

The image displays a grid of 100 small, individual diagrams or data tables, arranged in 10 rows and 10 columns. Each cell contains technical information, likely related to tape diagnostics. The content within each cell is dense and difficult to read due to the small size and low contrast, but it appears to consist of various data points, possibly including error codes, timing diagrams, or component specifications. The overall layout is a structured grid of these small units.

B01

DZTRA A MACY11 27,1006) 22-SEP-76 12:30 PAGE 2
DZTRA.HED 13-APR-76 00:00 SPECIAL TEST MACROS

.REM :

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTRA-A-D
PRODUCT NAME: TR79F TAPE DIAGNOSTIC
DATE RELEASED: 21-APRIL-1976
MAINTAINER: DIAGNOSTIC GROUP

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

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

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

COPYRIGHT (C) 1976
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

A2. REQUIREMENTS

- A. FDP-11 COMPUTER
- B. MAGNETIC TAPE TRANSPORT CONTROL INTERFACE
- C. 12K WORDS OF MEMORY
- D. CONSOLE TELEPRINTER
- E. PAPER TAPE READER

A3. LOADING PROCEDURE

THE DIAGNOSTIC IS IN ABSOLUTE BINARY FORMAT AND IS LOADED USING THE ABSOLUTE LOADER.

A4. STARTING PROCEDURE

- A. LOAD ADDRESS 200
- B. DEPRESS START

D01

DZTRAA MACY(1 27.1006) 22-SEP-76 12:30 PAGE 4
DZTRA.HED 13-APR-76 00:00 SPECIAL TEST MACROS

TEST SELECTION:

AFTER THE TYPEOUT "SELECT TEST NUMBER..." THE OPERATOR
SHOULD SELECT THE DESIRED TEST BY TYPING THE NUMBER FOLLOWED BY
A CARRIAGE RETURN.

NOTE: THE NORMAL SWITCH SETTINGS ARE ALL SWITCHES DOWN

E01

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 5
DZTRA.HED 13-APR-76 00:00 SPECIAL TEST MACROS

A6. TEST GROUP ABSTRACT

PRETST: PRETEST

THIS TEST GROUP CHECKS THE RESET FUNCTION. IT IS PERFORMED EACH TIME A TEST GROUP IS SELECTED.

NOTE: IN THE EVENT OF ERRORS IN THE PRETEST GROUP IT IS RECOMMENDED THAT THE RESET SWITCH ON THE TAPE TRANSPORT BE DEPRESSED AND THE TEST GROUP BE RESTARTED AT THE RESTART ADDRESS.

TEST 1

TEST ONE IS A MANUAL INTERVENTION TEST REQUIRING THE OPERATOR TO PERFORM THE TASKS AS DIRECTED BY THE TTY. IT IS SUGGESTED THAT THE EOT STICKER BE PLACED ABOUT 50 FEET FROM THE LOAD POINT.

TEST 2

TEST GROUP TWO IS A LOGIC TEST GROUP WHICH CHECKS THE INTERFACE LOGIC.

TEST 3

TEST GROUP THREE IS A LOGIC TEST GROUP WHICH TESTS THE TRANSPORTS RESPONSE TO ALL TYPES OF COMMANDS.

TEST 4

TEST GROUP FOUR IS A RELIABILITY TEST GROUP WHICH CHECKS THE ABILITY OF THE TRANSPORT AND THE INTERFACE OVER AN EXTENDED PERIOD OF TIME.

TEST 5

TEST GROUP FIVE IS THE READ PART OF A COMPATIBILITY TEST. THIS TEST GROUP READS TAPES WRITTEN BY TEST GROUP 4.

TEST 6

TEST GROUP SIX IS A MAINTENANCE AID THIS ROUTINE ALLOWS THE OPERATOR TO LOOP ON ANY SUBTEST BY ENTERING THE STARTING ADDRESS AT THE TTY. THE LOOP CONTROL WILL KEEP THE SELECTED TEST GROUP RUNNING IF LOOP IS NOT SELECTED THE TEST GROUP WILL EXECUTE ONCE AND ANOTHER STARTING ADDRESS MUST BE ENTERED.

NOTE: IF ANY ERRORS DETECTED IN BOTH TEST GROUP 3 AND 4 HAVE THE SAME PASS NUMBER A DEFECTIVE TAPE SHOULD BE THE FIRST CONSIDERATION. IF SWITCH #3 IS DOWN THEN TESTS 2 THROUGH 5 ARE LOOPED ON WHERE NO OPERATOR INTERVENTION IS REQUIRED. IF SWITCH #3 IS UP THEN ALL TESTS ARE PERFORMED.

A7. ERRORS

THERE ARE TWO BASIC ERROR TYPEOUTS ONE OF WHICH CONTAINS THE FOLLOWING INFORMATION. THE OTHER ONLY GIVES THE PASS, PC, AND SR. WHEN THIS SHORT ERROR PRINTOUT IS TYPED THE MESSAGE WHICH FOLLOWS WILL GIVE AMPLE INFORMATION ABOUT THE ERROR.

PASS	XXX				
SUBTEST	PC	STATUS	CORRECT	ACTUAL	LOCATION
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX

PASS= XXXXXX (THIS IS THE CURRENT PASS NUMBER)
SUBTEST=XXXXXX (THIS IS THE STARTING ADDRESS OF THE
CURRENT SUBTEST BEING EXECUTED)
PC= XXXXXX (THIS IS THE PC FROM WHICH ERROR WAS CALLED)
STATUS= XXXXXX (THIS IS THE CONTENTS OF THE PROCESSOR STATUS
REGISTER AT THE TIME OF THE ERROR)
CORRECT=XXXXXX (THIS IS WHAT SHOULD HAVE BEEN IN THE
REGISTER OR LOCATION BEING TESTED)
ACTUAL= XXXXXX (THIS IS WHAT WAS IN THE REGISTER OR
LOCATION AT THE TIME IT WAS EXAMINED)
LOCATION=XXXXXX (THIS IS THE LOCATION THAT WAS IN ERROR)
COMMENT (OPTIONAL)

GO1

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 7
DZTRA.HED 13-APR-76 00:00

INTRODUCTION TO TR79 DIAGNOSTIC

:*MAINDEC-11-DZTRAA /<377>/TR79F CHECKOUT TESTS
:*COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
:-----*

: STARTING PROCEDURE
: LOAD PROGRAM
: LOAD ADDRESS 000200
: PRESS START
: PROGRAM WILL TYPE "MAINDEC-11-DZTRAA /<377>/TR79F CHECKOUT TESTS"
: PROGRAM WILL TYPE "RUNNING" TO INDICATE THAT TESTING HAS STARTED
: AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
: AND THEN RESUME TESTING

: SWITCH REGISTER OPTIONS

:-----*

195	100000	SW15=100000	:=1, HALT ON ERROR
196	040000	SW14=40000	:=1, LOOP ON CURRENT TEST GROUP
197	020000	SW13=20000	:=1, INHIBIT ERROR TYPEOUT
198	010000	SW12=10000	:=1, BELL ON ERROR.
199	004000	SW11=4000	:=1, INHIBIT ITERATIONS
200	002000	SW10=2000	:=1, ESCAPE TO NEXT TEST ON ERROR
201	001000	SW09=1000	:=1, LOOP WITH CURRENT DATA
202	000400	SW08=400	:=1, LOOP ON ERROR
203			:NOTE...FOR TEST 4&5 LOOPS BACK TO BEGINING.
204	000200	SW07=200	
205	000100	SW06=100	
206	000040	SW05=40	
207	000020	SW04=20	
208	000010	SW03=10	:=0, LOOPS ON TESTS 2 THRU 5.
209			:=1, RUNS TESTS 1 THRU 6.
210			
211	000004	SW02=4	:=1, LOCK ON TEST SELECT
212	000002	SW01=2	:=1, RESTART PROGRAM AT SELECTED TEST
213	000001	SW00=1	:=1, SELECT DEVICE ADDRESS, VECTOR, ETC.

H01

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 8
DZTRA.HED 13-APR-76 00:00 GENERAL DEFINITIONS AND EQUIVALENCES

```
214  
215  
216 ;REGISTER DEFINITIONS  
217 -----  
218  
219 000000 RO=%0 ;GENERAL REGISTER  
220 000001 R1=%1 ;GENERAL REGISTER  
221 000002 R2=%2 ;GENERAL REGISTER  
222 000003 R3=%3 ;GENERAL REGISTER  
223 000004 R4=%4 ;GENERAL REGISTER  
224 000005 R5=%5 ;GENERAL REGISTER  
225 000006 SP=%6 ;PROCESSOR STACK POINTER  
226 000007 PC=%7 ;PROGRAM COUNTER  
227  
228 ;LOCATION EQUIVALENCES  
229 -----  
230  
231 177776 PS=177776 ;PROCESSOR STATUS WORD  
232 001200 STACK=1200 ;START OF PROCESSOR STACK  
233  
234 ;INSTRUCTION DEFINITIONS  
235 -----  
236  
237 005746 PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD  
238 005726 POP1SP=5726 ;INCREMENT PROCESSOR STACK 1 WORD  
239 010046 PUSHRO=10046 ;SAVE R0 ON STACK  
240 012600 POPRO=12600 ;RESTORE R0 FROM STACK  
241 024646 PUSH2SP=24646 ;DECREMENT STACK TWICE  
242 022626 POP2SP=22626 ;INCREMENT STACK TWICE  
243 .EQUIV EMT,ERROR ;BASIC DEFINITION OF ERROR CALL  
244  
245 ;BIT DEFINITIONS  
246 -----  
247  
248 100000 BIT15=100000  
249 040000 BIT14=40000  
250 020000 BIT13=20000  
251 010000 BIT12=10000  
252 004000 BIT11=4000  
253 002000 BIT10=2000  
254 001000 BIT9=1000  
255 000400 BIT8=400  
256 000200 BIT7=200  
257 000100 BIT6=100  
258 000040 BIT5=40  
259 000020 BIT4=20  
260 000010 BIT3=10  
261 000004 BIT2=4  
262 000002 BIT1=2  
263 000001 BIT0=1  
264  
265 ;PROCESSOR PRIORITY LEVELS  
266 -----  
267  
268 000340 LEVEL7=340  
269 000300 LEVEL6=300
```


I01

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 9
DZTRA.HED 13-APR-76 00:00

GENERAL DEFINITIONS AND EQUIVALENCES

270	000240	LEVEL5=240
271	000200	LEVEL4=200
272	000140	LEVEL3=140
273	000100	LEVEL2=100
274	000040	LEVEL1=040
275	000000	LEVEL0=000

;TR79 CONTROL AND STATUS REGISTER DEFINITIONS
;(TRCR) BIT DEFINITIONS

;TAPE TEST EQUALS

			UNIT	COMMAND
282	000000	ILC00 =0	0	ILLEGAL COMMAND
283	000402	WRITE =402	1	WRITE FORWARD
284	000004	READ =4	0	READ FORWARD
285	000406	ILC03 =406	1	ILLEGAL COMMAND
286	001410	SPACER =1410	3	SPACE REVERSE 1 RECORD
287	001012	ILC05 =1012	2	ILLEGAL COMMAND
288	001414	ILC06 =1414	3	ILLEGAL COMMAND
289	000016	ERASE =16	0	ERASE WORD COUNT FORWARD
290	001020	REWIND =1020	1	REWIND TAPE
291	000022	ILC11 =22	0	ILLEGAL COMMAND
292	000424	ILC12 =424	1	ILLEGAL COMMAND
293	004000	PWRCLR=BIT11		DEFINITION OF POWER CLEAR BIT
294	001026	GOEOT =1026	2	FAST FORWARD TO EOT
295	001030	ILC14 =1030	2	ILLEGAL COMMAND
296	000432	WIDB =432	1	WRITE I.D. BLOCK
297	001034	WEOF =1034	2	WRITE END OF FILE
298	001436	OFLINE =1436	3	TAKE UNIT OFF UNIT

301	164000	TCR=	164000	:DEFAULT CONTROL REGISTER ADDRESS
302	164002	TSR=	164002	:DEFAULT STATUS REGISTER ADDRESS
303	164004	TWC=	164004	:DEFAULT WORD COUNT ADDRESS
304	164006	TBA=	164006	:DEFAULT BUS ADDRESS ADDRESS
305	000170	TVA=	170	:DEFAULT TAPE VECTOR ADDRESS
306	000172	TSA=	172	:DEFAULT TAPE STATUS ADDRESS

J01

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 10
 DZTRA.HED 13-APR-76 00:00

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```

308      ;:*****
309      ;-----
310      ; TRAPCATCHER FOR ILLEGAL INTERRUPTS
311      ; THE STANDARD "TRAP CATCHER" IS PLACED
312      ; BETWEEN ADDRESS 0 TO ADDRESS 776.
313      ; IT LOOKS LIKE "PC+2 HALT".
314      ;-----
315      ;:*****
316
317      00000C      .=0
318      ; STANDARD INTERRUPT VECTORS
319      ;-----
320
321      000024      .=24
322      000024      004262      .PFAIL      ; POWER FAIL HANDLER
323      000026      000340      340          ; SERVICE AT LEVEL 7
324      000030      003704      .EROR      ; ERROR HANDLER
325      000032      000340      340          ; SERVICE AT LEVEL 7
326      000034      003640      .TRPSRV     ; GENERAL HANDLER DISPATCH SERVICE
327      000036      000340      340          ; SERVICE AT LEVEL 7
328
329      000040      000040      .=40
330      000042      000000      0          ; SAVE FOR ACT-11 OR XXDP
331      000044      000000      0          ; RETURN ADDRESS IF UNDER ACT-11 OR XXDP
332      000046      000000      0          ; SAVE FOR ACT-11 OR XXDP
333      000052      000052      .=52      $ENDAD      ; FOR USE WITH ACT-11 OR XXDP
334      000052      000000      0          ; ACT-1: PROGRAM CHARACTERISTICS
335
336      000174      000174      .=174
337      000174      000000      DISPREG:0    ; SOFTWARE DISPLAY REGISTER FOR SMALL 115
338      000176      000000      SWREG: 0      ; SOWTARE SWITCH REGISTER FOR SMALL 115
339      000200      000200      .=200
340      000200      000137      001510      JMP      .START      ; GO TO START OF PROGRAM
341
342
343
344      001000      001000      040515 047111  .=1000      MTITLE: .ASCIZ <377><12>/MAINDEC-11-DZTRAA '<377>/TR79F CHECKOUT TESTS/<377>
345      (2)
346      001200      001200      .=1200
347      001202      177570      DISPLAY:177570
348      001202      177570      SWR: 177570
349      ; INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
350      ;-----
351      001204      177560      TKCSR: 177560      ; TELETYPE KEYBOARD CONTROL REGISTER
352      001206      177562      TKDBR: 177562      ; TELETYPE KEYBOARD DATA BUFFER
353      001210      177564      TPCSR: 177564      ; TELEPRINTER CONTROL REGISTER
354      001212      177566      TPDBR: 177566      ; TELEPRINTER DATA BUFFER
355
356      ; PROGRAM CONTROL PARAMETERS
357      ;-----
358
359      001214      000000      RETURN: 0          ; SCOPE ADDRESS FOR LOOP ON TEST
360      001216      000000      NEXT: 0           ; ADDRESS OF NEXT TEST TO BE EXECUTED
361      001220      000000      LOCK: 0          ; ADDRESS FOR LOCK ON CURRENT DATA
362      001222      000001      ICOUNT: 1        ; NUMBER OF ITERATIONS THAT CLURRENT TEST WILL BE EXECUTED
  
```

K01

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 11
DZTRA.HED 13-APR-76 00:00 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
363 001224 000000 LPCNT: 0 ;NUMBER OF ITERATIONS COMPLETED
364 001226 000000 TSTNO: 0 ;NUMBER OF TEST IN PROGRESS
365 001230 000000 PASCNT: 0 ;NUMBER OF PASSES COMPLETED
366 001232 000000 ERRCNT: 0 ;TOTAL NUMBER OF ERRORS
367 001234 000000 LSTERR: 0 ;PC OF LAST ERROR CALL
368 001236 000000 SAVLIN: .WORD ;PROGRAM PARAMETER
369
370 ;PROGRAM VARIABLES
371 -----
372
373 001240 000010 HOLD: 10 ;DEFAULT DELAY TIME FOR 11/70 PROCESSOR
374 001242 000000 TRBASE: 0 ;BASE ADDRESS OF A GIVEN TR79 UNDER TEST
375 001244 000000 REGIST: 0 ;DEVICE ADDRESS STORAGE LOCATION
376 001246 000000 STAT: 0 ;TR STATUS WORD STORAGE
377 001250 000000 TSTPTR: 0
378 001252 000000 ERR: 0
379 001254 000000 XPC: 0
380 001256 000000 XSR: 0
381 001260 000000 COR: 0
382 001262 000000 ACT: 0
383 001264 000000 WRTFLG: 0
384 001266 000000 LOC: 0
385 001270 000215 CR: 215 ;TTY CARRIAGE RETURN
386 001272 000212 LF: 212 ;TTY LINE FEED
387 001274 000377 RO: 377 ;TTY RUBOUT
388 001276 000000 XTEST: 0
389 001300 000000 ZERO: 0
390 001302 000000 TC: 0 ;CONTROL STORAGE
391 001304 000000 TEMP1: 0 ;TEMPORARY STORAGE
392 001306 000000 TEMP2: 0 ;TEMPORARY STORAGE
393 001310 000000 TEMP3: 0 ;TEMPORARY STORAGE
394 001312 000000 TEMP4: 0 ;TEMPORARY STORAGE
395 001314 000000 TEMP5: 0 ;TEMPORARY STORAGE
396 001316 000000 SAVR0: 0 ;R0 STORAGE
397 001320 000000 SAVR1: 0 ;R1 STORAGE
398 001322 000000 SAVR2: 0 ;R2 STORAGE
399 001324 000000 SAVR3: 0 ;R3 STORAGE
400 001326 000000 SAVR4: 0 ;R4 STORAGE
401 001330 000000 SAVR5: 0 ;R5 STORAGE
402 001332 000000 SAVSP: 0 ;STACK POINTER STORAGE
403 001334 000000 SAVPC: 0 ;PROGRAM COUNTER STORAGE
404 .EVEN
405 001336 001500 ACTIVE: TR.MAF ;TABLE POINTER.
```

L01

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 12
DZTRA.HED 13-APR-76 00:00

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

406
407 ;PROGRAM CONTROL FLAGS
408 ;-----
409
410 001340 000 INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
411 001341 000 ERAFLG: .BYTE 0 ;ERROR OCCURRED FLAG
412 001342 000 LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
413 001343 000 QV.FLG: .BYTE 0 ;QUICK VERIFY FLAG.
414 001344 000 MNTFLG: .BYTE 0 ;MAINTENANCE BIT SET FLAG
415 ;ON FIRST PASS OF EACH TR79 ITERATIONS WILL BE SUPPRESSE
416 001346 .EVEN
417
418 ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
419 ;POINTERS TO SUBROUTINES CAN BE FOUND
420 ;IN THE TABLE IMMEDIATELY FOLLOWING THE DEFINITIONS
421
422 ;:*****
423 ;-----
424 001346 .TRPTAB:
425 104400 SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
426 001346 002556 .SCOPE
427 104401 SCOPE1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
428 001350 002704 .SCOPE1
429 104402 TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
430 001352 002730 .TYPE
431 104403 TYPEF=TRAP+3 ;CALL TO FAILURE MESSAGE OUTPUT ROUTINE
432 001354 003004 .TYPEF
433 104404 TYPEL=TRAP+4 ;CALL TO REPEAT MESSAGE OUTPUT ROUTINE
434 001356 003016 .TYPEL
435 104405 INSTR=TRAP+5 ;CALL TO ASCII STRING INPUT ROUTINE
436 001360 003040 .INSTR
437 104406 INSTER=TRAP+6 ;CALL TO INPUT ERROR HANDLER
438 001362 003144 .INSTER
439 104407 PARAM=TRAP+7 ;CALL TO NUMERICAL DATA INPUT ROUTINE
440 001364 003164 .PARAM
441 104410 SAVOS=TRAP+10 ;CALL TO REGISTER SAVE ROUTINE
442 001366 003364 .SAVOS
443 104411 RESOS=TRAP+11 ;CALL TO REGISTER RESTORE ROUTINE
444 001370 003424 .RESOS
445 104412 CONVRT=TRAP+12 ;CALL TO DATA OUTPUT ROUTINE
446 001372 003456 .CONVRT
447 104413 CNVRT=TRAP+13 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
448 001374 003462 .CNVRT
449
450 ;:*****
451

```

MO1

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 13
DZTRA.HED 13-APR-76 00:00

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

452					; R79 VECTOR AND REGISTER INDIRECT POINTERS
453					; WORKING AREA
454					
455	001376	164000	TRCR:	164000	; R/W
456	001400	164001	HTRCR:	164001	; R/W
457	001402	164002	TRSR:	164002	; READ ONLY
458	001404	164003	HTRSR:	164003	; READ ONLY
459	001406	164004	TRWC:	164004	; R/W
460	001410	164005	HTRWC:	164005	; R/W
461	001412	164006	TRBA:	164006	; READ ONLY
462					; DEFAULT TR VECTORS
463	001414	000170	TRVCT:	170	; REC INTR VECTOR
464	001416	000172	TRRIS:	172	; REC INTR STATUS
465					
466					

NO1.

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 14
DZTRA.HED 13-APR-76 00:00

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

; TR79 STATUS TABLE AND ADDRESS ASSIGNMENTS

467									
468									
469									
470		001500							
471	001500								
472									
473	001500	000001							
474	001502	000001							
475	001504	000001							
476									
477	001506	000000							

TR.MAP: .=1500

TRCRO: .BLKW 1 ;CONTROL REGISTER FOR TR79 NUMBER 0
TRVCO: .BLKW 1 ;BASE VECTOR FOR TR79 NUMBER 0
TRLVO: .BLKW 1 ;PRIORITY LEVEL

TR.END: 000000

480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533

:PROGRAM INITIALIZATION
:LOCK OUT INTERRUPTS
:SET UP PROCESSOR STACK
:SET UP POWER FAIL VECTOR
:CLEAR PROGRAM CONTROL FLAGS AND COUNTS
:TYPE TITLE MESSAGE

.START:

001510	012706	001200		MOV	#STACK,SP	:SET UP STACK
001510	012737	000300	177776	MOV	#LEVEL6,PS	:LOCK OUT INTERRUPTS
001514	012737	004262	000024	MOV	#PFAIL,2024	:SET UP POWER FAIL VECTOR
001530	005037	001230		CLR	PASCNT	:CLEAR PASS COUNT
001534	105037	001341		CLRB	ERRFLG	:CLEAR ERROR FLAG
001540	105037	001343		CLRB	QV.FLG	:ZERO QUICK VERIFY FLAG
001544	012737	001500	001336	MOV	#TR.MAP,ACTIVE	:GET MAP POINTER.
001552	005037	001232		CLR	ERRCNT	:CLEAR ERROR COUNT
001556	005037	001234		CLR	LSTERR	:CLEAR LAST ERROR POINTER
001562	012737	000001	001226	MOV	#1,TSTNO	:SET UP FOR TEST 1
001570	012737	001510	001214	MOV	#.START,RETRN	:SET UP FOR POWER FAIL BEFORE TESTING STARTS
001576	013746	000006		;SET UP FOR SMALL 11 SWITCH REGISTER COMPATIBILITY		
001602	013746	000004		MOV	6,-(SP)	:SAVE BUS ERROR PS
001606	012737	001626	000004	MOV	4,-(SP)	:SAVE BUS ERROR PC
001614	005037	177776		MOV	#22\$,4	:SET UP TO TRAP TO THIS ROUTINE
001620	005777	177356		CLR	PS	:ALLOW INTERRUPTS
001624	000407			TST	2SWR	:CAN 177570 BE REFERENCED?
001626	012737	000176	001202	BR	21\$:IF YES, SKIP AROUND THE SETUP
001634	012737	000174	001200	MOV	#SWREG,SWR	:IF NO, TRAP COMES HERE. POINT TO SOFTWARE SWR
001642	022626			MOV	#DISPREG,DISPLAY	:POINT TO SOFTWARE DISPLAY REGISTER
001644	012637	000004	21\$:	POP2SP		:REMOVE THE TRAP FROM THE STACK
001650	012637	000006		MOV	(SP)+,4	:RESTORE THE BUS ERROR VECTOR
001654	005737	000042		MOV	(SP)+,6	
001660	001402			TST	42	:WORKING UNDER A MONITOR ?
001662	000137	002154		BEQ	31\$:NO
001666	105737	001340	31\$:	JMP	66\$:IF YES, SKIP THE TERMINAL INTERROGATION
001672	001004			TSTB	INIFLG	:HAVE WE ALREADY BEEN HERE TODAY?
001674	104402	001000		BNE	29\$:IF SO, SKIP PRINTING THE TITLE
001700	105337	001340		TYPE	#TITLE	:PRINT THE DIAGNOSTIC'S NAME
001704	032777	000001	177270	DECB	INIFLG	:SET THE ONCE ONLY FLAG
001712	001013		29\$:	BIT	#SW00,2SWR	:RESELECT ?
001714	012737	164000	001500	BNE	32\$:IF YES, GO SET UP THE INFORMATION
001722	012737	000170	001502	MOV	#164000,TRCRO	:SET DEFAULT PARAMETERS
001730	012737	000004	001504	MOV	#170,TRVCO	
001736	000137	002154		MOV	#4,TRLVO	
001742	012700	001500	32\$:	JMP	66\$:IF NO, SKIP THE INTERROGATION
001746	005020		68\$:	MOV	#TR.MAP,RO	:POINT TO THE BEGINNING OF THE MAP TABLE
001750	020027	001506		CLR	(RO)+	:CLEAR A TABLE LOCATION
001754	001374			CMP	RO,#TR.END	:HAVE THE TABLE BOUNDARIES BEEN EXCEEDED?
				BNE	68\$:IF NOT, CLEAR THE NEXT LOCATION IN THE TABLE

:THE FOLLOWING ARE PARAMETERS USED TO FILL IN THE MAP TABLE AND SET UP THE DIAGNOSTIC.

:GET THE BASE ADDRESS OF THE TR79'S

```

534 001756          33$: INSTR          :CALL THE STRING INPUT ROUTINE
535 001756 104405 69$          :POINTER TO MESSAGE TO BE PRINTED
536 001760 002034 PARAM        :CALL THE OCTAL TO ASCII CONVERT ROUTINE
537 001762 104407 160000      :LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
538 001764 160000 175500      :HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
539 001766 175500 TRCRO       :POINTER TO MAP LOCATION TO BE FILLED
540 001770 001500 .BYTE 7      :MASK OF INVALID BITS FOR THIS PARAMETER
541 001772          .BYTE 1      :NUMBER OF PARAMETERS TO STORE
542 001773 001          :GET THE BASE VECTOR ADDRESS
543
544
545
546 001774          34$: INSTR          :CALL THE STRING INPUT ROUTINE
547 001774 104405 70$          :POINTER TO MESSAGE TO BE PRINTED
548 001776 002062 PARAM        :CALL THE OCTAL TO ASCII CONVERT ROUTINE
549 002000 104407 170         :LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
550 002002 000170 776         :HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
551 002004 000776 TRVCO       :POINTER TO MAP LOCATION TO BE FILLED
552 002006 001502 .BYTE 3      :MASK OF INVALID BITS FOR THIS PARAMETER
553 002010          .BYTE 1      :NUMBER OF PARAMETERS TO STORE
554 002011 001          :GET THE BUS REQUEST LEVEL
555
556
557
558 002012 104405 INSTR        :CALL THE STRING INPUT ROUTINE
559 002014 002110 72$          :POINTER TO MESSAGE TO BE PRINTED
560 002016 104407 PARAM        :CALL THE OCTAL TO ASCII CONVERT ROUTINE
561 002020 000004 4          :LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
562 002022 000007 7          :HIGHEST LEGITIMATE VALUE OF EXPECTED RESPONSE
563 002024 001504 TRLVO       :POINTER TO MAP LOCATION TO BE FILLED
564 002026          .BYTE 0      :MASK OF INVALID BITS FOR THIS PARAMETER
565 002027          .BYTE 1      :NUMBER OF PARAMETERS TO STORE
566 002030 000137 002154 JMP 66$ :GO CONTINUE PROCESSING
567
568 002034 041777 047117 051124 59$: .ASCIZ <377>/CONTROLLER ADDRESS /
569 (2) 002062 030777 052123 053040 70$: .ASCIZ <377>/1ST VECTOR ADDRESS /
570 (2) 002110 047377 052117 035105 72$: .ASCII <377>/NOTE: ONE TR79 ONLY /<377>
571 (2) 002137 377 051102 046040 71$: .ASCIZ <377>/BR LEVEL /
572 (2) 002154 002154 .EVEN
573 (2) 002154 105737 001340 66$:
574 002154 001027 TSTB INIFLG :HAS INITIALIZATION BEEN PERFORMED
575 002162 022737 002502 000042 BNE 1$ :BR IF YES
576 002170 001402 CMP #SENDAD,2#42 :IF ACT-11 AUTOMATIC MODE, DON'T TYPE ID
577 002172 104402 001000 BEQ 7$ :OTHERWISE, PRINT ID
578 002176 105337 001340 TYPE MTITLE :TYPE TITLE MESSAGE
579 002202 105777 176774 7$: DECB INIFLG :IF NOT SET FLAG AND DC
580 002206 002002 TSTB QSWR :BIT7=1??
581 002210 004737 007720 BGE 16$ :BR IF NO AUTO SIZE
582 002214 104402 004663 JSR PC,CSRMAP :GO DO THE AUTO SIZE
583 002220 012700 001500 TYPE XHEAD :TYPE MAP HEADER
584 002224 012037 001306 MOV @TR.MAP,RO :SET POINTER
585 002230 001403 BEQ 1$ :SET DATA
586 002232 104412 CONVRT :ALL DONE WITH DATA
587 002234 007356 XSTATG :CALL THE OCTAL TO ASCII CONVERSION ROUTINE
588 002236 000772 BR 5$ :CONVERT THE DATA AT THIS ADDRESS

```



```

585 002240 005737 000042      1$:  TST      0#42      ; IS PROGRAM RUNNING UNDER MONITOR
586 002244 001000                BNE      3$      ; YES
587 002246 012700 000210      3$:  MOV      #210,R0   ; PREPARE TO CLEAR THE REMAINING
588 002252 012701 000212      MOV      #212,R1   ; VECTOR AREA. 210-776
589 002256 010120      4$:  MOV      R1,(R0)+ ; START PUTTING "PC+2 - HALT"
590 002260 005021                CLR      (R1)+    ; IN VECTOR AREA.
591 002262 022021                CMP      (R0)+,(R1)+ ; POP POINTERS
592 002264 022700 001000      CMP      #1000,R0 ; ALL DONE??
593 002270 001372                BNE      4$      ; BR IF NO.
594
595                ; TEST START AND RESTART
596                ;-----
597
598 002272 012737 000340 177776 .BEGIN: MOV      #340,PS ; LOCK OUT INTERRUPTS
599 002300 012706 001200      MOV      #STACK,SP ; SET UP STACK
600 002304 005737 000042      TST      0#42      ; IS PROGRAM UNDER MONITOR CONTROL
601 002310 001023                BNE      2$      ; BR IF YES
602 002312 032777 000004 176662 BIT      #BIT2,DSWR ; CHECK FOR LOCK ON TEST
603 002320 001411                BEQ      1$      ; BR IF NO LOCK DESIRED.
604 002322 104402 004555      TYPE    ,MLOCK    ; TYPE LOCK SELECTED.
605 002326 012737 000240 002570 MOV      #NOP,TTST ; ADJUST SCOPE ROUTINE.
606 002334 012737 000240 002572 MOV      #NOP,TTST+2 ; SET UP TO LOCK
607 002342 000406                BR       2$      ; CONTINUE ALONG.
608 002344 013737 002700 002570 1$:  MOV      BRW,TTST  ; PREPARE NORMAL SCOPE ROUTINE
609 002352 013737 002702 002572 MOV      BRX,TTST+2 ; LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
610 002360 012737 007540 001214 2$:  MOV      #CYCLE,RETURN ; START AT "CYCLE" FIND WHICH DEVICE TO TEST
611 002366 104402 004436      TYPE    ,MR      ; TYPE "RUNNING"
612 002372 000177 176616      JMP      $RETURN  ; START TESTING
    
