEST ENABLE SUPERVISOR DATA SPACE BIT
2712      JMP      T11FIN
2713
2714      ;
2715      ;ROUTINE TO TEST ENABLE DATA SPACE BITS OF MMR3
2716      ;
2717      TS11: JSR      PC,MMU      ;INIT MMU
2718      MOV      R1,#0172516      ;DISABLE DATA SPACE OF MODE UNDER TEST
2719      MOV      #77400,(R3)      ;SETUP IPDR TO ABORT
2720      MOV      #100000,R2      ;
2721      MOV      #1,FLAG          ;SETUP FLAG FOR AN ABORT
2722      MOV      #1,#0177572      ;MMU ON
2723      MFPD      (R2).          ;DO A RELOCATION
2724      CMP      #0,FLAG          ;DID AN ABORT OCCUR
2725      BEQ      1$              ;YES GO ON
```


2725	013254	104000			ERROR			
2726	013256	000241			.WORD	241		;ALL ERRORS TO TRAP TO EMT VECTOR
2727	013260	001213			.WORD	MMUERR		;UNIQUE ERROR NUMBER
2728								;ADDRESS OF ERROR MESSAGE
2729	013262	010437	172516	1\$:	MOV	R4,0#172516		;NO GO TO ERROR
2730	013266	012702	100000		MOV	0#100000,R2		;ENABLE DATA SPACE OF MODE UNDER TEST
2731	013272	012767	000001	165542	MOV	0#1,FLAG		
2732	013300	012737	000001	177572	MOV	0#1,0#177572		;SETUP FLAG FOR AN ABORT
2733	013306	106522			MFPD	(R2),		;MMU ON
2734	013310	005726			TST	(SP),		;DO A RELOCATION
2735	013312	022767	000001	165522	CMP	0#1,FLAG		;POP THE STACK
2736	013320	001403			BEQ	2\$		;DID AN ABORT OCCUR
2737	013322	104000			ERROR			;NO GO ON
2738	013324	000242			.WORD	242		;ALL ERRORS TO TRAP TO EMT VECTOR
2739	013326	001213			.WORD	MMUERR		;UNIQUE ERROR NUMBER
2740								;ADDRESS OF ERROR MESSAGE
2741	013330	005067	165506	2\$:	CLR	FLAG		;YES GO TO ERROR
2742	013334	000207			RTS	PC		;CLEAR MMU ABORT FLAG
2743								;RETURN
2744	013336							
2745	013336				T11FIN:			
2746					TSM12:			
2747					*****			
2748					;*TEST 20	MMR1 FUNCTIONAL TEST		
2749	013336				*****			
2750	013336	005267	165442		TST20:			
2751	013342	005037	177572		INC	\$TESTN		;INCREMENT TEST NUMBER
2752	013346	005067	165470		CLR	0#177572		;MMU OFF
2753	013352	005067	165516		CLR	FLAG		;CLEAR MMU ABORT FLAG
2754	013356	004767	165724		CLR	SAVMR1		;CLEAR STATUS REG SAVE AREA
2755	013362	012737	030000	177776	JSR	PC,MMU		;INIT MMU
2756	013370	012704	100200		MOV	0#30000,0#177776		;INIT PSW
2757	013374	010401			MOV	0#100200,R4		;SETUP TEST LOCATIONS
2758	013376	012705	100101		MOV	R4,R1		
2759	013402	010502			MOV	0#100101,R5		
2760	013404	012737	000020	172516	MOV	R5,R2		
2761	013412	012737	077402	172310	MOV	0#20,0#172516		;INIT MMR3
2762	013420	012703	006414		MOV	0#77402,0#172310		;SETUP KIPDR4 TO ABORT
2763	013424	012767	000001	165410	MOV	0#6414,R3		;SETUP EXPECTED DATA FOR MMR1
2764	013432	012737	000001	177572	MOV	0#1,FLAG		;SETUP FLAG FOR AN ABORT
2765	013440	010767	165364		MOV	0#1,0#177572		;TURN MMU ON
2766	013444	112425			MOV	R7,SLOC00		;SAVE PC
2767	013446	004767	000206		MOVB	(R4),-(R5),		;DO A RELOCATION
2768					JSR	PC,TS12		;CHECK IF AN ABORT OCCURRED AND IF
2769	013452	012703	175011					;YES IF MMR1 EQUALS EXPECTED DATA
2770	013456	012767	000001	165356	MOV	0#175011,R3		;SETUP EXPECTED DATA FOR MMR1
2771	013464	012737	000001	177572	MOV	0#1,FLAG		;SETUP FLAG FOR AN ABORT
2772	013472	010767	165332		MOV	0#1,0#177572		;TURN MMU ON
2773	013476	112142			MOV	R7,SLOC00		;SAVE PC
2774	013500	004767	000154		MOVB	(R1),-(R2),		;DO A RELOCATION
2775					JSR	PC,TS12		;CHECK IF AN ABORT OCCURRED AND IF
2776	013504	012703	006771					;YES IF MMR1 EQUALS EXPECTED DATA
2777	013510	012767	000001	165324	MOV	0#6771,R3		;SETUP EXPECTED DATA FOR MMR1
2778	013516	012737	000001	177572	MOV	0#1,FLAG		;SETUP FLAG FOR AN ABORT
2779	013524	010767	165300		MOV	0#1,0#177572		;TURN MMU ON
2780	013530	114125			MOV	R7,SLOC00		;SAVE PC
					MOVB	-(R1),-(R5),		;DO A RELOCATION

```

2781 013532 004767 000122 JSR PC,TS12 ;CHECK IF AN ABORT OCCURRED AND IF
2782 ;YES IF MMR1 EQUALS EXPECTED DATA
2783 013536 012703 006411 MOV #6411,R3 ;SETUP EXPECTED DATA FOR MMR1
2784 013542 012767 000001 165272 MOV #1,FLAG ;SETUP FLAG FOR AN ABORT
2785 013550 012737 000001 177572 MOV #1,00177572 ;TURN MMU ON
2786 013556 010767 165246 MOV R7,SLOC00 ;SAVE PC
2787 013562 112125 MOVVB (R1)+,(R5)+ ;DO A RELOCATION
2788 013564 004767 000070 JSR PC,TS12 ;CHECK IF AN ABORT OCCURRED AND IF
2789 ;YES IF MMR1 EQUALS EXPECTED DATA
2790 013570 012703 171025 MOV #171025,R3 ;SETUP EXPECTED DATA FOR MMR1
2791 013574 012767 000001 165240 MOV #1,FLAG ;SETUP FLAG FOR AN ABORT
2792 013602 012737 000001 177572 MOV #1,00177572 ;TURN MMU ON
2793 013610 010767 165214 MOV R7,SLOC00 ;SAVE PC
2794 013614 012542 MOV (R5)+,(R2)+ ;DO A RELOCATION
2795 013616 004767 000036 JSR PC,TS12 ;CHECK IF AN ABORT OCCURRED AND IF
2796 ;YES IF MMR1 EQUALS EXPECTED DATA
2797 013622 012703 012762 MOV #12762,R3 ;SETUP EXPECTED DATA FOR MMR1
2798 013626 012767 000001 165206 MOV #1,FLAG ;SETUP FLAG FOR AN ABORT
2799 013634 012737 000001 177572 MOV #1,00177572 ;TURN MMU ON
2800 013642 010767 165162 MOV R7,SLOC00 ;SAVE PC
2801 013646 014225 MOV -(R2),(R5)+ ;DO A RELOCATION
2802 013650 004767 000004 JSR PC,TS12 ;CHECK IF AN ABORT OCCURRED AND IF
2803 ;YES IF MMR1 EQUALS EXPECTED DATA
2804
2805 013654 000167 000062 JMP T12FIN
2806
2807 ;ROUTINE TO CHECK IF AN ABORT OCCURRED AND IF MMR1 EQUALS EXPECTED DATA
2808
2809 013660 022767 000000 165154 TS12: CMP #0,FLAG ;DID AN ABORT OCCUR
2810 013666 001403 BEQ 1$ ;YES GO ON
2811 013670 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
2812 013672 000243 .WORD 243 ;UNIQUE ERROR NUMBER
2813 013674 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2814 ;NO GO TO ERROR
2815 013676 020367 165172 1$: CMP R3,SAVMR1 ;TEST MMR1 FOR EXPECTED DATA
2816 013702 001403 BEQ 2$ ;OK GO ON
2817 013704 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
2818 013706 000244 .WORD 244 ;UNIQUE ERROR NUMBER
2819 013710 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2820 ;NO GO TO ERROR
2821 013712 026767 165112 165156 2$: CMP SLOC00,SAVMR2 ;TEST MMR2 FOR EXPECTED DATA
2822 013720 001403 BEQ 3$ ;OK GO ON
2823 013722 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
2824 013724 000245 .WORD 245 ;UNIQUE ERROR NUMBER
2825 013726 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
2826 ;NO GO TO ERROR
2827 013730 005067 165140 3$: CLR SAVMR1 ;CLEAR STATUS REG SAVE AREA
2828 013734 005067 165136 CLR SAVMR2 ;
2829 013740 000207 RTS PC ;RETURN
2830
2831 013742 ;T12FIN:
2832 013742 TSMM13:
2833 ;*****
2834 ;*TEST 21 ADDER RELOCATION TEST PART A
2835 ;*****
2836 ;(NEED 16 BITS OF MEMORY ADDRESSING)

```



```
2837
2838 013742
2839 013742 005267 165036
2840 013746 005037 177572
2841 013752 005067 165064
2842 013756 005037 177776
2843 013762 004767 165320
2844 013766 012737 000020 172516
2845 013774 012703 014164
2846 014000 012701 014216
2847 014004 012133
2848 014006 021127 000333
2849 014012 001374
2850 014014 012703 014302
2851 014020 012701 014250
2852 014024 012702 014334
2853 014030 012237 177776
2854 014034 013305
2855 014036 012737 000001 177572
2856 014044 006531
2857 014046 012604
2858 014050 005037 177572
2859 014054 020504
2860 014056 001403
2861 014060 104000
2862 014062 000246
2863 014064 001213
2864
2865 014066 021327 000111
2866 014072 001356
2867 014074 005203
2868 014076 005203
2869 014100 005201
2870 014102 005201
2871 014104 005202
2872 014106 005202
2873 014110 012237 177776
2874 014114 013305
2875 014116 012737 000027 172516
2876 014124 012737 000001 177572
2877 014132 106531
2878 014134 012604
2879 014136 005037 177572
2880 014142 005037 172516
2881 014146 020504
2882 014150 001403
2883 014152 104000
2884 014154 000247
2885 014156 001213
2886
2887 014160
2888
2889 014160 000167 000202
2890
2891
2892
```

```
*****
TST21:
INC $TESTN ; INCREMENT TEST NUMBER
CLR @0177572 ; MMU OFF
CLR FLAG ; CLEAR MMU ABORT FLAG
CLR @0177776 ; INIT PSW
JSR PC,MMU ; INIT MMU
MOV @20,@0172516 ; INIT MMR3
MOV @PARAD1,R3 ; SETUP PARS WITH TEST VALUES
MOV @PARVA1,R1
1$: MOV (R1)+,@(R3)+
CMP (R1),#333
BNE 1$
MOV @PHY1,R3 ; SET POINTERS TO ADDER PART A
MOV @VIR1,R1 ; TEST TABLES.
MOV @MODE1,R2
2$: MOV (R2)+,@0177776
MOV @R3)+,R5 ; INIT PSW
MOV @1,@0177572 ; SAVE DATA AT PHYSICAL ADDRESS
MFPI @R1)+ ; TURN MMU ON
MOV (SP)+,R4 ; SAVE DATA AT RELOCATED VIRTUAL ADDRESS
CLR @0177572
CMP R5,R4 ; TURN MMU OFF
BEQ 3$ ; IS DATA EQUAL TO EXPECTED
ERROR ; YES GO ON
.WORD 246 ; ALL ERRORS TO TRAP TO EMT VECTOR
.WORD MMUERR ; UNIQUE ERROR NUMBER
; ADDRESS OF ERROR MESSAGE
; NO IT IS AN ADDER ERROR
3$: CMP (R3),#111
BNE 2$
INC R3
INC R3
INC R1
INC R1
INC R2
INC R2
MOV (R2)+,@0177776 ; INIT PSW
MOV @R3)+,R5 ; SAVE DATA AT PHYSICAL ADDRESS
MOV @27,@0172516 ; INIT MMR3
MOV @1,@0177572 ; TURN MMU ON
MFPI @R1)+ ; SAVE DATA AT RELOCATED VIRTUAL ADDRESS
MOV (SP)+,R4 ; POP THE STACK
CLR @0177572 ; TURN MMU OFF
CLR @0172516 ; CLEAR MMR3
CMP R5,R4 ; IS DATA EQUAL TO EXPECTED
BEQ 4$ ; YES GO ON
ERROR ; ALL ERRORS TO TRAP TO EMT VECTOR
.WORD 247 ; UNIQUE ERROR NUMBER
.WORD MMUERR ; ADDRESS OF ERROR MESSAGE
; NO IT IS AN ADDER ERROR
4$: JMP T13FIN
;
; ADDER TEST PART A TABLES
;
```

2893	014164	172240	PARAD1:	.WORD	172240
2894	014166	177642		.WORD	177642
2895	014170	172252		.WORD	172252
2896	014172	177640		.WORD	177640
2897	014174	172242		.WORD	172242
2898	014176	172254		.WORD	172254
2899	014200	177652		.WORD	177652
2900	014202	177644		.WORD	177644
2901	014204	172246		.WORD	172246
2902	014206	177654		.WORD	177654
2903	014210	172250		.WORD	172250
2904	014212	177660		.WORD	177660
2905	014214	000333		.WORD	333
2906	014216	000000	PARVA1:	.WORD	000000
2907	014220	000010		.WORD	000010
2908	014222	177777		.WORD	177777
2909	014224	177601		.WORD	177601
2910	014226	000010		.WORD	000010
2911	014230	000052		.WORD	000052
2912	014232	000070		.WORD	000070
2913	014234	000010		.WORD	000010
2914	014236	000010		.WORD	000010
2915	014240	000060		.WORD	000060
2916	014242	000000		.WORD	000000
2917	014244	000010		.WORD	000010
2918	014246	000333		.WORD	333
2919	014250	000000	VIR1:	.WORD	000000
2920	014252	025000		.WORD	025000
2921	014254	135224		.WORD	135224
2922	014256	017700		.WORD	017700
2923	014260	033000		.WORD	033000
2924	014262	145252		.WORD	145252
2925	014264	121000		.WORD	121000
2926	014266	043000		.WORD	043000
2927	014270	075000		.WORD	075000
2928	014272	142000		.WORD	142000
2929	014274	117700		.WORD	117700
2930	014276	000111		.WORD	111
2931	014300	007000		.WORD	007000
2932	014302	000000	PHY1:	.WORD	000000
2933	014304	006000		.WORD	006000
2934	014306	015124		.WORD	015124
2935	014310	000000		.WORD	000000
2936	014312	014000		.WORD	014000
2937	014314	012452		.WORD	012452
2938	014316	010000		.WORD	010000
2939	014320	004000		.WORD	004000
2940	014322	016000		.WORD	016000
2941	014324	010000		.WORD	010000
2942	014326	017700		.WORD	017700
2943	014330	000111		.WORD	111
2944	014332	010000		.WORD	010000
2945	014334	010000	MODE1:	.WORD	010000
2946	014336	030000		.WORD	030000
2947	014340	010000		.WORD	010000
2948	014342	030000		.WORD	030000