```

```

613                                     :END OF PASS
614                                     :TYPE NAME OF TEST
615                                     :UPDATE PASS COUNT
616                                     :CHECK FOR EXIT TO ACT-11
617                                     :RESTART TEST
618
619                                     .EOP:
620 002376 005037 001234 CLR LSTERR ;CLEAR LAST ERROR PC
621 002402 105037 001341 CLR ERRFLG ;CLEAR ERROR FLAG
622 002406 005237 001230 INC PASCNT ;UPDATE PASS COUNT
623 002412 013777 001230 176560 MOV PASCNT, QDISPLAY ;DISPLAY PASS COUNT
624 002420 104402 004413 TYPE ,MEPASS ;TYPE END PASS
625 002424 104402 004604 TYPE ,MCSRX ;TYPE CSR
626 002430 104413 002526 CNVRT ,XCSR ;SHOW IT
627 002434 104402 004612 TYPE ,MVECX ;TYPE VECTOR
628 002440 104413 002534 CNVRT ,XVEC ;SHOW IT
629 002444 104402 004620 TYPE ,MPASSX ;TYPE PASSES
630 002450 104413 002542 CNVRT ,XPASS ;SHOW IT
631 002454 104402 004631 TYPE ,MERRX ;TYPE ERRORS
632 002460 104413 002550 CNVRT ,XERR ;SHOW IT
633 002464 112737 000377 001343 MOV B377, QV.FLG ;SET THE QUICK VERIFY FLAG.
634 002472 013701 000042 MOV Q#42, R1 ;CHECK FOR ACT-11 OR DDP
635 002476 001406 BEQ RESTART ;IF NOT, CONTINUE TESTING
636 002500 000005 RESET ;STOP THE SHOW--CLEAR THE WORLD
637 002502
638 002502 004711 JSR PC, (R1)
639 002504 000240 NOP
640 002506 000240 NOP
641 002510 000240 NOP
642 002512 000240 NOP
643 002514 012737 007540 001214 RESTART: MOV #CYCLE, RETURN
644 002522 000137 007540 JMP CYCLE
645 002526 000001 XCSR: 1
646 002530 006 002 .BYTE 6,2
647 002532 001376 TRCR
648 002534 000001 XVEC: 1
649 002536 003 002 .BYTE 3,2
650 002540 001414 TRVCT
651 002542 000001 XPASS: 1
652 002544 006 002 .BYTE 6,2
653 002546 001230 PASCNT
654 002550 000001 XERR: 1
655 002552 006 002 .BYTE 6,2
656 002554 001232 ERRCNT
657
658                                     :SCOPE LOOP AND ITERATION HANDLER
659 -----
660
661 002556 005037 001234 .SCOPE: CLR LSTERR ;CLEAR LAST ERROR PC.
662 002562 032777 040000 176412 BIT #BIT14, QSWR ;"LOOP ON THIS TEST"?
663 002570 001407 TTST: BEQ 1$ ;BR IF NO. (IF LOCK SW01=1; THIS LOC =240)
664 002572 002437 BR 3$ ;GOTO 3$ (IF LOCK SW01=1; THIS LOC =240)
665 002574 105777 176404 TSTB QTKCSR ;KEYBOARD DONE?
666 002600 100034 BPL 3$ ;BR IF NO. (LOCK: HIT KEY TO GOTO NEXT TEST)
667 002602 017716 176400 MOV QTKDBR, (SP) ;CLEAR DONE BIT
668 002606 000415 BR 2$ ;CONTINUE
    
```

1 1.
 2 21)

```

669 002610 032777 004000 176364 1$: BIT #SW11,2SWR ;DELETE ITERATION? (QUICK PASS)
670 002616 001011 BNE 2$ ;BR IF YES
671 002620 105737 001343 TSTB QV.FLG ;HAVE PASSES BEEN COMPLETED?
672 002624 001406 BEQ 2$ ;BR IF QUICK PASS.
673 002626 005237 001224 INC LPCNT ;UPDATE ITERATION COUNTER
674 002632 023737 001224 001222 CMP LPCNT,ICOUNT ;ARE ALL ITERATIONS DONE??
675 002640 101414 SLOS 3$ ;BR IF NOT YET
676 002642 105037 001341 2$: CLRB ERRFLG ;PREPARE FOR NEW TEST
677 002646 005037 001224 CLR LPCNT ;START ICOUNTER AT 0
678 002652 005037 001220 CLR LOCK
679 002656 012737 000005 001222 MOV #5,ICOUNT ;RESET ITERATIONS
680 002664 013737 001216 001214 MOV NEXT,RETURN ;GET NEXT TEST
681 002672 022626 3$: POP2SP ;FAKE AN "RTI"
682 002674 000177 176314 4$: JMP @RETURN ;GO DO THE TEST
683 002700 001407 BRW: 1407
684 002702 000437 BRX: 437

;CHECK FOR FREEZE ON CURRENT DATA
;-----
689 002704 032777 001000 176270 .SCOPE1: BIT #SW09,2SWR ;IS SW09=1(SET)?
690 002712 001405 BEQ 1$ ;BR IF NOT SET.
691 002714 005737 001220 TST LOCK ;IS THER A TIGHT LOOP SPECIFIED?
692 002720 001402 BEQ 1$ ;I NO, RETURN
693 002722 013716 001220 MOV LOCK,(SP) ;IF YES, GOTO THE ADDRESS IN LOCK.
694 002726 000002 1$: RTI ;GO BACK.

;TELETYPE OUTPUT ROUTINE
;-----
699 002730 010546 .TYPE: MOV R5, -(SP) ;SAVE R5 ON THE STACK.
700 002732 017605 MOV @2(SP),R5 ;GET ADDRESS OF MESSAGE.
701 002736 062766 000002 000002 ADD #2,2(SP) ;POP OVER ADDRESS.
702 002744 032777 020000 176230 1$: BIT #SW13,2SWR ;INHIBIT ALL PRINT OUT??
703 002752 001012 BNE 3$ ;BR IF NO PRINT OUT WANTED (SW13=1)
704 002754 105715 TSTB (R5)
705 002756 100002 BPL 2$
706 002760 104402 004352 TYPE MCRLF
707 002764 105777 176220 2$: TSTB @TPCSR ;TTY READY?
708 002770 100375 BPL 2$ ;BR IF NO.
709 002772 112577 176214 MOVB (R5)+,@TPDBR ;PRINT CURRENT CHAR.
710 002776 001362 BNE 1$ ;IF NOT ZERO KEEP PRINTING!
711 003000 012605 3$: MOV (SP)+,R5 ;END OF OUTPUT. RESTORE R5
712 003002 000002 RTI ;GO HOME

;AUXILLIARY TELETYPE OUTPUT ROUTINES
;-----
717 003004 017637 000000 003020 .TYPEF: MOV @2(SP),LSTMSG ;GET THE ADDRESS OF THE MESSAGE TO PRINT
718 003012 062716 000002 ADD #2,(SP) ;POINT TO THE NEXT LOCATION IN THE MAIN PROCEDURE
719 003016 104402 .TYPEL: TYPE ;BOTH ROUTINES TYPE HERE
720 003020 000000 LSTMSG: .WORD 0 ;POINTER TO MESSAGE TO PRINT
721 003022 032777 020000 176152 BIT #SW13,2SWR ;INHIBIT PRINTOUT?
722 003030 001002 BNE 1$ ;YES
723 003032 104402 TYPE ;THIS PART PRINTS THE REMAINDER OF THE MESSAGE
724 003034 007344 MFAIL ;THIS IS THE 'FAILURE' PART OF IT
    
```



```

725 003036 000002      1$:      RTI                      ;RETURN TO THE MAIN PROCEDURE
726
727                      ;STRING INPUT ROUTINE
728
729 -----
730 003040 010346      .INSTR: MOV      R3,-(SP)          ;SAVE R3 ON STACK
731 003042 010446      MOV      R4,-(SP)          ;SAVE R4 ON STACK
732 003044 017637 000004 003062      MOV      @4(SP),MSG       ;GET THE ADDRESS OF THE MESSAGE TO BE PRINTED
733 003052 062766 000002 000004      ADD      #2,4(SP)        ;POINT TO INSTRUCTION AFTER THE ADDRESS POINTER
734 003060 104402      .INST1: TYPE          ;PRINT THE MESSAGE
735 003062 000000      .MSG:      0              ;MESSAGE IS POINTED TO FROM HERE
736 003064 012704 007370      MOV      #INBUF,R4       ;POINT R4 TO THE INPUT BUFFER
737 003070 012703 000007      MOV      #7,R3           ;SET THE MAXIMUM NUMBER OF CHARACTERS ALLOWED
738 003074 105777 176104      1$:      TSTB     @TKCSR     ;HAS A CHARACTER BEEN RECEIVED?
739 003100 100375      BPL      1$              ;IF NO, KEEP WAITING FOR IT
740 003102 117714 176100      MOVB     @TKDBR,(R4)     ;IF YES, SAVE IT IN THE INPUT BUFFER
741 003106 142714 000200      BICB     #200,(R4)      ;KEEP ONLY THE 7-BIT ASCII INFORMATION
742 003112 122427 000015      CMPB     (R4)+,#15      ;IS THIS CHARACTER A LINE FEED?
743 003116 001417      BEQ      INSTR2         ;IF SO, TERMINATE THE INPUT SEQUENCE
744 003120 105777 176064      2$:      TSTB     @TPCSR     ;IF NOT, CHECK TO SEE IF THE CHARACTER CAN PRINT
745 003124 100375      BPL      2$              ;IF WE CAN'T WAIT UNTIL WE CAN
746 003126 017777 176054 176056      MOV      @TKDBR,@TPDBR  ;ECHO THE CHARACTER BACK
747 003134 005303      DEC      R3              ;REDUCE THE NUMBER OF CHARACTERS RECEIVED
748 003136 001356      BNE      1$              ;IF WE DON'T HAVE 7, GO GET SOME MORE
749 003140 012504      MOV      (SP)+,R4       ;IF WE HAVE 7, RESTORE R4
750 003142 012603      MOV      (SP)+,R3       ;RESTORE R3
751 003144 010346      .INSTE: MOV      R3,-(SP)   ;SAVE R3 ON THE STACK
752 003146 010446      MOV      R4,-(SP)       ;SAVE R4 ON THE STACK
753 003150 104402 004346      TYPE     .MOM           ;PRINT A QUESTION MARK... WHAT'S GOING ON?
754 003154 000741      BR       .INST1         ;GO PRINT THE MESSAGE AGAIN
755 003156 012604      INSTR2: MOV      (SP)+,R4   ;RESTORE R4
756 003160 012603      MOV      (SP)+,R3       ;RESTORE R3
757 003162 000002      RTI                      ;RETURN TO THE MAIN PROCEDURE
758
759                      ;CONVERT ASCII STRING TO OCTAL
760 -----
761
762 003164 010546      .PARAM: MOV      R5,-(SP)   ;SAVE R5 ON THE STACK
763 003166 010446      MOV      R4,-(SP)       ;SAVE R4 ON THE STACK
764 003170 016605 000004      MOV      4(SP),R5       ;GET THE SETUP INFORMATION POINTER
765 003174 012537 003354      MOV      (R5)+,LOLIM    ;SET THE LOW LIMIT FOR THE INPUT
766 003200 012537 003356      MOV      (R5)+,HILIM    ;SET THE HIGH LIMIT FOR THE INPUT
767 003204 012537 003360      MOV      (R5)+,DEVADR   ;SAVE THE ADDRESS WHERE THE RESULT WILL BE STORED
768 003210 112537 003362      MOVB     (R5)+,LOBITS   ;GET THE MASK OF THE INCORRECT BITS.
769 003214 112537 003363      MOVB     (R5)+,ADRCNT   ;GET THE COUNT OF ITEMS TO BE STORED
770 003220 010566 000004      MOV      R5,4(SP)      ;POINT TO WHERE THE MAIN PROGRAM WILL RESUME
771 003224 005005      PARAM1: CLR     R5       ;INITIALIZE THE ASCII TO OCTAL RESULT WORD
772 003226 012704 007370      MOV      #INBUF,R4     ;POINT TO THE INPUT BUFFER
773 003232 122714 000015      CMPB     #15,(R4)      ;IS THIS CHARACTER A CARRIAGE RETURN?
774 003236 001420      BEQ      PARERR        ;IF SO, PRINT THE MESSAGE AGAIN
775 003240 121427 000060      1$:      CMPB     (R4),#60      ;IS THIS CHARACTER BELOW THE NUMERIC RANGE?
776 003244 002415      BLT      PARERR        ;IF SO, GO PRINT THE MESSAGE AGAIN
777 003246 121427 000067      CMPB     (R4),#67      ;IS THIS CHARACTER ABOVE THE NUMERIC RANGE?
778 003252 003012      BGT      PARERR        ;IF SO, GO PRINT THE MESSAGE AGAIN
779 003254 142714 000060      BICB     #60,(R4)      ;ISOLATE THE NUMBER THE CHARACTER REPRESENTS
780 003260 152405      BISB     (R4)+,R5      ;CONCATENATE THESE BITS TO THE EXISTING STRING
    
```

H02

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 21
 DZTRA.HED 13-APR-76 00:00 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

```

781 003262 122714 000015      CMPB    #15, (R4)      ; IS THE NEXT CHARACTER A CARRIAGE RETURN?
792 003266 001406      BEQ     LIMITS      ; IF SO, GO SEE IF THE NUMBER IS WITHIN LIMITS
783 003270 006305      ASL    R5           ; CLEAR BIT POSITION 0, MOVE THE EXISTING STRING TO THE L
794 003272 006305      ASL    R5           ; CLEAR POSITION 1, MOVE THE STRING TO THE LEFT AGAIN
795 003274 006305      ASL    R5           ; MOVE THE STRING ONE MORE TIME TO MAKE ROOM FOR
786 003276 000760      BR     1$          ; NEXT THREE BITS
787 003300 104406      PARERR: INSTER     ; GO GET THE NEXT CHARACTER
788 003302 000750      BR     PARAM1     ; THERE WAS AN ERROR- GO PRINT THE MESSAGE AGAIN
790 003302 000750      BR     PARAM1     ; TRY GETTING THE PARAMETERS AGAIN
791 003302 000750      ; TEST TO SEE IF NUMBER IS WITHIN LIMITS
792 003302 000750      -----
793 003304 020537 003356  LIMITS: CMP     R5, HILIM ; DOES RESULT EXCEED ITS MAXIMUM CORRECT VALUE?
794 003310 101373      BHI    PARERR     ; IF YES, GO PRINT THE MESSAGE AGAIN
795 003312 020537 003354  CMP     R5, LOLIM ; IS THE RESULT LOWER THAN ALLOWED?
796 003316 103770      BLO    PARERR     ; IF YES, GOR PRINT THE MESSAGE AGAIN
797 003320 133705 003362  BITB   LOBITS, R5 ; ARE ANY INCORRECT BITS SET IN THE RESULT?
798 003324 001365      BNE    PARERR     ; IF SO, GO PRINT THE MESSAGE AGAIN
800 003324 001365      ; STORE NUMBER AT SPECIFIED ADDRESS
801 003326 013704 003360  1$:    MOV     DEVADR, R4 ; POINT TO THE LOCATION WHERE THE RESULT WILL BE STORED
802 003332 010524      MOV     R5, (R4)+    ; STORE THE RESULT
803 003334 062705 000002  ADD    #2, R5       ; CALCULATE THE NEXT DATUM
804 003340 105337 003363  DECB   ADCRNT      ; REDUCE COUNT OF STORED RESULTS. IS IT EXCEEDED?
805 003344 001372      BNE    1$          ; IF NOT, GO STORE THE NEXT DATUM
806 003346 012604      MOV     (SP)+, R4   ; RESTORE R4
807 003350 012605      MOV     (SP)+, R5   ; RESTORE R5
808 003352 000002      RTI              ; RETURN TO THE MAIN PROGRAM
809 003354 000000  LOLIM:  0           ; LOWEST ACCEPTABLE VALUE
810 003356 000000  HILIM:  0           ; HIGHEST ACCEPTABLE
811 003360 000000  DEVADR: 0           ; LOCATION WHERE RESULT WILL BE STORED
812 003362 000      LOBITS: .BYTE  0   ; INCORRECT BITS MASK
813 003363 000      ADCRNT: .BYTE  0   ; COUNT OF ITEMS TO BE STORED
814 003364 016637 000004 001334 .SAV05: MOV    4(SP), SAVPC ; SAVE R7 (PC)
815 003364 016637 000004 001334 ; SAVE R0-R5
816 003372 010537 001330  SV05:  MOV    R5, SAVR5   ; SAVE R5
817 003376 010437 001326  MOV    R4, SAVR4   ; SAVE R4
818 003402 010337 001324  MOV    R3, SAVR3   ; SAVE R3
819 003406 010237 001322  MOV    R2, SAVR2   ; SAVE R2
820 003412 010137 001320  MOV    R1, SAVR1   ; SAVE R1
821 003416 010037 001316  MOV    R0, SAVR0   ; SAVE R0
822 003422 000002      PTI              ; LEAVE.
823 003424 000002      ; RESTORE R0-R5
824 003424 013700 001316  .RES05: MOV    SAVR0, R0 ; RESTORE R0
825 003430 013701 001320  MOV    SAVR1, R1   ; RESTORE R1

```

```

837 003434 013702 001322      MOV      SAVR2,R2      ;RESTORE R2
838 003440 013703 001324      MOV      SAVR3,R3      ;RESTORE R3
839 003444 013704 001326      MOV      SAVR4,R4      ;RESTORE R4
840 003450 013705 001330      MOV      SAVR5,R5      ;RESTORE R5
841 003454 000002      RTI                    ;LEAVE
842
843                                     ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
844 -----
845
846 003456 104402 004352      .CONVR: TYPE      MCRLF      ;PRINT A CARRIAGE RETURN
847 003462 010046      .CNVRT: MOV      R0,-(SP)      ;SAVE R0
848 003464 010146      MOV      R1,-(SP)      ;SAVE R1
849 003466 010346      MOV      R3,-(SP)      ;SAVE R3
850 003470 010446      MOV      R4,-(SP)      ;SAVE R4
851 003472 010546      MOV      R5,-(SP)      ;SAVE R5
852 003474 017601 000012      MOV      @12(SP),R1      ;PLACE THE ADDRESS OF THE ARGUMENTS IN R1
853 003500 062766 000002 000012      ADD      #2,12(SP)      ;POINT TO WHERE THE MAIN PROGRAM WILL RESUME
854 003506 012137 003632      MOV      (R1)+,WRDCNT      ;GET THE NUMBER OF WORDS TO BE PRINTED
855 003512 112105      1$:      MOV      (R1)+,R5      ;GET THE NUMBER OF CHARACTERS TO BE PRINTED
856 003514 112100      MOV      (R1)+,R0      ;GET THE NUMBER OF SPACES TO PRINT
857 003516 013104      MOV      @1(R1)+,R4      ;COPY THE WORD TO BE CONVERTED
858 003520 110537 003634      MOV      R5,CHRCNT      ;COPY THE CHARACTER COUNT
859 003524 010403      3$:      MOV      R4,R3      ;COPY THE ARGUMENT WORD AGAIN
860 003526 042703 177770      BIC      #177770,R3      ;ISOLATE THREE BITS TO BE TREATED AS A CHARACTER
861 003532 062703 000060      ADD      #060,R3      ;MAKE AN ASCII CHARACTER OUT OF THEM
862 003536 110346      MOV      R3,-(SP)      ;SAVE THAT CHARACTER
863 003540 006004      MOV      R4      ;MOVE THE NEXT THREE BITS INTO PLACE
864 003542 006204      ASR      R4      ;MOVE THEM AGAIN
865 003544 006204      ASR      R4      ;AND FINALLY A THIRD TIME
866 003546 005305      DEC      R5      ;REDUCE THE CHARACTER COUNT. ARE ALL CHARACTERS
867                                     ;BUILT?
868 003550 001365      BNE      3$      ;IF NO, GO BUILD THE NEXT ONE.
869 003552 012703 007476      MOV      #MDATA,R3      ;NOW POINT TO WHERE NUMBER WILL BE PRINTED FROM
870 003556 112623      4$:      MOV      (SP)+,(R3)+      ;STORE THE CHARACTER, STARTING WITH THE MOST
871 003560 105337 003634      DECB      CHRCNT      ;REDUCE COUNT. ARE ALL CHARACTERS TRANSFERRED?
872 003564 001374      BNE      4$      ;IF NO, GO TRANSFER ANOTHER
873 003566 105700      TSTB      R0      ;ARE ANY SPACES TO BE PRINTED?
874 003570 001404      BEQ      6$      ;IF NO, DON'T SET UP ANY
875 003572 112723 000040      5$:      MOV      #040,(R3)+      ;ADD A SPACE TO THE OUTPUT BUFFER
876 003576 105300      DECB      R0      ;REDUCE THE COUNT. SHOULD WE PRINT MORE?
877 003600 001374      BNE      5$      ;IF YES, GO ADD ANOTHER SPACE
878 003602 105013      6$:      CLRB      (R3)      ;TERMINATE THE OUTPUT BUFFER WITH A ZERO
879 003604 104402 007476      TYPE      ,MDATA      ;PRINT THE STRING WE JUST BUILT
880 003610 005337 003632      DEC      WRDCNT      ;REDUCE THE WORD COUNT. ARE ANY MORE WORDS LEFT?
881 003614 001336      BNE      1$      ;IF YES, GO CONVERT THEM
882 003616 012605      MOV      (SP)+,R5      ;RESTORE R5
883 003620 012604      MOV      (SP)+,R4      ;RESTORE R4
884 003622 012603      MOV      (SP)+,R3      ;RESTORE R3
885 003624 012601      MOV      (SP)+,R1      ;RESTORE R1
886 003626 012600      MOV      (SP)+,R0      ;RESTORE R0
887 003630 000002      RTI                    ;RETURN TO THE MAIN PROGRAM
888 003632 000000      WRDCNT: 0
889 003634 000      CHRCNT: .BYTE
890 003635 000      SPACNT: .BYTE 0
891
892 003636 000000      BINWRD: 0
    
```

```

893
894
895
896
897
898
899
900 003640 011646 .TRPSR: MOV (SP),-(SP) ;GET PC OF RETURN
901 003642 162716 000002 SUB #2,(SP) ;=PC OF TRAP
902 003646 017616 000000 MOV @2(SP),(SP) ;GET TRP
903 003652 006316 TRPOK: ASL (SP) ;MULTIPLY TRAP ARG BY 2
904 003654 042716 117001 BIC #117001,(SP) ;CLEAR UNWANTED BITS
905 003660 062716 001346 ADD #.TRPTAB,(SP) ;POINTER TO SUBROUTINE ADDRESS
906 003664 017616 000000 MOV @2(SP),(SP) ;SUBROUTINE ADDRESS
907 003670 000136 JMP @2(SP)+ ;GO TO SUBROUTINE
908
909
910
911
912
913 003672 005203 CLOCK: INC R3 ;COUNT A TIME TICK
914 003674 001376 BNE CLOCK ;KEEP GOING UNTIL WE REACH 0
915 003676 005237 001304 INC TEMP1 ;NOW ADD TO THE OVERALL TIMER
916 003702 000773 BR CLOCK ;KEEP GOING
917
918
919
920 003704 032777 010000 175270 .EROR: BIT #SW12,@SWR ;BELL ON ERROR?
921 003712 001406 BEQ XBX ;BR IF NO BELL
922 003714 105777 175270 TSTP @TPCSR ;TTY READY.
923 003720 100003 BPL XBX ;DON'T WAIT IF TTY NOT READY.
924 003722 112777 000207 175262 MOVJ #207,@TPDBR ;PUSH A BELL AT THE TTY.
925 003730 032777 020000 175244 XBX: BIT #SW13,@SWR ;DELETE ERROR PRINT OUT?
926 003736 001105 BNE HALTS ;BR IF NO PRINT OUT WANTED.
927 003740 021637 001234 CMP (SP),LSTERR ;WAS THIS ERROR FOUND LAST TIME?
928 003744 001404 BEQ IS ;BR IF YES
929 003746 011637 001234 MOV (SP),LSTERR ;RECORD BEING HERE
930 003752 105037 001341 CLRB ERRFLG ;PREPARE HEADER
931 003756 104410 IS: SAVOS ;SAVE ALL PROC REGISTERS
932 003760 011605 MOV (SP),R5 ;GET THE PC OF ERROR
933 003762 162705 000002 SUB #2,R5 ;GET ADDRESS OF TRAP CALL
934 003766 011504 MOV (R5),R4 ;GET ERROR INSTRUCTION
935 003770 006304 ASL R4 ;MULT BY TWO
936 003772 061504 ADD (R5),R4 ;DOUBLE IT
937 003774 006304 ASL R4 ;MULT AGAIN
938 003776 042704 137001 BIC #137001,R4 ;CLEAR JUNK
939 004002 062704 041124 ADD #.ERRTAB,R4 ;GET POINTER
940 004006 012437 004122 MOV (R4)+,ERRMSG ;GET ERROR MESSAGE
941 004012 012437 004134 MOV (R4)+,DATAHD ;GET DATA HEADRER
942 004016 011437 004146 MOV (R4),DATABP ;GET DATA TABLE
943 004022 105737 001341 TSTB ERRFLG ;TYPE HEADER
944 004026 001403 BEQ TYPMSG ;BR IF YES
945 004030 005737 004146 TST DATABP ;DOES DATA TABLE EXIST?
946 004034 001040 BNE TYPDAT ;BR IF YES.
947 004036 104402 004352 TYPMSG: TYPE ,MCRLF ;TYPE A CARRIAGE RETURN
948 004042 104402 004352 TYPE ;AND TYPE ANOTHER
    
```

K02

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 24
 DZTRA.HED 13-APR-76 00:00 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

949	004046	005737	001220		TST	LOCK	
950	004052	001402			BEQ	1\$	
951	004053	104402	004654		TYPE	,MASTEK	
952	004060	104402	004642	1\$:	TYPE	,MTSTN	
953	004064	104413	004254		CONVRT	,XTSTN	;SHOW IT
954	004070	104402	004656		TYPE	,MERRPC	;TYPE PC.
955	004074	104413	004246		CONVRT	,ERTAB0	;SHOW IT
956	004100	104402	004352		TYPE	,MCRLF	;GIVE A CR/LF
957	004104	112737	177777	001341	MOVB	#-1,ERRFLG	;NO MORE HEADER UNLESS NO DATA TABLE.
958	004112	005737	004122		TST	ERRMSG	;IS THERE AN ERROR MESSAGE?
959	004116	001402			BEQ	WTBS.FM	;BR IF NO.
960	004120	104402			TYPE		;TYPE
961	004122	000000			ERRMSG:	0	ERROR MESSAGE
962	004124				WTBS.FM:		
963	004124	005737	004134		TST	DATAHD	DATA HEADER?
964	004130	001402			BEQ	TYPDAT	BR IF NO
965	004132	104402			TYPE		TYPE
966	004134	000000			DATAHD:	0	DATA HEADER
967	004136	005737	004146		TYPDAT:	TST	DATA TABLE?
968	004142	001402			BEQ	DATABP	BR IF NO.
969	004144	104412			CONVRT	RESREG	SHOW
970	004146	000000			DATABP:	0	DATA TABLE
971	004150	104411			RESREG:	RES05	RESTORE PROC REGISTERS
972	004152	022737	002502	000042	HALTS:	CMP	#SENDAD, @#42
973	004160	001403			BEQ	1\$	CHECK TO SEE IF IN ACT-11 MODE
974	004162	005777	175014		TST	@SWR	IF SO, HANDLE ACCORDINGLY
975	004166	100005			BPL	EXITER	HALT ON ERROR?
976	004170	010046			1\$:	PUSHRO	BR IF NO HALT ON ERROR
977	004172	016600	000002		MOV	2(SP),RO	SAVE RO
978	004176	000000			HALT		SHOW ERROR PC IN DATA DISPLAY
979	004200	012600			POPPO		HALT
980	004202	005237	001232		EXITER:	INC	GET RO
981	004206	032777	000400	174766	BIT	ERRCNT	UPDATE ERROR COUNT
982	004214	001007			BNE	#SW08, @SWR	GOTO TOP OF TEST?
983	004216	032777	002000	174756	BIT	1\$	BR IF YES
984	004224	001407			BEQ	#SW10, @SWR	GOTO NEXT TEST
985	004226	013737	001216	001214	BEQ	2\$	BR IF NO
986	004234	012706	001200		MOV	NEXT, RETURN	SET FOR NEXT TEST
987	004240	000177	174750		1\$:	MOV	RESET SP
988	004244	000002			JMP	#STACK, SP	GOTO SPECIFIED TEST
989	004246	000001			2\$:	RTI	RETURN
990	004250	006	002		ERTAB0:	1	
991	004252	001334			.BYTE	6,2	
992	004254	000001			SAVPC		
993	004256	003	002		XTSTN:	1	
994	004260	001226			.BYTE	3,2	
995					TSTNO		
996					;ENTER HERE ON POWER FAILURE		
997					;-----		
998							
999	004262				.PFAIL:		
1000	004262	012737	004274	000024	MOV	#RESTART, 24	;SET UP FOR POWER UP TRAP
1001	004270	000000			HALT		;HALT ON POWER DOWN NORMAL
1002	004272	000777			BR	.	;PREVENT ANY FURTHER PROGRESS
1003							
1004							;PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED


```

1005
1006 004274
1007 004274 012737 004262 000024
1008 004302 012706 001200
1009 004306 005000
1010 004310 005200
1011 004312 001376
1012 004314 104402 004355
1013 004320 104413 004340
1014 004324 105037 001341
1015 004330 005037 001234
1016 004334 000177 174654
1017 004340 000001
1018 004342 003 002
1019 004344 001226
1020 004346 020040 000077
(2) 004352 005015 000
(2) 004355 377 053520 020122
(2) 004413 377 047105 020104
(2) 004436 051377 047125 044516
(2) 004450 050377 047522 051107
(2) 004517 377 047111 052523
(2) 004543 377 042524 052123
(2) 004555 377 047514 045503
(2) 004604 051503 035122 000040
(2) 004612 042526 035103 000040
(2) 004620 040520 051523 051505
(2) 004631 105 051122 051117
(2) 004642 042524 052123 047040
(2) 004654 000052
(2) 004656 041520 020072 000
(2) 004663 377 040515 020120
(2) 004710 051377 051505 052105
(2) 004721 377 051127 052111
(2) 004732 051377 040505 020104
(2) 004742 051777 040520 042503
(2) 004763 377 051105 051501
(2) 004774 020377 042522 047515
(2) 005054 044777 051516 051105
(2) 005132 044777 052116 051105
(2) 005156 044777 052116 051105
(2) 005203 377 047520 042527
(2) 005222 053777 044522 042524
(2) 005264 051377 053505 047111
(2) 005316 044777 046114 043505
(2) 005342 053777 044522 042524
(2) 005374 053777 044522 042524
(2) 005421 377 051127 052111
(2) 005467 377 046120 041501
(2) 005552 047777 020116 044514
(2) 005573 377 051127 052111
(2) 005635 377 047125 052111
(2) 005671 377 043117 020106
(2) 005713 377 042522 044527
(2) 005736 051377 040505 020104
(2) 005761 377 050123 041501

```

RESTART:

```

MOV #,PFAIL,24 ;SET UP FOR POWER FAILURE
MOV #STACK,SP ;RESET THE STACK POINTER
CLR RD ;READY FOR TIMER
IS: INC RD ;PLUS ONE TO THE TIMER!
SNE IS ;BR IF MORE TO GO
TYPE ,MPFAIL ;TYPE THE MESSAGE
CNVRT ,PFTAB ;TELL WHAT TEST TO RETURN TO.
CLRB ERRFLG ;START CLEAN
CLR LSTERR
JMP @RETJRN ;START DOING THAT TEST AGAIN.

```

PFTAB:

```

1
.BYTE 3,2
TSTNO
MQM: .ASCIZ / ?/
MCRFL: .ASCIZ <15><12>
MPFAIL: .ASCIZ <377>/PWR FAILED. RESTART AT TEST /
MEPASS: .ASCIZ <377>/END PASS DZTRAA /
MR: .ASCIZ <377>/RUNNING /
MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./
MERR3: .ASCIZ <377>/INSUFFICIENT DATA!/
MTSTPC: .ASCIZ <377>/TEST PC-/
MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/
MCSR: .ASCIZ /CSR: /
MVEC: .ASCIZ /VEC: /
MPASSX: .ASCIZ /PASSES: /
MERRX: .ASCIZ /ERRORS: /
MTSTN: .ASCIZ /TEST NO: /
MASTEK: .ASCIZ /*/
MERRPC: .ASCIZ /PC: /
XHEAD: .ASCIZ <377>/MAP OF TR79 STATUS/<377>
MRSTFL: .ASCIZ <377>/RESET /
MWRTFL: .ASCIZ <377>/WRITE /
MRDFL: .ASCIZ <377>/READ /
MSRVFL: .ASCIZ <377>/SPACE REVERSE /
MERFL: .ASCIZ <377>/ERASE /
MRMVRG: .ASCIZ <377>/ REMOVE WRITE ENABLE RING LOAD & PLACE ONLINE /
MINSRG: .ASCIZ <377>/INSERT WRITE ENABLE RING LOAD & PLACE ONLINE/
MIENFL: .ASCIZ <377>/INTERRUPT ENABLE /
MIDSFL: .ASCIZ <377>/INTERRUPT DISABLE /
MPCFL: .ASCIZ <377>/POWER CLEAR /
MWIPLFL: .ASCIZ <377>/WRITE ID BLOCK PAST LOAD POINT /
MRWLPFL: .ASCIZ <377>/REWIND FROM LOAD POINT /
MILFFL: .ASCIZ <377>/ILLEGAL FUNCTION /
MWLPFL: .ASCIZ <377> /WRITE FROM LOAD POINT /
MWEFFL: .ASCIZ <377>/WRITE END OF FILE /
MWELPFL: .ASCIZ <377>/WRITE END OF FILE FROM LOAD POINT /
MUOLN: .ASCIZ <377>/PLACE UNIT ON LINE AFTER TAPE MOTION HAS STOPPED /
MOLEFL: .ASCIZ <377>/ON LINE ERROR /
MWILPFL: .ASCIZ <377>/WRITE ID BLOCK FROM LOAD POINT /
MUNOFLN: .ASCIZ <377>/UNIT DID NOT GO OFF LINE /
MOFLEFL: .ASCIZ <377>/OFF LINE ERROR /
MRWOFL: .ASCIZ <377>/REWIND OFF LINE /
MRDERR: .ASCIZ <377>/READ DATA ERROR /
MSREFFL: .ASCIZ <377>/SPACE REVERSE OVER END OF FILE /

```

M02

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 26
 DZTRA.HED 13-APR-76 00:00

GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

```

(2) 006023      377 042522 042101 MREFFL: .ASCIZ <377>/READ END OF FILE /
(2) 006047      377 051127 052111 MWTOFL: .ASCIZ <377>/WRITE TIME OUT /
(2) 006071      377 042522 042101 MRTOFL: .ASCIZ <377>/READ TIME OUT /
(2) 006112      053777 044522 042524 MWXNXL: .ASCIZ <377>/WRITE NON-EXISTENT MEMORY /
(2) 006147      377 040505 041040 MBNXFL: .ASCIZ <377>/EA BIT 12- NON EXISTENT MEMORY /
(2) 006211      377 054105 042524 MNXBFL: .ASCIZ <377>/EXTENSION BIT 13 NON EXISTENT MEMORY /
(2) 006261      377 040505 041040 MNXMFL: .ASCIZ <377>/EA BITS 12 AND 13- NON EXISTENT MEMORY /
(2) 006333      377 042522 042101 MRCFL: .ASCIZ <377>/READ COUNT /
(2) 006351      377 047105 020104 METFL: .ASCIZ <377>/END OF TAPE /
(2) 006370      051377 051505 052105 MRSTU: .ASCIZ <377>/RESET TAPE UNIT /
(2) 006413      377 040515 052516 MMRSFL: .ASCIZ <377>/MANUAL RESET /
(2) 006433      377 051127 052111 MWPETFL: .ASCIZ <377>/WRITE PAST END OF TAPE /
(2) 006466      051777 040520 042503 MSRETFL: .ASCIZ <377>/SPACE REVERSE PAST END OF TAPE /
(2) 006530      051377 040505 020104 MRPETFL: .ASCIZ <377>/READ PAST END OF TAPE /
(2) 006561      377 051127 052111 MWEFETF: .ASCIZ <377>/WRITE END OF FILE PAST END OF TAPE /
(2) 006632      051377 040505 020104 MREFETF: .ASCIZ <377>/READ END OF FILE PAST END OF TAPE /
(2) 006702      053777 044522 042524 MWENFL: .ASCIZ <377>/WRITE ENABLE /
(2) 006722      051777 040520 042503 MSRLPFL: .ASCIZ <377>/SPACE REVERSE AT LOAD POINT /
(2) 006761      377 050123 041501 MSRIF: .ASCIZ <377>/SPACE REVERSE AT ID BLOCK /
(2) 007016      051377 040505 020104 MRI: .ASCIZ <377>/READ ID BLOCK /
(2) 007037      377 053105 047105 MPARFL: .ASCIZ <377>/EVEN PARITY /
(2) 007056      042777 042526 020116 MEPSRFL: .ASCIZ <377>/EVEN PARITY SPACE REVERSE /
(2) 007113      377 042522 042101 MREPFL: .ASCIZ <377>/READ EVEN PARITY /
(2) 007137      377 047527 042122 MWCFL: .ASCIZ <377>/WORD COUNT /
(2) 007155      377 047125 052111 MUNERR: .ASCIZ <377>/UNIT HAS ERROR FLAG SET /
(2) 007207      377 040524 042520 MUNRDY: .ASCIZ <377>/TAPE UNIT NOT READY /
(2) 007235      377 177777 042607 MEOT: .ASCIZ <377><377><377><207>/END OF TAPE... TAPE REWOUND /
(2) 007276      052377 054524 041040 MTBEX: .ASCIZ <377>/TTY BUFFER EXCEEDED. RESTART INPUT. /
(2) 007344      040506 046111 051125 MFAIL: .ASCIZ /FAILURE /
(2) 007356      000002
1021 007360      006 003
1022 007362      001304
1023 007364      006 002
1024 007366      001306
1025
1026
1027
1028
1029 007370      000000
1030
1031 007432      000000
1032 007434      000000
1033
1034 007476      000000
1035

```

```

XSTATQ: 2
        .BYTE 6,3
        TEMP1
        .BYTE 6,2
        TEMP2

```

```

.EVEN
;BUFFERS FOR INPUT-OUTPUT

```

```

INBUF: 0
.=.+40
TTYEND: 0
TEMP: 0
.=.+40
MDATA: 0
.=.+40

```

```

1036
1037
1038
1039
1040
1041
1042
1043
1044 007540 013700 001336      CYCLE:  MOV    ACTIVE, R0      ;GET ADDRESS POINTER.
1045 007544 012037 001242      MOV    (R0)+, TRBASE     ;LOAD SYSTEM CTRL. REG
1046 007550 012037 001414      MOV    (R0)+, TRVCT     ;LOAD VECTOR
1047 007554 012037 040764      MOV    (R0)+, TRPRT     ;LOAD PRIORITY
1048 007560 004737 040636      JSR    PC, TRLEV        ;SET UP
1049 007564 005737 000042      TST    @#42             ;ARE WE UNDER MONITOR CONTROL?
1050 007570 001046              BNE    4$               ;IF YES, SKIP THIS SETUP
1051 007572 032777 000002 171402  BIT    #SW01, @SWR      ;IF SW01=1, GET STARTING TEST #
1052 007600 001442              BEQ    4$               ;BR IF NO TEST IS TO BE INPUTTED
1053 007602 104402 004352      7$:   TYPE    ,MCRLF
1054              ;GET THE STARTING TEST NUMBER
1055 007606 104405      INSTR
1056 007610 004642      MTSTN
1057 007612 104407      PARAM
1058 007614 000001      1
1059 007616 001000      1000
1060 007620 001226      TSTNO
1061 007622      .BYTE    0
1062 007623      .BYTE    1
1063 007624 012700 010344      MOV    #TST1, R0
1064 007630 022710 012737      5$:   CMP    #12737, (R0)
1065 007634 001015      BNE    6$
1066 007636 023760 001226 000002      CMP    TSTNO, 2(R0)
1067 007644 001011      BNE    6$
1068 007646 022760 001226 000004      CMP    #TSTNO, 4(R0)
1069 007654 001005      BNE    6$
1070 007656 010037 001214      MOV    R0, RETURN      ;SAVE PC
1071 007662 104402 004352      TYPE    ,MCRLF
1072 007666 000412      BR     8$
1073 007670 005720      6$:   TST    (R0)+
1074 007672 020027 011324      JMP    R0, #TLAST+10
1075 007676 001354      BNE    5$
1076 007700 104402 004346      TYPE    ,MQM
1077 007704 000736      BR     7$
1078 007706 012737 010344 001214  4$:   MOV    #TST1, RETURN  ;PREPARE RETURN ADDRESS
1079 007714 000177 171274      8$:   JMP    @RETURN        ;GO START TESTING.
1080
1081
1082
1083
1084
1085 007720 012702 001500      CSRMAP: MOV    #TR.MAP, R2
1086 007724 012722 164000      MOV    #TCR, (R2)+     ;STORE CSR IN CORE TABLE.
1087 007730 005722      TST    (R2)+           ;POP OVER VECTOR STORE AREA
1088 007732 012722 000004      MOV    #4, (R2)+       ;SET THE DEFAULT BUS LEVEL
1089 007736 012737 000340 000022  VECMAP: MOV    #340, @#22  ;SET IOT TRAP PRIORITY TO 7
1090 007744 012737 010032 000020      MOV    #4$, @#20       ;SET IOT TRAP VECTOR
1091 007752 012702 001500      MOV    #TR.MAP, R2     ;SET SOFTWARE POINTER
    
```

;*ROUTINE USED TO "AUTO SIZE" THE TR79
 ;*CSR AND VECTOR.

```

:093 007756 012700 000210      MOV      #210,R0      :RANDOM VECTORS START HERE.
:094 007756 012701 000212      MOV      #212,R1      :PC OF IOT INSTR.
:095 007756 010120      15:  MOV      R1,(R0)+     :START FILLING VECTOR AREA
:096 007770 012721 000004      MOV      #4,(R1)+     :WITH .+2; IOT
:097 007774 022021      CMP      (R0)+(R1)+   :ADD 2 TO R0 +R1
:098 007776 020127 001000      CMP      R1,#1000     :HAS THE VECTOR AREA BEEN EXCEEDED?
:099 010002 101771      BLOS    15            :BR IF MORE TO FILL
:100 010004 005037 177776      CLR      PS           :ZERO CPU PRIO
:101 010010 012772 000100 000000      MOV      #BIT6,3(R2)
:102 010016 105200      INCB    R0           :ATTEMPT TO FORCE AN INTERUPT
:103 010020 001376      BNE     .-2          :STALL
:104 010022 012762 000170 000002      MOV      #170,2(R2)   :FOR TIME TO INTERUPT
:105 010030 000407      BR      SS           :NO INTERUPT ASSUME 210 AND FIX TR79 LATER
:106 010032 051662 000002 45:  BIS     (SP),2(R2)    :ALL DONE
:107 010036 042762 000007 000002      BIC     #7,2(R2)     :GET VECTOR ADDRESS
:108 010044 02262F      POP2SP :CLEAR JUNK
:109 010046 000002      RTI     :PCP IOT JUNK OFF STACK
:110 010050 000207      RTS     PC           :ALL DONE WITH "AUTO SIZING"

```

```

1112
1113      000062      MAXERR=50.      ;MAXIMUM NUMBER OF DEROR PRINTOUTS
1114      000006      MAXTST=6.      ;MAXIMUM NUMBER OF TESTS
1115      000012      PASRAT=10.     ;PASS RATE PRINT OUT
1116      040000      CORSIZ=40000 ;CORE REQUIRED
1117
1118
1119
1120
1121
1122
1123      010052      012737      010052      001250      PRETST:  MOV      #PRETST,TSTPTR ;SET TEST ADDRESS
1124      010050      004737      013662
1125      010064      013727      001240      JSR      PC,TESTN
1126      010070      000000      64$:      MOV      HOLD,(PC)+ ;PICK UP TIME PARAMETER
1127      010072      005227      000000      65$:      .WORD    0 ;USE THIS WORD AS A TIME COUNTER
1128      010072      005227      000000      INC      #0 ;IF NO,COUNT 1 OF 65535 TICKS
1129      010076      001375      BNE      66$ ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
1130      010100      005337      010070      DEC      64$ ;HAS THE TOTAL TIME ELAPSED?
1131      010104      001372      BNE      66$ ;IF NO,GO WAIT A LITTLE LONGER
1132      010106
1133      010106      052777      004000      171262      65$:      BIS      #PWRCLR,@TRCR ;DEVICE MASTER RESET
1134      010114      032777      004000      171254      67$:      BIT      #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
1135      010122      001374      BNE      67$ ;IF NO, WAIT FOR IT TO CLEAR
1136      010124      000005      RESET
1137      010126      022777      006026      171242      CMP      #6026,@TRCR ;CHECK COMMAND REGISTER
1138      010134      001412      BEQ      1$ ;SKIP IF OK
1139      010136      012705      006026      MOV      #6026,R5
1140      010142      017704      171230      MOV      @TRCR,R4
1141      010146      013737      001376      001244      MOV      TRCR,REGIST
1142      010154      104001      ERROR    1 ;INCORRECT REGISTER MATCHUP
1143      010156      104403      004710      TYPEF    ,MRSTFL
1144
1145      010162
1146      010162      012737      000050      001240      1$:      MOV      #50,HOLD ;SET UP FOR A MASSIVE DELAY
1147      010170      013727      001240      MOV      HOLD,(PC)+ ;PICK UP TIME PARAMETER
1148      010174      000000      68$:      .WORD    0 ;USE THIS WORD AS A TIME COUNTER
1149      010176      005227      000000      70$:      INC      #0 ;IF NO,COUNT 1 OF 65535 TICKS
1150      010202      001375      BNE      70$ ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
1151      010204      005337      010174      DEC      68$ ;HAS THE TOTAL TIME ELAPSED?
1152      010210      001372      BNE      70$ ;IF NO,GO WAIT A LITTLE LONGER
1153      010212
1154      010212      012737      000010      001240      69$:      MOV      #10,HOLD ;RESTORE THE NORMAL DELAY FACTOR
1155      010220      022777      002226      171150      CMP      #2226,@TRCR ;TEST COMMAND REGISTER
1156      010226      001412      BEQ      2$ ;SKIP IF OK
1157      010230      012705      002226      MOV      #2226,R5
1158      010234      017704      171136      MOV      @TRCR,R4
1159      010240      013737      001376      001244      MOV      TRCR,REGIST
1160      010246      104001      ERROR    1 ;INCORRECT REGISTER MATCHUP
1161      010250      104403      004710      TYPEF    ,MRSTFL
1162
1163
1164      010254      017701      171122      2$:      MOV      @TRSR,R1 ;GET STATUS
1165      010254      042701      002247      BIC      #2247,R1 ;CLEAR UNWANTED BITS
1166      010260      005701      TST      R1 ;TEST THE STATUS REGISTER
1167
    
```

```

1168 010256 001410      BEQ          3$      ;SKIP IF OK
1169 010270 012705 000000  MOV          #0,R5
1170 010274 010104      MOV          R1,R4
1171 010276 013737 001402 001244  MOV          TRSR,REGIST
1172 010304 104001      ERROR          1      ;INCORRECT REGISTER MATCHUP
1173 010306 104404      TYPEL
1174
1175 010310      3$:
1176 010310 005737 001252      TST          ERR      ;TEST THE DEROR FLAG
1177 010314 001406      BEQ          71$      ;SKIP IF NO DEROR
1178 010316 032777 040000 170656  BIT          #BIT14,DSWR ;TEST LOOP BIT
1179 010324 001402      BEQ          71$      ;BRANCH IF CLEAR
1180 010326 000137 010052      JMP          PRETST   ;JUMP IF SET
1181 010332 005037 001252 71$:      CLR          ERR      ;CLEAR THE DEROR FLAG
1182 010336 004737 027230      JSR          PC,FIXBUF
1183
1184 010342 000207      RTS          PC
1185
1186      ;***** TEST 1 *****
1187      ;* TEST GROUP 1
1188      ;*****
1189      ;*****
1190      ;*
1191      ;* TEST 1
1192      ;*
1193      ;*****
1194 010344 012737 000001 001226 1$T1:      MOV          #1,DSSTNO ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
1195 010352 012737 010530 001216      MOV          #TST2,NEXT ;POINT TO THE START OF THE NEXT TEST
1196 010360 032777 000010 170614      BIT          #10,DSWR  ;SKIP??
1197 010366 001002      BNE          2$      ;NO
1198 010370 000137 010464      JMP          END1
1199 010374 012737 010466 001302 2$:      MCV          #TABLE1,TC ;SET TEST CONTROL INDEX
1200 010402 012706 001200      MOV          #STACK,SP ;RESET THE STACK
1201 010406
1202 010406 013737 001302 001250 1$:      MOV          TC,TSTPTR ;SET DEROR SUBTEST ADDRESS
1203 010414 013705 001302      MOV          TC,R5      ;POINT TO NEXT TEST
1204 010420 062737 000002 001302  ADD          #2,TC      ;PERFORM THE TEST
1205 010426 004775 000000      JSR          PC,@(R5)  ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
1206 010432 000240      NOP                    ;EQUIPMENT STATUS OR PROGRAM OPERATION
1207
1208 010434 005737 001252      TST          ERR      ;TEST THE DEROR FLAG
1209 010440 001406      BEQ          64$      ;SKIP IF NO DEROR
1210 010442 032777 040000 170532  BIT          #BIT14,DSWR ;TEST LOOP BIT
1211 010450 001402      BEQ          64$      ;BRANCH IF CLEAR
1212 010452 000137 010406      JMP          1$      ;JUMP IF SET
1213 010456 005037 001252 64$:      CLR          ERR      ;CLEAR THE DEROR FLAG
1214 010462 000751      BR          1$      ;DO NEXT TEST
1215 010464 104400  END1:      SCOPE          ;PASS COMPLETED
1216
1217
1218
1219      ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 1
1220
1221 010466 011562  TABLE1: TESTB ;REMOVE WRITE RING
1222 010470 023012  RUNAS ;TRY TO WRITE WITHOUT RING
1223 010472 011644  TESTC ;INSERT WRITE RING

```

1224 010474 020750
1225 010476 014070
1226 010500 021022
1227 010502 012624
1228 010504 021256
1229 010506 021614
1230 010510 022022
1231 010512 022170
1232 010514 022376
1233 010516 022604
1234 010520 014212
1235 010522 000137 002514
1236 010526 010464

TESTAJ : DELAY 10 SECONDS
TESTP : TAKE UNIT OFF LINE
TESTAK : TELL OPERATOR TO RESET
TESTI : REWIND THE TAPE
TESTAL : WRITE PAST EOT
TESTAM : SPACE REVERSE PAST EOT
TESTAN : READ PAST EOT
TESTAO : WRITE EOF PAST EOT
TESTAP : SPACE REVERSE OVER EOF PAST EOT
TESTAQ : READ EOF PAST EOT
TESTQ : REWIND AND GO OFF LINE
JMP RESTRT
END!

***** TEST 2 *****
* TEST GROUP 2

1237
1238
1239
1240
1241
1242
1243
1244
1245
1246 010530 012737 000002 001226
1247 010536 012737 010670 001216
1248 010544 012737 010636 001302
1249 010552 012706 001200
1250 010556
1251 010556 013737 001302 001250
1252 010564 013705 001302
1253 010570 062737 000002 001302
1254 010576 004775 000000
1255 010602 000240
1256
1257 010604 005737 001252
1258 010610 001406
1259 010612 032777 040000 170362
1260 010620 001402
1261 010622 000137 010556
1262 010626 005037 001252
1263 010632 000751
1264 010634 104400
1265
1266
1267
1268
1269
1270 010636 012624
1271 010640 011644
1272 010642 011756
1273 010644 012144
1274 010646 012200
1275 010650 013032
1276 010652 013266
1277 010654 013546
1278 010656 023312
1279 010660 017066

* TEST 2 *

TST2: MOV #2, TSTNO : LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
MOV #TST3, NEXT : POINT TO THE START OF THE NEXT TEST
MOV #TABLE2, TC : SET TEST CONTROL INDEX
MOV #STACK, SP : RESET THE STACK

IS: MOV TC, TSTPTR
MOV TC, R5 : SET DEROR SUBTEST ADDRESS
ADD #2, TC : POINT TO NEXT TEST
JSR PC, J(R5) : PERFORM THE TEST
NOP : THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
EQUIPMENT STATUS OR PROGRAM OPERATION

TST ERR : TEST THE DEROR FLAG
BEQ 64\$: SKIP IF NO DEROR
BIT #BIT14, QSWR : TEST LOOP BIT
BEQ 64\$: BRANCH IF CLEAR
JMP IS : JUMP IF SET
64\$: CLR ERR : CLEAR THE DEROR FLAG
BR IS : DO NEXT TEST
END2: SCOPE : PASS COMPLETED

; THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 2

TABLE2: TESTI : REWIND THE TAPE
TESTC : INSURE WRITE RING
TESTO : UNIBUS INTENTION BIT TEST
TESTE : INTERRUPT BIT SET TEST
TESTF : INTERRUPT BIT CLEAR TEST
TESTJ : ILLEGAL COMMAND TEST
TESTK : WRITE FROM LOAD POINT
TESTM : WRITE EOF FROM LOAD POINT
TESTAT : SPACE REVERSE AT LOAD POINT
TESTAB : NXM TEST

F03

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 32
 DZTRA.HED 13-APR-76 00:00 PRETEST ROUTINE

1280 010662 023520
 1291 010664 025142
 1292 010666 010634

TESTAU ;BUSS ADDRESS BIT TEST
 TESTBA ;WORD COUNT BIT TEST
 END2

***** TEST 3 *****
 * TEST GROUP 3

1293 010670 012737 000003 001226
 1294 010676 012737 011042 001216
 1295 010704 012737 010776 001302
 1296 010712 012706 001200
 1297 010716
 1298 010716 013737 001302 001250
 1299 010724 013705 001302
 1300 010730 062737 000002 001302
 1301 010736 004775 000000
 1302 010742 000240
 1303
 1304 010744 005737 001252
 1305 010750 001406
 1306 010752 032777 040000 170222
 1307 010760 001402
 1308 010762 000137 010716
 1309 010766 005037 001252
 1310 010772 000751
 1311 010774 104400
 1312
 1313
 1314
 1315
 1316
 1317 010776 011644
 1318 011000 012624
 1319 011002 013762
 1320 011004 012476
 1321 011006 023560
 1322 011010 023770
 1323 011012 013412
 1324 011014 014704
 1325 011016 015114
 1326 011020 015320
 1327 011022 017346
 1328 011024 017606
 1329 011026 020046
 1330 011030 020240
 1331 011032 024234
 1332 011034 024476
 1333 011036 024730
 1334 011040 010774
 1335

```

*****
*
* TEST 3
*
*****
TST3:  MOV    #3,2#TSTNO      ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
        MOV    #TST4,NEXT    ;POINT TO THE START OF THE NEXT TEST
        MOV    #TABLE3,TC    ;SET TEST CONTROL INDEX
        MOV    #STACK,SP     ;RESET THE STACK

IS:    MOV    TC,TSTPTR      ;SET DEROR SUBTEST ADDRESS
        MOV    TC,R5         ;POINT TO NEXT TEST
        ADD    #2,TC         ;PERFORM THE TEST
        JSR    PC,3(R5)     ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
                                EQUIPMENT STATUS OR PROGRAM OPERATION
        TST    ERR           ;TEST THE DEROR FLAG
        BEQ    64$          ;SKIP IF NO DEROR
        BIT    #BIT14,2SWR  ;TEST LOOP BIT
        BEQ    64$          ;BRANCH IF CLEAR
        JMP    IS           ;JUMP IF SET
64$:   CLR    ERR           ;CLEAR THE DEROR FLAG
        BR    IS           ;DO NEXT TEST
END3:  SCOPE                ;PASS COMPLETED
  
```

; THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 3

```

TABLE3: TESTC ;INSURE WRITE ENABLE RING
        TESTI ;REWIND THE TAPE
        TESTO ;WRITE ID BLOCK
        TESTH ;ILLEGAL ID BLOCK TEST
        TESTAV ;SPACE REVERSE OVER ID BLOCK
        TESTAW ;READ ID BLOCK
        RUNL ;WRITE EOF MARKS
        RUNU ;SPACE REVERSE OVER EOF MARKS
        RUNV ;READ EOF MARKS
        TESTW ;TIME OUT TEST
        TESTAC ;READ COUNT
        TESTAD ;READ COUNT
        TESTAE ;READ COUNT
        TESTAF ;SPACE COUNT
        TESTAX ;WRITE EVEN PARITY
        TESTAY ;SPACE REVERSE OVER EVEN PARITY
        TESTAZ ;READ EVEN PARITY
        END3
  
```


G03

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 33
DZTRA.MED 13-APR-76 00:00 PRETEST ROUTINE

```
1336 ;***** TEST 4 *****  
1337 ;* TEST GROUP 4  
1338 ;*****  
1339 ;  
1340 ;* TEST 4  
1341 ;*  
1342 ;*  
1343 ;*****  
1344 011042 012737 000004 001226 TST4: MOV #4, @TSTNO ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR  
1345 011050 012737 011160 001216 MOV #TST5, NEXT ;POINT TO THE START OF THE NEXT TEST  
1346 011056 012737 011150 001302 MOV #TABLE4, TC ;SET TEST CONTROL INDEX  
1347 011064 012706 001200 MOV #STACK, SP ;RESET THE STACK  
1348 011070  
1349 011070 013737 001302 001250 AAA: MOV TC, TSTPTR ;SET DEROR SUBTEST ADDRESS  
1350 011076 013705 001302 MOV TC, R5 ;POINT TO NEXT TEST  
1351 011102 062737 000002 001302 ADD #2, TC ;PERFORM THE TEST  
1352 011110 004775 000000 JSR PC, @ (R5)  
1353 011114  
1354 011114 000240 RTA: NOP ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT  
1355 ;EQUIPMENT STATUS OR PROGRAM OPERATION  
1356 011116 005737 001252 TST ERR ;TEST THE DEROR FLAG  
1357 011122 001406 BEQ 64$ ;SKIP IF NO DEROR  
1358 011124 032777 040000 170050 BIT #BIT14, JSWR ;TEST LOOP BIT  
1359 011132 001402 BEQ 64$ ;BRANCH IF CLEAR  
1360 011134 000137 011070 JMP AAA ;JUMP IF SET  
1361 011140 005037 001252 64$: CLR ERR ;CLEAR THE DEROR FLAG  
1362 011144 000751 BR AAA ;DO NEXT TEST  
1363 011146 104400 END4: SCOPE ;PASS COMPLETED  
1364  
1365  
1366  
1367 ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 4  
1368  
1369 011150 011644 TABLE4: TESTC ;INSURE WRITE RING  
1370 011152 013762 TESTO ;REWIND AND WRITE ID BLOCK  
1371 011154 026206 TESTXX ;WRITE, ERASE, SPACE REV, READ AND TEST DATA  
1372 011156 011146 END4  
1373  
1374 ;***** TEST 5 *****  
1375 ;* TEST GROUP 5  
1376 ;*****  
1377 ;  
1378 ;* TEST 5  
1379 ;*  
1380 ;*  
1381 ;*****  
1382 011160 012737 000005 001226 TST5: MOV #5, @TSTNO ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR  
1383 011166 012737 011314 001216 MOV #TST6, NEXT ;POINT TO THE START OF THE NEXT TEST  
1384 011174 012737 011304 001302 MOV #TABLE5, TC ;SET TEST CONTROL INDEX  
1385 011202 012706 001200 MOV #STACK, SP ;RESET THE STACK  
1386 011206  
1387 011206 013737 001302 001250 BBB: MOV TC, TSTPTR ;SET DEROR SUBTEST ADDRESS  
1388 011214 013705 001302 MOV TC, R5 ;POINT TO NEXT TEST  
1389 011220 062737 000002 001302 ADD #2, TC ;PERFORM THE TEST  
1390 011226 004775 000000 JSR PC, @ (R5)  
1391 011232 RTB:
```

```

1392 011232 000240          NOP          ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
1393                                     ;EQUIPMENT STATUS OR PROGRAM OPERATION
1394 011234 005737 001252    TST      ERR          ;TEST THE DEROR FLAG
1395 011240 001406          BEQ      64$         ;SKIP IF NO DEROR
1396 011242 032777 040000 167732 BIT      #BIT14,2SWR ;TEST LOOP BIT
1397 011250 001402          BEQ      64$         ;BRANCH IF CLEAR
1398 011252 000137 011206    JMP      BBB         ;JUMP IF SET
1399 011256 005037 001252    CLR      ERR         ;CLEAR THE DEROR FLAG
1400 011262 000751          BR       BBB         ;DO NEXT TEST
1401 011264
1402 011264 032777 000010 167710 BIT      #10,2SWR    ;SKIP NEXT TEST?
1403 011272 001003          BNE     1$          ;NO
1404 011274 012737 002376 001216 MOV     #.EOP,NEXT ;SET EOP
1405 011302 104400          1$:    SCOPE        ;PASS COMPLETED
1406
1407                                     ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 5
1408
1409 011304 012624          TABLE5: TESTI    ;REWIND THE TAPE
1410 011306 023770          TESTAW   ;READ THE ID BLOCK
1411 011310 026434          TESTXY  ;READ THE TAPE
1412 011312 011264          ENDS
1413
1414                                     ;***** TEST 6 *****
1415                                     ;* TEST GROUP 6
1416                                     ;*****
1417                                     ;*****
1418                                     ;*
1419                                     ;* TEST 6
1420                                     ;*
1421                                     ;*****
1422 011314 012737 000006 001226 TST6:  MOV     #6,2#TSTNO ;LOAD THE NUMBER OF THIS TEST IN THE INDICATOR
1423 011322 012737 002376 001216 MOV     #.EOP,NEXT ;POINT TO THE END-OF-PASS HANDLER
1424 011330 012737 011516 001302 MOV     #TABLE6,TC ;SET TEST CONTROL INDEX
1425 011336 012706 001200          MOV     #STACK,$P ;RESET THE STACK
1426 011342
1427 011342 104405          1$:    INSTR      ;CALL THE STRING INPUT ROUTINE
1428 011344 011454          MESTADD ;POINTER TO MESSAGE TO BE PRINTED
1429 011346 104407          PARAM    ;CALL THE OCTAL TO ASCII CONVERT ROUTINE
1430 011350 010344          TSTI    ;LOWEST LEGITIMATE VALUE OF EXPECTED RESPONSE
1431 011352 026206          TESTXX  ;HIGHEST LEGITIMATE VALUE OF EXPECTED REPOSE
1432 011354 011516          TABLE6 ;POINTER TO MAP LOCATION TO BE FILLED
1433 011356 001          .BYTE   1          ;MASK OF INVALID BITS FOR THIS PARAMETER
1434 011357 001          .BYTE   1          ;NUMBER OF PARAMETERS TO STORE
1435 011360 012737 011516 001302 2$:    MOV     #TABLE6,TC ;SET TEST CONTROL INDEX
1436 011366 017737 167710 001250 MOV     @TC,TSTPTR ;SET DEROR SUBTEST ADDRESS
1437 011374 013705 001302 MOV     TC,R5
1438 011400 062737 000002 001302 ADD     #2,TC
1439 011406 004775 000000 JSR     PC,@(R5)
1440 011412 000240          NOP
1441                                     ;PERFORM THE TEST
1442 011414 012737 177777 001252 MOV     #-1,ERR ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
1443 011422 005737 001252    TST     ERR         ;EQUIPMENT STATUS OR PROGRAM OPERATION
1444 011426 001406          BEQ     64$         ;SET DEROR FLAG
1445 011430 032777 040000 167544 BIT     #BIT14,2SWR ;TEST THE DEROR FLAG
1446 011436 001402          BEQ     64$         ;SKIP IF NO DEROR
1447 011440 000137 011360    JMP     2$          ;TEST LOOP BIT
                                     ;BRANCH IF CLEAR
                                     ;JUMP IF SET

```

```

1448 011444 005037 001252      64$: CLR ERR ;CLEAR THE DEROR FLAG
1449 011450 000734              BR 1$ ;DO NEXT TEST
1450 011452 104400      END6: SCOPE ;PASS COMPLETED
1451
1452 011454 042777 052116 051105 MESTADD: .ASCIZ <377>/ENTER STARTING ADDRESS OF TEST /
1453 011462 051440 040524 052122
1454 011470 047111 020107 042101
1455 011476 051104 051505 020123
1456 011504 043117 052040 051505
1457 011512 020124 000
1458 011516 .EVEN
1459
1460
1461 ;THIS TABLE CONTAINS THE SUBTESTS TO BE PERFORMED BY TEST 6
1462
1463 011516 000000 TABLE6: 0
1464 011520 011452 END6
1465
1466 ;-----*
1467
1468 .SBTTL SUB-TEST ROUTINES
1469
1470 ;THIS ROUTINE WAITS FOR TAPE UNIT TO GET READY OR DEROR
1471 ;OR TIMES OUT AND EXITS WHEN IT DOES
1472
1473 011522 005001 TESTA: CLR R1 ;CLEAR TIMER
1474 011524 1$:
1475 011524 005201 INC R1 ;COUNT UP
1476 011526 001414 BEQ 2$ ;SKIP IF ZERO
1477 011530 032777 100000 167640 BIT #BIT15,@TRCR
1478 011536 001010 BNE 2$ ;SKIP IF SET
1479 011540 032777 002000 167630 BIT #BIT10,@TRCR
1480 011546 001766 BEQ 1$ ;NO-BRANCH
1481 011550 032777 000200 167620 BIT #BIT7,@TRCR
1482 011556 001762 BEQ 1$ ;NO-BRANCH
1483 011560
1484 011560 000207 2$: RTS PC
1485
1486 ;-----*
1487 ;THIS ROUTINE INSURES THE WRITE ENABLE RING
1488 ;REMOVED AND EXITS WHEN IT IS
1489
1490 011562 TESTB:
1491 011562 032777 000004 167612 BIT #BIT2,@TRSR
1492 011570 001015 BNE 2$ ;SKIP IF ONE
1493 011572 104402 004774 TYPE ,MRMVRG
1494 011576 1$:
1495 011576 032777 000004 167576 BIT #BIT2,@TRSR
1496 011604 001007 BNE 2$ ;SKIP IF ONE
1497 011606 005237 007434 INC TEMP
1498 011612 001003 BNE 64$
1499 011614 012777 000207 167370 MOV #207,@TPDBR ;RING THE BELL
1500 011622 64$:
1501 011622 000765 BR 1$
1502 011624 2$:
1503 011624 052777 004000 167544 BIS #PWRCLR,@TRCR ;DEVICE MASTER RESET
  
```

```

1504 011632 032777 004000 167536 65$: BIT #PWRCLR, @TRCR ; INSTRUCTION CLEAR ?
1505 011640 001374 BNE 65$ ; IF NO, WAIT FOR IT TO CLEAR
1506 011642 000207 RTS PC
1507
1508

```

```

-----*
; THIS ROUTINE INSURES THE WRITE ENABLE RING
; IS INSERTED AND EXITS WHEN IT IS

```

```

1509
1510
1511
1512 011644 TESTC:
1513 011644 032777 000004 167530 BIT #BIT2, @TRSR
1514 011652 001420 BEQ 2$ ; SKIP IF ZERO
1515 011654 104402 005054 TYPE ,MINSRG
1516 011650 1$:
1517 011660 005237 007434 INC TEMP
1518 011664 001003 BNE 64$
1519 011666 012777 000207 167316 MOV #207, @TPDBR ; RING THE BELL
1520 011674 64$:
1521 011674 032777 000004 167500 BIT #BIT2, @TRSR
1522 011702 001366 BNE 1$ ; LOOP TILL ZERO
1523 011704 032777 002000 167464 BIT #BIT10, @TRCR
1524 011712 001762 BEQ 1$ ; BRANCH IF CLEAR
1525 011714 2$:
1526 011714 052777 004000 167454 BIS #PWRCLR, @TRCR ; DEVICE MASTER RESET
1527 011722 032777 004000 167446 65$: BIT #PWRCLR, @TRCR ; INSTRUCTION CLEAR ?
1528 011730 001374 BNE 65$ ; IF NO, WAIT FOR IT TO CLEAR
1529 011732 004737 013662 JSR PC, TESTN
1530 011736 052777 004000 167432 BIS #PWRCLR, @TRCR ; DEVICE MASTER RESET
1531 011744 032777 004000 167424 66$: BIT #PWRCLR, @TRCR ; INSTRUCTION CLEAR ?
1532 011752 001374 BNE 66$ ; IF NO, WAIT FOR IT TO CLEAR
1533 011754 000207 RTS PC
1534

```

```

-----+
; THIS ROUTINE TESTS THE READ WRITE ABILITY OF THE
; UNIBUS EXTENSION ADDRESS BITS

```

```

1535
1536
1537
1538
1539
1540 011756 TESTD:
1541 011756 052777 004000 167412 BIS #PWRCLR, @TRCR ; DEVICE MASTER RESET
1542 011764 032777 004000 167404 64$: BIT #PWRCLR, @TRCR ; INSTRUCTION CLEAR ?
1543 011772 001374 BNE 64$ ; IF NO, WAIT FOR IT TO CLEAR
1544 011774 013737 001376 001244 MOV TRCR, REGIST ; INDICATE THE ERROR REGISTER
1545 012002 052777 010000 167366 BIS #010000, @TRCR ; SET UNIBUS EXTENSION BIT 16
1546 012010 032777 010000 167360 BIT #010000, @TRCR ; TEST UNIBUS EXTENSION BIT 16
1547 012016 001001 BNE 1$ ; SKIP IF OK
1548 012020 104000 ERROR ; ERROR-BIT WAS NOT CORRECT
1549
1550 012022 1$:
1551
1552 012022 042777 010000 167346 BIC #010000, @TRCR ; CLEAR THE BIT
1553 012030 032777 010000 167340 BIT #010000, @TRCR ; TEST THE BIT
1554 012036 001401 BEQ 2$ ; SKIP IF CLEAR
1555 012040 104000 ERROR ; ERROR-BIT WAS NOT CORRECT
1556
1557 012042 2$:
1558
1559

```


L03

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 38
 DZTRA.MED 13-APR-76 00:00 SUB-TEST ROUTINES

```

1616 012220 104403 005156          TYPEF  .MIDSFL
1617
1618 012224          1$:
1619 012224 005037 177776          CLR    PS          ;SET PROCESSOR PRIORITY TO 00
1620 012230 000207          RTS    PC
1621
1622          ;-----
1623
1624          ;THIS ROUTINE TESTS THE FUNCTION OF THE POWER CLEAR
1625
1626 012232 000005          TESTG: RESET
1627 012234 012777 000026 167134  MOV    #26,@TRCR    ;SET COMMAND REGISTER
1628 012242 052777 004000 167126  BIS    #BIT11,@TRCR
1629 012250 012701 177300          MOV    #-500,R1    ;SET DELAY
1630 012254          1$:
1631 012254 005201          INC    R1          ;DELAY
1632 012256 001376          BNE    1$         ;CONTINUE
1633
1634 012260 032777 004000 167110  BIT    #BIT11,@TRCR
1635 012266 001001          BNE    2$         ;SKIP IF SET
1636 012270 104000          ERROR   2$       ;ERROR-BIT WAS NOT CORRECT
1637
1638 012272          2$:
1639
1640 012272 013727 001240          MOV    HOLD,(PC)+ ;PICK UP TIME PARAMETER
1641 012276 000000          64$: .WORD 0        ;USE THIS WORD AS A TIME COUNTER
1642 012300          66$:
1643 012300 005227 000000          INC    #0         ;IF NO,COUNT 1 OF 65535 TICKS
1644 012304 001375          BNE    66$       ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
1645 012306 005337 012276          DEC    64$       ;HAS THE TOTAL TIME ELAPSED?
1646 012312 001372          BNE    66$       ;IF NO,GO WAIT A LITTLE LONGER
1647 012314          65$:
1648 012314 013727 001240          MOV    HOLD,(PC)+ ;PICK UP TIME PARAMETER
1649 012320 000000          67$: .WORD 0        ;USE THIS WORD AS A TIME COUNTER
1650 012322          69$:
1651 012322 005227 000000          INC    #0         ;IF NO,COUNT 1 OF 65535 TICKS
1652 012326 001375          BNE    69$       ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
1653 012330 005337 012320          DEC    67$       ;HAS THE TOTAL TIME ELAPSED?
1654 012334 001372          BNE    69$       ;IF NO,GO WAIT A LITTLE LONGER
1655 012336          68$:
1656 012336          3$:
1657 012336 032777 004000 167032  BIT    #BIT11,@TRCR
1658 012344 001403          BEQ    4$         ;SKIP IF CLEAR
1659 012346 005237 007434          INC    TEMP
1660 012352 001371          BNE    3$         ;DELAY
1661
1662 012354          4$:
1663
1664 012354 032777 004000 167014  BIT    #BIT11,@TRCR
1665 012362 001401          BEQ    5$         ;SKIP IF CLEAR
1666 012364 104000          ERROR   5$       ;ERROR-BIT WAS NOT CORRECT
1667
1668 012366          5$:
1669
1670 012366 032777 002000 167006  BIT    #BIT10,@TRSR
1671 012374 001001          BNE    6$         ;SKIP IF ON LINE
  
```

```

1672 012376 000207          RTS      PC
1673 012400          6$:
1674
1675 012400 017701 166772      MOV      @TRCR,R1          ;GET COMMAND REGISTER
1676 012404 042701 000036      BIC      #00036,R1       ;CLEAR UNWANTED BITS
1677 012410 022701 002200      CMP      #2200,R1       ;TEST COMMAND REGISTER
1678 012414 001411          BEQ      7$              ;SKIP IF OK
1679 012416 012705 002200      MOV      #2200,R5
1680 012422 010104          MOV      R1,R4
1681 012424 013737 001376 001244  MOV      TRCR,REGIST
1682 012432 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
1683 012434 104403 005203      TYPEF   ,MPCFL
1684
1685 012440          7$:
1686
1687 012440 017701 166736      MOV      @TRSR,R1          ;GET STATUS REGISTER
1688 012444 042701 002247      BIC      #2247,R1       ;CLEAR UNWANTED BITS
1689 012450 005701          TST      R1              ;TEST STATUS REGISTER
1690 012452 001410          BEQ      8$              ;SKIP IF OK
1691 012454 012705 002247      MOV      #2247,R5
1692 012460 010104          MOV      R1,R4
1693 012462 013737 001402 001244  MOV      TRSR,REGIST
1694 012470 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
1695 012472 104404          TYPEL
1696
1697 012474          8$:
1698
1699 012474 000207          RTS      PC
1700
1701 -----
1702 ; THIS ROUTINE IS USED TO TEST THE FUNCTION OF WRITING A ID
1703 ; BLOCK FROM LOAD POINT AND AFTER LOAD POINT.
1704
1705 TESTH:
1706 012476 004737 013762      JSR      PC,TESTO
1707 012502 004737 025605      JSR      PC,TESTCK
1708
1709 012506 004537 026544      JSR      R5,TRYIT
1710 012512 000432          .WORD   WIOB            ;:FUNCTION
1711 012514 000000          .WORD   -0              ;:WORD COUNT
1712 012516 000000          .WORD   0                ;:BUS ADDRESS
1713
1714 012520 022777 142632 166650      CMP      #142632,@TRCR   ;TEST COMMAND REGISTER
1715 012526 001412          BEQ      1$              ;SKIP IF OK
1716 012530 012705 142632      MOV      #142632,R5
1717 012534 017704 166636      MOV      @TRCR,R4
1718 012540 013737 001376 001244  MOV      TRCR,REGIST
1719 012546 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
1720 012550 104403 005222      TYPEF   ,MWIPLFL
1721
1722 012554          1$:
1723
1724 012554 022777 002000 166620      CMP      #2000,@TRSR     ;TEST THE STATUS REGISTER
1725 012562 001410          BEQ      2$              ;SKIP IF OK
1726 012564 012705 002000      MOV      #2000,R5
1727 012570 017704 166606      MOV      @TRSR,R4
    
```

```

1728 012574 013737 001402 001244      MOV   TRSR,REGIST
1729 012602 104001                      ERROR   1      ;INCORRECT REGISTER MATCHUP
1730
1731 012604                      2$:
1732
1733 012604 052777 004000 166564      BIS   #PWRCLR,@TRCR ;DEVICE MASTER RESET
1734 012612 032777 004000 166556 64$:  BIT   #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
1735 012620 001374                      BNE   64$      ;IF NO, WAIT FOR IT TO CLEAR
1736 012622 000207                      RTS   PC
1737
1738                                     ;+-----+
1739                                     ;THIS ROUTINE TESTS THE FUNCTION OF REWINDING
1740                                     ;WHILE AT LOAD POINT THIS IS THE MAIN REWIND ROUTINE
1741
1742 012624                      TESTI:
1743 012624 052777 004000 166544      BIS   #PWRCLR,@TRCR ;DEVICE MASTER RESET
1744 012632 032777 004000 166536 64$:  BIT   #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
1745 012640 001374                      BNE   64$      ;IF NO, WAIT FOR IT TO CLEAR
1746 012642 004537 026544      JSR   R5,TRYIT
1747 012646 001020                      .WORD REWIND
1748 012650 000000                      .WORD -0
1749 012652 000000                      .WORD 0
1750                                     A=.
1751 012654 032777 000002 166520      BIT   #BIT1,@TRSR
1752 012662 001374                      BNE   A      ;BRANCH IF SET
1753 012664 032777 000040 166510      BIT   #BITS,@TRSR
1754 012672 001770                      BEQ   A      ;BRANCH IF CLEAR
1755 012674 052777 004000 166474      BIS   #PWRCLR,@TRCR ;DEVICE MASTER RESET
1756 012702 032777 004000 166466 65$:  BIT   #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
1757 012710 001374                      BNE   65$      ;IF NO, WAIT FOR IT TO CLEAR
1758 012712 004537 026544      JSR   R5,TRYIT
1759 012716 001020                      .WORD REWIND
1760 012720 000000                      .WORD -0
1761 012722 000000                      .WORD 0
1762
1763 012724 022777 003220 166444      CMP   #3220,@TRCR ;TEST COMMAND REGISTER
1764 012732 001412                      BEQ   A$0
1765 012734 012705 003220      MOV   #3220,R5
1766 012740 017704 166432      MOV   @TRCR,R4
1767 012744 013737 001376 001244      MOV   TRCR,REGIST
1768 012752 104001                      ERROR   1      ;INCORRECT REGISTER MATCHUP
1769 012754 104403 005264      TYPEF ,MRWLPFL
1770
1771 012760 000001                      A$0:  $TAGAS=$TAGAS+1
1772
1773 012760 022777 002040 166414      CMP   #2040,@TRSR ;TEST STATUS REGISTER
1774 012766 001411                      BEQ   A$1
1775 012770 012705 002040      MOV   #2040,R5
1776 012774 017704 166402      MOV   @TRSR,R4
1777 013000 013737 001402 001244      MOV   TRSR,REGIST
1778 013006 104001                      ERROR   1      ;INCORRECT REGISTER MATCHUP
1779 013010 104404      TYPEL
1780
1781 013012 000002                      A$1:  $TAGAS=$TAGAS+1
1782
1783 013012 052777 004000 166356      BIS   #PWRCLR,@TRCR ;DEVICE MASTER RESET

```



```

1794 013020 032777 004000 166350 64S: BIT #PWRCLR, JTRCR ; INSTRUCTION CLEAR /
1795 013026 001374 BNE 64S ; IF NO, WAIT FOR IT TO CLEAR
1796 013030 000207 RTS PC
-----
; THIS ROUTINE TESTS THE FUNCTIONS OF ILLEGAL COMMANDS
1797
1798 013032 TESTJ: JSR PC, TESTI
1799 013032 004737 012624 JSR PC, ENAINT
1800 013036 004737 030210 MOV #ILCTAB, R3 ; POINT TO THE ILLEGAL INSTRUCTION CODE TABLE
1801 013042 012703 013252 MOV #6, R4 ; COUNT THE NUMBER OF ILLEGAL INSTRUCTIONS
1802 013046 012704 000006 MOV (R3)+, 2S ; INSERT AN ILLEGAL INSTRUCTION CODE
1803 013052 012337 013062 1S: JSR R5, TRYIT ; GO DO THE FUNCTION
1804 013056 004537 026544 JSR .WORD ; LOCATION OF FUNCTION TO DO
1805 013062 000000 2S: .WORD 0, 0 ; PARAMETERS TO TRYIT
1806 013064 000000 000000 JSR PC, TESTJ1 ; CHECK IT OUT
1807 013070 004737 013102 JSR PC, TESTJ1 ; CHECK IT OUT
1808 013074 005304 DEC R4 ; REDUCE THE COUNT. ARE WE DONE?
1809 013076 001365 BNE 1S ; IF NOT, CONTINUE
1810 013106 000207 RTS PC
1811
1812 013102 012737 000030 001240 TESTJ1: MOV #30, HOLD ; DELAY PAST TIME OUT
1813 013110 013727 001240 MOV HOLD, (PC)+ ; PICK UP TIME PARAMETER
1814 013114 000000 64S: .WORD 0 ; USE THIS WORD AS A TIME COUNTER
1815 013116 005227 000000 66S: INC #0 ; IF NO, COUNT 1 OF 65535 TICKS
1816 013122 001375 BNE 66S ; HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
1817 013124 005337 013114 DEC 64S ; HAS THE TOTAL TIME ELAPSED?
1818 013130 001372 BNE 66S ; IF NO, GO WAIT A LITTLE LONGER
1819
1820 013132 012737 000010 001240 65S: MOV #10, HOLD ; RESTORE NORMAL DELAY FACTOR
1821 013140 017701 166232 MOV JTRCR, R1 ; GET COMMAND REGISTER
1822 013144 042701 001436 BIC #1436, R1 ; CLEAR UNWANTED BITS
1823 013150 022701 142200 CMP #142200, R1 ; TEST COMMAND REGISTER
1824 013154 001411 BEQ AS2
1825 013156 012705 142200 MOV #142200, R5
1826 013162 010104 MOV R1, R4
1827 013164 013737 001376 001244 MOV TRCR, REGIST
1828 013172 104001 ERROR 1 ; INCORRECT REGISTER MATCHUP
1829 013174 104403 005316 TYPEF ,MILFFL
1830
1831 013200 000003 AS2: STAGAS=STAGAS+1
1832 013200 017701 166176 MOV JTRSR, R1 ; GET STATUS REGISTER
1833 013204 022701 002040 CMP #2040, R1 ; TEST STATUS REGISTER
1834 013210 001410 BEQ AS3
1835 013212 012705 002040 MOV #2040, R5
1836 013216 010104 MOV R1, R4
1837 013220 013737 001402 001244 MOV TRSR, REGIST
1838 013226 104001 ERROR 1 ; INCORRECT REGISTER MATCHUP
1839 013230 104404 TYPEL
1840
1841 013232 000004 AS3: STAGAS=STAGAS+1
    
```

```

1840 013232 052777 004000 166136      BIS      #PWRCLR, @TRCR ;DEVICE MASTER RESET
1841 013240 032777 004000 166130 64$:    BIT      #PWRCLR, @TRCR ;INSTRUCTION CLEAR ?
1842 013246 001374          BNE     64$      ;IF NO, WAIT FOR IT TO CLEAR
1843 013250 000207          RTS     PC
1844 013252 000496 001012 001414 ILCTAB: .WORD ILC03, ILC05, ILC06, ILC11, ILC12, ILC14
1845 013260 000022 000424 001030
  
```

 ; THIS ROUTINE TESTS THE DEROR CONDITIONS OF WRITING
 ; FROM LOAD POINT

```

1852 013266          TESTK:
1853 013266 004737 012624      JSR     PC, TESTI
1854 013272 004537 026544      JSR     R5, TRYIT
1855 013276 000402          .WORD  WRITE      ;FUNCTION
1856 013300 177700          .WORD  -100       ;WORD COUNT
1857 013302 030576          .WORD  OUTPUT     ;BUS ADDRESS
1858
1859 013304 022777 142602 166064      CMP     #142602, @TRCR ;TEST CONTROL REGISTER
1860 013312 001412          BEQ     A$4
1861 013314 012705 142602          MOV     #142602, R5
1862 013320 017704 166052          MOV     @TRCR, R4
1863 013324 013737 001376 001244      MOV     TRCR, REGIST
1864 013332 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
1865 013334 104403 005342          TYPEF  ,MWPFL
  
```

A\$4: STAGAS=STAGAS+1

```

1867 013340 000005
1868
1869 013340 022777 002040 166034      CMP     #2040, @TRSR ;TEST STATUS REGISTER
1870 013346 001411          BEQ     A$5
1871 013350 012705 002040          MOV     #2040, R5
1872 013354 017704 166022          MOV     @TRSR, R4
1873 013360 013737 001402 001244      MOV     TRSR, REGIST
1874 013366 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
1875 013370 104404          TYPEF  ,MWPFL
  
```

A\$5: STAGAS=STAGAS+1

```

1877 013372 000006
1878
1879 013372 052777 004000 165776      BIS      #PWRCLR, @TRCR ;DEVICE MASTER RESET
1880 013400 032777 004000 165770 64$:    BIT      #PWRCLR, @TRCR ;INSTRUCTION CLEAR ?
1881 013406 001374          BNE     64$      ;IF NO, WAIT FOR IT TO CLEAR
1882 013410 000207          RTS     PC
  
```

 ; THIS ROUTINE WRITES A END OF FILE MARK
 ; TAPE MUST BE POSITIONED PRIOR TO ENTERING THIS ROUTINE

```

1888 013412          RUNL:
1889 013412 004737 020710      JSR     PC, TESTAH
1890 013416 012737 000062 013442      MOV     #50, 3$
1891 013424 005737 013442 2$:     TST     3$
1892 013430 001001          BNE     1$
1893 013432 000207          RTS     PC
1894 013434 005337 013442 1$:     DEC     3$
1895 013440 000401          BR     4$
  
```

```

1896 013442 000000 3S: 0
1897 013444 4S:
1898
1899 013444 TESTL:
1900 013444 004537 026544 JSR R5,TRYIT ;:
1901 013450 001034 .WORD WEOF ;FUNCTION
1902 013452 000000 .WORD -0 ;WORD COUNT
1903 013454 000000 .WORD 0 ;BUS ADDRESS
1904 013456 022777 003234 165712 CMP #3234,@TRCR ;TEST COMMAND REGISTER
1905 013464 001412 BEQ AS6
1906 013466 012705 003234 MOV #3234,R5
1907 013472 017704 165700 MOV @TRCR,R4
1908 013476 013737 001376 001244 MOV TRCR,REGIST
1909 013504 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
1910 013506 104403 005374 TYPEF ,MWEFFL
1911
1912 013512 000007 AS6: STAGAS=STAGAS+1
1913
1914 013512 022777 002011 165662 CMP #2011,@TRSR ;TEST THE STATUS REGISTER
1915 013520 001411 BEQ AS7
1916 013522 012705 002011 MOV #2011,R5
1917 013526 017704 165650 MOV @TRSR,R4
1918 013532 013737 001402 001244 MOV TRSR,REGIST
1919 013540 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
1920 013542 104404 TYPEF
1921
1922 013544 000010 AS7: STAGAS=STAGAS+1
1923
1924
1925 013544 000207 RTS PC
1926
1927 ;-----
1928
1929 ;THIS ROUTINE ROUTINE TESTS THE DEROR CONDITION OF WRITING A END
1930 ;OF FILE MARK FROM LOAD POINT.
1931
1932 013546 TESTM:
1933 013546 004737 012624 JSR PC,TESTI
1934 013552 004537 026544 JSR R5,TRYIT ;:
1935 013556 001034 .WORD WEOF ;FUNCTION
1936 013560 000000 .WORD -0 ;WORD COUNT
1937 013562 000000 .WORD 0 ;BUS ADDRESS
1938 013564 022777 143234 165604 CMP #143234,@TRCR ;TEST COMMAND REGISTER
1939 013572 001412 BEQ AS10
1940 013574 012705 143234 MOV #143234,R5
1941 013600 017704 165572 MOV @TRCR,R4
1942 013604 013737 001376 001244 MOV TRCR,REGIST
1943 013612 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
1944 013614 104403 005421 TYPEF ,MWELPFL
1945
1946 013620 000011 AS10: STAGAS=STAGAS+1
1947
1948 013620 022777 002040 165554 CMP #2040,@TRSR ;TEST THE STATUS REGISTER
1949 013626 001411 BEQ AS11
1950 013630 012705 002040 MOV #2040,R5
1951 013634 017704 165542 MOV @TRSR,R4

```

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 44
 DZTRA.MED 13-APR-76 00:00 SUB-TEST ROUTINES

```

1952 013640 013737 001402 001244      MOV   TRSR,REGIST
1953 013646 104001      ERROR 1          ;INCORRECT REGISTER MATCHUP
1954 013650 104404      TYPEL
1955
1956 013652 000012      AS11:  STAGAS=STAGAS+1
1957
1958 013652 004737 012232      JSR   PC,TESTG
1959 013656 000207      RTS   PC
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969 013660 000207      A=.
1970 013662
1971 013662 032777 002000 165512      TESTN:  RTS   PC
1972 013670 001373      BIT   #BIT10,@TRSR
1973 013672 104402 005467      BNE   A          ;EXIT IF ON LINE
1974
1975 013676 032777 002000 165476      A=.
1976 013704 001007      BIT   #BIT10,@TRSR
1977 013706 005237 007434      BNE   AS12
1978 013712 001003      INC   TEMP
1979 013714 012777 000207 165270      BNE   64$
1980 013722      MOV   #207,@TPDDBR ;RING THE BELL
1981 013722 000765      64$:
1982 013724 000013      AS12:  BR    A
1983 013724 032777 100000 165444      STAGAS=STAGAS+1
1984 013732 001403      BIT   #BIT15,@TRCR
1985 013734 104000      BEQ   AS13
1986 013736 104403 005552      ERROR
1987
1988 013742 000014      TYPEF ,MOLEFL ;ERROR-BIT WAS NOT CORRECT
1989 013742 052777 004000 165426      AS13:  STAGAS=STAGAS+1
1990 013750 032777 004000 165420      BIS   #PWRCLR,@TRCR ;DEVICE MASTER RESET
1991 013756 001374      BIT   #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
1992 013760 000207      BNE   64$       ;IF NO, WAIT FOR IT TO CLEAR
1993
1994
1995
1996
1997
1998 013762
1999 013762 004737 012624      TESTO:  JSR   PC,TESTI
2000 013766 004537 026544      JSR   R5,TRYIT
2001 013772 000432      .WORD WIOB      ;FUNCTION
2002 013774 000000      .WORD -0        ;WORD COUNT
2003 013776 000000      .WORD 0         ;BUS ADDRESS
2004 014000 022777 002632 165370      CMP   #2632,@TRCR ;TEST COMMAND REGISTER
2005 014006 001412      BEQ   AS14
2006 014010 012705 002632      MOV   #2632,R5
2007 014014 017704 165356      MOV   @TRCR,R4

```

-----*

;THIS ROUTINE TESTS FOR TAPE TRANSPORT ON LINE AND EXITS WHEN
 ;IT IS.