```
2949 014344 010000 .WORD 010000
2950 014346 010000 .WORD 010000
2951 014350 030000 .WORD 030000
2952 014352 030000 .WORD 030000
2953 014354 010000 .WORD 010000
2954 014356 030000 .WORD 030000
2955 014360 010000 .WORD 010000
2956 014362 000111 .WORD 111
2957 014364 030000 .WORD 030000
2958
2959 014366
2960 014366
2961
2962
2963
2964
2965
2966
2967
2968 014366
2969 014366 005267 164412
2970 014372 032777 000400 164446
2971 014400 001405
2972 014402 062737 000001 001004
2973
2974 014410 000167 001472
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987 014414 013767 000004 164406
2988 014422 013767 000006 164402
2989 014430 012737 014514 000004
2990 014436 012737 000340 000006
2991 014444 012767 000020 156044
2992 014452 012767 170000 155674
2993 014460 012767 000001 163104
2994 014466 005067 163306
2995 014472 012737 177777 140000
2996
2997 014500 005767 163274
2998
2999
3000 014504 001407
3001
3002 014506 104000
3003 014510 000250
3004 014512 001213

;
T13FIN:
TS1822:
;*****
;TEST 22 TEST 22/18 BIT ADDRESS OPTION
;*****
;CHECK THE SOFTWARE SWITCH REGISTER TO DETERMINE IF THIS IS A 22 BIT OR AN
;18 BIT ADDRESS SYSTEM. BIT 08 IN THE SWR=1 INDICATES AN 18 BIT SYSTEM.
;IF WE'RE IN A 22 BIT SYSTEM WE CAN PERFORM SOME EXTRA TESTS.
;*****
TST22:
INC $TESTN ;INCREMENT TEST NUMBER
BIT @BIT08,@SWR ;IS BIT 08 SET?
BEQ 100$ ;BRANCH IF ITS NOT
ADD @1,@$TESTN ;KEEP TEST NUMBERS IN ORDER
;ADD 1 FOR THE TESTS WE'RE SKIPPING
;SKIP OVER THESE TESTS IF IT IS
JMP T14FIN

;
;IF THIS IS A 22 BIT SYSTEM CHECK THE 22/18 BIT ADDRESS OPTION
;
;TO TEST 22 BIT ADDRESSING WE DO THE FOLLOWING:
; A. ENABLE 22 BIT ADDRESSING MODE
; B. CLEAR ADDRESS 0
; C. WRITE PHYSICAL ADDRESS 17000000 WITH ALL ONES
; D. CHECK ADDRESS 0
;IF ADDRESS 0 IS UNCHANGED (=0) OR A TIME OUT OCCURRED, IT INDICATES
;22 BIT MODE IS FUNCTIONING.
;IF ADDRESS 0 =177777 IT INDICATES THAT 22 BIT MODE IS NOT FUNCTIONING.
100$: MOV @4,SLOC00 ;SAVE VECTOR
MOV @6,SLOC01 ;SAVE VECTOR
MOV @1,@4 ;SET VECTOR FOR NXM TRAP
MOV @340,@6
MOV @20,SR3 ;ENABLE 22 BIT MODE ADDRESSING
MOV @170000,KIPAR6 ;SET KIPAR6 FOR 1920-1924KW ADDR RANGE
MOV @1,SRO ;ENABLE MMU
CLR 0 ;CLEAR ADDR 0
MOV @177777,@140000 ;MOVE ALL ONES TO ADDR 17000000 VIA KIPAR6
;A TIME OUT ERROR OR
;ADDR 0 REMAINING CLEAR INDICATES
;THAT 22 BIT ADDRESS MODE IS WORKING AND
;THAT SOME FURTHER TESTS SHOULD BE PERFORMED
;IF ADDR 0 =177777
;ERROR! 22 BIT ADDRESS MODE BAD
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
TST 0
BEQ 2$
ERROR
.WORD 250
.WORD MMUERR
```

PC	Inst	Op1	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op419	Op420	Op421	Op422	Op423	Op424	Op425	Op426	Op427	Op428	Op429	Op430	Op431	Op432	Op433	Op434	Op435	Op436	Op437	Op438	Op439	Op440	Op441	Op442	Op443	Op444	Op445	Op446	Op447	Op448	Op449	Op450	Op451	Op452	Op453	Op454	Op455	Op456	Op457	Op458	Op459	Op460	Op461	Op462	Op463	Op464	Op4
----	------	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-----

3061	015000	012767	100000	155346	MOV	#100000,KIPAR6	;TEST ADDRESS BIT 21
3062	015006	012737	177777	140000	MOV	#177777,#140000	;WRITE ALL ONES AT ADDR 10000000
3063	015014	005767	162760		TST	0	;CHECK ADDRESS 0. SHOULD = 0
3064	015020	001405			BEQ	8:	;BRANCH IF ADDR 0 = 0
3065	015022	005067	162544		CLR	SRO	;DISABLE MMU BEFORE ERROR
3066	015026	104000			ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
3067	015030	000255			.WORD	255	;UNIQUE ERROR NUMBER
3068	015032	001213			.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
3069							;ERROR! ADDR BIT 21 DID NOT ASSERT
3070	015034	005067	162532		8: CLR	SRO	;DISABLE MMU
3071	015040	005037	177766		CLR	#177766	;CLEAR CPU ERROR REGISTER
3072	015044	012706	001000		MOV	#STBOT,R6	;RESET STACK POINTER
3073	015050	013737	001030	000004	MOV	#SLOC00,#4	;RESTORE VECTORS
3074	015056	013737	001032	000006	MOV	#SLOC01,#6	
3075							
3076	015064						
3077							
3078							
3079							
3080							
3081							
3082	015064						
3083	015064	005267	163714				
3084	015070	005037	177572				
3085	015074	005067	162666				
3086	015100	013737	000004	001030			
3087	015106	013737	000006	001032			
3088	015114	012737	015620	000004			
3089	015122	012737	000340	000006			
3090	015130	005037	172340				
3091	015134	012767	077406	155136			
3092	015142	012737	177500	172354			
3093							
3094							
3095	015150	012767	077406	155136			
3096	015156	012737	000020	172516			
3097	015164	012737	000001	177572			
3098	015172	005737	157776				
3099							
3100	015176	013737	001030	000004			
3101	015204	013737	001032	000006			
3102	015212	005037	177572				
3103	015216	005037	001042				
3104	015222	004767	164060				
3105	015226	012737	010000	177776			
3106	015234	012737	000020	172516			
3107	015242	052737	001000	177746			
3108	015250	012704	016022				
3109	015254	012701	016054				
3110	015260	012437	172246				
3111	015264	012737	000001	177572			
3112	015272	012746	125252				
3113	015276	006671	000000				
3114	015302	006531					
3115	015304	022726	125252				
3116	015310	001405					


```

TSM14:
;*****
;TEST 23      ADDER RELOCATION TEST PART B
;*****
;(NEED 22 BITS OF MEMORY ADDRESSING)
;*****
TST23:
      INC      $TESTN          ;INCREMENT TEST NUMBER
      CLR      @SRO            ;TURN OFF MMU.
      CLR      CPREG           ;CLEAR THE CPU ERROR REGISTER
      MOV      @4,@SLOC00      ;SAVE LOC 4 IN SLOC00.
      MOV      @6,@SLOC01      ;SAVE LOC 6 IN SLOC01.
      MOV      @NXMTRP,@4      ;SET UP FOR TIMEOUT TRAP
      MOV      @340,@6         ;SET UP FOR TIMEOUT TRAP
      CLR      @KIPAR0         ;SET KER PAR0 FOR 1ST 4KW OF MEMORY.
      MOV      @77406,KIPDR0   ;SET KER PDR FOR 4KW R/W ACCESS.
      MOV      @177500,@KIPAR6 ;SET UP KERNEL PAGE ADDR REG 6
                                   ;FOR HIGHEST 4K WORDS OF NON-I/O
                                   ;FOR 2 MEG WORDS OF MEMORY.
      MOV      @77406,KIPDR6   ;SET KER PDR6 FOR 4KW R/W ACCESS.
      MOV      @20,@SR3        ;ENABLE 22 BIT ADDRESSING.
      MOV      @1,@SRO         ;TURN ON THE MMU.
      TST      @157776         ;ATTEMPT TO ADDRESS LAST MEMORY ADDR.
;*****WILL TRAP TO 4 IF 2 MEG WORDS OF MEMORY NOT AVAILABLE*****
      MOV      @SLOC00,@4      ;RESTORE LOC 4
      MOV      @SLOC01,@6      ;RESTORE LOC 6
      CLR      @177572         ;MMU OFF
      CLR      @FLAG           ;CLEAR MMU ABORT FLAG
      JSR      PC,MMU          ;INIT MMU
      MOV      @10000,@177776  ;INIT PSW
      MOV      @20,@172516     ;INIT MMR3
      BIS      @1000,@177746   ;TURN CACHE TEST FEATURE ON
      MOV      @PARVA3,R4      ;SET POINTERS TO INIT TABLES
      MOV      @VIR3,R1
      MOV      (R4),@172246    ;INIT SIPAR3
      MOV      @1,@177572      ;TURN MMU ON
      MOV      @125252,-(SP)   ;PUSH BACKGROUND DATA ON TO THE STACK
      MTPI     @0(R1)          ;WRITE DATA TO PHYSICAL ADDRESS
      MFPI     @0(R1)          ;WRITE DATA AT PHYSICAL ADDRESS TO STACK
      CMP      @125252,(SP)    ;IS DATA EQUAL TO EXPECTED
      BEQ      2$              ;YES GO ON

```

3117 015312 005037 177572 CLR @0177572 ;TURN MMU OFF
3118 015316 104000 ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
3119 015320 000256 .WORD 256 ;UNIQUE ERROR NUMBER
3120 015322 001213 .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
3121 ;NOT EQUAL GO TO ERROR
3122 015324 005037 177572 2\$: CLR @0177572 ;TURN MMU OFF
3123 015330 021427 000333 CMP (R4),@333 ;ARE WE DONE
3124 015334 001351 BNE 1\$;NO GO TO 1\$
3125 015336 012704 015704 MOV @PARVA2,R4 ;SET POINTERS TO PAR INIT TABLES
3126 015342 012701 015652 MOV @PARAD2,R1 ;
3127 015346 012431 3\$: MOV (R4)+,@(R1)+ ;INIT PARS
3128 015350 021127 000333 CMP (R1),@333 ;ARE WE DONE
3129 015354 001374 BNE 3\$;NO, GO TO 3\$
3130 015356 012704 015770 MOV @MODE2,R4 ;SET POINTERS TO ADDER PART B TABLES
3131 015362 012701 015736 MOV @VIR2,R1 ;
3132 015366 012702 016022 MOV @PARVA3,R2 ;
3133 015372 012703 016054 MOV @VIR3,R3 ;
3134 015376 004767 000076 4\$: JSR PC,T514 ;
3135 ;WRITE DATA TO PHYSICAL ADDRESS AND THEN
3136 ;CHECK IF DATA AT PHYSICAL ADDRESS IS
3137 ;EQUAL TO EXPECTED AND IF NOT DETERMINE
3138 015402 021127 000111 CMP (R1),@111 ;IF IT IS AN ADDER ERROR OR A MEMORY ERROR
3139 ;HAVE WE DONE ALL THE 22 BIT MODE I SPACE
3140 015406 001373 BNE 4\$;CASES
3141 015410 005201 INC R1 ;NO GO TO 4\$
3142 015412 005201 INC R1 ;POINT TO 22 BIT MODE D SPACE CASE
3143 015414 005204 INC R4 ;
3144 015416 005204 INC R4 ;
3145 015420 012737 000027 172516 MOV @27,@0172516 ;INIT MMR3
3146 015426 012437 177776 MOV (R4)+,@0177776 ;INIT PSW
3147 015432 012746 052525 MOV @52525,-(SP) ;PUSH DATA ONTO STACK
3148 015436 012737 000001 177572 MOV @1,@0177572 ;TURN MMU ON
3149 015444 106631 MTPD @(R1)+ ;WRITE DATA TO PHYSICAL ADDRESS
3150 015446 005037 177572 CLR @0177572 ;TURN MMU OFF
3151 015452 012737 000020 172516 MOV @20,@0172516 ;INIT MMR3
3152 015460 004767 000040 JSR PC,T14 ;CHECK IF DATA AT PHYSICAL ADDRESS IS EQUAL
3153 ;TO EXPECTED AND IF NOT DETERMINE IF IT
3154 ;IS AN ADDER ERROR OR A MEMORY ERROR
3155 015464 005037 172516 CLR @0172516 ;INIT MMR3 FOR 18 BIT MODE
3156 015470 004767 000004 JSR PC,T514 ;WRITE DATA TO PHYSICAL ADDRESS AND THEN
3157 ;CHECK IF DATA AT PHYSICAL ADDRESS IS
3158 ;EQUAL TO EXPECTED AND IF NOT DETERMINE IF
3159 ;IT IS AN ADDER ERROR OR A MEMORY ERROR
3160
3161 015474 000167 000406 JMP T14FIN
3162 ;
3163 ;ROUTINE TO WRITE DATA TO PHYSICAL ADDRESS AND TO CHECK IF DATA AT
3164 ;PHYSICAL ADDRESS IS EQUAL TO EXPECTED AND IF NOT DETERMINE IF IT IS
3165 ;AN ADDER ERROR OR A MEMORY ERROR
3166 ;
3167 015500 012437 177776 T514: MOV (R4)+,@0177776 ;INIT PSW
3168 015504 012737 000001 177572 MOV @1,@0177572 ;TURN MMU ON
3169 015512 012746 052525 MOV @52525,-(SP) ;WRITE DATA ONTO STACK
3170 015516 006631 MTPD @(R1)+ ;WRITE DATA TO PHYSICAL ADDRESS VIA STACK
3171 015520 005037 177572 CLR @0177572 ;TURN MMU OFF
3172 015524 012737 010000 177776 T14: MOV @10000,@0177776 ;INIT PSW