```

A=.
TESTN:  RTS   PC
        BIT   #BIT10,@TRSR
        BNE   A          ;EXIT IF ON LINE
        TYPE ,MUOLN
A=.
        BIT   #BIT10,@TRSR
        BNE   AS12
        INC   TEMP
        BNE   64$
        MOV   #207,@TPDDBR ;RING THE BELL
64$:
AS12:  BR    A
        STAGAS=STAGAS+1
        BIT   #BIT15,@TRCR
        BEQ   AS13
        ERROR
        TYPEF ,MOLEFL ;ERROR-BIT WAS NOT CORRECT
AS13:  STAGAS=STAGAS+1
        BIS   #PWRCLR,@TRCR ;DEVICE MASTER RESET
64$:  BIT   #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
        BNE   64$       ;IF NO, WAIT FOR IT TO CLEAR
        RTS   PC

```

-----*
 ;THIS ROUTINE IS USED TO TEST THE FUNCTION OF WRITING A ID
 ;BLOCK FROM LOAD POINT THIS IS THE MAIN ID BLOCK ROUTINE

```

TESTO:  JSR   PC,TESTI
        JSR   R5,TRYIT
        .WORD WIOB      ;FUNCTION
        .WORD -0        ;WORD COUNT
        .WORD 0         ;BUS ADDRESS
        .WORD 0         ;TEST COMMAND REGISTER
        CMP   #2632,@TRCR
        BEQ   AS14
        MOV   #2632,R5
        MOV   @TRCR,R4

```

```

2008 014020 013737 001376 001244      MOV      TRCR,REGIST
2009 014026 104001                      ERROR    1          ;INCORRECT REGISTER MATCHUP
2010 014030 104403 005573                      TYPEF    ,MWILPFL
2011
2012 014034 000015                      AS14:   STAGAS=STAGAS+1
2013
2014 014034 022777 002021 165340      CMP      #2021,2TRSR          ;TEST THE STATUS REGISTER
2015 014042 001411                      BEQ      AS15
2016 014044 012705 002021      MOV      #2021,R5
2017 014050 017704 165326      MOV      2TRSR,R4
2018 014054 013737 001402 001244      MOV      TRSR,REGIST
2019 014052 104001                      ERROR    1          ;INCORRECT REGISTER MATCHUP
2020 014054 104404                      TYPEF
2021
2022 014066 000016                      AS15:   STAGAS=STAGAS+1
2023
2024
2025 014066 000207                      RTS      PC
2026
2027
2028
2029
2030
2031
2032
2033
2034 014070 004537 026544      JSR      R5,TRYIT
2035 014074 001436                      .WORD   OFFLINE      ;:FUNCTION
2036 014076 000000                      .WORD   -0            ;:WORD COUNT
2037 014100 000000                      .WORD   0             ;:BUS ADDRESS
2038
2039 014102 032777 002000 165272      BIT      #BIT10,2TRSR
2040 014110 001403                      BEQ      AS16
2041 014112 104000                      ERROR
2042 014114 104403 005635                      TYPEF    ,MUNOFLN      ;:ERROR-BIT WAS NOT CORRECT
2043
2044 014120 000017                      AS16:   STAGAS=STAGAS+1
2045 014120 032777 100000 165250      BIT      #BIT15,2TRCR
2046 014126 001003                      BNE     AS17
2047 014130 104000                      ERROR
2048 014132 104403 005671                      TYPEF    ,MOFLEFL      ;:ERROR-BIT WAS NOT CORRECT
2049
2050 014136 000020                      AS17:   STAGAS=STAGAS+1
2051 014136 052777 004000 165232      BIS     #BIT11,2TRCR
2052 014144 013727 001240      MOV     HOLD,(PC)+
2053 014150 000000                      64$:   .WORD   0          ;:PICK UP TIME PARAMETER
2054 014152                      66$:   ;:USE THIS WORD AS A TIME COUNTER
2055 014152 005227 000000                      INC     #0            ;:IF NO,COUNT 1 OF 65535 TICKS
2056 014156 001375                      BNE     66$          ;:HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2057 014160 005337 014150                      DEC     64$          ;:HAS THE TOTAL TIME ELAPSED?
2058 014164 001372                      BNE     66$          ;:IF NO,GO WAIT A LITTLE LONGER
2059 014166
2060 014166 004737 013662                      65$:   JSR      PC,TESTN
2061 014172 052777 004000 165176      BIS     #PWRCLR,2TRCR      ;:DEVICE MASTER RESET
2062 014200 032777 004000 165170      67$:   BIT      #PWRCLR,2TRCR  ;:INSTRUCTION CLEAR ?
2063 014206 001374                      BNE     67$          ;:IF NO, WAIT FOR IT TO CLEAR

```

Address	PC	RS	PC	Code	Comments
2064	014210	000207			
2065					
2066					
2067					
2068					
2069					
2070					
2071	014212			TESTQ:	
2072	014212	012737	001026	026534	MOV #GOEOT, USEA ; SET FUNCTION
2073	014220	012737	000000	026536	MOV #-0, USEB ; SET WORD COUNT
2074	014226	012737	000000	026540	MOV #0, USEC ; SET BUS ADDRESS
2075	014234	004737	027010		JSR PC, EOTTST
2076	014240	013727	001240		MOV HOLD, (PC)+ ; PICK UP TIME PARAMETER
2077	014244	000000			.WORD 0 ; USE THIS WORD AS A TIME COUNTER
2078	014245				64\$:
2079	014246	005227	000000		66\$: INC #0 ; IF NO, COUNT 1 OF 65535 TICKS
2080	014252	001375			BNE 66\$; HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2081	014254	005337	014244		DEC 64\$; HAS THE TOTAL TIME ELAPSED?
2082	014260	001372			BNE 66\$; IF NO, GO WAIT A LITTLE LONGER
2083	014262				65\$:
2084	014262	013727	001240		MOV HOLD, (PC)+ ; PICK UP TIME PARAMETER
2085	014266	000000			.WORD 0 ; USE THIS WORD AS A TIME COUNTER
2086	014270				67\$:
2087	014270	005227	000000		69\$: INC #0 ; IF NO, COUNT 1 OF 65535 TICKS
2088	014274	001375			BNE 69\$; HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2089	014276	005337	014266		DEC 67\$; HAS THE TOTAL TIME ELAPSED?
2090	014302	001372			BNE 69\$; IF NO, GO WAIT A LITTLE LONGER
2091	014304				68\$:
2092	014304	013727	001240		MOV HOLD, (PC)+ ; PICK UP TIME PARAMETER
2093	014310	000000			.WORD 0 ; USE THIS WORD AS A TIME COUNTER
2094	014312				72\$:
2095	014312	005227	000000		INC #0 ; IF NO, COUNT 1 OF 65535 TICKS
2096	014316	001375			BNE 72\$; HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2097	014320	005337	014310		DEC 70\$; HAS THE TOTAL TIME ELAPSED?
2098	014324	001372			BNE 72\$; IF NO, GO WAIT A LITTLE LONGER
2099	014326				71\$:
2100	014326	052777	004000	165042	BIS #PWRCLR, @TRCR ; DEVICE MASTER RESET
2101	014334	032777	004000	165034	73\$: BIT #PWRCLR, @TRCR ; INSTRUCTION CLEAR ?
2102	014342	001374			BNE 73\$; IF NO, WAIT FOR IT TO CLEAR
2103	014344	004537	026544		JSR R5, TRYIT
2104	014350	001020			.WORD REWIND ; FUNCTION
2105	014352	000000			.WORD -0 ; WORD COUNT
2106	014354	000000			.WORD 0 ; BUS ADDRESS
2107	014356	052777	004000	165012	BIS #BIT11, @TRCR
2108					
2109	014364	013727	001240		MOV HOLD, (PC)+ ; PICK UP TIME PARAMETER
2110	014370	000000			.WORD 0 ; USE THIS WORD AS A TIME COUNTER
2111	014372				74\$:
2112	014372	005227	000000		76\$: INC #0 ; IF NO, COUNT 1 OF 65535 TICKS
2113	014376	001375			BNE 76\$; HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2114	014400	005337	014370		DEC 74\$; HAS THE TOTAL TIME ELAPSED?
2115	014404	001372			BNE 76\$; IF NO, GO WAIT A LITTLE LONGER
2116	014406				75\$:
2117	014406	022777	004020	164762	CMP #4020, @TRCR ; TEST COMMAND REGISTER
2118	014414	001412			BEG A\$20
2119	014416	012705	000020		MOV #20, R5

```

2120 014422 017704 164750          MOV    @TRCR,R4
2121 014426 013737 001376 001244    MOV    TRCR,REGIST
2122 014434 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
2123 014436 104403 005713          TYPEF  ,MRWOFI
2124
2125 014442 000021          AS20:  STAGAS=STAGAS+1
2126
2127 014442 022777 000001 164732    CMP    #1,@TRSR          ;TEST THE STATUS REGISTER
2128 014450 001411          BEQ    AS21
2129 014452 012705 000001          MOV    #1,R5
2130 014456 017704 164720          MOV    @TRSR,R4
2131 014462 013737 001402 001244    MOV    TRSR,REGIST
2132 014470 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
2133 014472 104404          TYPEL
2134
2135 014474 000022          AS21:  STAGAS=STAGAS+1
2136
2137 014474 005777 164712          TST    @TRBA          ;TEST BUFFER ADDRESS
2138 014500 001411          BEQ    AS22
2139 014502 012705 000000          MOV    #0,R5
2140 014506 017704 164700          MOV    @TRBA,R4
2141 014512 013737 001412 001244    MOV    TRBA,REGIST
2142 014520 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
2143 014522 104404          TYPEL
2144
2145 014524 000023          AS22:  STAGAS=STAGAS+1
2146
2147 014524 005777 164656          TST    @TRWC          ;TEST THE WORD COUNT
2148 014530 001411          BEQ    AS23
2149 014532 012705 000000          MOV    #0,R5
2150 014536 017704 164644          MOV    @TRWC,R4
2151 014542 013737 001406 001244    MOV    TRWC,REGIST
2152 014550 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
2153 014552 104404          TYPEL
2154
2155 014554 000024          AS23:  STAGAS=STAGAS+1
2156
2157 014554 032777 000002 164620    A=.    BIT    #BIT1,@TRSR
2158 014562 001374          BNE   A          ;WAIT FOR REWIND TO FINISH
2159 014564 032777 000040 164610    BIT    #BITS,@TRSR
2160 014572 001770          BEQ   A          ;WAIT FOR LOADPOINT TO SET
2161 014574 004737 013662          JSR   PC,TESTN
2162 014600 052777 004000 164570    BIS    #PWRCLR,@TRCR ;DEVICE MASTER RESET
2163 014606 032777 004000 164562    BIT    #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
2164 014614 001374          BNE   64$        ;IF NO, WAIT FOR IT TO CLEAR
2165 014616 000207          RTS    PC
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175

```

-----*

```

;THIS ROUTINE CHECKS THE DATA JUST READ BY THE
;MAG TAPE UNIT AND REPORTS ANY DERORS THAT EXIST
;THIS ROUTINE ALSO CLEARS THE INPUT AREA AFTER TESTING
;THE DATA

```

2176					
2177	014620	012701	034636		TESTS: MOV #INPUT,R1 ;SET INDEX
2178	014624	012702	030576		MOV #OUTPUT,R2 ;SET INDEX
2179	014630	013703	026536		MOV USEB,R3 ;GET THE WORD COUNT
2180		014634		A=.	
2181	014634	021112			CMP @R1,@R2 ;TEST INPUT
2182	014636	001410			SEQ AS24
2183	014640	011205			MOV @R2,R5
2184	014642	011104			MOV @R1,R4
2185	014644	010137	001244		MOV R1,REGIST
2186	014650	104001			ERROR 1 ;INCORRECT REGISTER MATCHUP
2187	014652	104402	005736		TYPE #ADERR
2188	014656	010200			MOV @R2,R0
2189					:----- CONVRT CALL SHOULD GO HERE
2190					
2191	014660	000025			AS24: STAGAS=STAGAS+1
2192	014660	022122			CMP (R1)+,(R2)+ ;ADVANCE INDEXES
2193	014662	005203			INC R3 ;INCREMENT WORD COUNT
2194	014664	001363			BNE A ;CONTINUE TESTING INPUT
2195					
2196	014666	012700	034636		MOV #INPUT,R0 ;SET INDEX
2197	014672	012701	177000		MOV #-1000,R1 ;SET COUNTER
2198	014676	000026			AS25: STAGAS=STAGAS+1
2199		014676		A=.	
2200	014676	005020			CLR (R0)+ ;CLEAR INPUT AREA
2201	014700	001376			BNE A ;CONTINUE CLEARING
2202	014702	000207			RTS PC
2203					
2204					
2205					
2206					
2207					
2208					
2209					
2210					
2211					
2212					:+-----
2213					: THIS ROUTINE CHECKS THE FUNCTION OF SPACING REVERSE OVER A
2214					: END OF FILE
2215	014704				
2216	014704	012737	000062	014730	RUNU: MOV #50..3\$
2217	014712	005737	014730		2\$: TST 3\$
2218	014716	001001			BNE 1\$
2219	014720	000207			RTS PC
2220	014722	005337	014730		1\$: DEC 3\$
2221	014726	000401			BR 4\$
2222	014730	000000			3\$: 0
2223	014732				4\$:
2224					
2225	014732				TESTU:
2226	014732	004537	026544		JSR R5,TRYIT
2227	014736	001410			.WORD SPACER ;FUNCTION
2228	014740	000000			.WORD -0 ;WORD COUNT
2229	014742	000000			.WORD 0 ;BUS ADDRESS
2230	014744	022777	003610	164424	CMP #3610,@TRCR ;TEST COMMAND REGISTER
2231	014752	001412			BEG AS26


```

2232 014754 012705 003610          MOV    #3610,R5
2233 014760 017704 164412          MOV    @TRCR,R4
2234 014764 013737 001376 001244    MOV    TRCR,REGIST
2235 014772 104001                    ERROR  1          ;INCORRECT REGISTER MATCHUP
2236 014774 104403 005761          TYPEF  ,MSREFFL
2237
2238 015000 000027          AS26:  STAGAS=STAGAS+1
2239
2240 015000 022777 002011 164374    CMP    #2011,@TRSR ;TEST THE STATUS REGISTER
2241 015006 001411          BEQ    AS27
2242 015010 012705 002011          MOV    #2011,R5
2243 015014 017704 164362          MOV    @TRSR,R4
2244 015020 013737 001402 001244    MOV    TRSR,REGIST
2245 015026 104001                    ERROR  1          ;INCORRECT REGISTER MATCHUP
2246 015030 104404          TYPEL
2247
2248 015032 000030          AS27:  STAGAS=STAGAS+1
2249
2250 015032 005777 164354          TST    @TRBA      ;TEST BUFFER ADDRESS
2251 015036 001411          BEQ    AS30
2252 015040 012705 000000          MOV    #0,R5
2253 015044 017704 164342          MOV    @TRBA,R4
2254 015050 013737 001412 001244    MOV    TRBA,REGIST
2255 015056 104001                    ERROR  1          ;INCORRECT REGISTER MATCHUP
2256 015060 104404          TYPEL
2257
2258 015062 000031          AS30:  STAGAS=STAGAS+1
2259
2260 015062 005777 164320          TST    @TRWC      ;TEST THE WORD COUNT
2261 015066 001411          BEQ    AS31
2262 015070 012705 000000          MOV    #0,R5
2263 015074 017704 164306          MOV    @TRWC,R4
2264 015100 013737 001406 001244    MOV    TRWC,REGIST
2265 015106 104001                    ERROR  1          ;INCORRECT REGISTER MATCHUP
2266 015110 104404          TYPEL
2267
2268 015112 000032          AS31:  STAGAS=STAGAS+1
2269
2270 015112 000207          RTS    PC
2271
2272
2273 ;-----
2274 ;THIS ROUTINE CHECKS THE FUNCTION OF READING A END OF FILE
2275
2275 015114          RUNV:
2276 015114 012737 000062 015140    MOV    #50.,3$
2277 015122 005737 015140    2$:   TST    3$
2278 015126 001001          BNE   1$
2279 015130 000207          RTS   PC
2280 015132 005337 015140    1$:   DEC    3$
2281 015136 000401          BR    4$
2282 015140 000000          3$:   0
2283 015142          4$:
2284
2285 015142          TESTV:
2286 015142 004537 026544    JSR   R5,TRYIT
2287 015146 000004          .WORD READ      ;:FUNCTION
  
```

```

2298 015150 000000 .WORD -0 ;WORD COUNT
2299 015152 034636 .WORD INPUT ;BUS ADDRESS
2290 015154 022777 002204 164214 CMP #2204,@TRCR ;TEST COMMAND REGISTER
2291 015162 001412 BEQ A$32
2292 015164 012705 002204 MOV #2204,R5
2293 015170 017704 164202 MOV @TRCR,R4
2294 015174 013737 001376 001244 MOV TRCR,REGIST
2295 015202 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
2296 015204 104403 006023 TYPEF ,MREFFL
2297
2298 015210 000033 A$32: $TAGAS=$TAGAS+1
2299
2300 015210 022777 002011 164164 CMP #2011,@TRSR ;TEST THE STATUS REGISTER
2301 015216 001410 BEQ A$33
2302 015220 012705 002011 MOV #2011,R5
2303 015224 017704 164152 MOV @TRSR,R4
2304 015230 013737 001402 001244 MOV TRSR,REGIST
2305 015236 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
2306
2307 015240 000034 A$33: $TAGAS=$TAGAS+1
2308
2309 015240 022777 034636 164144 CMP #INPUT,@TRBA ;TEST BUFFER ADDRESS
2310 015246 001410 BEQ A$34
2311 015250 012705 034636 MOV #INPUT,R5
2312 015254 017704 164132 MOV @TRBA,R4
2313 015260 013737 001412 001244 MOV TRBA,REGIST
2314 015266 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
2315
2316 015270 000035 A$34: $TAGAS=$TAGAS+1
2317
2318 015270 005777 164112 TST @TRWC ;TEST THE WORD COUNT
2319 015274 001410 BEQ A$35
2320 015276 012705 000000 MOV #0,R5
2321 015302 017704 164100 MOV @TRWC,R4
2322 015306 013737 001406 001244 MOV TRWC,REGIST
2323 015314 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
2324 015316 000036 A$35: $TAGAS=$TAGAS+1
2325
2326 015316 000207 RTS PC
2327
2328 -----*
2329 ;THIS ROUTINE TESTS THE FUNCTION OF THE TIME OUT WHILE WRITING
2330
2331 015320 TESTW:
2332 015320 004737 020710 JSR PC,TESTAH
2333 015324 042777 000001 164050 BIC #BIT0,@TRSR
2334 015332 005077 164050 CLR @TRWC ;SET WORD COUNT TO MAX
2335 015336 012777 030576 164046 MOV #OUTPUT,@TRBA ;SET BUS ADDRESS AT START
2336 015344 012777 000402 164024 MOV #WRITE,@TRCR ;SET COMMAND REGISTER
2337 015352 052777 000001 164016 BIS #BIT0,@TRCR ;SET THE GO BIT
2338 015360 005037 007434 CLR TEMP ;CLEAR THE TIMER
2339 015364 A=.
2340 015364 005077 164016 CLR @TRWC ;RESET THE WORD COUNT
2341 015370 012777 030576 164014 MOV #OUTPUT,@TRBA ;RESET THE BUS ADDRESS
2342 015376 005237 007434 INC TEMP ;INCREMENT TIME
2343 015402 001370 BNE A ;DELAY MORE
  
```

2344		015404			A=.			
2345	015404	005077	163776			CLR	@TRWC	; RESET THE WORD COUNT
2346	015410	012777	030576	163774		MOV	#OUTPUT, @TRBA	; RESET THE BUS ADDRESS
2347	015416	005237	007434			INC	TEMP	; INCREMENT TIME
2348	015422	001370				BNE	A	; DELAY MORE
2349		015424			A=.			
2350	015424	005077	163756			CLR	@TRWC	; RESET THE WORD COUNT
2351	015430	012777	030576	163754		MOV	#OUTPUT, @TRBA	; RESET THE BUS ADDRESS
2352	015436	005237	007434			INC	TEMP	; INCREMENT TIME
2353	015442	001370				BNE	A	; DELAY MORE
2354	015444	022777	102602	163724		CMP	#102602, @TRCR	; TEST COMMAND REGISTER
2355	015452	001412				BEQ	A\$36	
2356	015454	012705	102602			MOV	#102602, R5	
2357	015460	017704	163712			MOV	@TRCR, R4	
2358	015464	013737	001376	001244		MOV	TRCR, REGIST	
2359	015472	104001				ERROR	1	; INCORRECT REGISTER MATCHUP
2360	015474	104403	006047			TYPEF	, MWTOFL	
2361								
2362	015500	000037			A\$36:		\$TAGAS=\$TAGAS+1	
2363								
2364	015500	017701	163676			MOV	@TRSR, R1	; GET THE STATUS
2365	015504	042701	170000			BIC	#170000, R1	; CLEAR BAD BITS
2366	015510	022701	002401			CMP	#2401, R1	; TEST THE STATUS REGISTER
2367	015514	001411				BEQ	A\$37	
2368	015516	012705	002401			MOV	#2401, R5	
2369	015522	017704	163654			MOV	@TRSR, R4	
2370	015526	013737	001402	001244		MOV	TRSR, REGIST	
2371	015534	104001				ERROR	1	; INCORRECT REGISTER MATCHUP
2372	015536	104404				TYPEL		
2373								
2374	015540	000040			A\$37:		\$TAGAS=\$TAGAS+1	
2375								
2376	015540	022777	030576	163644		CMP	#OUTPUT, @TRBA	; TEST BUFFER ADDRESS
2377	015546	001411				BEQ	A\$40	
2378	015550	012705	030576			MOV	#OUTPUT, R5	
2379	015554	017704	163632			MOV	@TRBA, R4	
2380	015560	013737	001412	001244		MOV	TRBA, REGIST	
2381	015566	104001				ERROR	1	; INCORRECT REGISTER MATCHUP
2382	015570	104404				TYPEL		
2383								
2384	015572	000041			A\$40:		\$TAGAS=\$TAGAS+1	
2385								
2386	015572	005777	163610			TST	@TRWC	; TEST THE WORD COUNT
2387	015576	001411				BEQ	A\$41	
2388	015600	012705	000000			MOV	#0, R5	
2389	015604	017704	163576			MOV	@TRWC, R4	
2390	015610	013737	001406	001244		MOV	TRWC, REGIST	
2391	015616	104001				ERROR	1	; INCORRECT REGISTER MATCHUP
2392	015620	104404				TYPEL		
2393								
2394	015622	000042			A\$41:		\$TAGAS=\$TAGAS+1	
2395								
2396	015622	052777	004000	163546		BIS	#PWRCLR, @TRCR	; DEVICE MASTER RESET
2397	015630	032777	004000	163540	64\$:	BIT	#PWRCLR, @TRCR	; INSTRUCTION CLEAR ?
2398	015636	001374				BNE	64\$; IF NO, WAIT FOR IT TO CLEAR
2399	015640	004737	013762			JSR	PC, TESTO	

			RIS	PC		
2400	015644	000207				
2401						
2402						
2403						
2404						
2405	015646					
2406	015646	004737	020710			
2407	015652	004737	025556			
2408	015656	004737	025646			
2409	015662	004737	025646			
2410	015666	0 2777	000001	163506		
2411	015674	005077	163506			
2412	015700	012777	034636	163504		
2413	015706	012777	001410	163462		
2414	015714	052777	000001	163454		
2415	015722	004737	011522			
2416						
2417	015726	022777	102050	163442		
2418	015734	001412				
2419	015736	012705	102050			
2420	015742	017704	163430			
2421	015746	013737	001376	001244		
2422	015754	104001				
2423	015756	104403	006071			
2424						
2425	015762	000043				
2426						
2427	015762	022777	002400	163412		
2428	015770	001411				
2429	015772	012705	002400			
2430	015776	017704	163400			
2431	016002	013737	001402	001244		
2432	016010	104001				
2433	016012	104404				
2434						
2435	016014	000044				
2436						
2437	016014	022777	034636	163370		
2438	016022	001411				
2439	016024	012705	034636			
2440	016030	017704	163356			
2441	016034	013737	001412	001244		
2442	016042	104001				
2443	016044	104404				
2444						
2445	016046	000045				
2446						
2447	016046	005777	163334			
2448	016052	001411				
2449	016054	012705	000000			
2450	016060	017704	163322			
2451	016064	013737	001406	001244		
2452	016072	104001				
2453	016074	104404				
2454						
2455	016076	000046				

-----*
 ; THIS ROUTINE CHECKS THE FUNCTION OF THE TIME OUT WHILE READING

TESTX:
 JSR PC, TESTAH
 JSR PC, TESTCA
 JSR PC, TESTFO
 JSR PC, TESTFO
 BIC #BIT0, @TRSR
 CLR @TRWC ; SET WORD COUNT TO MAX
 MOV #INPUT, @TRBA ; SET BUS ADDRESS AT START
 MOV #SPACER, @TRCR ; SET COMMAND REGISTER
 BIS #BIT0, @TRCR ; SET THE GO BIT
 JSR PC, TESTA

CMP #102050, @TRCR ; TEST COMMAND REGISTER
 BEQ AS42
 MOV #102050, R5
 MOV @TRCR, R4
 MOV TRCR, REGIST
 ERROR 1 ; INCORRECT REGISTER MATCHUP
 TYPEF , MRTOFL

AS42: STAGAS=STAGAS+1
 CMP #2400, @TRSR ; TEST THE STATUS REGISTER
 BEQ AS43
 MOV #2400, R5
 MOV @TRSR, R4
 MOV TRSR, REGIST
 ERROR 1 ; INCORRECT REGISTER MATCHUP
 TYPEL

AS43: STAGAS=STAGAS+1
 CMP #INPUT, @TRBA ; TEST BUFFER ADDRESS
 BEQ AS44
 MOV #INPUT, R5
 MOV @TRBA, R4
 MOV TRBA, REGIST
 ERROR 1 ; INCORRECT REGISTER MATCHUP
 TYPEL

AS44: STAGAS=STAGAS+1
 TST @TRWC ; TEST THE WORD COUNT
 BEQ AS45
 MOV #0, R5
 MOV @TRWC, R4
 MOV TRWC, REGIST
 ERROR 1 ; INCORRECT REGISTER MATCHUP
 TYPEL

AS45: STAGAS=STAGAS+1

```

2456
2457 016076 052777 004000 163272      BIS      #PWRCLR,@TRCR ;DEVICE MASTER RESET
2458 016104 032777 004000 163264 64$: BIT      #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
2459 016112 001374          BNE      64$      ;IF NO, WAIT FOR IT TO CLEAR
2460 016114 000207          RTS      PC
2461
2462
2463 -----*
2464 ;THIS ROUTINE FORCES A NXM AND BGL DEROR WHILE WRITING
2465
2466 016116          TESTY:
2467 016116 004737 020710      JSR      PC,TESTAH
2468 016122 004537 026544      JSR      R5,TRYIT
2469 016126 000402          .WORD   WRITE      ;FUNCTION
2470 016130 000000          .WORD   -0         ;WORD COUNT
2471 016132 177777          .WORD   -1         ;BUS ADDRESS
2472 016134 013727 001240      MOV      HOLD,(PC)+ ;PICK UP TIME PARAMETER
2473 016140 000000 64$: .WORD   0         ;USE THIS WORD AS A TIME COUNTER
2474 016142          66$:
2475 016142 005227 000000      INC      #0         ;IF NO,COUNT 1 OF 65535 TICKS
2476 016146 001375          BNE      66$      ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2477 016150 005337 016140      DEC      64$      ;HAS THE TOTAL TIME ELAPSED?
2478 016154 001372          BNE      66$      ;IF NO,GO WAIT A LITTLE LONGER
2479 016156          65$:
2480 016156 022777 112602 163212      CMP      #112602,@TRCR ;TEST COMMAND REGISTER
2481 016164 001412          BEQ      AS46
2482 016166 012705 112602      MOV      #112602,R5
2483 016172 017704 163200      MOV      @TRCR,R4
2484 016176 013737 001376 001244      MOV      TRCR,REGIST
2485 016204 104001          ERROR   1         ;INCORRECT REGISTER MATCHUP
2486 016206 104403 006112          TYPEF   ,MWNXFL
2487
2488 016212 000047          AS46: $TAGAS=$TAGAS+1
2489
2490 016212 022777 006101 163162      CMP      #6101,@TRSR ;TEST THE STATUS REGISTER
2491 016220 001411          BEQ      AS47
2492 016222 012705 006101      MOV      #6101,R5
2493 016226 017704 163150      MOV      @TRSR,R4
2494 016232 013737 001402 001244      MOV      TRSR,REGIST
2495 016240 104001          ERROR   1         ;INCORRECT REGISTER MATCHUP
2496 016242 104404          TYPEL
2497
2498 016244 000050          AS47: $TAGAS=$TAGAS+1
2499
2500 016244 022777 000001 163140      CMP      #1,@TRBA   ;TEST BUFFER ADDRESS
2501 016252 001411          BEQ      AS50
2502 016254 012705 000001      MOV      #1,R5
2503 016260 017704 163126      MOV      @TRBA,R4
2504 016264 013737 001412 001244      MOV      TRBA,REGIST
2505 016272 104001          ERROR   1         ;INCORRECT REGISTER MATCHUP
2506 016274 104404          TYPEL
2507
2508 016276 000051          AS50: $TAGAS=$TAGAS+1
2509
2510 016276 022777 000001 163102      CMP      #1,@TRWC   ;TEST THE WORD COUNT
2511 016304 001411          BEQ      AS51
    
```

```

2512 016306 012705 000301      MOV      #1,R5
2513 016312 017704 163070      MOV      @TRWC,R4
2514 016316 013737 001406 001244      MOV      TRWC,REGIST
2515 016324 104001      ERROR    1      ;INCORRECT REGISTER MATCHUP
2516 016326 104404      TYPEL
2518 016330 000C52      ASS1:   STAGAS=STAGAS+1
2519 016330 0002C7      RTS     PC
-----*
;THIS ROUTINE FORCES A NXM DEROR USING BIT 12
TESTZ:
2525 016332 004737 020710      JSR     PC,TESTAH
2526 016332 004737 020710      JSR     PC,TESTAH
2527 016336 012777 000402 163032      MOV     @WRITE,@TRCR ;SET COMMAND
2528 016344 052777 010000 163024      BIS     #BIT12,@TRCR
2529 016352 042777 000001 163022      BIC     #BIT0,@TRSR
2530 016360 005077 163022      CLR     @TRWC ;CLEAR THE WORD COUNT
2531 016364 005077 163022      CLR     @TRBA ;CLEAR BUS ADDRESS
2532 016370 005277 163002      INC     @TRCR ;SET GO BIT
2534 016374 013727 001240      MOV     HOLD,(PC)+ ;PICK UP TIME PARAMETER
2535 016400 000000      64$:   .WORD 0 ;USE THIS WORD AS A TIME COUNTER
2536 016402 005227 000000      66$:   INC     #0 ;IF NO,COUNT 1 OF 65535 TICKS
2537 016406 001375      BNE     66$ ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2538 016410 005337 016400      DEC     64$ ;HAS THE TOTAL TIME ELAPSED?
2539 016414 001372      BNE     66$ ;IF NO,GO WAIT A LITTLE LONGER
2541 016416 022777 112602 162752      65$:   CMP     #112602,@TRCR ;TEST COMMAND REGISTER
2542 016416 022777 112602 162752      BEQ     ASS2
2543 016424 001412      MOV     #112602,R5
2544 016426 012705 112602      MOV     @TRCR,R4
2545 016432 017704 162740      MOV     TRCR,REGIST
2546 016436 013737 001376 001244      MOV     TRCR,REGIST
2547 016444 104001      ERROR    1      ;INCORRECT REGISTER MATCHUP
2548 016446 104403 006147      TYPEF   ,MBNXL
2550 016452 000053      ASS2:   STAGAS=STAGAS+1
2551 016452 022777 006101 162722      CMP     #6101,@TRSR ;TEST THE STATUS REGISTER
2552 016460 001411      BEQ     ASS3
2553 016462 012705 006101      MOV     #6101,R5
2554 016466 017704 162710      MOV     @TRSR,R4
2555 016472 013737 001402 001244      MOV     TRSR,REGIST
2556 016500 104001      ERROR    1      ;INCORRECT REGISTER MATCHUP
2557 016502 104404      TYPEL
2559 016504 000054      ASS3:   STAGAS=STAGAS+1
2561 016504 022777 000002 162700      CMP     #2,@TRBA ;TEST BUFFER ADDRESS
2562 016512 001411      BEQ     ASS4
2563 016514 012705 000002      MOV     #2,R5
2564 016520 017704 162666      MOV     @TRBA,R4
2565 016524 013737 001412 001244      MOV     TRBA,REGIST
2566 016532 104001      ERROR    1      ;INCORRECT REGISTER MATCHUP

```

```

2568 016534 104404          TYPEL
2569
2570 016536 000055          ASS4:  STAGAS=STAGAS+1
2571
2572 016536 022777 000001 162642  CMP      #1,2TRWC          ;TEST THE WORD COUNT
2573 016544 001411          BEQ      ASS5
2574 016546 012705 000001          MOV      #1,R5
2575 016552 017704 162630          MOV      2TRWC,R4
2576 016556 013737 001406 001244  MOV      TRWC,REGIST
2577 016564 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2578 016566 104404          TYPEL
2579
2580 016570 000056          ASS5:  STAGAS=STAGAS+1
2581
2582 016570 052777 004000 162600  BIS      #PWRCLR,2TRCR ;DEVICE MASTER RESET
2583 016576 032777 004000 162572 645:    BIT      #PWRCLR,2TRCR ;INSTRUCTION CLEAR ?
2584 016604 001374          BNE     645          ;IF NO, WAIT FOR IT TO CLEAR
2585 016606 000207          RTS     PC
2586
2587 -----*
2588 ;THIS ROUTINE FORCES A NXM DEROR USING BIT 13
2589
2590
2591 016610          TESTAA:
2592 016610 004737 020710          JSR     PC,TESTAA
2593 016614 012777 000402 162554  MOV     #WRITE,2TRCR ;SET COMMAND
2594 016622 052777 020000 162546  BIS     #BIT13,2TRCR
2595 016630 042777 000001 162544  BIC     #BIT0,2TRSR
2596 016636 005077 162544          CLR     2TRWC          ;CLEAR THE WORD COUNT
2597 016642 005077 162544          CLR     2TRBA          ;CLEAR BUS ADDRESS
2598 016646 005277 162524          INC     2TRCR          ;SET GO BIT
2599
2600 016652 013727 001240          MOV     HOLD,(PC)+ ;PICK UP TIME PARAMETER
2601 016656 000000 645:    .WORD  0          ;USE THIS WORD AS A TIME COUNTER
2602 016660          665:
2603 016660 005227 000000          INC     #0          ;IF NO,COUNT 1 OF 65535 TICKS
2604 016664 001375          BNE     665          ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2605 016666 005337 016656          DEC     645          ;HAS THE TOTAL TIME ELAPSED?
2606 016674 001372          BNE     665          ;IF NO,GO WAIT A LITTLE LONGER
2607 016674 022777 122602 162474 655:    CMP     #122602,2TRCR ;TEST COMMAND REGISTER
2608 016702 001412          BEQ     ASS6
2609 016704 012705 122602          MOV     #122602,R5
2610 016710 017704 162462          MOV     2TRCR,R4
2611 016714 013737 001376 001244  MOV     TRCR,REGIST
2612 016722 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2613 016724 104403 006211          TYPEF  ,MNXBFL
2614
2615 016730 000057          ASS6:  STAGAS=STAGAS+1
2616
2617 016730 022777 006101 162444  CMP     #6101,2TRSR ;TEST THE STATUS REGISTER
2618 016736 001411          BEQ     ASS7
2619 016740 012705 006101          MOV     #6101,R5
2620 016744 017704 162432          MOV     2TRSR,R4
2621 016750 013737 001402 001244  MOV     TRSR,REGIST
2622 016756 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2623 016760 104404          TYPEL
    
```

```

2624
2625 016762 000060          AS57:  STAGAS=STAGAS+1
2626
2627 016762 022777 000002 162422      CMP      #2,@TRBA          ;TEST BUFFER ADDRESS
2628 016770 001411          BEQ      AS60
2629 016772 012705 000002          MOV      #2,R5
2630 016776 017704 162410          MOV      @TRBA,R4
2631 017002 013737 001412 001244      MOV      TRBA,REGIST
2632 017010 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2633 017012 104404          TYPEL
2634
2635 017014 000061          AS60:  STAGAS=STAGAS+1
2636
2637 017014 022777 000001 162364      CMP      #1,@TRWC          ;TEST THE WORD COUNT
2638 017022 001411          BEQ      AS61
2639 017024 012705 000001          MOV      #1,R5
2640 017030 017704 162352          MOV      @TRWC,R4
2641 017034 013737 001406 001244      MOV      TRWC,REGIST
2642 017042 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2643 017044 104404          TYPEL
2644
2645 017046 000062          AS61:  STAGAS=STAGAS+1
2646
2647 017046 052777 004000 162322      BIS      #PWRCLR,@TRCR    ;DEVICE MASTER RESET
2648 017054 032777 004000 162314      64$:    BIT      #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
2649 017062 001374          BNE     64$          ;IF NO, WAIT FOR IT TO CLEAR
2650 017064 000207          RTS      PC
2651
2652 -----*
2653 :THIS ROUTINE FORCES A NXM DEROR USING BIT 12-13
2654
2655 017066          TESTAB:
2656 017066 004737 020710          JSR      PC,TESTAB
2657 017072 012777 000402 162276      MOV      #WRITE,@TRCR    ;SET COMMAND
2658 017100 052777 030000 162270      BIS      #BIT12!BIT13,@TRCR ;SET THE EXTENDED ADDRESS BITS
2659 017106 042777 000001 162266      BIC      #BIT0,@TRSR
2660 017114 005077 162266          CLR      @TRWC          ;CLEAR THE WORD COUNT
2661 017120 012777 160000 162264      MOV      #160000,@TRBA   ;SET UP AN I/O PAGE REFERENCE
2662 017126 005277 162244          INC      @TRCR          ;SET GO BIT
2663 017132 013727 001240          MOV      HOLD.(PC)+     ;PICK UP TIME PARAMETER
2664 017136 000000          64$:    .WORD 0          ;USE THIS WORD AS A TIME COUNTER
2665 017140          66$:
2666 017140 005227 000000          INC      #0          ;IF NO,COUNT 1 OF 65535 TICKS
2667 017144 001375          BNE     66$          ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2668 017146 005337 017136          DEC      64$          ;HAS THE TOTAL TIME ELAPSED?
2669 017152 001372          BNE     66$          ;IF NO,GO WAIT A LITTLE LONGER
2670 017154          65$:
2671 017154 022777 132602 162214      CMP      #132602,@TRCR   ;TEST COMMAND REGISTER
2672 017162 001412          BEQ      AS62
2673 017164 012705 132602          MOV      #132602,R5
2674 017170 017704 162202          MOV      @TRCR,R4
2675 017174 013737 001376 001244      MOV      TRCR,REGIST
2676 017202 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2677 017204 104403 006261          TYPEL  MNXMFL
2678 017210 000063          AS62:  STAGAS=STAGAS+1
2679 017210 022777 006101 162164      CMP      #6101,@TRSR    ;TEST THE STATUS REGISTER
  
```



```

2680 017216 001411      BEQ      A$63
2681 017220 012705 006101    MOV      #6101,R5
2682 017224 017704 162152    MOV      JTRSR,R4
2683 017230 013737 001402 001244    MOV      TRSR,REGIST
2684 017236 104001      ERROR   1          ;INCORRECT REGISTER MATCHUP
2685 017240 104404      TYPEL
2686
2687 017242 000064      A$63:   STAGAS=STAGAS+1
2688
2689 017242 032777 000002 162142    BIT      #2,JTRBA          ;TEST BUFFER ADDRESS
2690 017250 001011      BNE      A$64
2691 017252 012705 000002      MOV      #2,R5
2692 017256 017704 162130    MOV      JTRBA,R4
2693 017262 013737 001412 001244    MOV      TRBA,REGIST
2694 017270 104001      ERROR   1          ;INCORRECT REGISTER MATCHUP
2695 017272 104404      TYPEL
2696
2697 017274 000065      A$64:   STAGAS=STAGAS+1
2698
2699 017274 022777 000001 162154    CMP      #1,JTRWC          ;TEST THE WORD COUNT
2700 017302 001411      BEQ      A$65
2701 017304 012705 000001      MOV      #1,R5
2702 017310 017704 162072    MOV      JTRWC,R4
2703 017314 013737 001406 001244    MOV      TRWC,REGIST
2704 017322 104001      ERROR   1          ;INCORRECT REGISTER MATCHUP
2705 017324 104404      TYPEL
2706
2707 017326 000066      A$65:   STAGAS=STAGAS+1
2708
2709 017326 052777 004000 162042    BIS      #PWRCLR,JTRCR    ;DEVICE MASTER RESET
2710 017334 032777 004000 162034 64$:    BIT      #PWRCLR,JTRCR    ;INSTRUCTION CLEAR ?
2711 017342 001374      BNE      64$           ;IF NO, WAIT FOR IT TO CLEAR
2712 017344 000207      RTS      PC
2713
2714 -----*
2715 ;THIS ROUTINE FORCS A READ COUNT DEROR WHILE READING A LONG RECORD
2716
2717 017346      TESTAC:
2718 017346 004737 020710    JSR      PC,TESTAH
2719 017352 004737 025606    JSR      PC,TESTCK
2720 017356 004737 025626    JSR      PC,TESTEK
2721 017362 004537 026544    JSR      R5,TRYIT
2722 017366 000004      .WORD   READ           ;FUNCTION
2723 017370 177701      .WORD   -77           ;WORD COUNT
2724 017372 034636      .WORD   INPUT         ;BUS DDDRESS
2725 017374 013727 001240    MOV      HOLD,(PC)+     ;PICK UP TIME PARAMETER
2726 017400 000000 64$:      .WORD   0             ;USE THIS WORD AS A TIME COUNTER
2727 017402 66$:
2728 017402 005227 000000      INC      #0           ;IF NO,COUNT 1 OF 65535 TICKS
2729 017406 001375      BNE      66$          ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2730 017410 005337 017400      DEC      64$         ;HAS THE TOTAL TIME ELAPSED?
2731 017414 001372      BNE      66$         ;IF NO,GO WAIT A LITTLE LONGER
2732 017416 65$:
2733
2734 017416 022777 102204 161752    CMP      #102204,JTRCR   ;TEST COMMAND REGISTER
2735 017424 001412      BEQ      A$66
    
```

```

2736 017426 012705 102204      MOV      #102204,R5
2737 017432 017704 161740      MOV      @TRCR,R4
2738 017436 013737 001376 001244      MOV      TRCR,REGIST
2739 017444 104001      ERROR   1          ;INCORRECT REGISTER MATCHUP
2740 017446 104403 006333      TYPEF   ,MRCFL
2741
2742 017452 000067      AS66:   STAGAS=STAGAS+1
2743
2744 017452 022777 003001 161722      CMP      #3001,@TRSR      ;TEST THE STATUS REGISTER
2745 017460 001411      BEQ     AS67
2746 017462 012705 003001      MOV      #3001,R5
2747 017466 017704 161710      MOV      @TRSR,R4
2748 017472 013737 001402 001244      MOV      TRSR,REGIST
2749 017500 104001      ERROR   1          ;INCORRECT REGISTER MATCHUP
2750 017502 104404      TYPEL
2751
2752 017504 000070      AS67:   STAGAS=STAGAS+1
2753
2754 017504 022777 035034 161700      CMP      #INPUT+176,@TRBA      ;TEST BUFFER ADDRESS
2755 017512 001411      BEQ     AS70
2756 017514 012705 035034      MOV      #INPUT+176,R5
2757 017520 017704 161666      MOV      @TRBA,R4
2758 017524 013737 001412 001244      MOV      TRBA,REGIST
2759 017532 104001      ERROR   1          ;INCORRECT REGISTER MATCHUP
2760 017534 104404      TYPEL
2761
2762 017536 000071      AS70:   STAGAS=STAGAS+1
2763
2764 017536 005777 161644      TST     @TRWC      ;TEST THE WORD COUNT
2765 017542 001411      BEQ     AS71
2766 017544 012705 000000      MOV      #0,R5
2767 017550 017704 161632      MOV      @TRWC,R4
2768 017554 013737 001406 001244      MOV      TRWC,REGIST
2769 017562 104001      ERROR   1          ;INCORRECT REGISTER MATCHUP
2770 017564 104404      TYPEL
2771
2772 017566 000072      AS71:   STAGAS=STAGAS+1
2773
2774 017566 052777 004000 161602      BIS     #PWRCLR,@TRCR      ;DEVICE MASTER RESET
2775 017574 032777 004000 161574 64$:   BIT     #PWRCLR,@TRCR      ;INSTRUCTION CLEAR ?
2776 017602 001374      BNE     64$          ;IF NO, WAIT FOR IT TO CLEAR
2777 017604 000207      RTS     PC
2778
2779 -----*
2780 ;THIS ROUTINE FORCES A READ COUNT DEROR WHILE READING A SHORT RECORD
2781
2782 TESTAD:
2783 017606 004737 020710      JSR     PC,TESTAH
2784 017612 004737 025606      JSR     PC,TESTCK
2785 017616 004737 025626      JSR     PC,TESTEK
2786 017622 004537 026544      JSR     R5,TRYIT
2787 017626 000004      .WORD  READ          ;FUNCTION
2788 017630 177677      .WORD  -101         ;WORD COUNT
2789 017632 034636      .WORD  INPUT        ;BUS ADDRESS
2790 017634 013727 001240      MOV     HOLD,(PC)+    ;PICK UP TIME PARAMETER
2791 017640 000000 64$:   .WORD  0           ;USE THIS WORD AS A TIME COUNTER
    
```

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 59
 DZTRA.MED 13-APR-76 00:00 SUB-TEST ROUTINES

```

2792 017642          66$:
2793 017642 005227 000000          INC      #0          ;IF NO,COUNT 1 OF 65535 TICKS
2794 017646 001375          BNE      66$         ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2795 017650 005337 017640          DEC      64$         ;HAS THE TOTAL TIME ELAPSED?
2796 017654 001372          BNE      66$         ;IF NO,GO WAIT A LITTLE LONGER
2797 017656
2798
2799 017656 022777 102204 161512          CMP      #102204,@TRCR ;TEST COMMAND REGISTER
2800 017664 001411          BEQ      A$72
2801 017666 012705 102204          MOV      #102204,R5
2802 017672 017704 161500          MOV      @TRCR,R4
2803 017676 013737 001376 001244          MOV      TRCR,REGIST
2804 017704 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2805 017706 104404          TYPEL
2806
2807 017710 000073          A$72:  STAGAS=STAGAS+1
2808
2809 017710 022777 003001 161464          CMP      #3001,@TRSR ;TEST THE STATUS REGISTER
2810 017716 001411          BEQ      A$73
2811 017720 012705 003001          MOV      #3001,R5
2812 017724 017704 161452          MOV      @TRSR,R4
2813 017730 013737 001402 001244          MOV      TRSR,REGIST
2814 017736 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2815 017740 104404          TYPEL
2816
2817 017742 000074          A$73:  STAGAS=STAGAS+1
2818
2819 017742 022777 035036 161442          CMP      #INPUT+200,@TRBA ;TEST BUFFER ADDRESS
2820 017750 001411          BEQ      A$74
2821 017752 012705 035036          MOV      #INPUT+200,R5
2822 017756 017704 161430          MOV      @TRBA,R4
2823 017762 013737 001412 001244          MOV      TRBA,REGIST
2824 017770 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2825 017772 104404          TYPEL
2826
2827 017774 000075          A$74:  STAGAS=STAGAS+1
2828
2829 017774 022777 177777 161404          CMP      #-1,@TRWC ;TEST THE WORD COUNT
2830 020002 001411          BEQ      A$75
2831 020004 012705 177777          MOV      #-1,R5
2832 020010 017704 161372          MOV      @TRWC,R4
2833 020014 013737 001406 001244          MOV      TRWC,REGIST
2834 020022 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2835 020024 104404          TYPEL
2836
2837 020026 000076          A$75:  STAGAS=STAGAS+1
2838
2839 020026 052777 004000 161342          BIS      #PWRCLR,@TRCR ;DEVICE MASTER RESET
2840 020034 032777 004000 161334 64$:  BIT      #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
2841 020042 001374          BNE      64$         ;IF NO, WAIT FOR IT TO CLEAR
2842 020044 000207          RTS      PC
2843
2844 -----*
2845 ;THIS ROUTINE CHECKS THE FUNCTION OF SPACING REVERSE WITH A
2846 ;SHORT WORD COUNT
2847

```

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 60
 DZTRA.HED 13-APR-76 00:00 SUB-TEST ROUTINES

```

2848 020046          TESTAE:
2849 020046 004737 020710      JSR      PC,TESTAH
2850 020052 004737 025606      JSR      PC,TESTCK
2851 020056 004537 026544      JSR      R5,TRYIT
2852 020062 001410          .WORD   SPACER          ;:FUNCTION
2853 020064 177701          .WORD   -77            ;:WORD COUNT
2854 020066 000000          .WORD   0              ;:BUS ADDRESS
2855
2855 020070 022777 003610 161300  CMP      #3610,@TRCR    ;:TEST COMMAND REGISTER
2857 020076 001411          BEQ      AS76
2858 020100 012705 003610      MOV      #3610,R5
2859 020104 017704 161266      MOV      @TRCR,R4
2860 020110 013737 001376 001244  MOV      TRCR,REGIST
2861 020116 104001          ERROR   1              ;:INCORRECT REGISTER MATCHUP
2862 020120 104404          TYPEL
2863
2864 020122 000077          AS76:  STAGAS=STAGAS+1
2865
2866 020122 022777 002001 161252  CMP      #2001,@TRSR    ;:TEST THE STATUS REGISTER
2867 020130 001411          BEQ      AS77
2868 020132 012705 002001      MOV      #2001,R5
2869 020136 017704 161240      MOV      @TRSR,R4
2870 020142 013737 001402 001244  MOV      TRSR,REGIST
2871 020150 104001          ERROR   1              ;:INCORRECT REGISTER MATCHUP
2872 020152 104404          TYPEL
2873
2874 020154 000100          AS77:  STAGAS=STAGAS+1
2875
2876 020154 005777 161232      TST      @TRBA          ;:TEST BUFFER ADDRESS
2877 020160 001411          BEQ      AS100
2878 020162 012705 000000      MOV      #0,R5
2879 020166 017704 161220      MOV      @TRBA,R4
2880 020172 013737 001412 001244  MOV      TRBA,REGIST
2881 020200 104001          ERROR   1              ;:INCORRECT REGISTER MATCHUP
2882 020202 104404          TYPEL
2883
2884 020204 000101          AS100: STAGAS=STAGAS+1
2885
2886 020204 022777 177701 161174  CMP      #177701,@TRWC ;:TEST THE WORD COUNT
2887 020212 001411          BEQ      AS101
2888 020214 012705 177701      MOV      #177701,R5
2889 020220 017704 161162      MOV      @TRWC,R4
2890 020224 013737 001406 001244  MOV      TRWC,REGIST
2891 020232 104001          ERROR   1              ;:INCORRECT REGISTER MATCHUP
2892 020234 104404          TYPEL
2893
2894 020236 000102          AS101: STAGAS=STAGAS+1
2895
2896 020236 000207          RTS      PC
2897
2898
2899 ;-----*
2900 ;THIS ROUTINE CHECKS THE FUNCTION OF SPACING WITH A LONG WORD COUNT
2901
2901 020240          TESTAF:
2902 020240 004737 020710      JSR      PC,TESTAH
2903 020244 004737 025606      JSR      PC,TESTCK

```

```

2904 020250 004537 026544 JSR R5,TRYIT ;;;
2905 020254 001410 .WORD SPACER ;FUNCTION
2906 020256 177677 .WORD -101 ;WORD COUNT
2907 020260 000000 .WORD 0 ;BUS ADDRESS
2908
2909 020262 022777 003610 161106 CMP #3610,@TRCR ;TEST COMMAND REGISTER
2910 020270 001411 BEQ A$102
2911 020272 012705 003610 MOV #3610,R5
2912 020276 017704 161074 MOV @TRCR,R4
2913 020302 013737 001376 001244 MOV TRCR,REGIST
2914 020310 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
2915 020312 104404 TYPEL
2916
2917 020314 000103 A$102: STAGAS=STAGAS+1
2918
2919 020314 022777 002001 161060 CMP #2001,@TRSR ;TEST THE STATUS REGISTER
2920 020322 001410 BEQ A$103
2921 020324 012705 002001 MOV #2001,R5
2922 020330 017704 161046 MOV @TRSR,R4
2923 020334 013737 001402 001244 MOV TRSR,REGIST
2924 020342 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
2925
2926 020344 000104 A$103: STAGAS=STAGAS+1
2927
2928 020344 005777 161042 TST @TRBA ;TEST BUFFER ADDRESS
2929 020350 001410 BEQ A$104
2930 020352 012705 000000 MOV #0,R5
2931 020356 017704 161030 MOV @TRBA,R4
2932 020362 013737 001412 001244 MOV TRBA,REGIST
2933 020370 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
2934
2935 020372 000105 A$104: STAGAS=STAGAS+1
2936
2937 020372 022777 177677 161006 CMP #177677,@TRWC ;TEST THE WORD COUNT
2938 020400 001410 BEQ A$105
2939 020402 012705 177677 MOV #177677,R5
2940 020406 017704 160774 MOV @TRWC,R4
2941 020412 013737 001406 001244 MOV TRWC,REGIST
2942 020420 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
2943
2944 020422 000106 A$105: STAGAS=STAGAS+1
2945
2946 020422 000207 RTS PC
2947
2948
2949 ;-----*
2949 ;THIS ROUTINE TESTS THE END OF TAPE SENCING FUNCTION
2950 ;BY GOING FAST FORWARD TO END OF TAPE
2951
2952 020424 TESTAG:
2953 020424 052777 004000 160744 645: BIS #PWRCLR,@TRCR ;DEVICE MASTER RESET
2954 020432 032777 004000 160736 BIT #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
2955 020440 001374 BNE 645 ;IF NO, WAIT FOR IT TO CLEAR
2956 020442 012777 001026 160726 MOV #GOEOT,@TRCR ;SET COMMAND REGISTER
2957 020450 042777 000001 160724 BIC #BIT0,@TRSR ;CLEAR INHIBIT
2958 020456 012777 177777 160726 MOV #-1,@TRBA ;SET BUFFER ADDRESS
2959 020464 012777 177777 160714 MOV #-1,@TRWC ;SET WORD COUNT
    
```

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 62
 DZTRA.HED 13-APR-76 00:00 SUB-TEST ROUTINES

```

2960 020472 005277 160700          INC      @TRCR          ;SET THE GO BIT
2961
2962          020476          A=.
2963 020476 032777 000200 160676      BIT      #BIT7,@TRSR
2964 020504 001774          BEQ      A          ;BRANCH IF CLEAR
2965
2966 020506 013727 001240          MOV      HOLD,(PC)+    ;PICK UP TIME PARAMETER
2967 020512 000000          .WORD   0          ;USE THIS WORD AS A TIME COUNTER
2968 020514
2969 020514 005227 000000          INC      #0          ;IF NO,COUNT 1 OF 65535 TICKS
2970 020520 001375          BNE     67$         ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
2971 020522 005337 020512          DEC     65$         ;HAS THE TOTAL TIME ELAPSED?
2972 020526 001372          BNE     67$         ;IF NO,GO WAIT A LITTLE LONGER
2973 020530
2974 020530 004737 011522          JSR     PC,TESTA
2975 020534 022777 003226 160634      CMP     #3226,@TRCR   ;TEST COMMAND REGISTER
2976 020542 001412          BEQ     AS106
2977 020544 012705 003226          MOV     #3226,R5
2978 020550 017704 160622          MOV     @TRCR,R4
2979 020554 013737 001376 001244      MOV     TRCR,REGIST
2980 020562 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2981 020564 104403 006351          TYPEF  ,METFL
2982
2983 020570 000107          AS106: STAGAS=STAGAS+1
2984
2985 020570 022777 002201 160604      CMP     #2201,@TRSR   ;TEST THE STATUS REGISTER
2986 020576 001411          BEQ     AS107
2987 020600 012705 002201          MOV     #2201,R5
2988 020604 017704 160572          MOV     @TRSR,R4
2989 020610 013737 001402 001244      MOV     TRSR,REGIST
2990 020616 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
2991 020620 104404          TYPEL
2992
2993 020622 000110          AS107: STAGAS=STAGAS+1
2994
2995 020622 022777 177777 160562      CMP     #-1,@TRBA     ;TEST BUFFER ADDRESS
2996 020630 001411          BEQ     AS110
2997 020632 012705 177777          MOV     #-1,R5
2998 020636 017704 160550          MOV     @TRBA,R4
2999 020642 013737 001412 001244      MOV     TRBA,REGIST
3000 020650 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
3001 020652 104404          TYPEL
3002
3003 020654 000111          AS110: STAGAS=STAGAS+1
3004
3005 020654 022777 177777 160524      CMP     #-1,@TRWC     ;TEST THE WORD COUNT
3006 020662 001411          BEQ     AS111
3007 020664 012705 177777          MOV     #-1,R5
3008 020670 017704 160512          MOV     @TRWC,R4
3009 020674 013737 001406 001244      MOV     TRWC,REGIST
3010 020702 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
3011 020704 104404          TYPEL
3012
3013 020706 000112          AS111: STAGAS=STAGAS+1
3014
3015 020706 000207          RTS     PC

```

```

3016
3017
3018
3019
3020
3021 020710
3022 020710 052777 004000 160460
3023 020716 032777 004000 160452
3024 020724 001374
3025 020726 032777 000040 160446
3026 020734 001402
3027 020736 004737 013762
3028 020742 000113
3029 020742 004737 025646
3030 020746 000207
3031
3032
3033
3034
3035 020750
3036 020750 012737 000024 020774
3037 020756 005737 020774
3038 020762 001001
3039 020764 000207
3040 020766 005337 020774
3041 020772 000401
3042 020774 000000
3043 020776
3044 020776 013727 001240
3045 021002 000000
3046 021004
3047 021004 005227 000000
3048 021010 001375
3049 021012 005337 021002
3050 021016 001372
3051 021020
3052 021020 000207
3053
3054
3055
3056
3057
3058
3059 021022
3060 021022 004737 012624
3061 021026 104402 006370
3062
3063
3064 021032 005237 007434
3065 021036 001003
3066 021040 012777 000207 160144
3067 021046
3068 021046 032777 002000 160326
3069 021054 001366
3070
3071

```

```

-----*
; THIS ROUTINE INSURES THE TAPE IS NOT AT LOAD POINT AND THERE
; IS GOOD DATA ON THE TAPE
TESTAH:
64$:  BIS      #PWRCLR, @TRCR  ; DEVICE MASTER RESET
      BIT      #PWRCLR, @TRCR  ; INSTRUCTION CLEAR ?
      BNE      64$             ; IF NO, WAIT FOR IT TO CLEAR
      BIT      #BITS, @TRSR
      BEQ      AS112
      JSR      PC, TESTO
AS112: $TAGAS=$TAGAS+1
      JSR      PC, TESTFO
      RTS      PC
-----*
; THIS ROUTINE DELAYS ABOUT 10 SECONDS
TESTAJ:
2$:  MOV      #20., 3$
      TST     3$
      BNE     1$
      RTS     PC
1$:  DEC     3$
      BR     4$
3$:  0
4$:
64$: MOV      HOLD, (PC)+      ; PICK UP TIME PARAMETER
      .WORD   0                ; USE THIS WORD AS A TIME COUNTER
66$: INC     #0                ; IF NO, COUNT 1 OF 65535 TICKS
      BNE     66$              ; HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
      DEC     64$              ; HAS THE TOTAL TIME ELAPSED?
      BNE     66$              ; IF NO, GO WAIT A LITTLE LONGER
65$: RTS     PC
-----*
; THIS ROUTINE TELLS THE OPERATOR TO RESET THE
; TAPE TRANSPORT AND TESTS THE REGISTERS AFTER HE DOES
TESTAK:
      JSR     PC, TESTI
      TYPE   .MRSTU
A=:  INC     TEMP
      BNE     64$
64$: MOV     #207, @TPDBR      ; RING THE BELL
      BIT     #BIT10, @TRSR
      BNE     A                ; BRANCH IF CLEAR

```

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 64
 DZTRA.HED 13-APR-76 00:00 SUB-TEST ROUTINES