```
3173 015532 012237 172246      MOV      (R2)+, @0172246      ;INIT SIPAR3
3174 015536 012737 000001 177572 MOV      @1, @0177572      ;TURN MMU ON
3175 015544 006573 000000      MFPI     @0(R3)          ;DO RELOCATION
3176 015550 022726 052525      CMP      @52525.(SP)+    ;IS DATA EQUAL TO EXPECTED
3177 015554 001414              BEQ      2$                ;YES GO ON
3178 015556 006573 000000      MFPI     @0(R3)          ;WHAT TYPE OF ERROR IS IT
3179 015562 022726 125252      CMP      @125252.(SP)+    ;
3180 015566 001404              BEQ      1$                ;
3181 015570 104000              ERROR                     ;ALL ERRORS TO TRAP TO EMT VECTOR
3182 015572 000257              .WORD    257              ;UNIQUE ERROR NUMBER
3183 015574 001213              .WORD    MMUERR           ;ADDRESS OF ERROR MESSAGE
3184                                     ;IT IS A MEMORY ERROR
3185 015576 000403              BR        2$                ;
3186 015600              1$: ERROR                     ;ALL ERRORS TO TRAP TO EMT VECTOR
3187 015600 104000              .WORD    260              ;UNIQUE ERROR NUMBER
3188 015602 000260              .WORD    MMUERR           ;ADDRESS OF ERROR MESSAGE
3189 015604 001213              ;IT IS AN ADDER ERROR
3190                                     ;TURN MMU OFF
3191 015606 005037 177572      2$: CLR      @0177572      ;
3192 015612 005203              INC      R3                ;
3193 015614 005203              INC      R3                ;
3194 015616 000207              RTS      PC                ;RETURN
3195                                     ;
3196                                     ;NON-EXISTANT MEMORY TRAP ROUTINE
3197                                     ;
3198 015620 005037 177572      NXMTRP: CLR      @0SR0      ;TURN OFF MMU.
3199 015624 012716 016106      MOV      @T14FIN.(SP)    ;SET UP STACK WITH RETURN ADDR.
3200 015630 013737 001030      MOV      @0SLOC00, @04    ;RESTORE LOC 4
3201 015636 013737 001032      MOV      @0SLOC01, @06    ;RESTORE LOC 6
3202 015644 005037 177766      CLR      @0177766      ;CLEAR TIME OUT INDICATION FROM
3203                                     ;CPU ERROR REGISTER.
3204 015650 000006      RTT                                     ;RETURN FROM TRAP; GO TO NEXT TEST.
3205                                     ;
3206                                     ;ADDER TEST PART B TABLES
3207                                     ;
3208 015652 177646      PARAD2: .WORD    177646
3209 015654 177650      .WORD    177650
3210 015656 177652      .WORD    177652
3211 015660 172240      .WORD    172240
3212 015662 177640      .WORD    177640
3213 015664 177642      .WORD    177642
3214 015666 172244      .WORD    172244
3215 015670 177644      .WORD    177644
3216 015672 172252      .WORD    172252
3217 015674 172352      .WORD    172352
3218 015676 177662      .WORD    177662
3219 015700 172242      .WORD    172242
3220 015702 000333      .WORD    333
3221 015704 157700      PARVA2: .WORD    157700
3222 015706 137700      .WORD    137700
3223 015710 077700      .WORD    077700
3224 015712 176777      .WORD    176777
3225 015714 007600      .WORD    007600
3226 015716 167700      .WORD    167700
3227 015720 175700      .WORD    175700
3228 015722 177425      .WORD    177425
```

3229	015724	177220		.WORD	177220
3230	015726	173700		.WORD	173700
3231	015730	176700		.WORD	176700
3232	015732	077400		.WORD	077400
3233	015734	000333		.WORD	333
3234	015736	070000	VIR2:	.WORD	070000
3235	015740	110000		.WORD	110000
3236	015742	130000		.WORD	130000
3237	015744	000000		.WORD	000000
3238	015746	000000		.WORD	000000
3239	015750	030000		.WORD	030000
3240	015752	050000		.WORD	050000
3241	015754	052524		.WORD	052524
3242	015756	136000		.WORD	136000
3243	015760	130000		.WORD	130000
3244	015762	000111		.WORD	111
3245	015764	030000		.WORD	030000
3246	015766	030000		.WORD	030000
3247	015770	030000	MODE2:	.WORD	030000
3248	015772	030000		.WORD	030000
3249	015774	030000		.WORD	030000
3250	015776	010000		.WORD	010000
3251	016000	030000		.WORD	030000
3252	016002	030000		.WORD	030000
3253	016004	010000		.WORD	010000
3254	016006	030000		.WORD	030000
3255	016010	010000		.WORD	010000
3256	016012	000000		.WORD	000000
3257	016014	000111		.WORD	111
3258	016016	030000		.WORD	030000
3259	016020	010000		.WORD	010000
3260	016022	160000	PARVA3:	.WORD	160000
3261	016024	140000		.WORD	140000
3262	016026	100000		.WORD	100000
3263	016030	176770		.WORD	176770
3264	016032	007600		.WORD	007600
3265	016034	170000		.WORD	170000
3266	016036	176000		.WORD	176000
3267	016040	177552		.WORD	177552
3268	016042	177400		.WORD	177400
3269	016044	174000		.WORD	174000
3270	016046	177000		.WORD	177000
3271	016050	007500		.WORD	007500
3272	016052	000333		.WORD	333
3273	016054	060000	VIR3:	.WORD	060000
3274	016056	060000		.WORD	060000
3275	016060	060000		.WORD	060000
3276	016062	060700		.WORD	060700
3277	016064	060000		.WORD	060000
3278	016066	060000		.WORD	060000
3279	016070	060000		.WORD	060000
3280	016072	060024		.WORD	060024
3281	016074	060000		.WORD	060000
3282	016076	060000		.WORD	060000
3283	016100	060000		.WORD	060000
3284	016102	060000		.WORD	060000


```

3285 016104 000333 .WORD 333
3286
3287
3288
3289 016106
3290
3291
3292
3293
3294
3295
3296
3297
3298 016106
3299 016106 005267 162672
3300 016112 004767 163170
3301 016116 012737 177400 172354
3302 016124 016767 161654 162676
3303 016132 016767 000026 161644
3304 016140 052767 000001 161424
3305 016146 005067 161614
3306 016152 005067 161620
3307 016156 005737 157776
3308 016162 000423
3309
3310
3311 016164 022767 000040 161574 2$:
3312 016172 001403
3313 016174 104000
3314 016176 000261
3315 016200 001213
3316 016202 022726 016162 3$:
3317 016206 001403
3318 016210 104000
3319 016212 000262
3320 016214 001213
3321 016216 022726 000000 4$:
3322 016222 001403
3323 016224 104000
3324 016226 000263
3325 016230 001213
3326 016232 005067 161334
3327 016236 005067 161524
3328 016242 016767 162562 161534
3329
3330
3331 016250
3332
3333
3334
3335 016250
3336 016250 005267 162530
3337 016254 005037 177572
3338 016260 005067 162556
3339 016264 004767 163016
3340 016270 005037 177776

;
;T14FIN:
;*****
;*TEST 24 TEST NON-EXISTANT MEMORY TRAP
;*****
;WE ARE ASSUMING THAT THE NON-EXISTANT MEMORY TIME OUT
;FEATURE IS WORKING SINCE WE CAN'T GUARANTEE THAT
;THE SYSTEM BEING TESTED HAS A NON-EXISTANT MEMORY LOCATION.
;AT THIS TIME WE WILL ATTEMPT TO TEST THE NXM FUNCTION
;*****
TST24:
INC $TESTN ;INCREMENT TEST NUMBER
JSR PC,MMU ;INIT THE MMU
MOV #177400,$#KIPAR6 ;SET KIPAR6 TO RELOCATE TO HIGHEST MEMORY
MOV 4,SLOC00 ;SAVE VECTOR
MOV 2$,4 ;LOAD VEC WITH ADDR OF TRAP HANDLER
BIS #BIT00,SRO ;TURN ON THE MMU
CLR CPEREG ;CLEAR THE CPU ERROR REGISTER
CLR PSW ;CLEAR THE PSW
TST #157776 ;ACCESS PHYSICAL ADDR 17757776
BR NXMFIN ;IF IT DOESN'T TRAP WE'LL ASSUME
;THAT THIS IS A 4 MEGABYTE SYSTEM
;AND GO TO THE NEXT TEST
;IS CPU ERROR REGISTER CORRECT?
;
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;IS CONTENTS OF STACK CORRECT?
;
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;IS CONTENTS OF STACK CORRECT?
;
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;IS CONTENTS OF STACK CORRECT?
;
;ALL ERRORS TO TRAP TO EMT VECTOR
;UNIQUE ERROR NUMBER
;ADDRESS OF ERROR MESSAGE
;TURN OFF THE MMU
;CLEAR THE CPU ERROR REGISTER
;RESTORE THE VECTOR

;
;TSM15:
;*****
;*TEST 25 PAGE WRITTEN BIT TEST
;*****
TST25:
INC $TESTN ;INCREMENT TEST NUMBER
CLR #177572 ;MMU OFF
CLR FLAG ;CLEAR MMU ABORT FLAG
JSR PC,MMU ;INIT MMU
CLR #177776 ;INIT PSW

```

```

3341 016274 012704 172300      MOV    #172300,R4
3342 016300 004767 000114      JSR    PC,T15
3343
3344
3345 016304 004767 000164      JSR    PC,T15
3346
3347
3348 016310 012737 050000 177776  MOV    #50000,#177776
3349 016316 012704 172200      MOV    #172200,R4
3350 016322 004767 000072      JSR    PC,T15
3351
3352
3353 016326 004767 000142      JSR    PC,T15
3354
3355
3356 016332 005037 177776      CLR    #177776
3357 016336 012737 170000 177776  MOV    #170000,#177776
3358 016344 012704 177600      MOV    #177600,R4
3359 016350 004767 000044      JSR    PC,T15
3360
3361
3362 016354 004767 000114      JSR    PC,T15
3363
3364
3365 016360 005037 177776      CLR    #177776
3366 016364 012704 172300      MOV    #172300,R4
3367 016370 004767 000162      JSR    PC,T15A
3368
3369
3370 016374 012704 172200      MOV    #172200,R4
3371 016400 004767 000152      JSR    PC,T15A
3372
3373
3374 016404 012704 177600      MOV    #177600,R4
3375 016410 004767 000142      JSR    PC,T15A
3376
3377
3378
3379 016414 000167 000170      JMP    T15FIN
3380
3381      ; ROUTINE TO DO RELOCATIONS AND TEST IPDRS FOR PAGE WRITTEN BIT BEING
3382      ; SET AND IF NOT SET REPORT AN ERROR
3383
3384 016420 005001      T15:  CLR    R1
3385 016422 012737 000020 172516  MOV    #20,#172516
3386 016430 012737 000001 177572 18:  MOV    #1,#177572
3387 016436 011111      MOV    (R1),(R1)
3388 016440 005037 177572      CLR    #177572
3389 016444 022427 077506      CMP    (R4),#077506
3390 016450 001403      BEQ    29
3391 016452 104000      ERROR
3392 016454 000264      .WORD 264
3393 016456 001213      .WORD MMUERR
3394
3395 016460 062701 020000      29:  ADD    #20000,R1
3396 016464 020127 160000      CMP    R1,#160000

```

;SET POINTER TO KPDRS
 ;DO RELOCATIONS AND TEST KIPDRS FOR
 ;PAGE WRITTEN BIT BEING SET AND IF
 ;NOT SET GO TO ERROR
 ;DO RELOCATIONS AND TEST KDPDRS FOR
 ;PAGE WRITTEN BIT BEING SET AND IF NOT
 ;SET GO TO ERROR
 ;INIT PSW
 ;SET POINTER TO SPDRS
 ;DO RELOCATIONS AND TEST SIPDRS FOR
 ;PAGE WRITTEN BIT BEING SET AND IF NOT
 ;SET GO TO ERROR
 ;DO RELOCATIONS AND TEST SDPDRS FOR
 ;PAGE WRITTEN BIT BEING SET AND IF NOT
 ;SET GO TO ERROR
 ;INIT PSW TO A KNOWN STATE
 ;INIT PSW
 ;SET POINTER TO UPDRS
 ;DO RELOCATIONS AND TEST UIPDRS FOR
 ;PAGE WRITTEN BIT BEING SET AND IF
 ;NOT SET GO TO ERROR
 ;DO RELOCATIONS AND TEST UOPDRS FOR
 ;PAGE WRITTEN BIT BEING SET AND IF NOT
 ;SET GO TO ERROR
 ;INIT PSW TO A KNOWN STATE
 ;SET POINTER TO KPDRS
 ;EXPLICITLY WRITE TO KPDRS AND TEST
 ;FOR PAGE WRITTEN BIT BEING CLEARED
 ;AND IF NOT CLEARED GO TO ERROR
 ;SET POINTER TO SPDRS
 ;EXPLICITLY WRITE TO SPDRS AND TEST
 ;FOR PAGE WRITTEN BIT BEING CLEARED
 ;AND IF NOT CLEARED GO TO ERROR
 ;SET POINTER TO UPDRS
 ;EXPLICITLY WRITE TO UPDRS AND TEST
 ;FOR PAGE WRITTEN BIT BEING CLEARED
 ;AND IF NOT CLEARED GO TO ERROR

;SET POINTER TO VIRTUAL ADDRESS
 ;INIT MMU3
 ;TURN MMU ON
 ;DO A RELOCATION
 ;TURN MMU OFF
 ;IS DATA EQUAL TO EXPECTED
 ;OK GO ON
 ;ALL ERRORS TO TRAP TO EMT VECTOR
 ;UNIQUE ERROR NUMBER
 ;ADDRESS OF ERROR MESSAGE
 ;NO GO TO ERROR
 ;POINT TO NEXT VIRTUAL ADDRESS
 ;ARE WE DONE