```

3072 021056 013727 001240          MOV    HOLD,(PC)+    ;PICK UP TIME PARAMETER
3073 021062 000000          65$: .WORD    0      ;USE THIS WORD AS A TIME COUNTER
3074 021064                67$:
3075 021064 005227 000000          INC    #0            ;IF NO,COUNT 1 OF 65535 TICKS
3076 021070 001375                BNE    67$          ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3077 021072 005337 021062          DEC    65$         ;HAS THE TOTAL TIME ELAPSED?
3078 021076 001372                BNE    67$         ;IF NO,GO WAIT A LITTLE LONGER
3079 021100
3080 021100 032777 100000 160270    66$: BIT    #100000,@TRCR ;TEST COMMAND REGISTER
3081 021106 001012                BNE    AS113
3082 021110 012705 100026          MOV    #100026,R5
3083 021114 017704 160256          MOV    @TRCR,R4
3084 021120 013737 001376 001244    MOV    TRCR,REGIST
3085 021126 104001                ERROR   1           ;INCORRECT REGISTER MATCHUP
3086 021130 104403 006413          TYPEF  ,MMRSFL
3087
3088 021134 000114                AS113: $TAGAS=$TAGAS+1
3089
3090 021134 017701 160242          MOV    @TRSR,R1    ;GET STATUS
3091 021140 042701 000040          BIC    #BITS,R1   ;CLEAR BIT 5
3092
3093 021144 022701 000001          CMP    #1,R1      ;TEST THE STATUS
3094 021150 001411                BEQ    AS114
3095 021152 012705 000001          MOV    #1,R5
3096 021156 017704 160220          MOV    @TRSR,R4
3097 021162 013737 001402 001244    MOV    TRSR,REGIST
3098 021170 104001                ERROR   1           ;INCORRECT REGISTER MATCHUP
3099 021172 104404          TYPEL
3100
3101 021174 000115                AS114: $TAGAS=$TAGAS+1
3102
3103 021174 005777 160212          TST    @TRBA      ;TEST BUFFER ADDRESS
3104 021200 001411                BEQ    AS115
3105 021202 012705 000000          MOV    #0,R5
3106 021206 017704 160200          MOV    @TRBA,R4
3107 021212 013737 001412 001244    MOV    TRBA,REGIST
3108 021220 104001                ERROR   1           ;INCORRECT REGISTER MATCHUP
3109 021222 104404          TYPEL
3110
3111 021224 000116                AS115: $TAGAS=$TAGAS+1
3112
3113 021224 005777 160156          TST    @TRWC      ;TEST THE WORD COUNT
3114 021230 001411                BEQ    AS116
3115 021232 012705 000000          MOV    #0,R5
3116 021236 017704 160144          MOV    @TRWC,R4
3117 021242 013737 001406 001244    MOV    TRWC,REGIST
3118 021250 104001                ERROR   1           ;INCORRECT REGISTER MATCHUP
3119 021252 104404          TYPEL
3120
3121 021254 000117                AS116: $TAGAS=$TAGAS+1
3122
3123 021254 000207                RTS    PC
3124
3125 -----*
3126 ;THIS ROUTINE WRITES RECORDS PAST EOT
3127

```


3128	021256				TESTAL:			
3129	021256	032777	000200	160116	BIT	#BIT7,@TRSR		
3130	021264	001033			BNE	AS117		
3131	021266	012737	001026	026534	MOV	#GOEOT,USEA		;SET FUNCTION
3132	021274	012737	000001	026536	MOV	#--1,USEB		;SET WORD COUNT
3133	021302	012737	177777	026540	MOV	#-1,USEC		;SET BUS ADDRESS
3134	021310	004737	027010		JSR	PC,EOTTST		
3135								
3136		021314			A=.			
3137	021314	032777	000200	160060	BIT	#BIT7,@TRSR		
3138	021322	001774			BEQ	A		;BRANCH IS CLEAR
3139	021324	032777	000040	160044	BIT	#BITS,@TRCR		
3140	021332	001370			BNE	A		;BRANCH IF SET
3141	021334	032777	002000	160034	BIT	#BIT10,@TRCR		
3142	021342	001764			BEQ	A		;BRANCH IF CLEAR
3143	021344	032777	000200	160024	BIT	#BIT7,@TRCR		
3144	021352	001760			BEQ	A		;BRANCH IF CLEAR
3145								
3146	021354	000120			AS117:	\$TAGAS=\$TAGAS+1		
3147	021354	012737	000016	026534	MOV	#ERASE,USEA		;SET FUNCTION
3148	021362	012737	177000	026536	MOV	#-1000,USEB		;SET WORD COUNT
3149	021370	012737	030576	026540	MOV	#OUTPUT,USEC		;SET BUS ADDRESS
3150	021376	004737	027010		JSR	PC,EOTTST		
3151	021402	052777	004000	157766	BIS	#PWRCLR,@TRCR		;DEVICE MASTER RESET
3152	021410	032777	004000	157760	64\$:	BIT	#PWRCLR,@TRCR	;INSTRUCTION CLEAR ?
3153	021416	001374			BNE	64\$;IF NO, WAIT FOR IT TO CLEAR
3154								
3155	021420	012737	000012	021444	MOV	#10.,3\$		
3156	021426	005737	021444		2\$:	TST	3\$	
3157	021432	001001			BNE	1\$		
3158	021434	000207			RTS	PC		
3159	021436	005337	021444		1\$:	DEC	3\$	
3160	021442	000401			BR	4\$		
3161	021444	000000			3\$:	0		
3162	021446				4\$:			
3163	021446	012737	000402	026534	MOV	#WRITE,USEA		;SET FUNCTION
3164	021454	012737	177700	026536	MOV	#-100,USEB		;SET WORD COUNT
3165	021462	012737	030576	026540	MOV	#OUTPUT,USEC		;SET BUS ADDRESS
3166	021470	004737	027010		JSR	PC,EOTTST		
3167								
3168	021474	022777	002602	157674	CMP	#2602,@TRCR		;TEST COMMAND REGISTER
3169	021502	001412			BEQ	AS120		
3170	021504	012705	002602		MOV	#2602,R5		
3171	021510	017704	157662		MOV	@TRCR,R4		
3172	021514	013737	001376	001244	MOV	TRCR,REGIST		
3173	021522	104001			ERROR	1		;INCORRECT REGISTER MATCHUP
3174	021524	104403	006433		TYPEF	,MWPETFL		
3175								
3176	021530	000121			AS120:	\$TAGAS=\$TAGAS+1		
3177								
3178	021530	022777	002201	157644	CMP	#2201,@TRSR		;TEST THE STATUS REGISTER
3179	021536	001411			BEQ	AS121		
3180	021540	012705	002201		MOV	#2201,R5		
3181	021544	017704	157632		MOV	@TRSR,R4		
3182	021550	013737	001402	001244	MOV	TRSR,REGIST		
3183	021556	104001			ERROR	1		;INCORRECT REGISTER MATCHUP

```

3184 021560 104404          TYPEL
3185
3186 021562 000122          AS121: $TAGAS=$TAGAS+1
3187
3188
3189 021562 005777 157620      TST      @TRWC          ;TEST THE WORD COUNT
3190 021566 001411          BEQ      AS122
3191 021570 012705 000000      MOV      #0,R5
3192 021574 017704 157606      MOV      @TRWC,R4
3193 021600 013737 001406 001244  MOV      TRWC,REGIST
3194 021606 104001          ERROR    1          ;INCORRECT REGISTER MATCHUP
3195 021610 104404          TYPEL
3196
3197 021612 000123          AS122: $TAGAS=$TAGAS+1
3198
3199 021612 000207          RTS      PC
3200
3201 -----*
3202 ;:THIS ROUTINE TESTS THE FUNCTION OF SPACE REVERSE PAST EOT
3203
3204 021614          TESTAM:
3205 021614 004737 021256      JSR      PC,TESTAL
3206
3207 021620 012737 001410 026534  MOV      #SPACER,USER      ;SET FUNCTION
3208 021626 012737 000001 026536  MOV      #--1,USEB        ;SET WORD COUNT
3209 021634 012737 177777 026540  MOV      #-1,USEC        ;SET BUS ADDRESS
3210 021642 004737 027010      JSR      PC,EOTTST
3211
3212 021646 022777 003610 157522  CMP      #3610,@TRCR      ;TEST COMMAND REGISTER
3213 021654 001412          BEQ      AS123
3214 021656 012705 003610      MOV      #3610,R5
3215 021662 017704 157510      MOV      @TRCR,R4
3216 021666 013737 001376 001244  MOV      TRCR,REGIST
3217 021674 104001          ERROR    1          ;INCORRECT REGISTER MATCHUP
3218 021676 104403 006466      TYPEF    ,MSRETFL
3219
3220 021702 000124          AS123: $TAGAS=$TAGAS+1
3221
3222 021702 022777 002201 157472  CMP      #2201,@TRSR      ;TEST THE STATUS REGISTER
3223 021710 001411          BEQ      AS124
3224 021712 012705 002201      MOV      #2201,R5
3225 021716 017704 157460      MOV      @TRSR,R4
3226 021722 013737 001402 001244  MOV      TRSR,REGIST
3227 021730 104001          ERROR    1          ;INCORRECT REGISTER MATCHUP
3228 021732 104404          TYPEL
3229
3230 021734 000125          AS124: $TAGAS=$TAGAS+1
3231
3232 021734 022777 177777 157450  CMP      #-1,@TRBA        ;TEST BUFFER ADDRESS
3233 021742 001411          BEQ      AS125
3234 021744 012705 177777      MOV      #-1,R5
3235 021750 017704 157436      MOV      @TRBA,R4
3236 021754 013737 001412 001244  MOV      TRBA,REGIST
3237 021762 104001          ERROR    1          ;INCORRECT REGISTER MATCHUP
3238 021764 104404          TYPEL
3239

```

```

3240 021766 000126          AS125: STAGAS=STAGAS+1
3241 021766 022777 000001 157412      CMP      #1, @TRWC          ;TEST THE WORD COUNT
3242 021774 001411          BEQ      AS126
3243 021776 012705 000001      MOV      #1, R5
3244 022002 017704 157400      MOV      @TRWC, R4
3245 022006 013737 001406 001244      MOV      TRWC, REGIST
3246 022014 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
3247 022016 104404          TYPEL
3248
3249
3250 022020 000127          AS126: STAGAS=STAGAS+1
3251
3252 022023 000207          RTS      PC
3253
3254 -----*
3255 :THIS ROUTINE TESTS THE FUNCTION OF READING PAST EOT
3256 :TAPE MUST BE POSITIONED FIRST
3257
3258 TESTAN:
3259 022022 012737 000004 026534      MOV      #READ, USEA      ;SET FUNCTION
3260 022030 012737 177700 026536      MOV      #-100, USEB     ;SET WORD COUNT
3261 022036 012737 034636 026540      MOV      #INPUT, USEC    ;SET BUS ADDRESS
3262 022044 004737 027010      JSR      PC, EOT1ST
3263
3264 022050 022777 002204 157320      CMP      #2204, @TRCR     ;TEST COMMAND REGISTER
3265 022056 001412          BEQ      AS127
3266 022060 012705 002204      MOV      #2204, R5
3267 022064 017704 157306      MOV      @TRCR, R4
3268 022070 013737 001376 001244      MOV      TRCR, REGIST
3269 022076 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
3270 022100 104403 006530      TYPEF   ,MRPETFL
3271
3272 022104 000130          AS127: STAGAS=STAGAS+1
3273
3274 022104 022777 002201 157270      CMP      #2201, @TRSR     ;TEST THE STATUS REGISTER
3275 022112 001411          BEQ      AS130
3276 022114 012705 002201      MOV      #2201, R5
3277 022120 017704 157256      MOV      @TRSR, R4
3278 022124 013737 001402 001244      MOV      TRSR, REGIST
3279 022132 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
3280 022134 104404          TYPEL
3281
3282 022136 000131          AS130: STAGAS=STAGAS+1
3283
3284
3285 022136 005777 157244      TST      @TRWC          ;TEST THE WORD COUNT
3286 022142 001411          BEQ      AS131
3287 022144 012705 000000      MOV      #0, R5
3288 022150 017704 157232      MOV      @TRWC, R4
3289 022154 013737 001406 001244      MOV      TRWC, REGIST
3290 022162 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
3291 022164 104404          TYPEL
3292
3293 022166 000132          AS131: STAGAS=STAGAS+1
3294
3295 022166 000207          RTS      PC
    
```

```

3296
3297
3298
3299
3300 022170
3301 022170 004737 021256
3302
3303 022174 012737 001034 026534
3304 022202 012737 000001 026536
3305 022210 012737 177777 026540
3306 022216 004737 027010
3307
3308 022222 022777 003234 157146
3309 022230 001412
3310 022232 012705 003234
3311 022236 017704 157134
3312 022242 013737 001376 001244
3313 022250 104001
3314 022252 104403 006561
3315
3316 022256 000133
3317
3318 022256 022777 002211 157116
3319 022264 001411
3320 022266 012705 002211
3321 022272 017704 157104
3322 022276 013737 001402 001244
3323 022304 104001
3324 022306 104404
3325
3326 022310 000134
3327
3328 022310 022777 177777 157074
3329 022316 001411
3330 022320 012705 177777
3331 022324 017704 157062
3332 022330 013737 001412 001244
3333 022336 104001
3334 022340 104404
3335
3336 022342 000135
3337
3338 022342 022777 000001 157036
3339 022350 001411
3340 022352 012705 000001
3341 022356 017704 157024
3342 022362 013737 001406 001244
3343 022370 104001
3344 022372 104404
3345
3346 022374 000136
3347
3348 022374 000207
3349
3350
3351

```

-----*

```

; THIS ROUTINE WRITES A EOF PAST END OF TAPE
TESTAO:
      JSR      PC, TESTAL
      MOV      #WEOF, USEA          ; SET FUNCTION
      MOV      #-1, USEB          ; SET WORD COUNT
      MOV      #-1, USEC          ; SET BUS ADDRESS
      JSR      PC, EOTTST
      CMP      #3234, @TRCR        ; TEST COMMAND REGISTER
      BEQ      AS132
      MOV      #3234, R5
      MOV      @TRCR, R4
      MOV      TRCR, REGIST
      ERROR   1                    ; INCORRECT REGISTER MATCHUP
      TYPEL   ,MWEFETF
AS132: STAGAS=STAGAS+1
      CMP      #2211, @TRSR        ; TEST THE STATUS REGISTER
      BEQ      AS133
      MOV      #2211, R5
      MOV      @TRSR, R4
      MOV      TRSR, REGIST
      ERROR   1                    ; INCORRECT REGISTER MATCHUP
      TYPEL
AS133: STAGAS=STAGAS+1
      CMP      #-1, @TRBA          ; TEST BLFFER ADDRESS
      BEQ      AS134
      MOV      #-1, R5
      MOV      @TRBA, R4
      MOV      TRBA, REGIST
      ERROR   1                    ; INCORRECT REGISTER MATCHUP
      TYPEL
AS134: STAGAS=STAGAS+1
      CMP      #1, @TRWC          ; TEST THE WORD COUNT
      BEQ      AS135
      MOV      #1, R5
      MOV      @TRWC, R4
      MOV      TRWC, REGIST
      ERROR   1                    ; INCORRECT REGISTER MATCHUP
      TYPEL
AS135: STAGAS=STAGAS+1
      RTS      PC

```

-----*

```

; THIS ROUTINE TESTS THE FUNCTION OF SPACING REVERSE OVER

```

```

3352                                     :END OF FILE PAST EOT
3353
3354 022376                                TESTAP:
3355 022376 004737 022170                   JSR      PC,TESTA0
3356
3357 022402 012737 001410 026534           MOV      #SPACER,USEA           ;SET FUNCTION
3358 022410 012737 000001 026536           MOV      #-1,USEB             ;SET WORD COUNT
3359 022416 012737 177777 026540           MOV      #-1,USEC             ;SET BUS ADDRESS
3360 022424 004737 027010                   JSR      PC,EOTTST
3361
3362 022430 022777 003610 156740           CMP      #3610,@TRCR          ;TEST COMMAND REGISTER
3363 022436 001412                               BEQ      AS136
3364 022440 012705 003610                   MOV      #3610,R5
3365 022444 017704 156726                   MOV      @TRCR,R4
3366 022450 013737 001376 001244           MOV      TRCR,REGIST
3367 022456 104001                               ERROR   1                       ;INCORRECT REGISTER MATCHUP
3368 022460 104403 006632                   TYPEL   ,MREFETF
3369
3370 022464 000137                   AS136: STAGAS=STAGAS+1
3371
3372 022464 022777 002211 156710           CMP      #2211,@TRSR          ;TEST THE STATUS REGISTER
3373 022472 001411                               BEQ      AS137
3374 022474 012705 002211                   MOV      #2211,R5
3375 022500 017704 156676                   MOV      @TRSR,R4
3376 022504 013737 001402 001244           MOV      TRSR,REGIST
3377 022512 104001                               ERROR   1                       ;INCORRECT REGISTER MATCHUP
3378 022514 104404                               TYPEL
3379
3380 022516 000140                   AS137: STAGAS=STAGAS+1
3381
3382 022516 022777 177777 156666           CMP      #-1,@TRBA           ;TEST BUFFER ADDRESS
3383 022524 001411                               BEQ      AS140
3384 022526 012705 177777                   MOV      #-1,R5
3385 022532 017704 156654                   MOV      @TRBA,R4
3386 022536 013737 001412 001244           MOV      TRBA,REGIST
3387 022544 104001                               ERROR   1                       ;INCORRECT REGISTER MATCHUP
3388 022546 104404                               TYPEL
3389
3390 022550 000141                   AS140: STAGAS=STAGAS+1
3391
3392 022550 022777 000001 156630           CMP      #1,@TRWC           ;TEST THE WORD COUNT
3393 022556 001411                               BEQ      AS141
3394 022560 012705 000001                   MOV      #1,R5
3395 022564 017704 156616                   MOV      @TRWC,R4
3396 022570 013737 001406 001244           MOV      TRWC,REGIST
3397 022576 104001                               ERROR   1                       ;INCORRECT REGISTER MATCHUP
3398 022600 104404                               TYPEL
3399
3400 022602 000142                   AS141: STAGAS=STAGAS+1
3401
3402 022602 000207                               RTS      PC
3403
3404 ----->
3405 ;THIS ROUTINE TESTS THE FUNCTION OF READING A EOF PAST EOT
3406
3407 022604                                TESTAQ:

```

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 70
 DZTRA.HED 13-APR-76 00:00 SUB-TEST ROUTINES

```

3408 022604 004737 022376      JSR      PC,TESTAP
3409
3410 022610 012737 000004 026534      MOV      #READ,USEA      ;SET FUNCTION
3411 022616 012737 000001 026536      MOV      #-1,USEB      ;SET WORD COUNT
3412 022624 012737 177777 026540      MOV      #-1,USEC      ;SET BUS ADDRESS
3413 022632 004737 027010      JSR      PC,EOTTST
3414
3415 022636 022777 002204 156532      CMP      #2204,ATRCR    ;TEST COMMAND REGISTER
3416 022644 001412      BEQ      AS142
3417 022646 012705 002204      MOV      #2204,R5
3418 022652 017704 156520      MOV      ATRCR,R4
3419 022656 013737 001376 001244      MOV      TRCR,REGIST
3420 022664 104001      ERROR    1      ;INCORRECT REGISTER MATCHUP
3421 022666 104403 006632      TYPEF      ,MREFETF
3422
3423 022672 000143      AS142:  STAGAS=STAGAS+1
3424
3425 022672 022777 002211 156502      CMP      #2211,ATRSR    ;TEST THE STATUS REGISTER
3426 022700 001411      BEQ      AS143
3427 022702 012705 002211      MOV      #2211,R5
3428 022706 017704 156470      MOV      ATRSR,R4
3429 022712 013737 001402 001244      MOV      TRSR,REGIST
3430 022720 104001      ERROR    1      ;INCORRECT REGISTER MATCHUP
3431 022722 104404      TYPEL
3432
3433 022724 000144      AS143:  STAGAS=STAGAS+1
3434
3435 022724 022777 177777 156460      CMP      #-1,ATRBA      ;TEST BUFFER ADDRESS
3436 022732 001411      BEQ      AS144
3437 022734 012705 177777      MOV      #-1,R5
3438 022740 017704 156446      MOV      ATRBA,R4
3439 022744 013737 001412 001244      MOV      TRBA,REGIST
3440 022752 104001      ERROR    1      ;INCORRECT REGISTER MATCHUP
3441 022754 104404      TYPEL
3442
3443 022756 000145      AS144:  STAGAS=STAGAS+1
3444
3445 022756 022777 000001 156422      CMP      #1,ATRWC      ;TEST THE WORD COUNT
3446 022764 001411      BEQ      AS145
3447 022766 012705 000001      MOV      #1,R5
3448 022772 017704 156410      MOV      ATRWC,R4
3449 022776 013737 001406 001244      MOV      TRWC,REGIST
3450 023004 104001      ERROR    1      ;INCORRECT REGISTER MATCHUP
3451 023006 104404      TYPEL
3452
3453 023010 000146      AS145:  STAGAS=STAGAS+1
3454
3455 023010 000207      RTS      PC
3456
3457 -----*
3458 ;THIS ROUTINE TRIES TO WRITE ON TAPE WITHOUT A WRITE
3459 ;RING
3460
3461 023012      RUNAS:
3462 023012 012737 000012 023036      MOV      #10.,3S
3463 023020 005737 023036      TST      3S

```

3464	023024	001001			BNE	1\$	
3465	023026	000207			RTS	PC	
3466	023030	005337	023036		1\$: DEC	3\$	
3467	023034	000401			BR	4\$	
3468	023036	000000			3\$: 0		
3469	023040				4\$:		
3470							
3471	023040				TESTAS:		
3472	023040	052777	004000	156330	RIS	#PWRCLR, @TRCR	; DEVICE MASTER RESET
3473	023046	032777	004000	156322	64\$: BIT	#PWRCLR, @TRCR	; INSTRUCTION CLEAR ?
3474	023054	001374			BNE	64\$; IF NO, WAIT FOR IT TO CLEAR
3475	023056	004537	026544		JSR	R5, TRYIT	
3476	023062	000402			.WORD	WRITE	; FUNCTION
3477	023064	177700			.WORD	-100	; WORD COUNT
3478	023066	030576			.WORD	OUTPUT	; BUS ADDRESS
3479							
3480	023070	013727	001240		MOV	HOLD, (PC)+	; PICK UP TIME PARAMETER
3481	023074	000000			65\$: .WORD	0	; USE THIS WORD AS A TIME COUNTER
3482	023076				67\$:		
3483	023076	005227	000000		INC	#0	; IF NO, COUNT 1 OF 65535 TICKS
3484	023102	001375			BNE	67\$; HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3485	023104	005337	023074		DEC	65\$; HAS THE TOTAL TIME ELAPSED?
3486	023110	001372			BNE	67\$; IF NO, GO WAIT A LITTLE LONGER
3487	023112				66\$:		
3488							
3489	023112	022777	142602	156256	CMP	#142602, @TRCR	; TEST COMMAND REGISTER
3490	023120	001412			BEQ	AS146	
3491	023122	012705	142602		MOV	#142602, R5	
3492	023126	017704	156244		MOV	@TRCR, R4	
3493	023132	013737	001376	001244	MOV	TRCR, REGIST	
3494	023140	104001			ERROR	1	; INCORRECT REGISTER MATCHUP
3495	023142	104403	006702		TYPEF	, MWENFL	
3496							
3497	023146	000147			AS146: STAGAS=STAGAS+1		
3498							
3499	023146	017701	156230		MOV	@TRSR, R1	; GET STATUS REGISTER
3500	023152	042701	000040		BIC	#BITS, R1	; CLEAR UNWANTED BITS
3501							
3502	023156	022701	002004		CMP	#2004, R1	; TEST THE STATUS REGISTER
3503	023162	001411			BEQ	AS147	
3504	023164	012705	002004		MOV	#2004, R5	
3505	023170	017704	156206		MOV	@TRSR, R4	
3506	023174	013737	001402	001244	MOV	TRSR, REGIST	
3507	023202	104001			ERROR	1	; INCORRECT REGISTER MATCHUP
3508	023204	104404			TYPEL		
3509							
3510	023206	000150			AS147: STAGAS=STAGAS+1		
3511							
3512							
3513	023206	022777	030576	156176	CMP	#OUTPUT, @TRBA	; TEST BUFFER ADDRESS
3514	023214	001411			BEQ	AS150	
3515	023216	012705	030576		MOV	#OUTPUT, R5	
3516	023222	017704	156164		MOV	@TRBA, R4	
3517	023226	013737	001412	001244	MOV	TRBA, REGIST	
3518	023234	104001			ERROR	1	; INCORRECT REGISTER MATCHUP
3519	023236	104404			TYPEL		

```

3520
3521 023240 000151          AS150:  STAGAS=STAGAS+1
3522
3523 023240 022777 177700 156140      CMP      #177700, @TRWC      ;TEST THE WORD COUNT
3524 023246 001411          BEQ      AS151
3525 023350 012705 177700          MOV      #177700, R5
3526 023254 017704 156126          MOV      @TRWC, R4
3527 023260 013737 001406 001244      MOV      TRWC, REGIST
3528 023266 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
3529 023270 104404          TYPEL
3530
3531 023272 000152          AS151:  STAGAS=STAGAS+1
3532
3533 023272 052777 004000 156076      BIS      #PWRCLR, @TRCR    ;DEVICE MASTER RESET
3534 023300 032777 004000 156070 64$:  BIT      #PWRCLR, @TRCR    ;INSTRUCTION CLEAR ?
3535 023306 001374          BNE     64$              ;IF NO, WAIT FOR IT TO CLEAR
3536 023310 000207          RTS      PC
3537
3538 -----*
3539 ;THIS ROUTINE TRIES TO SPACE REVERSE AT LOAD POINT
3540
3541 023312          TESTAT:
3542 023312 004737 012624      JSR      PC, TESTI
3543 023316 004537 026544      JSR      R5, TRYIT
3544 023322 001410          .WORD   SPACER          ;:FUNCTION
3545 023324 177777          .WORD   -1              ;:WORD COUNT
3546 023326 000000          .WORD   0                ;:BUS ADDRESS
3547
3548
3549 023330 022777 143610 156040      CMP      #143610, @TRCR    ;TEST COMMAND REGISTER
3550 023336 001412          BEQ      AS152
3551 023340 012705 143610          MOV      #143610, R5
3552 023344 017704 156026          MOV      @TRCR, R4
3553 023350 013737 001376 001244      MOV      TRCR, REGIST
3554 023356 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
3555 023360 104403 006722          TYPEF   ,MSRLPFL
3556
3557 023364 000153          AS152:  STAGAS=STAGAS+1
3558
3559 023364 022777 002040 156010      CMP      #2040, @TRSR     ;TEST THE STATUS REGISTER
3560 023372 001411          BEQ      AS153
3561 023374 012705 002040          MOV      #2040, R5
3562 023400 017704 155776          MOV      @TRSR, R4
3563 023404 013737 001402 001244      MOV      TRSR, REGIST
3564 023412 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
3565 023414 104404          TYPEL
3566
3567 023416 000154          AS153:  STAGAS=STAGAS+1
3568
3569 023416 005777 155770          TST     @TRBA            ;TEST BUFFER ADDRESS
3570 023422 001411          BEQ      AS154
3571 023424 012705 000000          MOV      #0, R5
3572 023430 017704 155756          MOV      @TRBA, R4
3573 023434 013737 001412 001244      MOV      TRBA, REGIST
3574 023442 104001          ERROR   1                ;INCORRECT REGISTER MATCHUP
3575 023444 104404          TYPEL
    
```


DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 73
 DZTRA.HED 13-APR-76 00:00 SUB-TEST ROUTINES

```

3576
3577 023446 000155          AS154: $TAGAS=$TAGAS+1
3578
3579 023446 022777 177777 155732      CMP      #-1,@TRWC          ;TEST THE WORD COUNT
3580 023454 001411          BEQ      AS155
3581 023456 012705 177777          MOV      #-1,R5
3582 023462 017704 155720          MOV      @TRWC,R4
3583 023466 013737 001406 001244      MOV      TRWC,REGIST
3584 023474 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
3585 023476 104404          TYPEL
3586
3587 023500 000156          AS155: $TAGAS=$TAGAS+1
3588
3589 023500 052777 004000 155670      BIS      #PWRCLR,@TRCR    ;DEVICE MASTER RESET
3590 023506 032777 004000 155662      64$: BIT      #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
3591 023514 001374          BNE     64$        ;IF NO, WAIT FOR IT TO CLEAR
3592 023516 000207          RTS      PC
3593
3594
3595 -----*
3596 ;THIS ROUTINE TESTS THE BUS ADDRESS BITS BY USING A COUNT PATTERN
3597
3598
3599 023520 005001          TESTAU: CLR     R1          ;CLEAR COUNT
3600          A=.
3601 023522 010177 155664          MOV      R1,@TRBA        ;SET BUS ADDRESS
3602 023526 020177 155660          CMP      R1,@TRBA        ;TEST BUS ADDRESS
3603 023532 001407          BEQ     AS156
3604 023534 010105          MOV      R1,R5
3605 023536 017704 155650          MOV      @TRBA,R4
3606 023542 013737 001412 001244      MOV      TRBA,REGIST
3607 023550 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
3608
3609 023552 000157          AS156: $TAGAS=$TAGAS+1
3610 023552 005201          INC     R1          ;INCREMENT COUNT
3611 023554 001362          BNE     A          ;CONTINUE
3612 023556 000207          RTS      PC
3613
3614
3615 -----*
3616 ;THIS ROUTINE SPACES REVERSE OVER ID BLOCK
3617
3618
3619 023560 004737 013762          TESTAV: JSR     PC,TESTO
3620
3621 023564 004537 026544          JSR     R5,TRYIT
3622 023570 001410          .WORD  SPACER          ;:FUNCTION
3623 023572 177777          .WORD  -1              ;:WORD COUNT
3624 023574 177777          .WORD  -1              ;:BUS ADDRESS
3625
3626
3627 023576 022777 003610 155572      CMP      #3610,@TRCR    ;TEST COMMAND REGISTER
3628 023604 001412          BEQ     AS157
3629 023606 012705 003610          MOV      #3610,R5
3630 023612 017704 155560          MOV      @TRCR,R4
3631 023616 013737 001376 001244      MOV      TRCR,REGIST

```

```

3632 023624 104001          ERROR 1          ;INCORRECT REGISTER MATCHUP
3633 023626 104403 006761  TYPEF ,MSRIFL
3634
3635 023632 000160          AS157: STAGAS=STAGAS+1
3636
3637 023632 022777 002061 155542  CMP      #2061,@TRSR  ;TEST THE STATUS REGISTER
3638 023640 001411          BEQ      AS160
3639 023642 012705 002061  MOV      #2061,R5
3640 023646 017704 155530  MOV      @TRSR,R4
3641 023652 013737 001402 001244  MOV      TRSR,REGIST
3642 023660 104001          ERROR 1          ;INCORRECT REGISTER MATCHUP
3643 023662 104404          TYPEL
3644
3645 023664 000161          AS160: STAGAS=STAGAS+1
3646
3647 023664 022777 177777 155520  CMP      #-1,@TRBA  ;TEST BUFFER ADDRESS
3648 023672 001411          BEQ      AS161
3649 023674 012705 177777  MOV      #-1,R5
3650 023700 017704 155506  MOV      @TRBA,R4
3651 023704 013737 001412 001244  MOV      TRBA,REGIST
3652 023712 104001          ERROR 1          ;INCORRECT REGISTER MATCHUP
3653 023714 104404          TYPEL
3654
3655 023716 000162          AS161: STAGAS=STAGAS+1
3656
3657 023716 022777 177777 155462  CMP      #-1,@TRWC  ;TEST THE WORD COUNT
3658 023724 001411          BEQ      AS162
3659 023726 012705 177777  MOV      #-1,R5
3660 023732 017704 155450  MOV      @TRWC,R4
3661 023736 013737 001406 001244  MOV      TRWC,REGIST
3662 023744 104001          ERROR 1          ;INCORRECT REGISTER MATCHUP
3663 023746 104404          TYPEL
3664
3665 023750 000163          AS162: STAGAS=STAGAS+1
3666
3667 023750 052777 004000 155420  BIS      #PWRCLR,@TRCR ;DEVICE MASTER RESET
3668 023756 032777 004000 155412 645: BIT      #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
3669 023764 001374          BNE      645      ;IF NO, WAIT FOR IT TO CLEAR
3670 023766 000207          RTS      PC
3671
3672
3673 ;-----*
3674 ;THIS ROUTINE READS A ID BLOCK
3675
3676 TESTAW:
3677 JSR      PC,TESTI
3678 CLR      INPUT          ;CLEAR INPUT AREA
3679
3680 JSR      R5,TRYIT      ;;;
3681 .WORD   READ          ;FUNCTION
3682 .WORD   -1           ;WORD COUNT
3683 .WORD   INPUT        ;BUS ADDRESS
3684
3685 024012 022777 002204 155356  CMP      #2204,@TRCR  ;TEST COMMAND REGISTER
3686 024020 001412          BEQ      AS163
3687 024022 012705 002204  MOV      #2204,R5
    