```

3397 016470 001357      BNE 1$      ;NO GO TO 1$
3398 016472 000207      RTS PC      ;RETURN
3399
3400      ;ROUTINE TO DO RELOCATIONS AND TEST DPDRS FOR PAGE WRITTEN BIT BEING SET
3401      ;AND IF NOT SET REPORT AN ERROR
3402
3403 016474 005001      T15: CLR R1      ;SET POINTER TO VIRTUAL ADDRESS
3404 016476 062704 000002 ADD #2,R4      ;POINT TO FIRST DPDR
3405 016502 012737 000027 172516 MOV #27,#0172516 ;INIT MMR3
3406 016510 012737 000001 177572 1$: MOV #1,#0177572 ;TURN MMU ON
3407 016516 011146      MOV (R1),-(SP) ;PUSH DATA ONTO THE STACK
3408 016520 106611      MTPD (R1) ;DO A RELOCATION
3409 016522 005037 177572 CLR #0177572 ;TURN MMU OFF
3410 016526 022427 077506 CMP (R4),#077506 ;IS DATA EQUAL TO EXPECTED
3411 016532 001403      BEQ 2$      ;OK GO ON
3412 016534 104000      ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
3413 016536 000265      .WORD ;UNIQUE ERROR NUMBER
3414 016540 001213      .WORD 265 ;ADDRESS OF ERROR MESSAGE
3415      .WORD MMUERR
3416 016542 062701 020000 2$: ADD #20000,R1 ;NO GO TO ERROR
3417 016546 020127 160000 CMP R1,#160000 ;POINT TO NEXT VIRTUAL ADDRESS
3418 016552 001356      BNE 1$      ;ARE WE DONE
3419 016554 000207      RTS PC      ;NO GO TO 1$
3420      ;RETURN
3421      ;ROUTINE TO EXPLICITLY WRITE TO PDRS AND TEST PAGE WRITTEN BIT FOR BEING
3422      ;CLEARED AND IF NOT CLEARED REPORT AN ERROR
3423
3424 016556 005002      T15A: CLR R2      ;CLEAR COUNTER
3425 016560 011414      1$: MOV (R4),(R4) ;DO AN EXPLICIT WRITE TO PDR
3426 016562 022427 077406 CMP (R4),#077406 ;IS DATA EQUAL TO EXPECTED
3427 016566 001403      BEQ 2$      ;OK GO ON
3428 016570 104000      ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
3429 016572 000266      .WORD ;UNIQUE ERROR NUMBER
3430 016574 001213      .WORD 266 ;ADDRESS OF ERROR MESSAGE
3431      .WORD MMUERR
3432 016576 005202      2$: INC R2      ;NO GO TO ERROR
3433 016600 020227 000020 CMP R2,#20 ;INCREMENT POINTER
3434 016604 001365      BNE 1$      ;ARE WE DONE
3435 016606 000207      RTS PC      ;NO GO TO 1$
3436 016610      ;RETURN
3437
3438 016610      T15FIN:
3439      TSM16:
3440      ;*****
3441      ;*TEST 26 TEST CSM (CALL SUPERVISOR MODE)
3442      ;*****
3443 016610 005267 162170 TST26: INC $TESTN ;INCREMENT TEST NUMBER
3444 016614 005037 177572 CLR #0177572 ;MMU OFF
3445 016620 005037 001042 CLR #0FLAG ;CLEAR MMU ABORT FLAG
3446 016624 012704 017200 MOV #TMM16E,R4 ;INIT R4
3447 016630 004767 162452 JSR PC,MMU ;INIT MMU
3448 016634 012737 000037 172516 MOV #37,#0172516 ;ENABLE CSM INSTRUCTION
3449 016642 005037 177776 CLR #0177776 ;SET PS TO KER MODE
3450 016646 013746 000010 MOV #010,-(SP) ;SAVE VECTORS
3451 016652 013746 000014 MOV #014,-(SP)
3452 016656 013746 000016 MOV #016,-(SP)

```

3453	016662	012737	017054	000010	MOV	#TMM16B,@#10	;SETUP NEW VECTORS
3454	016670	012737	000137	000014	MOV	#137,@#14	;
3455	016676	012737	017214	000016	MOV	#TMM16A,@#16	;
3456	016704	007014			.WORD	7014	; TEST INSTRUCTION
3457	016706	104000			ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
3458	016710	000267			.WORD	267	;UNIQUE ERROR NUMBER
3459	016712	001213			.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
3460							;GO TO ERROR IF NOT TRAPPED
3461	016714	012737	017110	000010	TSM16A: MOV	#TMM16C,@#10	;SETUP NEW VECTOR
3462	016722	012737	000027	172516	MOV	#27,@#172516	;DISABLE CSM INSTRUCTION
3463	016730	012737	140000	177776	MOV	#140000,@#177776	;SET PS TO USER MODE
3464	016736	007014			.WORD	7014	; TEST INSTRUCTION
3465	016740	104000			ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
3466	016742	000270			.WORD	270	;UNIQUE ERROR NUMBER
3467	016744	001213			.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
3468							;GO TO ERROR IF NOT TRAPPED
3469	016746	012737	017144	000010	TSM16B: MOV	#TMM16D,@#10	;SETUP NEW VECTOR
3470	016754	012737	000027	172516	MOV	#27,@#172516	;DISABLE CSM INSTRUCTION
3471	016762	005037	177776		CLR	#177776	;SET PS TO KER MODE
3472	016766	007014			.WORD	7014	; TEST INSTRUCTION
3473	016770	104000			ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
3474	016772	000271			.WORD	271	;UNIQUE ERROR NUMBER
3475	016774	001213			.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
3476							;GO TO ERROR IF NOT TRAPPED
3477	016776	012737	000037	172516	TSM16C: MOV	#37,@#172516	;ENABLE CSM INSTRUCTION
3478	017004	012737	040000	177776	MOV	#40000,@#177776	;SET PS TO SUP MODE
3479	017012	012706	000700		MOV	#700,R6	;INIT SUP SP
3480	017016	012737	140000	177776	MOV	#140000,@#177776	;SET PS TO USER MODE
3481	017024	012706	000600		MOV	#600,R6	;INIT USER SP
3482	017030	012737	000014	000010	MOV	#14,@#10	;SETUP NEW VECTOR
3483	017036	000277			SCC		;SET ALL CC BITS
3484	017040	007024			.WORD	7024	; TEST INSTRUCTION
3485	017042				TSM16D: ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
3486	017042	104000			.WORD	272	;UNIQUE ERROR NUMBER
3487	017044	000272			.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
3488	017046	001213					;GO TO ERROR IF NOT TRAPPED
3490	017050	000167	000634		JMP	TM16A	
3491							
3492							
3493	017054	042737	007777	177776	TMM16B: BIC	#7777,@#177776	;CLEAR UNWANTED BITS
3494	017062	022737	000000	177776	CMP	#0,@#177776	;IS PS CORRECT
3495	017070	001403			BEQ	1\$	;YES GO ON
3496	017072	104000			ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
3497	017074	000273			.WORD	273	;UNIQUE ERROR NUMBER
3498	017076	001213			.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE
3499							;NO GO TO ERROR
3500	017100	005726			1\$: TST	(SP).	;CLEAN UP STACK
3501	017102	005726			TST	(SP).	;
3502	017104	000167	177604		JMP	TSM16A	;CONTINUE TESTING
3503	017110	042737	007777	177776	TMM16C: BIC	#7777,@#177776	;CLEAR UNWANTED BITS
3504	017116	022737	030000	177776	CMP	#30000,@#177776	;IS PS CORRECT
3505	017124	001403			BEQ	1\$	;YES GO ON
3506	017126	104000			ERROR		;ALL ERRORS TO TRAP TO EMT VECTOR
3507	017130	000274			.WORD	274	;UNIQUE ERROR NUMBER
3508	017132	001213			.WORD	MMUERR	;ADDRESS OF ERROR MESSAGE


```
3509
3510 017134 005726          1$: TST (SP)+      ;NO GO TO ERROR
3511 017136 005726          TST (SP)+      ;CLEAN UP STACK
3512 017140 000167 177602    JMP TSM16B      ;
3513 017144 042737 007777 177776 TMM16D: BIC 07777,00177776 ;CONTINUE TESTING
3514 017152 022737 000000 177776    CMP 00,00177776 ;CLEAR UNWANTED BITS
3515 017160 001403          BEQ 1$          ;IS PS CORRECT
3516 017162 104000          ERROR           ;YES GO ON
3517 017164 000275          .WORD 275       ;ALL ERRORS TO TRAP TO EMT VECTOR
3518 017166 001213          .WORD MMUERR    ;UNIQUE ERROR NUMBER
3519                                     ;ADDRESS OF ERROR MESSAGE
3520 017170 005726          1$: TST (SP)+      ;NO GO TO ERROR
3521 017172 005726          TST (SP)+      ;CLEAN UP STACK
3522 017174 000167 177576    JMP TSM16C      ;
3523 017200 156430          TMM16E: .WORD 156430 ;CONTINUE TESTING
3524 017202          TMM16F:                ;TEST LOCATION
3525 017202 104000          ERROR           ;ALL ERRORS TO TRAP TO EMT VECTOR
3526 017204 000276          .WORD 276       ;UNIQUE ERROR NUMBER
3527 017206 001213          .WORD MMUERR    ;ADDRESS OF ERROR MESSAGE
3528                                     ;GO TO ERROR IF DIDN'T ABORT
3529 017210 000167 000474    JMP TM16A      ;
3530 017214 022737 070017 177776 TMM16A: CMP 070017,00177776 ;IS PS CORRECT
3531 017222 001403          BEQ 1$          ;YES GO ON
3532 017224 104000          ERROR           ;ALL ERRORS TO TRAP TO EMT VECTOR
3533 017226 000277          .WORD 277       ;UNIQUE ERROR NUMBER
3534 017230 001213          .WORD MMUERR    ;ADDRESS OF ERROR MESSAGE
3535                                     ;NO GO TO ERROR
3536 017232 020627 000572    1$: CMP R6,0572 ;IS SP CORRECT
3537 017236 001403          BEQ 2$          ;YES GO ON
3538 017240 104000          ERROR           ;ALL ERRORS TO TRAP TO EMT VECTOR
3539 017242 000300          .WORD 300       ;UNIQUE ERROR NUMBER
3540 017244 001213          .WORD MMUERR    ;ADDRESS OF ERROR MESSAGE
3541                                     ;NO GO TO ERROR
3542 017246 020427 017202    2$: CMP R4,0TMM16E+2 ;IS R4 CORRECT
3543 017252 001403          BEQ 3$          ;YES GO ON
3544 017254 104000          ERROR           ;ALL ERRORS TO TRAP TO EMT VECTOR
3545 017256 000301          .WORD 301       ;UNIQUE ERROR NUMBER
3546 017260 001213          .WORD MMUERR    ;ADDRESS OF ERROR MESSAGE
3547                                     ;NO GO TO ERROR
3548 017262 023727 017200 156430 3$: CMP 00TMM16E,0156430 ;IS TEST LOCATION OK
3549 017270 001403          BEQ 4$          ;YES GO ON
3550 017272 104000          ERROR           ;ALL ERRORS TO TRAP TO EMT VECTOR
3551 017274 000302          .WORD 302       ;UNIQUE ERROR NUMBER
3552 017276 001213          .WORD MMUERR    ;ADDRESS OF ERROR MESSAGE
3553                                     ;NO GO TO ERROR
3554 017300 022627 156430    4$: CMP (SP)+,0156430 ;IS STACK CORRECT
3555 017304 001403          BEQ 5$          ;YES GO ON
3556 017306 104000          ERROR           ;ALL ERRORS TO TRAP TO EMT VECTOR
3557 017310 000303          .WORD 303       ;UNIQUE ERROR NUMBER
3558 017312 001213          .WORD MMUERR    ;ADDRESS OF ERROR MESSAGE
3559                                     ;NO GO TO ERROR
3560 017314 022627 017042    5$: CMP (SP)+,0TSM16D ;IS STACK CORRECT
3561 017320 001403          BEQ 6$          ;YES GO ON
3562 017322 104000          ERROR           ;ALL ERRORS TO TRAP TO EMT VECTOR
3563 017324 000304          .WORD 304       ;UNIQUE ERROR NUMBER
3564 017326 001213          .WORD MMUERR    ;ADDRESS OF ERROR MESSAGE
```

```
3565
3566 017330 022627 140000      6$: CMP (SP)+, #140000 ;NO GO TO ERROR
3567 017334 001403      7$: BEQ 7$ ;IS STACK CORRECT
3568 017336 104000      ERROR ;YES GO ON
3569 017340 000305      .WORD 305 ;ALL ERRORS TO TRAP TO EMT VECTOR
3570 017342 001213      .WORD MMUERR ;UNIQUE ERROR NUMBER
3571 ;ADDRESS OF ERROR MESSAGE
3572 017344 012706 000700      7$: MOV #700, R6 ;NO GO TO ERROR
3573 017350 012737 140000 177776 MOV #140000, #177776 ;RESTORE SUP SP
3574 017356 020627 000600      CMP R6, #600 ;SET PS TO USER MODE
3575 017362 001403      BEQ 8$ ;IS USER SP CORRECT
3576 017364 104000      ERROR ;YES GO ON
3577 017366 000306      .WORD 306 ;ALL ERRORS TO TRAP TO EMT VECTOR
3578 017370 001213      .WORD MMUERR ;UNIQUE ERROR NUMBER
3579 ;ADDRESS OF ERROR MESSAGE
3580 017372 012767 077400 152600 8$: MOV #77400, SIPDRO ;NO GO TO ERROR
3581 017400 012737 017202 000016 MOV #TMM16F, #16 ;SETUP SIPDRO TO ABORT
3582 017406 012737 000001 001042 MOV #1, #FLAG ;SETUP VECTOR
3583 017414 012737 000001 177572 MOV #1, #177572 ;SETUP FLAG FOR AN ABORT
3584 017422 010701      MOV R7, R1 ;TURN MMU ON
3585 017424 007014      .WORD 7014 ;SAVE OLD PC
3586 017426 022737 000000 001042 CMP #0, #FLAG ;TEST INSTRUCTION
3587 017434 001403      BEQ 9$ ;DID AN ABORT OCCUR
3588 017436 104000      ERROR ;YES GO ON
3589 017440 000307      .WORD 307 ;ALL ERRORS TO TRAP TO EMT VECTOR
3590 017442 001213      .WORD MMUERR ;UNIQUE ERROR NUMBER
3591 ;ADDRESS OF ERROR MESSAGE
3592 017444 023701 001076      9$: CMP #SAVMR2, R1 ;NO GO TO ERROR
3593 017450 001403      BEQ 10$ ;IS MMR2 CORRECT
3594 017452 104000      ERROR ;YES GO ON
3595 017454 000310      .WORD 310 ;ALL ERRORS TO TRAP TO EMT VECTOR
3596 017456 001213      .WORD MMUERR ;UNIQUE ERROR NUMBER
3597 ;ADDRESS OF ERROR MESSAGE
3598 017460 023727 001072 100041 10$: CMP #SAVMR0, #100041 ;NO GO TO ERROR
3599 017466 001403      BEQ 11$ ;IS MMRO CORRECT
3600 017470 104000      ERROR ;YES GO ON
3601 017472 000311      .WORD 311 ;ALL ERRORS TO TRAP TO EMT VECTOR
3602 017474 001213      .WORD MMUERR ;UNIQUE ERROR NUMBER
3603 ;ADDRESS OF ERROR MESSAGE
3604 017476 012737 000037 172516 11$: MOV #37, #172516 ;NO GO TO ERROR
3605 017504 012737 040000 177776 MOV #40000, #177776 ;ENABLE CSM
3606 017512 012706 000700      MOV #700, R6 ;SET PSW TO SUP
3607 017516 012737 140000 177776 MOV #140000, #177776 ;SETUP SUP SP
3608 017524 012706 000600      MOV #600, R6 ;SET PSW TO USE
3609 017530 012737 000014 000010 MOV #14, #10 ;SETUP USE SP
3610 017536 012737 017564 000016 MOV #TS16, #16 ;SETUP NEW VECTOR
3611 017544 000277      SCC ;SETUP NEW VECTOR
3612 017546 007027      .WORD 7027 ;SET ALL CC BITS
3613 017550 045712      .WORD 45712 ;TEST INSTRUCTION
3614 017552
3615 017552 104000      TS16A: ERROR ;ALL ERRORS TO TRAP TO EMT VECTOR
3616 017554 000312      .WORD 312 ;UNIQUE ERROR NUMBER
3617 017556 001213      .WORD MMUERR ;ADDRESS OF ERROR MESSAGE
3618
3619 017560 000167 000124      JMP TM16A ;GO TO ERROR IF DIDN'T TRAP
3620 017564 022737 070017 177776 TS16: CMP #70017, #177776 ;IS PSW CORRECT
```


Address	Instruction	Op-Code	Register	Value	Comment
3621	BEQ	200\$			;YES GO ON
3622	ERROR				;ALL ERRORS TO TRAP TO EMT VECTOR
3623	.WORD	313			;UNIQUE ERROR NUMBER
3624	.WORD	MMUERR			;ADDRESS OF ERROR MESSAGE
3625					;NO GO TO ERROR
3626	CMP	R6,#572			;IS SP CORRECT
3627	BEQ	201\$			;YES GO ON
3628	ERROR				;ALL ERRORS TO TRAP TO EMT VECTOR
3629	.WORD	314			;UNIQUE ERROR NUMBER
3630	.WORD	MMUERR			;ADDRESS OF ERROR MESSAGE
3631					;NO GO TO ERROR
3632	CMP	(SP)+,#45712			;IS STACK CORRECT
3633	BEQ	202\$			;YES GO ON
3634	ERROR				;ALL ERRORS TO TRAP TO EMT VECTOR
3635	.WORD	315			;UNIQUE ERROR NUMBER
3636	.WORD	MMUERR			;ADDRESS OF ERROR MESSAGE
3637					;NO GO TO ERROR
3638	CMP	(SP)+,#TS16A			;IS STACK CORRECT
3639	BEQ	203\$			;YES GO ON
3640	ERROR				;ALL ERRORS TO TRAP TO EMT VECTOR
3641	.WORD	316			;UNIQUE ERROR NUMBER
3642	.WORD	MMUERR			;ADDRESS OF ERROR MESSAGE
3643					;NO GO TO ERROR
3644	CMP	(SP)+,#140000			;IS STACK CORRECT
3645	BEQ	204\$			;YES GO ON
3646	ERROR				;ALL ERRORS TO TRAP TO EMT VECTOR
3647	.WORD	317			;UNIQUE ERROR NUMBER
3648	.WORD	MMUERR			;ADDRESS OF ERROR MESSAGE
3649					;NO GO TO ERROR
3650	MOV	#700,R6			;RESTORE SUP SP
3651	MOV	#140000,#177776			;SET PSW TO USER MODE
3652	CMP	R6,#600			;IS USER SP CORRECT
3653	BEQ	TM16A			;YES GO ON
3654	ERROR				;ALL ERRORS TO TRAP TO EMT VECTOR
3655	.WORD	320			;UNIQUE ERROR NUMBER
3656	.WORD	MMUERR			;ADDRESS OF ERROR MESSAGE
3657					;NO GO TO ERROR
3658	CLR	#177776			;SET PS TO KER MODE
3659	CLR	SRO			;TURN OFF MMU
3660	CLR	SR3			;TURN OFF 22 BIT ADDRESSING, I & D SPACE
3661	MOV	(SP)+,#16			;RESTORE VECTORS
3662	MOV	(SP)+,#14			;
3663	MOV	(SP)+,#10			;
3664					;
3665					;
3666					;
3667					;

```
3668 .MCALL IDMSG,ENDPAS
3669 .SBTTL END OF PASS ROUTINE
3670
3671 ;*****
3672 ;*INCREMENT THE PASS NUMBER ($PASS)
3673 ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
3674 ;*IF THERES A MONITOR GO TO IT
3675 ;*IF THERE ISN'T JUMP TO RESTART
3676
3677 $EOP:
3678     TST     $PASS
3679     BNE     SKIPID
3680     TYPE    ,MSG1
3681
3682     SKIPID: INC     $PASS
3683             BIC     @100000,$PASS
3684             DEC     (PC)+
3685             $EOPCT: .WORD 1
3686             BGT     $DOAGN
3687             MOV     (PC)+,@(PC)+
3688             $ENDCT: .WORD 1
3689             $EOPCT
3690             TYPE    ,MSG2
3691             MOV     $PASS,-(SP)
3692             TYPDS   , $ENULL
3693             TYPE    , $ENULL
3694             $GET42: MOV     @42,R0
3695             BEQ     $DOAGN
3696             RESET
3697             $ENDAD: JSR     PC,(R0)
3698             NOP
3699             NOP
3700             NOP
3701             $DOAGN: JMP     @(PC)+
3702             $RTNAD: .WORD  RESTART
3703             $ENULL: .BYTE  -1,-1,0
3704             .EVEN
3705             .ASCIZ  <CR><LF>/CZKDK-B-0 KDJ11 MEMORY MANAGEMENT DIAGNOSTIC/
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
```

017740 005767 161042
017744 001002
017746 104401 020046
017752
017752 005267 161030
017756 042767 100000 161022
017764 005327
017766 000001
017770 003022
017772 012737
017774 000001
017776 017766
020000 104401 020125
020004 016746 160776
020010 104405
020012 104401 020042
020016 013700 000042
020022 001405
020024 000005
020026 004710
020030 000240
020032 000240
020034 000240
020036
020036 000137
020040 002066
020042 377 377 000
020046
020046 005015 055103 042113
020054 026513 026502 020060
020062 042113 030512 020061
020070 042515 047515 054522
020076 046440 047101 043501
020104 046505 047105 020124
020112 044504 043501 047516
020120 052123 041511 000
020125 015 041412 045532
020132 045504 020102 047105
020140 020104 040520 051523
020146 021440 000
020152

ONLY TYPE MESSAGE AT END OF FIRST PASS
IF >0 THEN SKIP THE ID MESSAGE
ELSE TYPE THE ID MESSAGE
INCREMENT THE PASS NUMBER
DON'T ALLOW A NEG. NUMBER
LOOP?
YES
RESTORE COUNTER
SAVE \$PASS FOR TYPEOUT
GO TYPE--DECIMAL ASCII WITH SIGN
GET MONITOR ADDRESS
BRANCH IF NO MONITOR
CLEAR THE WORLD
GO TO MONITOR
SAVE ROOM
FOR
ACT11
RETURN
NULL CHARACTER STRING
CZKDK-B-0 KDJ11 MEMORY MANAGEMENT DIAGNOSTIC/
CZKDKB END PASS #/
EVEN
SBTTL TYPE ROUTINE

ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.


```
3724      ;*NOTE1:      $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
3725      ;*NOTE2:      $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
3726      ;*NOTE3:      $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
3727      ;*
3728      ;*CALL:
3729      ;*1) USING A TRAP INSTRUCTION
3730      ;*      TYPE      ,MESADR      ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
3731      ;*OR
3732      ;*      TYPE
3733      ;*      MESADR
3734      ;*
3735
3736      020152  105767  000343      $TYPE:  TSTB      $TPFLG      ;;IS THERE A TERMINAL?
3737      020156  100002              BPL      1$          ;;BR IF YES
3738      020160  000000              HALT              ;;HALT HERE IF NO TERMINAL
3739      020162  000430              BR      3$          ;;LEAVE
3740      020164  010046      1$:      MOV      R0,-(SP)      ;;SAVE R0
3741      020166  017600      000002      MOV      @2(SP),R0      ;;GET ADDRESS OF ASCIZ STRING
3742      020172  122767  000001  160620      CMPB      @APTENV,$ENV      ;;RUNNING IN APT MODE
3743      020200  001011              BNE      62$          ;;NO,GO CHECK FOR APT CONSOLE
3744      020202  132767  000100  160611      BITB      @APTSPool,$ENVM      ;;SPOOL MESSAGE TO APT
3745      020210  001405              BEQ      62$          ;;NO,GO CHECK FOR CONSOLE
3746      020212  010067  000004      MOV      R0,61$      ;;SETUP MESSAGE ADDRESS FOR APT
3747      020216  004767  001622      JSR      PC,$ATY3      ;;SPOOL MESSAGE TO APT
3748      020222  000000      61$:      .WORD      0          ;;MESSAGE ADDRESS
3749      020224  132767  000040  160567  62$:      BITB      @APTCsup,$ENVM      ;;APT CONSOLE SUPPRESSED
3750      020232  001003              BNE      60$          ;;YES,SKIP TYPE OUT
3751      020234  112046      2$:      MOVB      (R0),-(SP)      ;;PUSH CHARACTER TO BE TYPED ONTO STACK
3752      020236  001005              BNE      4$          ;;BR IF IT ISN'T THE TERMINATOR
3753      020240  005726              TST      (SP),      ;;IF TERMINATOR POP IT OFF THE STACK
3754      020242  012600      60$:      MOV      (SP),R0      ;;RESTORE R0
3755      020244  062716  000002      3$:      ADD      @2,(SP)      ;;ADJUST RETURN PC
3756      020250  000002              RTI              ;;RETURN
3757      020252  122716  000011      4$:      CMPB      @HT,(SP)      ;;BRANCH IF <HT>
3758      020256  001430              BEQ      8$          ;;BRANCH IF NOT <CRLF>
3759      020260  122716  000200      CMPB      @CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
3760      020264  001006              BNE      5$          ;;POP <CR><LF> EQUIV
3761      020266  005726              TST      (SP),      ;;TYPE A CR AND LF
3762      020270  104401      TYPE
3763      020272  001277      $CRLF
3764      020274  105067  000202      CLRB      $CHARCNT      ;;CLEAR CHARACTER COUNT
3765      020300  000755      BR      2$          ;;GET NEXT CHARACTER
3766      020302  004767  000056      5$:      JSR      PC,$TYPEC      ;;GO TYPE THIS CHARACTER
3767      020306  126726  000206      6$:      CMPB      $FILLC,(SP),      ;;IS IT TIME FOR FILLER CHARS.?
3768      020312  001350              BNE      2$          ;;IF NO GO GET NEXT CHAR.
3769      020314  016746  000176      MOV      $NULL,-(SP)      ;;GET # OF FILLER CHARS. NEEDED
3770      020320  105366  000001      7$:      DECB      1(SP)      ;;AND THE NULL CHAR.
3771      020324  002770              BLT      6$          ;;DOES A NULL NEED TO BE TYPED?
3772      020326  004767  000032      JSR      PC,$TYPEC      ;;BR IF NO--GO POP THE NULL OFF OF STACK
3773      020332  105367  000144      DECB      $CHARCNT      ;;GO TYPE A NULL
3774      020336  000770      BR      7$          ;;DO NOT COUNT AS A COUNT
3775      020336  000770
3776
3777      ;HORIZONTAL TAB PROCESSOR
3778
3779      020340  112716  000040      8$:      MOVB      @' ,(SP)      ;;REPLACE TAB WITH SPACE
```

```

3780 020344 004767 000014 9$: JSR PC,$TYPEC ;;TYPE A SPACE
3781 020350 132767 000007 000124 BITB #7,$CHARCNT ;;BRANCH IF NOT AT
3782 020356 001372 BNE 9$ ;;TAB STOP
3783 020360 005726 TST (SP), ;;POP SPACE OFF STACK
3784 020362 000724 BR 2$ ;;GET NEXT CHARACTER
3785 020364 $TYPEC:
3786 020364 105777 000116 TSTB @TKS ;;CHAR IN KYBD BUFFER? ;MJD001
3787 020370 100022 BPL 10$ ;;BR IF NOT ;MJD001
3788 020372 017746 000112 MOV @TKB,-(SP) ;;GET CHAR ;MJD001
3789 020376 042716 177600 BIC #177600,(SP) ;;STRIP EXTRANEIOUS BITS ;MJD001
3790 020402 122716 000023 CMPB @XOFF,(SP) ;;WAS CHAR XOFF ;MJD001
3791 020406 001012 BNE 102$ ;;BR IF NOT ;MJD001
3792 020410 101$:
3793 020410 105777 000072 TSTB @TKS ;;WAIT FOR CHAR ;MJD001
3794 020414 100375 BPL 101$ ;MJD001
3795 020416 117716 000066 MOVB @TKB,(SP) ;;GET CHAR ;MJD001
3796 020422 042716 177600 BIC #177600,(SP) ;;STRIP IT ;MJD001
3797 020426 122716 000021 CMPB @XON,(SP) ;;WAS IT XON? ;MJD001
3798 020432 001366 BNE 101$ ;;BR IF NOT ;MJD001
3799 020434 102$:
3800 020434 005726 TST (SP), ;;FIX STACK ;MJD001
3801 020436 10$:
3802 020436 105777 000050 TSTB @TPS ;;WAIT UNTIL PRINTER IS READY ;MJD001
3803 020442 100375 BPL 10$ ;MJD001
3804 020444 116677 000002 000042 MOVB 2(SP),@TPB ;;LOAD CHAR TO BE TYPED INTO DATA REG. ;MJD001
3805 020452 122766 000015 000002 CMPB @CR,2(SP) ;;IS CHARACTER A CARRIAGE RETURN?
3806 020460 001003 BNE 1$ ;;BRANCH IF NO
3807 020462 105067 000014 CLRB $CHARCNT ;;YES--CLEAR CHARACTER COUNT
3808 020466 000406 BR $TYPEX ;;EXIT
3809 020470 122766 000012 000002 1$: CMPB @LF,2(SP) ;;IS CHARACTER A LINE FEED?
3810 020476 001402 BEQ $TYPEX ;;BRANCH IF YES
3811 020500 105227 INCB (PC), ;;COUNT THE CHARACTER
3812 020502 000000 $CHARCNT: .WORD 0 ;;CHARACTER COUNT STORAGE
3813 020504 000207 $TYPEX: RTS PC
3814
3815 020506 177560 $TKS: .WORD 177560 ;;TTY KDB STATUS ;MJD001
3816 020510 177562 $TKB: .WORD 177562 ;;TTY KBD BUFFER ;MJD001
3817 020512 177564 $TPS: .WORD 177564 ;;TTY PRINTER STATUS REG. ADDRESS
3818 020514 177566 $TPB: .WORD 177566 ;;TTY PRINTER BUFFER REG. ADDRESS
3819 020516 000 $NULL: .BYTE 0 ;;CONTAINS NULL CHARACTER FOR FILLS
3820 020517 002 $FILLS: .BYTE 2 ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
3821 020520 012 $FILLC: .BYTE 12 ;;INSERT FILL CHARS. AFTER A "LINE FEED"
3822 020521 000 $TPFLG: .BYTE 0 ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
3823 020522 077 $QUES: .ASCII "?" ;;QUESTION MARK
3824 020523 012 000 $LF: .ASCIIZ <12> ;;LINEFEED
3825 020526 .EVEN
3826
3827 .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
3828
3829 ;*****
3830 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
3831 ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
3832 ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
3833 ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
3834 ;*REPLACED WITH SPACES.
3835 ;*CALL:
;* MOV NUM,-(SP) ;;PUT THE BINARY NUMBER ON THE STACK

```


3836
3837
3838 020526
3839 020526 010046
3840 020530 010146
3841 020532 010246
3842 020534 010346
3843 020536 010546
3844 020540 012746 020200
3845 020544 016605 000020
3846 020550 100004
3847 020552 005405
3848 020554 112766 000055 000001
3849 020562 005000
3850 020564 012703 020742
3851 020570 112723 000040
3852 020574 005002
3853 020576 016001 020732
3854 020602 160105
3855 020604 002402
3856 020606 005202
3857 020610 000774
3858 020612 060105
3859 020614 005702
3860 020616 001002
3861 020620 105716
3862 020622 100407
3863 020624 106316
3864 020626 103003
3865 020630 116663 000001 177777
3866 020636 052702 000060
3867 020642 052702 000040
3868 020646 110223
3869 020650 005720
3870 020652 020027 000010
3871 020656 002746
3872 020660 003002
3873 020662 010502
3874 020664 000764
3875 020666 105726
3876 020670 100003
3877 020672 116663 177777 177776
3878 020700 105013
3879 020702 012605
3880 020704 012603
3881 020706 012602
3882 020710 012601
3883 020712 012600
3884 020714 104401 020742
3885 020720 016666 000002 000004
3886 020726 012616
3887 020730 000002
3888 020732 023420
3889 020734 001750
3890 020736 000144
3891 020740 000012

;* TYPDS
\$TYPDS:
MOV R0,-(SP) ;;PUSH R0 ON STACK
MOV R1,-(SP) ;;PUSH R1 ON STACK
MOV R2,-(SP) ;;PUSH R2 ON STACK
MOV R3,-(SP) ;;PUSH R3 ON STACK
MOV R5,-(SP) ;;PUSH R5 ON STACK
MOV #20200,-(SP) ;;SET BLANK SWITCH AND SIGN
MOV 20(SP),R5 ;;GET THE INPUT NUMBER
BPL 1\$;;BR IF INPUT IS POS.
NEG R5 ;;MAKE THE BINARY NUMBER POS.
MOVB #'-.1(SP) ;;MAKE THE ASCII NUMBER NEG.
CLR R0 ;;ZERO THE CONSTANTS INDEX
MOV #DBLK,R3 ;;SETUP THE OUTPUT POINTER
MOVB #' ,(R3)+ ;;SET THE FIRST CHARACTER TO A BLANK
CLR R2 ;;CLEAR THE BCD NUMBER
MOV \$DTBL(R0),R1 ;;GET THE CONSTANT
SUB R1,R5 ;;FORM THIS BCD DIGIT
BLT 4\$;;BR IF DONE
INC R2 ;;INCREASE THE BCD DIGIT BY 1
BR 3\$
4\$: ADD R1,R5 ;;ADD BACK THE CONSTANT
TST R2 ;;CHECK IF BCD DIGIT=0
BNE 5\$;;FALL THROUGH IF 0
TSTB (SP) ;;STILL DOING LEADING 0'S?
BMI 7\$;;BR IF YES
5\$: ASLB (SP) ;;MSD?
BCC 6\$;;BR IF NO
MOVB 1(SP),-1(R3) ;;YES--SET THE SIGN
6\$: BIS #'0,R2 ;;MAKE THE BCD DIGIT ASCII
7\$: BIS #' ,R2 ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
MOVB R2,(R3)+ ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
TST (R0)+ ;;JUST INCREMENTING
CMP R0,#10 ;;CHECK THE TABLE INDEX
BLT 2\$;;GO DO THE NEXT DIGIT
BGT 8\$;;GO TO EXIT
MOV R5,R2 ;;GET THE LSD
BR 6\$;;GO CHANGE TO ASCII
8\$: TSTB (SP)+ ;;WAS THE LSD THE FIRST NON-ZERO?
BPL 9\$;;BR IF NO
MOVB -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
9\$: CLRB (R3) ;;SET THE TERMINATOR
MOV (SP)+,R5 ;;POP STACK INTO R5
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
TYPE ,DBLK ;;NOW TYPE THE NUMBER
MOV 2(SP),4(SP) ;;ADJUST THE STACK
MOV (SP)+,(SP)
RTI ;;RETURN TO USER
\$DTBL: 10000.
1000.
100.
10.

```
3892 020742 000004      $DBLK: .BLKW 4
3893                      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
3894
3895                      ;;*****
3896                      ;;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3897                      ;;OCTAL (ASCII) NUMBER AND TYPE IT.
3898                      ;;$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3899                      ;;CALL:
3900                      ;;      MOV      NUM, -(SP)          ;;NUMBER TO BE TYPED
3901                      ;;      TYPOS    ;;CALL FOR TYPEOUT
3902                      ;;      .BYTE   N                  ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3903                      ;;      .BYTE   M                  ;;M=1 OR 0
3904                      ;;                      ;;1=TYPE LEADING ZEROS
3905                      ;;                      ;;0=SUPPRESS LEADING ZEROS
3906                      ;;
3907                      ;;$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3908                      ;;$TYPOS OR $TYPOC
3909                      ;;CALL:
3910                      ;;      MOV      NUM, -(SP)          ;;NUMBER TO BE TYPED
3911                      ;;      TYPON    ;;CALL FOR TYPEOUT
3912                      ;;
3913                      ;;$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3914                      ;;CALL:
3915                      ;;      MOV      NUM, -(SP)          ;;NUMBER TO BE TYPED
3916                      ;;      TYPOC    ;;CALL FOR TYPEOUT
3917
3918 020752 017646 000000      $TYPOS: MOV      @ (SP), -(SP)      ;;PICKUP THE MODE
3919 020756 116667 000001 000211  MOV      1 (SP), $OFILL    ;;LOAD ZERO FILL SWITCH
3920 020764 112667 000207      MOV      (SP), $OMODE+1    ;;NUMBER OF DIGITS TO TYPE
3921 020770 062716 000002      ADD      @2, (SP)          ;;ADJUST RETURN ADDRESS
3922 020774 000406          BR      $TYPON
3923 020776 112767 000001 000171 $TYPOC: MOV      @1, $OFILL    ;;SET THE ZERO FILL SWITCH
3924 021004 112767 000006 000165  MOV      @6, $OMODE+1    ;;SET FOR SIX(6) DIGITS
3925 021012 112767 000005 000154 $TYPON: MOV      @5, $OCNT    ;;SET THE ITERATION COUNT
3926 021020 010346          MOV      R3, -(SP)          ;;SAVE R3
3927 021022 010446          MOV      R4, -(SP)          ;;SAVE R4
3928 021024 010546          MOV      R5, -(SP)          ;;SAVE R5
3929 021026 116704 000145      MOV      $OMODE+1, R4      ;;GET THE NUMBER OF DIGITS TO TYPE
3930 021032 005404          NEG      R4
3931 021034 062704 000006      ADD      @6, R4          ;;SUBTRACT IT FOR MAX. ALLOWED
3932 021040 110467 000132      MOV      R4, $OMODE      ;;SAVE IT FOR USE
3933 021044 116704 000125      MOV      $OFILL, R4      ;;GET THE ZERO FILL SWITCH
3934 021050 016605 000012      MOV      12 (SP), R5      ;;PICKUP THE INPUT NUMBER
3935 021054 005003          CLR      R3              ;;CLEAR THE OUTPUT WORD
3936 021056 006105          ROL      R5              ;;ROTATE MSB INTO "C"
3937 021060 000404          BR      3$              ;;GO DO MSB
3938 021062 006105          ROL      R5              ;;FORM THIS DIGIT
3939 021064 006105          ROL      R5
3940 021066 006105          ROL      R5
3941 021070 010503          MOV      R5, R3
3942 021072 006103          ROL      R3              ;;GET LSB OF THIS DIGIT
3943 021074 105367 000076      DECB     $OMODE          ;;TYPE THIS DIGIT?
3944 021100 100016          BPL      7$              ;;BR IF NO
3945 021102 042703 177770      BIC      @177770, R3      ;;GET RID OF JUNK
3946 021106 001002          BNE      4$              ;;TEST FOR 0
3947 021110 005704          TST      R4              ;;SUPPRESS THIS 0?
```



```
3948 021112 001403
3949 021114 005204
3950 021116 052703 000060
3951 021122 052703 000040
3952 021126 110367 000040
3953 021132 104401 021172
3954 021136 105367 000032
3955 021142 003347
3956 021144 002402
3957 021146 005204
3958 021150 000744
3959 021152 012605
3960 021154 012604
3961 021156 012603
3962 021160 016666 000002 000004
3963 021166 012616
3964 021170 000002
3965 021172 000
3966 021173 000
3967 021174 000
3968 021175 000
3969 021176 000000
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980 021200 022767 000176 157640
3981 021206 001074
3982 021210 105777 177272
3983 021214 100071
3984 021216 117746 177266
3985 021222 042716 177600
3986 021226 022726 000007
3987 021232 001062
3988 021234 126727 000514 000001
3989 021242 001456
3990
3991 021244 104401 021725
3992 021250 104401 021732
3993 021254 016746 156716
3994 021260 104402
3995 021262 104401 021743
3996 021266 005046
3997 021270 005046
3998 021272 105777 177210
3999 021276 100375
4000
4001 021300 117746 177204
4002 021304 042716 177600
4003
```

```
4$: BEQ 5$
INC R4
BIS #0,R3
5$: BIS #0,R3
MOVB R3,8$
TYPE 8$
7$: DECB $OCNT
BGT 2$
BLT 6$
INC R4
BR 2$
6$: MOV (SP)+,R5
MOV (SP)+,R4
MOV (SP)+,R3
MOV 2(SP),4(SP)
MOV (SP)+,(SP)
RTI
8$: .BYTE 0
.BYTE 0
$OCNT: .BYTE 0
$OFILL: .BYTE 0
$OMODE: .WORD 0
.SBTTL TTY INPUT ROUTINE

;;*****
.ENABL LSB

;;*****
;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
;*WHEN OPERATING IN TTY FLAG MODE.
$CKSWR: CMP $SWREG,SWR
BNE 15$
TSTB @TKS
BPL 15$
MOVB @TKB,-(SP)
BIC #C177,(SP)
CMP #7,(SP)
BNE 15$
CMPSB $AUTOB,#1
BEQ 15$
;;IS THE SOFT-SWR SELECTED?
;;BRANCH IF NO
;;CHAR THERE?
;;IF NO, DON'T WAIT AROUND
;;SAVE THE CHAR
;;STRIP-OFF THE ASCII
;;IS IT A CONTROL G?
;;NO, RETURN TO USER
;;ARE WE RUNNING IN AUTO-MODE?
;;BRANCH IF YES

$GTSWR: TYPE $CNTLG
TYPE $MSWR
MOV SWREG,-(SP)
TYPOC
TYPE $MNEW
19$: CLR -(SP)
CLR -(SP)
7$: TSTB @TKS
BPL 7$
MOVB @TKB,-(SP)
BIC #C177,(SP)
;;ECHO THE CONTROL-G (+G)
;;TYPE CURRENT CONTENTS
;;SAVE SWREG FOR TYPEOUT
;;GO TYPE--OCTAL ASCII(ALL DIGITS)
;;PROMPT FOR NEW SWR
;;CLEAR COUNTER
;;THE NEW SWR
;;CHAR THERE?
;;IF NOT TRY AGAIN
;;PICK UP CHAR
;;MAKE IT 7-BIT ASCII
```

```
4004
4005
4006 021310 021627 000025      9$:    CMP      (SP),#25      ;;IS IT A CONTROL-U?
4007 021314 001005              BNE      10$      ;;BRANCH IF NOT
4008 021316 104401 021720      TYPE    ,#CNTLU      ;;YES, ECHO CONTROL-U (+U)
4009 021322 062706 000006      20$:    ADD      #6,SP      ;;IGNORE PREVIOUS INPUT
4010 021326 000757              BR       19$      ;;LET'S TRY IT AGAIN
4011
4012
4013 021330 021627 000015      10$:    CMP      (SP),#15      ;;IS IT A <CR>?
4014 021334 001022              BNE      16$      ;;BRANCH IF NO
4015 021336 005766 000004      TST      4(SP)      ;;YES, IS IT THE FIRST CHAR?
4016 021342 001403              BEQ      11$      ;;BRANCH IF YES
4017 021344 016677 000002 157474      MOV      2(SP),@SWR      ;;SAVE NEW SWR
4018 021352 062706 000006      11$:    ADD      #6,SP      ;;CLEAR UP STACK
4019 021356 104401 001277      14$:    TYPE    ,#CRLF      ;;ECHO <CR> AND <LF>
4020 021362 126727 000367 000001      CMPB     $INTAG,#1      ;;RE-ENABLE TTY KBD INTERRUPTS?
4021 021370 001003              BNE      15$      ;;BRANCH IF NOT
4022 021372 012777 000100 177106      MOV      #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
4023 021400 000002              RTI              ;;RETURN
4024 021402 004767 176756      15$:    JSR      PC,$TYPEC      ;;ECHO CHAR
4025 021406 021627 000060      16$:    CMP      (SP),#60      ;;CHAR < 0?
4026 021412 002420              BLT      18$      ;;BRANCH IF YES
4027 021414 021627 000067      CMP      (SP),#67      ;;CHAR > 7?
4028 021420 003015              BGT      18$      ;;BRANCH IF YES
4029 021422 042726 000060      BIC      #60,(SP),      ;;STRIP-OFF ASCII
4030 021426 005766 000002      TST      2(SP)      ;;IS THIS THE FIRST CHAR
4031 021432 001403              BEQ      17$      ;;BRANCH IF YES
4032 021434 006316              ASL      (SP)      ;;NO, SHIFT PRESENT
4033 021436 006316              ASL      (SP)      ;;CHAR OVER TO MAKE
4034 021440 006316              ASL      (SP)      ;;ROOM FOR NEW ONE.
4035 021442 005266 000002      17$:    INC      2(SP)      ;;KEEP COUNT OF CHAR
4036 021446 056616 177776      BIS      -2(SP), (SP)      ;;SET IN NEW CHAR
4037 021452 000707              BR       7$      ;;GET THE NEXT ONE
4038 021454 104401 020522      18$:    TYPE    ,#QUES      ;;TYPE ?<CR><LF>
4039 021460 000720              BR       20$      ;;SIMULATE CONTROL-U
4040      .DSABL  LSB
4041
4042
4043      ;*****
4044      ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
4045      ;*CALL:
4046      ;*      RDCHR      ;;INPUT A SINGLE CHARACTER FROM THE TTY
4047      ;*      RETURN HERE  ;;CHARACTER IS ON THE STACK
4048      ;*      ;;WITH PARITY BIT STRIPPED OFF
4049      ;
4050
4051 021462 011646      $RDCHR: MOV      (SP),-(SP)      ;;PUSH DOWN THE PC
4052 021464 016666 000004 000002      MOV      4(SP),2(SP)      ;;SAVE THE PS
4053 021472 105777 177010      1$:    TSTB     @TKS      ;;WAIT FOR
4054 021476 100375              BPL      1$      ;;A CHARACTER
4055 021500 117766 177004 000004      MOVB     @TKB,4(SP)      ;;READ THE TTY
4056 021506 042766 177600 000004      BIC      #C<177>,4(SP)      ;;GET RID OF JUNK IF ANY
4057 021514 026627 000004 000023      CMP      4(SP),#23      ;;IS IT A CONTROL-S?
4058 021522 001013              BNE      3$      ;;BRANCH IF NO
4059 021524 105777 176756      2$:    TSTB     @TKS      ;;WAIT FOR A CHARACTER
```



```

4060 021530 100375      BPL      2%      ;;LOOP UNTIL ITS THERE
4061 021532 117746 176752  MOVB    0%TKB,-(SP)  ;;GET CHARACTER
4062 021536 042716 177600  BIC     0%C177,(SP)  ;;MAKE IT 7-BIT ASCII
4063 021542 022627 000021  CMP     (SP),021  ;;IS IT A CONTROL-Q?
4064 021546 001366      BNE     2%      ;;IF NOT DISCARD IT
4065 021550 000750      BR      1%      ;;YES, RESUME
4066 021552 026627 000004 000140 3%:  CMP     4(SP),0140  ;;IS IT UPPER CASE?
4067 021560 002407      BLT     4%      ;;BRANCH IF YES
4068 021562 026627 000004 000175  CMP     4(SP),0175  ;;IS IT A SPECIAL CHAR?
4069 021570 003003      BGT     4%      ;;BRANCH IF YES
4070 021572 042766 000040 000004  BIC     040,4(SP)  ;;MAKE IT UPPER CASE
4071 021600 000002      4%:  RTI      ;;GO BACK TO USER
4072
4073      ;;*****
4074      ;;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4075      ;;CALL:
4076      ;;      RDLIN      ;;INPUT A STRING FROM THE TTY
4077      ;;      RETURN HERE  ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4078      ;;      ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
4079 021602 010346      $RDLIN: MOV     R3,-(SP)  ;;SAVE R3
4080 021604 012703 021710 1%:  MOV     0%TTYIN,R3  ;;GET ADDRESS
4081 021610 022703 021720 2%:  CMP     0%TTYIN-8,R3  ;;BUFFER FULL?
4082 021614 101405      BLOS    4%      ;;BR IF YES
4083 021616 104410      RDCHR   ;;GO READ ONE CHARACTER FROM THE TTY
4084 021620 112613      MOVB    (SP),R3  ;;GET CHARACTER
4085 021622 122713 000177 10%:  CMPB    0177,R3  ;;IS IT A RUBOUT
4086 021626 001003      BNE     3%      ;;SKIP IF NOT
4087 021630 104401 020522 4%:  TYPE     ,%QUES  ;;TYPE A '?'
4088 021634 000763      BR      1%      ;;CLEAR THE BUFFER AND LOOP
4089 021636 111367 000044 3%:  MOVB    (R3),9%  ;;ECHO THE CHARACTER
4090 021642 104401 021706      TYPE     ,9%
4091 021646 122723 000015      CMPB    015,R3  ;;CHECK FOR RETURN
4092 021652 001356      BNE     2%      ;;LOOP IF NOT RETURN
4093 021654 105063 177777      CLRB    -1(R3)  ;;CLEAR RETURN (THE 15)
4094 021660 104401 020523      TYPE     ,%LF  ;;TYPE A LINE FEED
4095 021664 012603      MOV     (SP),R3  ;;RESTORE R3
4096 021666 011646      MOV     (SP),-(SP)  ;;ADJUST THE STACK AND PUT ADDRESS OF THE
4097 021670 016666 000004 000002  MOV     4(SP),2(SP)  ;;FIRST ASCII CHARACTER ON IT
4098 021676 012766 021710 000004  MOV     0%TTYIN,4(SP)
4099 021704 000002      RTI
4100 021706 000      9%:  .BYTE     0      ;;RETURN
4101 021707 000      .BYTE     0      ;;STORAGE FOR ASCII CHAR. TO TYPE
4102 021710 000010      $TTYIN: .BLKB    8.  ;;TERMINATOR
4103 021720 052536 005015 000      $CNTLU: .ASCIZ  /%U/<15><12>  ;;RESERVE 8 BYTES FOR TTY INPUT
4104 021725 136 006507 000012      $CNTLG: .ASCIZ  /%G/<15><12>  ;;CONTROL "U"
4105 021732 005015 053523 020122      $MSWR: .ASCIZ  <15><12>/SWR = /  ;;CONTROL "G"
4106 021740 020075 000
4107 021743 040 047040 053505      $MNEW: .ASCIZ  / NEW = /
4108 021750 036440 000040
4109 021754 000      $AUTOB: .BYTE     0      ;;AUTO MODE FLAG
4110 021755 000      $INTAG: .BYTE     0      ;;INTERRUPT MODE FLAG
4111      .SBTTL  TRAP DECODER
4112
4113      ;;*****
4114      ;;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
4115      ;;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS

```

```

4116
4117
4118
4119 021756 010046
4120 021760 016600 000002
4121 021764 005740
4122 021766 111000
4123 021770 006300
4124 021772 016000 022012
4125 021776 000200
4126
4127
4128
4129
4130 022000 011646
4131 022002 016666 000004 000002
4132 022010 000002
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 022012 022000
4167 022014 020152
4168 022016 020776
4169 022020 020752
4170 022022 021012
4171 022024 020526

; *OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
; *GO TO THAT ROUTINE.

$TRAP: MOV R0, -(SP) ;:SAVE R0
        MOV 2(SP), R0 ;:GET TRAP ADDRESS
        TST -(R0) ;:BACKUP BY 2
        MOVB (R0), R0 ;:GET RIGHT BYTE OF TRAP
        ASL R0 ;:POSITION FOR INDEXING
        MOV $TRPAD(R0), R0 ;:INDEX TO TABLE
        RTS R0 ;:GO TO ROUTINE

;:THIS IS USE TO HANDLE THE "GETPRI" MACRO

$TRAP2: MOV (SP), -(SP) ;:MOVE THE PC DOWN
        MOV 4(SP), 2(SP) ;:MOVE THE PSW DOWN
        RTI ;:RESTORE THE PSW

.MACRO SETTRAP A,B,MSG
        $$SET A,B,\<TRAP,$TRP>,\<$TRP,<MSG>
.NLIST
$TRP=$TRP+1
.LIST
.ENDM SETTRAP
.MACRO $$SET A,B,C,D,COMNT
.IF EQ $TRP-1
.SBTTL TRAP TABLE

; *THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
; *BY THE "TRAP" INSTRUCTION.

; ROUTINE
; -----
$TRPAD: .WORD $TRAP2
.ENDC
.IIF NDF GNS,.NLIST
        A= C
.IIF NDF GNS,.LIST
        B ;:CALL=A TRAP=D(C) COMNT
.ENDM $$SET
.MACRO TRMTRP
$TERM=-$TRPAD
.ENDM TRMTRP
.SBTTL TRAP TABLE

; *THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
; *BY THE "TRAP" INSTRUCTION.

; ROUTINE
; -----
$TRPAD: .WORD $TRAP2
        $TYPE ;:CALL=TYPE TRAP+1(104401) TTY TYPEOUT ROUTINE
        $TYPOC ;:CALL=TYPOC TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOS ;:CALL=TYPOS TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPON ;:CALL=TYPON TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
        $TYPDS ;:CALL=TYPDS TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)

```


4172										
4173	022026	021250				\$GTSWR	::CALL=GTSWR	TRAP+6(104406)	GET SOFT-SWR SETTING	
4174										
4175	022030	021200				\$CKSWR	::CALL=CKSWR	TRAP+7(104407)	TEST FOR CHANGE IN SOFT-SWR	
4176	022032	021462				\$RDCHR	::CALL=RDCHR	TRAP+10(104410)	TTY TYPEIN CHARACTER ROUTINE	
4177	022034	021602				\$RDLIN	::CALL=RDLIN	TRAP+11(104411)	TTY TYPEIN STRING ROUTINE	
4178										
4179						.SBTTL	APT COMMUNICATIONS ROUTINE			
4180										
4181	022036	112767	000001	000236		\$ATY1:	MOVB	#1,\$FFLG	::TO REPORT FATAL ERROR	
4182	022044	112767	000001	000226		\$ATY3:	MOVB	#1,\$MFLG	::TO TYPE A MESSAGE	
4183	022052	000403					BR	\$ATYC		
4184	022054	112767	000001	000220		\$ATY4:	MOVB	#1,\$FFLG	::TO ONLY REPORT FATAL ERROR	
4185	022062					\$ATYC:				
4186	022062	010046					MOV	RO,-(SP)	::PUSH RO ON STACK	
4187	022064	010146					MOV	R1,-(SP)	::PUSH R1 ON STACK	
4188	022066	105767	000206				TSTB	\$MFLG	::SHOULD TYPE A MESSAGE?	
4189	022072	001450					BEQ	5\$	::IF NOT: BR	
4190	022074	122767	000001	156716			CMPB	#APTEHV,\$ENV	::OPERATING UNDER APT?	
4191	022102	001031					BNE	3\$	::IF NOT: BR	
4192	022104	132767	000100	156707			BITB	#APTSPOOL,\$ENVH	::SHOULD SPOOL MESSAGES?	
4193	022112	001425					BEQ	3\$	::IF NOT: BR	
4194	022114	017600	000004				MOV	#4(SP),RO	::GET MESSAGE ADDR.	
4195	022120	062766	000002	000004			ADD	#2,4(SP)	::BUMP RETURN ADDR.	
4196	022126	005767	156646		1\$:		TST	\$MSGTYPE	::SEE IF DONE W/ LAST XMISSION?	
4197	022132	001375					BNE	1\$	::IF NOT: WAIT	
4198	022134	010067	156654				MOV	RO,\$MSGAD	::PUT ADDR IN MAILBOX	
4199	022140	105720			2\$:		TSTB	(RO).	::FIND END OF MESSAGE	
4200	022142	001376					BNE	2\$		
4201	022144	166700	156644				SUB	\$MSGAD,RO	::SUB START OF MESSAGE	
4202	022150	006200					ASR	RO	::GET MESSAGE LGTH IN WORDS	
4203	022152	010067	156640				MOV	RO,\$MSGLG	::PUT LENGTH IN MAILBOX	
4204	022156	012767	000004	156614			MOV	#4,\$MSGTYPE	::TELL APT TO TAKE MSG.	
4205	022164	000413					BR	5\$		
4206	022166	017667	000004	000016	3\$:		MOV	#4(SP),4\$	::PUT MSG ADDR IN JSR LINKAGE	
4207	022174	062766	000002	000004			ADD	#2,4(SP)	::BUMP RETURN ADDRESS	
4208	022202	016746	155570				MOV	177776,-(SP)	::PUSH 177776 ON STACK	
4209	022206	004767	175740				JSR	PC,\$TYPE	::CALL TYPE MACRO	
4210	022212	000000			4\$:		.WORD	0		
4211	022214				5\$:					
4212	022214	105767	000062		10\$:		TSTB	\$FFLG	::SHOULD REPORT FATAL ERROR?	
4213	022220	001416					BEQ	12\$	::IF NOT: BR	
4214	022222	005767	156572				TST	\$ENV	::RUNNING UNDER APT?	
4215	022226	001413					BEQ	12\$	::IF NOT: BR	
4216	022230	005767	156544		11\$:		TST	\$MSGTYPE	::FINISHED LAST MESSAGE?	
4217	022234	001375					BNE	11\$	::IF NOT: WAIT	
4218	022236	017667	000004	156536			MOV	#4(SP),\$FATAL	::GET ERROR #	
4219	022244	062766	000002	000004			ADD	#2,4(SP)	::BUMP RETURN ADDR.	
4220	022252	005267	156522				INC	\$MSGTYPE	::TELL APT TO TAKE ERROR	
4221	022256	105067	000020		12\$:		CLRB	\$FFLG	::CLEAR FATAL FLAG	
4222	022262	105067	000013				CLRB	\$LFLG	::CLEAR LOG FLAG	
4223	022266	105067	000006				CLRB	\$MFLG	::CLEAR MESSAGE FLAG	
4224	022272	012601					MOV	(SP)+,R1	::POP STACK INTO R1	
4225	022274	012600					MOV	(SP)+,RO	::POP STACK INTO RO	
4226	022276	000207					RTS	PC	::RETURN	
4227	022300	000				\$MFLG:	.BYTE	0	::MESSG. FLAG	

```

4228 022301 000 $LFLG: .BYTE 0 ;;LOG FLAG
4229 022302 000 $FFLG: .BYTE 0 ;;FATAL FLAG
4230 022304 .EVEN
4231 000200 APTSIZE=200
4232 000001 APTENV=001
4233 000100 APTSPool=100
4234 000040 APTCSUP=040
4235 ;;*****
4236 ;THIS ROUTINE WILL INCREMENT THE ERROR COUNT AND THEN PASS THE UNIQUE
4237 ;ERROR NUMBER TO THE APT ERROR ROUTINE TO BE REPORTED TO THE APT SYSTEM.
4238
4239 022304 005267 156542 $ERROR: INC $ERFLG ;INCREMENT ERROR FLAG
4240 022310 001775 BEQ $ERROR ;DON'T LET IT GO TO ZERO
4241 022312 005267 156526 INC ERRCNT ;INCREMENT THE ERROR COUNT
4242 022316 021627 001002 CMP (SP), #1002 ;IS ERROR FROM VECTOR AREA
4243 022322 101010 BHI 1$ ;IF YES THEN
4244 022324 012767 007777 000106 MOV #7777, 3$ ;REPORT AN UNEXPECTED TRAP
4245 022332 012637 001062 MOV (SP)+, @SAVSP1 ;SAVE UNEXPECTED TRAP DATA
4246 022336 012637 001064 MOV (SP)+, @SAVSP2 ;AND RESTORE SP
4247 022342 000430 BR 2$ ;ELSE
4248 022344 017667 000000 000066 1$: MOV @SP, 3$ ;REPORT UNIQUE ERROR NUMBER TO APT
4249 022352 011667 000072 MOV (SP), 101$ ;SAVE ERROR PC
4250 022356 062716 000002 ADD #2, (SP) ;GET OVER UNIQUE ERROR NUMBER FOR RETURN
4251 022362 017637 000000 022372 100$: MOV @SP, @102$
4252 022370 104401 TYPE ;TYPE ERROR MESSAGE
4253 022372 000000 102$: .WORD 0 ;GET OVER ERROR MESSAGE
4254 022374 062716 000002 ADD #2, (SP) ;
4255 022400 104401 001246 TYPE ,ERR1 ;PUSH UNIQUE ERROR NUMBER ON THE STACK
4256 022404 016746 000030 MOV 3$, -(SP) ;TYPE OCTAL ERROR NUMBER
4257 022410 104402 TYPOC ;
4258 022412 104401 001262 TYPE ,ERR2 ;PUSH ERROR PC ON THE STACK
4259 022416 016746 000026 MOV 101$, -(SP) ;TYPE THE ERROR PC
4260 022422 104402 TYPOC ;CHECK TO MAKE SURE WE'RE IN APT MODE
4261 022424 122767 000001 156366 2$: CMPB @APTENV, $ENV ;IF YES THEN
4262 022432 001004 BNE 5$ ;GO REPORT ERROR TO APT
4263 022434 004767 177414 JSR PC, $ATY4 ;STORAGE FOR ERROR NUMBER
4264 022440 000000 3$: .WORD 0 ;LOOP HERE AFTER REPORTING ERROR TO APT
4265 022442 000777 4$: BR 4$ ;IF NOT APT THEN HALT
4266 022444 000000 5$: HALT ;ALLOW RECOVERY FROM HALT
4267 022446 000002 RTI
4268 022450 000000 101$: .WORD 0
4269 022452 $PATCH: .BLKW 10
4270 022452 000010
4271 000001 .END

```


GLOBAL AREAS
KDJ11A.MACMACY11 30A(1052)
20-MAR-84 11:19

20-MAR-84 11:31

PAGE 84
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0083

ABASE = 000000	576			
ABORT0 012214	2383	25090		
ABORT1 012402	2394	25710		
ACDW1 = 000000	576			
ACDW2 = 000000	576			
ACPUOP = 000000	576	591		
ADDTRP 001454	7360	840	869	
ADDW0 = 000000	576			
ADDW1 = 000000	576			
ADDW10 = 000000	576			
ADDW11 = 000000	576			
ADDW12 = 000000	576			
ADDW13 = 000000	576			
ADDW14 = 000000	576			
ADDW15 = 000000	576			
ADDW2 = 000000	576			
ADDW3 = 000000	576			
ADDW4 = 000000	576			
ADDW5 = 000000	576			
ADDW6 = 000000	576			
ADDW7 = 000000	576			
ADDW8 = 000000	576			
ADDW9 = 000000	576			
ADEVCT = 000000	576	582		
ADEVN = 000000	576			
AENV = 000000	576	587		
AENVN = 000000	576	588		
AFATAL = 000000	576	579		
ALLCTR 001056	6180			
AMADR1 = 000000	576			
AMADR2 = 000000	576			
AMADR3 = 000000	576			
AMADR4 = 000000	576			
AMAMS1 = 000000	576			
AMAMS2 = 000000	576			
AMAMS3 = 000000	576			
AMAMS4 = 000000	576			
AMSGAD = 000000	576	584		
AMSGLG = 000000	576	585		
AMSGTY = 000000	576	578		
AMTYP1 = 000000	576			
AMTYP2 = 000000	576			
AMTYP3 = 000000	576			
AMTYP4 = 000000	576			
APASS = 000000	576	581		
APRIOR = 000000	576			
APTCU = 000040	3749	42340		
APTENV = 000001	3742	4190	42320	4261
APTSIZ = 000200	801	42310		
APTSP0 = 000100	3744	4192	42330	
ASWREG = 002000	4950	576	589	
ATESTN = 000000	576	580		
AUNIT = 000000	576	583		
AUSWR = 000000	576	590		
AVECT1 = 000000	576			
AVECT2 = 000000	576			

[illegible]


```
FLO      001110
FLOAT    001100
FPPTST=  000001
FPVEC   = 000244
GNS      = ***** U
GTSWR   = 104406
HITMIS=  177752
HT       = 000011
HYBRID=  000001
IOTVEC=  000020
KDPAR0=  172360
KDPAR1=  172362
KDPAR2=  172364
KDPAR3=  172366
KDPAR4=  172370
KDPAR5=  172372
KDPAR6=  172374
KDPAR7=  172376
KDPDR0=  172320
KDPDR1=  172322
```

1359	1362	1367	1370	1373	1376	1412	1415	1418	1421	1426	1429	1432
1435	1440	1443	1446	1449	1454	1457	1460	1463	1469	1472	1475	1478
1483	1486	1489	1492	1508	1511	1517	1520	1526	1529	1535	1538	1551
1554	1557	1560	1563	1566	1569	1572	1600	1603	1608	1611	1615	1618
1621	1624	1627	1630	1634	1637	1640	1643	1651	1654	1658	1661	1664
1667	1671	1674	1677	1680	1686	1689	1693	1696	1699	1702	1705	1708
1736	1739	1744	1747	1751	1754	1757	1760	1763	1766	1770	1773	1776
1779	1787	1790	1794	1797	1800	1803	1807	1810	1813	1816	1824	1827
1831	1834	1837	1840	1844	1847	1850	1853	1859	1862	1866	1869	1872
1875	1878	1881	1910	1913	1918	1921	1925	1928	1931	1934	1937	1940
1944	1947	1955	1958	1962	1965	1968	1971	1975	1978	1986	1989	1993
1996	1999	2002	2006	2009	2015	2018	2022	2025	2028	2031	2060	2063
2068	2071	2075	2078	2081	2084	2087	2090	2094	2097	2105	2108	2112
2115	2118	2121	2125	2128	2136	2139	2143	2146	2149	2152	2156	2159
2165	2168	2172	2175	2178	2181	2256	2259	2263	2266	2269	2272	2275
2278	2288	2291	2295	2298	2301	2304	2333	2336	2340	2343	2346	2349
2352	2355	2359	2362	2424	2427	2431	2434	2438	2441	2444	2447	2450
2453	2660	2663	2666	2669	2672	2675	2678	2681	2725	2728	2737	2740
2811	2814	2817	2820	2823	2826	2861	2864	2883	2886	3002	3005	3018
3021	3033	3036	3044	3047	3055	3058	3066	3069	3118	3121	3181	3184
3186	3190	3313	3316	3318	3321	3323	3326	3391	3394	3412	3415	3428
3431	3457	3460	3465	3468	3473	3476	3485	3489	3496	3499	3506	3509
3516	3519	3524	3528	3532	3535	3538	3541	3544	3547	3550	3553	3556
3559	3562	3565	3568	3571	3576	3579	3588	3591	3594	3597	3600	3603
3614	3618	3622	3625	3628	3631	3634	3637	3640	3643	3646	3649	3654
3657												
492												
311	779	780	791									
668	4255											
670	4258											
608												
611	741	762	838	863	1172	1265	1503	1547	1581	1719	1892	2042
2192	2218	2230	2250	2254	2286	2315	2326	2330	2375	2415	2422	2429
2454	2599	2607	2620	2631	2644	2658	2695	2720	2723	2731	2735	2741
2752	2763	2770	2777	2784	2791	2798	2809	2841	3103	3338	3445	3582
3586												
630	893	894	934	978	991	1004	1017	1052				
629	880	885	935	948	963							

[illegible]

SR2	= 177576	3330							
SR3	= 172516	3340	2991*	3009*	3025*	3096*	3660*		
STACK	= 001000	2160	768						
START	001604	521	527	7650					
STBOT	= 001000	4990	1583	1586	3005	3072			
STKLMT	= 177774	2270							
SWR	001046	6130	781*	783	789*	803*	815	2970	3980
SWREG	000176	5250	789	815	3980	3993			4017*
SW0	= 000001	2800							
SW00	= 000001	2700	280						
SW01	= 000002	2690	279						
SW02	= 000004	2680	278						
SW03	= 000010	2670	277						
SW04	= 000020	2660	276						
SW05	= 000040	2650	275						
SW06	= 000100	2640	274						
SW07	= 000200	2630	273						
SW08	= 000400	2620	272						
SW09	= 001000	2610	271						
SW1	= 000002	2790							
SW10	= 002000	2600							
SW11	= 004000	2590							
SW12	= 010000	2580							
SW13	= 020000	2570							
SW14	= 040000	2560							
SW15	= 100000	2550							
SW2	= 000004	2780							
SW3	= 000010	2770							
SW4	= 000020	2760							
SW5	= 000040	2750							
SW6	= 000100	2740							
SW7	= 000200	2730							
SW8	= 000400	2720							
SW9	= 001000	2710							
TAGRAM	= 000001	10	810						
TBITVE	= 000014	3130							
TKVEC	= 000060	3200							
TMM16A	017214	3455	35300						
TMM16B	017054	3453	34930						
TMM16C	017110	3461	35030						
TMM16D	017144	3469	35130						
TMM16E	017200	3446	35230	3542	3548				
TMM16F	017202	35240	3581						
TM16A	017710	3490	3529	3619	3653	36580			
TPVEC	= 000064	3210							
TRAPVE	= 000034	3190	772*	773*					
TRTVEC	= 000014	3140							
TSMA	011214	2287	22920						
TSMB	011240	2294	22990						
TSMC	011254	2300	23050						
TSMMU1	002100	8300							
TSMMU2	002174	8550							
TSMMU3	003550	11650							
TSMMU4	004114	12580							
TSMMU5	005210	14960							
TSMMU6	005366	15400							

[illegible]

TYP0C = 104402	4087	4090	4094	4167#	4252	4255	4258
TYP0N = 104404	3994	4168#	4257	4260			
TYP0S = 104403	4170#						
T10FIN 013064	4169#						
T11FIN 013336	2653	2687#					
T12FIN 013742	2712	2744#					
T13FIN 014366	2805	2831#					
T14 015524	2889	2959#					
T14FIN 016106	3152	3172#					
T15 016474	2974	3161	3199	3289#			
T15A 016556	3345	3353	3362	3403#			
T15FIN 016610	3367	3371	3375	3424#			
UDPAR0= 177660	3379	3436#					
UDPAR1= 177662	371#						
UDPAR2= 177664	372#						
UDPAR3= 177666	373#						
UDPAR4= 177670	374#						
UDPAR5= 177672	375#						
UDPAR6= 177674	376#						
UDPAR7= 177676	377#						
UDPDR0= 177620	378#						
UDPDR1= 177622	349#	1594*	1904*				
UDPDR2= 177624	350#						
UDPDR3= 177626	351#						
UDPDR4= 177630	352#						
UDPDR5= 177632	353#						
UDPDR6= 177634	354#						
UDPDR7= 177636	355#						
UIPAR0= 177640	356#						
UIPAR1= 177642	360#						
UIPAR2= 177644	361#						
UIPAR3= 177646	362#						
UIPAR4= 177650	363#						
UIPAR5= 177652	364#						
UIPAR6= 177654	365#						
UIPAR7= 177656	366#						
UIPDR0= 177600	367#						
UIPDR1= 177602	338#	1730*	2054*				
UIPDR2= 177604	339#						
UIPDR3= 177606	340#						
UIPDR4= 177610	341#						
UIPDR5= 177612	342#						
UIPDR6= 177614	343#						
UIPDR7= 177616	344#						
VIR1 014250	345#						
VIR2 015736	2851	2919#					
VIR3 016054	3131	3234#					
XBUF = 177566	3109	3133	3273#				
XCSR = 177564	491#						
\$APTHD 000204	490#						
\$ASTAT= ***** U	547	553#					
\$ATYC 022062	4212	4227					
\$ATY1 022036	4183	4185#					
\$ATY3 022044	4181#						
\$ATY4 022054	3747	4182#					
	4184#	4263					

[illegible]

[illegible]

ADOPMS	29600	2963													
BGNMOD	1890	823													
BGNSUB	1900														
BGNTST	1890	834	859	1169	1262	1500	1544	1578	1716	1889	2039	2189	2312	2372	2596
CKLOOP	2692	2749	2838	2968	3082	3298	3335	3442							
COMMEN	1890														
DEFPRG	3230														
ENDCOM	10														
ENDMOD	3230														
ENDPAS	1890	3666													
ENDSUB	36680	3690													
ENDTST	1900														
ERRDEF	1890	853	1164	1257	1495	1538	1572	1710	1883	2033	2183	2245	2367	2408	2652
ERRDF	2711	2804	2887	3075	3160	3329	3378	3664							
	1880														
	1890	743	849	897	903	913	926	942	957	970	983	996	1009	1021	1032
	1044	1058	1065	1076	1083	1091	1098	1108	1115	1125	1132	1142	1149	1157	1184
	1192	1200	1208	1217	1225	1242	1251	1296	1302	1310	1316	1324	1330	1338	1344
	1353	1359	1367	1373	1412	1418	1426	1432	1440	1446	1454	1460	1469	1475	1483
	1489	1508	1517	1526	1535	1551	1557	1563	1569	1600	1608	1615	1621	1627	1634
	1640	1651	1658	1664	1671	1677	1686	1693	1699	1705	1736	1744	1751	1757	1763
	1770	1776	1787	1794	1800	1807	1813	1824	1831	1837	1844	1850	1859	1866	1872
	1878	1910	1918	1925	1931	1937	1944	1955	1962	1968	1975	1986	1993	1999	2006
	2015	2022	2028	2060	2068	2075	2081	2087	2094	2105	2112	2118	2125	2136	2143
	2149	2156	2165	2172	2178	2256	2263	2269	2275	2288	2295	2301	2333	2340	2346
	2352	2359	2424	2431	2438	2444	2450	2660	2666	2672	2678	2725	2737	2811	2817
	2823	2861	2883	3002	3018	3033	3044	3055	3066	3118	3181	3186	3313	3318	3323
	3391	3412	3428	3457	3465	3473	3485	3496	3506	3516	3524	3532	3538	3544	3550
ERROR	3556	3562	3568	3576	3588	3594	3600	3614	3622	3628	3634	3640	3646	3654	
	2170	743	849	897	903	913	926	942	957	970	983	996	1009	1021	1032
	1044	1058	1065	1076	1083	1091	1098	1108	1115	1125	1132	1142	1149	1157	1184
	1192	1200	1208	1217	1225	1242	1251	1296	1302	1310	1316	1324	1330	1338	1344
	1353	1359	1367	1373	1412	1418	1426	1432	1440	1446	1454	1460	1469	1475	1483
	1489	1508	1517	1526	1535	1551	1557	1563	1569	1600	1608	1615	1621	1627	1634
	1640	1651	1658	1664	1671	1677	1686	1693	1699	1705	1736	1744	1751	1757	1763
	1770	1776	1787	1794	1800	1807	1813	1824	1831	1837	1844	1850	1859	1866	1872
	1878	1910	1918	1925	1931	1937	1944	1955	1962	1968	1975	1986	1993	1999	2006
	2015	2022	2028	2060	2068	2075	2081	2087	2094	2105	2112	2118	2125	2136	2143
	2149	2156	2165	2172	2178	2256	2263	2269	2275	2288	2295	2301	2333	2340	2346
	2352	2359	2424	2431	2438	2444	2450	2660	2666	2672	2678	2725	2737	2811	2817
	2823	2861	2883	3002	3018	3033	3044	3055	3066	3118	3181	3187	3313	3318	3323
	3391	3412	3428	3457	3465	3473	3486	3496	3506	3516	3525	3532	3538	3544	3550
	3556	3562	3568	3576	3588	3594	3600	3615	3622	3628	3634	3640	3646	3654	
ESCAPE	3230														
FRONT	10														
FRONT1	10	1850													
GETPRI	3230														
GETSWR	3230	810													
IDMSG	36680	3678													
MMMS13	28320	2835													
MMMS14	30760	3079													
MULT	3230														
NEWTST	1880	3230	830	855	1165	1258	1496	1540	1574	1712	1885	2035	2185	2308	2368
	2592	2688	2745	2832	2960	3076	3290	3331	3438						
NXMMMS	32900	3292													
POP	3230	3879	4224	4225											

PUSH	3230	3938	4185	4187	4208										
REPORT	3230														
SCOPE	2180														
SETPRI	3230														
SETTRA	41340	4159	4168	4169	4170	4171	4173	4175	4176	4177					
SETUP	1880	3230	767												
SKIP	3230														
SLASH	3230														
SPACE	3230														
STARS	3230	530	540	542	549	575	831	833	856	858	1166	1168	1259	1261	1497
	1499	1541	1543	1575	1577	1713	1715	1886	1888	2036	2038	2186	2188	2309	2311
	2369	2371	2593	2595	2689	2691	2746	2748	2833	2835	2837	2961	2963	2967	3077
	3079	3081	3290	3292	3297	3332	3334	3439	3441	3671	3721	3828	3895	3972	3975
	4043	4072	4113	4180	4235										
SWRSU	3230	7770													
TAIL	10	3668													
TRMTRP	41560														
TYPBIN	3230														
TYPDEC	3230	3691													
TYPNAM	3230														
TYPNUM	3230														
TYPOCS	3230														
TYPOCT	3230	3993													
TYPTXT	3230														
\$\$ESCA	3230														
\$\$NEWT	3230	831	856	1166	1259	1497	1541	1575	1713	1886	2036	2186	2309	2369	2593
	2689	2746	2833	2961	3077	3290	3332	3439							
\$\$SET	41400	4159	4168	4169	4170	4171	4173	4175	4176	4177					
\$\$SETM	7930	800													
\$\$SKIP	3230														
.EQUAT	1880	213													
.HEADE	1900	195													
.KT1.1	1880	323													
.SETUP	1900	500													
.\$ACT1	1900	528													
.\$APT8	1880	573													
.\$APTH	1900	538													
.\$APTY	1910	4178													
.\$EOP	1880	3669													
.\$ERRO	1910														
.\$READ	1910	3970													
.\$TRAP	1900	4111													
.\$TYPD	1890	3826													
.\$TYPE	1890	3719													
.\$TYPO	1910	3893													
.\$4OCA	1880	501													

. ABS. 022472 000

ERRORS DETECTED: 0

CZKDKB/EN:ABS,CZKDKB.SEG/CRF/DOC/SOL/NL:TOC=SYSMAC.SML/ML,CZKDKB.MAC/ML,KDJ11A.MAC
 RUN-TIME: 297 115 5 SECONDS
 RUN-TIME RATIO: 941/418=2.2

CORE USED: 52K (103 PAGES)

DOCUMENT PAGES: 96