```

```

3688 024026 017704 155344          MOV    @TRCR,R4
3689 024032 013737 001376 001244    MOV    TRCR,REGIST
3690 024040 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
3691 024042 104403 007016    TYPEL  ,MRIBFL
3692
3693 024046 000164          AS163: STAGAS=STAGAS+1
3694
3695 024046 022777 002021 155326    CMP    #2021,@TRSR ;TEST THE STATUS REGISTER
3696 024054 001411          BEQ    AS164
3697 024056 012705 002021          MOV    #2021,R5
3698 024062 017704 155314          MOV    @TRSR,R4
3699 024066 013737 001402 001244    MOV    TRSR,REGIST
3700 024074 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
3701 024076 104404          TYPEL
3702
3703 024100 000165          AS164: STAGAS=STAGAS+1
3704
3705 024100 022777 034636 155304    CMP    #INPUT,@TRBA ;TEST BUFFER ADDRESS
3706 024106 001411          BEQ    AS165
3707 024110 012705 034636          MOV    #INPUT,R5
3708 024114 017704 155272          MOV    @TRBA,R4
3709 024120 013737 001412 001244    MOV    TRBA,REGIST
3710 024126 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
3711 024130 104404          TYPEL
3712
3713 024132 000166          AS165: STAGAS=STAGAS+1
3714
3715 024132 022777 177777 155246    CMP    #-1,@TRWC   ;TEST THE WORD COUNT
3716 024140 001411          BEQ    AS166
3717 024142 012705 177777          MOV    #-1,R5
3718 024146 017704 155234          MOV    @TRWC,R4
3719 024152 013737 001406 001244    MOV    TRWC,REGIST
3720 024160 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
3721 024162 104404          TYPEL
3722
3723 024164 000167          AS166: STAGAS=STAGAS+1
3724
3725 024164 005737 034636          TST    INPUT      ;TEST INPUT AREA
3726 024170 001411          BEQ    AS167
3727 024172 012705 000000          MOV    #0,R5
3728 024176 013704 034636          MOV    INPUT,R4
3729 024202 012737 034636 001244    MOV    #INPUT,REGIST
3730 024210 104001          ERROR  1          ;INCORRECT REGISTER MATCHUP
3731 024212 104404          TYPEL
3732 024214 000170          AS167: STAGAS=STAGAS+1
3733 024214 052777 004000 155154    BIS    #PWRCLR,@TRCR ;DEVICE MASTER RESET
3734 024222 032777 004000 155146 64$: BIT    #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
3735 024230 001374          BNE    64$        ;IF NO, WAIT FOR IT TO CLEAR
3736
3737 024232 000207          RTS    PC
3738
3739
3740 -----*
3741 ;THIS ROUTINE WRITES EVEN PARITY
3742
3743 TESTAX:
3744          JSR    PC,TESTAH
    
```

```

3744
3745 024240 004537 026544      JSR    R5,TRYIT      ;:
3746 024244 000402              .WORD  WRITE        ;FUNCTION
3747 024246 177700              .WORD  -100         ;WORD COUNT
3748 024250 030572              .WORD  BADOUT       ;BUS ADDRESS
3749 024252 013727 001240      MOV    HOLD,(PC)+   ;PICK UP TIME PARAMETER
3750 024256 000000      64$: .WORD  0        ;USE THIS WORD AS A TIME COUNTER
3751 024260 66$:
3752 024260 005227 000000      INC    #0           ;IF NO,COUNT 1 OF 65535 TICKS
3753 024264 001375              BNE    66$         ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3754 024266 005337 024256      DEC    64$        ;HAS THE TOTAL TIME ELAPSED?
3755 024272 001372              BNE    66$        ;IF NO.GO WAIT A LITTLE LONGER
3756 024274 65$:
3757
3758 024274 022777 102602 155074  CMP    #102602,@TRCR ;TEST COMMAND REGISTER
3759 024302 001412              BEQ    AS170
3760 024304 012705 102602      MOV    #102602,R5
3761 024310 017704 155062      MOV    @TRCR,R4
3762 024314 013737 001376 001244  MOV    TRCR,REGIST
3763 024322 104001              ERROR  1          ;INCORRECT REGISTER MATCHUP
3764 024324 104403 007037      TYPEF ,MPARFL
3765
3766 024330 000171      AS170: $TAGAS=$TAGAS+1
3767
3768 024330 022777 162001 155044  CMP    #162001,@TRSR ;TEST THE STATUS REGISTER
3769 024336 001411              BEQ    AS171
3770 024340 012705 162001      MOV    #162001,R5
3771 024344 017704 155032      MOV    @TRSR,R4
3772 024350 013737 001402 001244  MOV    TRSR,REGIST
3773 024356 104001              ERROR  1          ;INCORRECT REGISTER MATCHUP
3774 024360 104404      TYPEL
3775
3776 024362 000172      AS171: $TAGAS=$TAGAS+1
3777
3778 024362 013701 026536      MOV    USEB,R1     ;GET WORD COUNT
3779 024366 005101              COM    R1          ;FIX IT
3780 024370 005201              INC    R1          ;INCREMENT IT
3781 024372 006301              ASL   R1           ;DOUBLE IT
3782 024374 062701 030572      ADD    #BADOUT,R1 ;FINISH IT
3783
3784 024400 020177 155006      CMP    R1,@TRBA   ;TEST BUFFER ADDRESS
3785 024404 001410              BEQ    AS172
3786 024406 010105              MOV    R1,R5
3787 024410 017704 154776      MOV    @TRBA,R4
3788 024414 013737 001412 001244  MOV    TRBA,REGIST
3789 024422 104001              ERROR  1          ;INCORRECT REGISTER MATCHUP
3790 024424 104404      TYPEL
3791
3792 024426 000173      AS172: $TAGAS=$TAGAS+1
3793
3794 024426 005777 154754      TST   @TRWC       ;TEST THE WORD COUNT
3795 024432 001411              BEQ    AS173
3796 024434 012705 000000      MOV    #0,R5
3797 024440 017704 154742      MOV    @TRWC,R4
3798 024444 013737 001406 001244  MOV    TRWC,REGIST
3799 024452 104001              ERROR  1          ;INCORRECT REGISTER MATCHUP

```

```

3800 024454 104404          TYPEL
3801
3802 024456 000174          AS173: $TAGAS=$TAGAS+1
3803
3804 024456 052777 004000 154712  BIS      #PWRCLR,@TRCR ;DEVICE MASTER RESET
3805 024464 032777 004000 154704 64$: BIT      #PWRCLR,@TRCR ;INSTRUCTION CLEAR?
3806 024472 001374          BNE      64$      ;IF NO, WAIT FOR IT TO CLEAR
3807 024474 000207          RTS      PC
3808
3809
3810          -----*
3810          ;THIS ROUTINE SPACES REVERSE OVER EVEN PARITY
3811
3812 024476          TESTAY:
3813 024476 004737 024234          JSR      PC,TESTAX
3814
3815 024502 004537 026544          JSR      R5,TRYIT
3816 024506 001410          .WORD   SPACER      ;FUNCTION
3817 024510 177700          .WORD   -100        ;WORD COUNT
3818 024512 034636          .WORD   INPUT       ;BUS ADDRESS
3819 024514 013727 001240          MOV      HOLD,(PC)+ ;PICK UP TIME PARAMETER
3820 024520 000000          64$: .WORD   0        ;USE THIS WORD AS A TIME COUNTER
3821 024522
3822 024522 005227 000000          INC      #0          ;IF NO,COUNT 1 OF 65535 TICKS
3823 024526 001375          BNE      66$          ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3824 024530 005337 024520          DEC      64$          ;HAS THE TOTAL TIME ELAPSED?
3825 024534 001372          BNE      65$          ;IF NO,GO WAIT A LITTLE LONGER
3826 024536          65$:
3827
3828
3829 024536 022777 103610 154632          CMP      #103610,@TRC. ;TEST COMMAND REGISTER
3830 024544 001412          BEQ
3831 024546 012705 103610          MOV      #103610,R5
3832 024552 017704 154620          MOV      @TRCR,R4
3833 024556 013737 001376 001244          MOV      TRCR,REGIST
3834 024564 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
3835 024566 104403 007056          TYPEF   ,MEPSRFL
3836
3837 024572 000175          AS174: $TAGAS=$TAGAS+1
3838
3839 024572 022777 122001 154602          CMP      #122001,@TRSR ;TEST THE STATUS REGISTER
3840 024600 001411          BEQ      AS175
3841 024602 012705 122001          MOV      #122001,R5
3842 024606 017704 154570          MOV      @TRSR,R4
3843 024612 013737 001402 001244          MOV      TRSR,REGIST
3844 024620 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
3845 024622 104404          TYPEL
3846
3847 024624 000176          AS175: $TAGAS=$TAGAS+1
3848
3849 024624 022777 034636 154560          CMP      #INPUT,@TRBA ;TEST BUFFER ADDRESS
3850 024632 001411          BEQ      AS176
3851 024634 012705 034636          MOV      #INPUT,R5
3852 024640 017704 154546          MOV      @TRBA,R4
3853 024644 013737 001412 001244          MOV      TRBA,REGIST
3854 024652 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
3855 024654 104404          TYPEL

```

```

3856
3857 024656 000177          AS176: $TAGAS=$TAGAS+1
3858
3859 024656 022777 177700 154522      CMP      #177700,@TRWC ;TEST THE WORD COUNT
3860 024664 001411          BEQ      AS177
3861 024666 012705 177700          MOV      #177700,R5
3862 024672 017704 154510          MOV      @TRWC,R4
3863 024676 013737 001406 001244      MOV      TRWC,REGIST
3864 024704 104001          ERROR   1 ;INCORRECT REGISTER MATCHUP
3865 024706 104404          TYPEL
3866
3867 024710 000200          AS177: $TAGAS=$TAGAS+1
3868
3869 024710 052777 004000 154460      BIS      #PWRCLR,@TRCR ;DEVICE MASTER RESET
3870 024716 032777 004000 154452      B4$:    BIT      #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
3871 024724 001374          BNE     B4$ ;IF NO, WAIT FOR IT TO CLEAR
3872 024726 000207          RTS     PC
3873
3874 ;-----*
3875 ;THIS ROUTINE READS EVEN PARITY
3876
3877 024730          TESTAZ:
3878 024730 004737 024476          JSR     PC,TESTAY
3879
3880 024734 004537 026544          JSR     R5,TRYIT ;;;
3881 024740 000004          .WORD  READ ;FUNCTION
3882 024742 177700          .WORD  -100 ;WORD COUNT
3883 024744 034636          .WORD  INPUT ;BUS ADDRESS
3884 024746 013727 001240      MOV     HOLD,(PC)+ ;PICK UP TIME PARAMETER
3885 024752 000000      B4$:    .WORD  0 ;USE THIS WORD AS A TIME COUNTER
3886 024754          B6$:
3887 024754 005227 000000          INC     #0 ;IF NO,COUNT 1 OF 65535 TICKS
3888 024760 001375          BNE     B6$ ;HAS THE TIME EXPIRED? IF NO, LOOP AGAIN
3889 024762 005337 024752          DEC     B4$ ;HAS THE TOTAL TIME ELAPSED?
3890 024766 001372          BNE     B6$ ;IF NO,GO WAIT A LITTLE LONGER
3891 024770          B5$:
3892
3893
3894 024770 022777 102204 154400      CMP      #102204,@TRCR ;TEST COMMAND REGISTER
3895 024776 001412          BEQ     AS200
3896 025000 012705 102204          MOV     #102204,R5
3897 025004 017704 154366          MOV     @TRCR,R4
3898 025010 013737 001376 001244      MOV     TRCR,REGIST
3899 025016 104001          ERROR   1 ;INCORRECT REGISTER MATCHUP
3900 025020 104403 007113          TYPEF  ,MREPFL
3901
3902 025024 000201          AS200: $TAGAS=$TAGAS+1
3903
3904 025024 022777 163001 154350      CMP      #163001,@TRSR ;TEST THE STATUS REGISTER
3905 025032 001411          BEQ     AS201
3906 025034 012705 163001          MOV     #163001,R5
3907 025040 017704 154336          MOV     @TRSR,R4
3908 025044 013737 001402 001244      MOV     TRSR,REGIST
3909 025052 104001          ERROR   1 ;INCORRECT REGISTER MATCHUP
3910 025054 104404          TYPEL
3911

```

```

3912 025056 000202          AS201: $TAGAS=$TAGAS+1
3913
3914 025056 022777 034640 154326      CMP      #INPUT+2, @TRBA ;TEST BUFFER ADDRESS
3915 025064 001410          BEQ      AS202
3916 025066 010105          MOV      R1, R5
3917 025070 017704 154316          MOV      @TRBA, R4
3918 025074 013737 001412 001244      MOV      TRBA, REGIST
3919 025102 104001          ERROR   1 ;INCORRECT REGISTER MATCHUP
3920 025104 104404          TYPEL
3921
3922 025106 000203          AS202: $TAGAS=$TAGAS+1
3923
3924 025106 022777 177701 154272      CMP      #177701, @TRWC ;TEST THE WORD COUNT
3925 025114 001411          BEQ      AS203
3926 025116 012705 177701          MOV      #177701, R5
3927 025122 017704 154260          MOV      @TRWC, R4
3928 025126 013737 001406 001244      MOV      TRWC, REGIST
3929 025134 104001          ERROR   1 ;INCORRECT REGISTER MATCHUP
3930 025136 104404          TYPEL
3931
3932 025140 000204          AS203: $TAGAS=$TAGAS+1
3933
3934 025140 000207          RTS      PC
3935
3936 ;-----*
3937 ;THIS ROUTINE TESTS THE BITS IN THE WORD COUNT USING A COUNT PATTERN
3938
3939 025142 005001          TESTBA: CLR      R1 ;CLEAR COUNT
3940 025144 025144          A=.
3941 025144 010177 154236          MOV      R1, @TRWC ;SET WORD COUNT
3942 025150 020177 154232          CMP      R1, @TRWC ;TEST WORD COUNT
3943 025154 001411          BEQ      AS204
3944 025156 010105          MOV      R1, R5
3945 025160 017704 154222          MOV      @TRWC, R4
3946 025164 013737 001406 001244      MOV      TRWC, REGIST
3947 025172 104001          ERROR   1 ;INCORRECT REGISTER MATCHUP
3948 025174 104403 007137          TYPEF  ,MWCFL
3949
3950 025200 000205          AS204: $TAGAS=$TAGAS+1
3951 025200 005201          INC      R1 ;INCREMENT COUNT
3952 025202 001360          BNE      A ;CONTINUE
3953 025204 000207          RTS      PC
3954
3955 ;-----*
3956
3957 ;THIS ROUTINE TESTS THE STATUS AND COMMAND REGISTERS
3958 ;AFTER VALID WRITE FUNCTIONS
3959
3960
3961
3962
3963 025206 022777 002602 154162      TESTCX: CMP      #2602, @TRCR ;TEST COMMAND REGISTER
3964 025214 001422          BEQ      AS205
3965 025216 032777 000200 154156      BIT      #200, @TRSR ;EOT??
3966 025224 001404          BEQ      1$ ;NO
3967 025226 022626          CMP      (SP)+, (SP)+
    
```

```

3978 025230 022626          CMP      (SP)+,(SP)+
3979 025232 022626          CMP      (SP)+,(SP)+
3980 025234 000207          RTS      PC
3981 025236          IS:
3982 025236 012705 002602          MOV      #2602,R5
3983 025242 017704 154130          MOV      @TRCR,R4
3984 025246 013737 001376 001244      MOV      TRCR,REGIST
3985 025254 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
3986 025256 104403 004721          TYPEF   ,MWRFL
3987 025262 000206          AS205: STAGAS=STAGAS+1
3988 025262 022777 002001 154112          CMP      #2001,@TRSR          ;TEST THE STATUS REGISTER
3989 025270 001422          BEQ     AS206
3990 025272 032777 000200 154102          BIT      #200,@TRSR          ;AT EOT??
3991 025300 001404          BEQ     IS              ;NO
3992 025302 022626          CMP      (SP)+,(SP)+
3993 025304 022626          CMP      (SP)+,(SP)+
3994 025306 022626          CMP      (SP)+,(SP)+
3995 025310 000207          RTS      PC
3996 025312          IS:
3997 025312 012705 002001          MOV      #2001,R5
3998 025316 017704 154060          MOV      @TRSR,R4
3999 025322 013737 001402 001244      MOV      TRSR,REGIST
4000 025330 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
4001 025332 104403 004721          TYPEF   ,MWRFL
4002 025336 000207          AS206: STAGAS=STAGAS+1
4003 025336 005777 154044          TST     @TRWC              ;TEST WORD COUNT
4004 025342 001412          BEQ     AS207
4005 025344 012705 000000          MOV     #0,R5
4006 025350 017704 154032          MOV     @TRWC,R4
4007 025354 013737 001406 001244      MOV     TRWC,REGIST
4008 025362 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
4009 025364 104403 004721          TYPEF   ,MWRFL
4010 025370 000210          AS207: STAGAS=STAGAS+1
4011 025370 000207          RTS      PC
4012 ----->
4013 ;THIS ROUTINE TESTS THE COMMAND AND STATUS AFTER VALID
4014 ;READ FUNCTIONS
4015 TESTDX: CMP      #2204,@TRCR          ;TEST COMMAND REGISTER
4016          BEQ     AS210
4017          BIT      #200,@TRSR          ;AT EOT??
4018          BEQ     IS              ;NO
4019          CMP      (SP)+,(SP)+
4020          TST     (SP)+
4021          RTS      PC
4022          IS:
4023 025420 012705 002204          MOV      #2204,R5
4024 025424 017704 153746          MOV      @TRCR,R4
4025 025430 013737 001375 001244      MOV      TRCR,REGIST
4026 025436 104001          ERROR   1          ;INCORRECT REGISTER MATCHUP
    
```



```

4024 025440 104403 004732          TYPEF ,MRDFL
4025
4026 025444 000211          AS210: STAGAS=STAGAS+1
4027
4028 025444 022777 002001 153730      CMP      #2001,2TRSR      ;TEST THE STATUS REGISTER
4029 025452 001421          BEQ      AS211
4030 025454 032777 000200 153720      BIT      #200,2TRSR      ;AT EOT??
4031 025462 001403          BEQ      IS              ;NO
4032 025464 022626          CMP      (SP)+,(SP)+
4033 025466 005726          TST      (SP)+
4034 025470 000207          RTS      PC
4035 025472
4036 025472 012705 002001          IS:   MOV      #2001,R5
4037 025476 017704 153700          MOV      2TRSR,R4
4038 025502 013737 001402 001244      MOV      TRSR,REGIST
4039 025510 104001          ERROR   1              ;INCORRECT REGISTER MATCHUP
4040 025512 104403 004732          TYPEF ,MRDFL
4041
4042 025516 000212          AS211: STAGAS=STAGAS+1
4043
4044 025516 005777 153664          TST      2TRWC          ;TEST WORD COUNT
4045 025522 001412          BEQ      AS212
4046 025524 012705 000000          MOV      #0,R5
4047 025530 017704 153652          MOV      2TRWC,R4
4048 025534 013737 001406 001244      MOV      TRWC,REGIST
4049 025542 104001          ERROR   1              ;INCORRECT REGISTER MATCHUP
4050 025544 104403 004732          TYPEF ,MRDFL
4051
4052 025550 000213          AS212: STAGAS=STAGAS+1
4053
4054 025550 004737 014620          JSR      PC,TESTS
4055 025554 000207          RTS      PC
4056
4057
4058 025556          TESTCA:
4059 025556 004737 030210          JSR      PC,ENAINIT
4060 025562 004537 026544          JSR      R5,TRYIT
4061 025566 000402          .WORD   WRITE          ;FUNCTION
4062 025570 177777          .WORD   -1             ;WORD COUNT
4063 025572 030576          .WORD   OUTPUT         ;BUS ADDRESS
4064 025574 004737 030250          JSR      PC,CHKINT
4065 025600 004737 025206          JSR      PC,TESTCX
4066 025604 000207          RTS      PC
4067
4068 025606          TESTCK:
4069 025606 004537 026544          JSR      R5,TRYIT
4070 025612 000402          .WORD   WRITE          ;FUNCTION
4071 025614 177700          .WORD   -100          ;WORD COUNT
4072 025616 030576          .WORD   OUTPUT         ;BUS ADDRESS
4073 025620 004737 025206          JSR      PC,TESTCX
4074 025624 000207          RTS      PC
4075 025626          TESTEK:
4076 025626 004537 026544          JSR      R5,TRYIT
4077 025632 001410          .WORD   SPACER        ;FUNCTION
4078 025634 177700          .WORD   -100          ;WORD COUNT
4079 025636 000000          .WORD   0             ;BUS ADDRESS

```

```

4080 025640 004737 025666          JSR    PC,TESTEX
4091 025644 000207                    RTS    PC
4082
4093
4084
4085 025646          TESTFO:
4086 025646 004537 026544          JSR    R5,TRYIT
4087 025652 000016          .WORD  ERASE          ;FUNCTION
4089 025654 177400          .WORD  -400          ;WORD COUNT
4099 025656 030576          .WORD  OUTPUT        ;BUS ADDRESS
4090 025660 004737 026014          JSR    PC,TESTFX
4091 025664 000207                    RTS    PC
4092
4093
4094
4095
4096
4097
4098 025666 022777 003610 153502 TESTEX: CMP    #3610,@TRCR ;TEST COMMAND REGISTER
4099 025674 001412          BEQ    A$213
4100 025676 012705 003610          MOV    #3610,R5
4101 025702 017704 153470          MOV    @TRCR,R4
4102 025706 013737 001376 001244          MOV    TRCR,REGIST
4103 025714 104001          ERROR  !          ;INCORRECT REGISTER MATCHUP
4104 025716 104403 004742          TYPEF  ,MSRVFL
4105
4106 025722 000214          A$213: STAGAS=STAGAS+1
4107
4108 025722 022777 002001 153452          CMP    #2001,@TRSR ;TEST THE STATUS REGISTER
4109 025730 001412          BEQ    A$214
4110 025732 012705 002001          MOV    #2001,R5
4111 025736 017704 153440          MOV    @TRSR,R4
4112 025742 013737 001402 001244          MOV    TRSR,REGIST
4113 025750 104001          ERROR  !          ;INCORRECT REGISTER MATCHUP
4114 025752 104403 004742          TYPEF  ,MSRVFL
4115
4116 025756 000215          A$214: STAGAS=STAGAS+1
4117
4118 025756 023777 026536 153422          CMP    USEB,@TRWC ;TEST WORD COUNT
4119 025764 001412          BEQ    A$215
4120 025766 013705 026536          MOV    USEB,R5
4121 025772 017704 153410          MOV    @TRWC,R4
4122 025776 013737 001406 001244          MOV    TRWC,REGIST
4123 026004 104001          ERROR  !          ;INCORRECT REGISTER MATCHUP
4124 026006 104403 004742          TYPEF  ,MSRVFL
4125
4126 026012 000216          A$215: STAGAS=STAGAS+1
4127
4128 026012 000207                    RTS    PC
4129
4130
4131
4132
4133
4134
4135

```

-----*

```

;THIS ROUTINE TESTS THE COMMAND AND STATUS AFTER VALID
;SPACE REVERSE FUNCTIONS

```

-----*

```

;THIS ROUTINE TESTS THE COMMAND AND STATUS REGISTERS AFTER
;AN ERASE FUNCTION

```

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 83
 DZTRA.MED 13-APR-76 00:00 SUB-TEST ROUTINES

```

4136 026014 022777 002216 153354 TESTFX: CMP #2216, @TRCR ;TEST COMMAND REGISTER
4137 026022 001412 BEQ AS216
4138 026024 012705 002216 MOV #2216, R5
4139 026030 017704 153342 MOV @TRCR, R4
4140 026034 013737 001376 001244 MOV TRCR, REGIST
4141 026042 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
4142 026044 104403 004763 TYPEF ,MERFL
4143
4144 026050 000217 AS216: STAGAS=STAGAS+1
4145
4146 026050 022777 002001 153324 CMP #2001, @TRSR ;TEST THE STATUS REGISTER
4147 026056 001412 BEQ AS217
4148 026060 012705 002001 MOV #2001, R5
4149 026064 017704 153312 MOV @TRSR, R4
4150 026070 013737 001402 001244 MOV TRSR, REGIST
4151 026076 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
4152 026100 104403 004763 TYPEF ,MERFL
4153
4154 026104 000220 AS217: STAGAS=STAGAS+1
4155
4156 026104 013701 026536 MOV USEB, R1 ;GET WORD COUNT
4157 026110 005101 COM R1 ;FIX IT
4158 026112 005201 INC R1
4159 026114 006301 ASL R1 ;DOUBLE IT
4160 026116 062701 030575 ADD #OUTPUT, R1 ;COMPUTE BUS ADDRESS
4161
4162 026122 020177 153264 CMP R1, @TRBA ;TEST BUFFER ADDRESS
4163 026126 001411 BEQ AS220
4164 026130 010105 MOV R1, R5
4165 026132 017704 153254 MOV @TRBA, R4
4166 026136 013737 001412 001244 MOV TRBA, REGIST
4167 026144 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
4168 026146 104403 004763 TYPEF ,MERFL
4169
4170 026152 000221 AS220: STAGAS=STAGAS+1
4171
4172 026152 005777 153230 TST @TRWC ;TEST THE WORD COUNT
4173 026156 001412 BEQ AS221
4174 026160 012705 000000 MOV #0, R5
4175 026164 017704 153216 MOV @TRWC, R4
4176 026170 013737 001406 001244 MOV TRWC, REGIST
4177 026176 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
4178 026200 104403 004763 TYPEF ,MERFL
4179
4180 026204 000222 AS221: STAGAS=STAGAS+1
4181
4182 026204 000207 RTS PC
4183
4184 -----*
4185 ;THIS ROUTINE WRITES, SPACES REVERSE, AND READS
4186
4187 026206 012704 000017 TESTXX: MOV #15, R4 ;LOAD THE TOTAL SUBTEST PASS COUNT
4188 026212 012703 026354 MOV #SIZTAB, R3 ;POINT TO TABLE OF BLOCK SIZES
4189 026216 012737 000001 001264 MOV #1, WRTFLG ;SET FLAG FOR RETURN
4190 026224 012337 026304 1$: MOV (R3)+, R5
4191 026230 012702 000003 MOV #3, R2 ;KEEP TRACK OF THE NUMBER OF FUNCTIONS TO DO

```

```

4192 026234 012705 026420      MOV      #BUFTAB,R5      ;POINT TO TABLE OF BUFFERS
4193 026240 012701 026412      MOV      #FUNCTB,R1     ;POINT TO TABLE OF FUNCTIONS
4194 026244 012700 026426      MOV      #SUBTST,R0     ;POINT TO TABLE OF FOLLOWUP TESTS
4195 026250 004737 030210      JSR      PC,ENAINI      ;
4196 026254 012137 026302      2$:     MOV      (R1)+,3$     ;LOAD A FUNCTION IN SUBROUTINE INTERFACE
4197 026260 012537 026306      MOV      (R5)+,5$     ;LOAD THE NAME OF THE BUFFER
4198 026264 010146                MOV      R1,-(SP)      ;SAVE REGISTERS.
4199 026266 010246                MOV      R2,-(SP)
4200 026270 010346                MOV      R3,-(SP)
4201 026272 010446                MOV      R4,-(SP)
4202 026274 010546                MOV      R5,-(SP)
4203 026276 004537 026544      JSR      R5,TRYIT      ;GO EXECUTE THE FUNCTION
4204 026302 000000      3$:     .WORD      0
4205 026304 000000      4$:     .WORD      0
4206 026306 000000      5$:     .WORD      0
4207 026310 004730                JSR      PC,2(R0)+     ;DO THE APPROPRIATE FOLLOWUP TEST
4208 026312 012605                MOV      (SP)+,R5     ;RESTORE THE REGISTERS
4209 026314 012604                MOV      (SP)+,R4
4210 026316 012603                MOV      (SP)+,R3
4211 026320 012602                MOV      (SP)+,R2
4212 026322 012601                MOV      (SP)+,R1
4213 026324 005302                DEC      R2           ;REDUCE THE COUNT. ALL FUNCTIONS DONE?
4214 026326 001352                BNE     2$           ;IF NOT, GO DO THE NEXT ONE
4215 026330 005304                DEC      R4           ;REDUCE THE OVERALL COUNT. ALL BLOCK SIZES DONE?
4216 026332 001334                BNE     1$           ;IF NOT, GO DO THE NEXT ONE
4217 026334 004737 013444      JSR      PC,TESTL
4218 026340 004737 014732      JSR      PC,TESTU
4219 026344 004737 015142      JSR      PC,TESTV
4220 026350 000137 026206      JMP      TESTXX
4221 026354 177777 177776 177775  SIZTAB: .WORD  -1,-2,-3,-4,-5,-6,-7,-10,-20,-40,-100,-200,-400,-1000,-2000
(1) 026412 000402 001410 000004  FUNCTB: .WORD  WRITE,SPACER,READ
(1) 026420 030576 000000 034636  BUFTAB: .WORD  OUTPUT,0,INPUT
(1) 026426 025206 025666 025372  SUBTST: .WORD  TESTCX,TESTEX,TESTDX

-----*
4222 ;THIS ROUTINE READS DATA WRITTEN BY TESTXX.
4223
4224
4225 026434 012704 000017      TESTXY: MOV      #15,R4      ;COUNT THE NUMBER OF ITERATIONS
4226 026440 012703 026354      MOV      #SIZTAB,R3     ;POINT TO THE TABLE OF BLOCK SIZES
4227 026444 012737 000002 001264      MOV      #2,WRTFLG     ;SET RETURN FLAG
4228 026452 004737 030210      JSR      PC,ENAINI
4229 026456 012337 026474      1$:     MOV      (R3)+,2$     ;LOAD SUBROUTINE INTERFACE
4230 026462 010346                MOV      R3,-(SP)     ;SAVE REGISTERS
4231 026464 010446                MOV      R4,-(SP)
4232 026466 004537 026544      JSR      R5,TRYIT      ;GO DO A READ FUNCTION
4233 026472 000004                .WORD      READ       ;EXECUTE A READ
4234 026474 000000      2$:     .WORD      0         ;BLOCK SIZE
4235 026476 034636                INPUT       ;BUFFER TO USE
4236 026500 004737 025372      JSR      PC,TESTDX     ;EXECUTE THE FOLLOWUP ROUTINE
4237 026504 012604                MOV      (SP)+,R4     ;RESTORE REGISTERS.
4238 026506 012603                MOV      (SP)+,R3
4239 026510 005304                DEC      R4           ;REDUE COUNT. ALL BLOCK SIZES DONE?
4240 026512 001361                BNE     1$           ;IF NO, GO DO THE NEXT ONE
4241 026514 004737 015142      JSR      PC,TESTV
4242 026520 000137 026434      JMP      TESTXY
4243
    
```

```

4244
4245
4246
4247
4248
4249
4250
4251
4252 026524 000000 USEAO: 0
4253 026526 000000 USEBO: 0
4254 026530 000000 USECO: 0
4255 026532 000000 USEDO: 0
4256
4257 026534 000000 USEA: 0 ;FUNCTION
4258 026536 000000 USEB: 0 ;WORD COUNT
4259 026540 000000 JSEC: 0 ;INITIAL BUFFER ADDRESS
4260 026542 000000 USED: 0 ;TEMP STORAGE LOCATION
4261
4262 026544 012537 026534 TRYIT: MOV (R5)+,USEA ;GET THE FUNCTION
4263 026550 012537 026536 MOV (R5)+,USEB ;GET THE WORD COUNT
4264 026554 012537 026540 MOV (R5)+,USEC ;GET THE BUS ADDRESS
4265 026560 000223 AS222: STAGAS=STAGAS+1
4266 026560 013737 026534 026524 MOV USEA,USEAO ;SAVE LAST FUNCTION
4267 026566 013737 026536 026526 MOV USEB,USEBO ;SAVE LAST WORD COUNT
4268 026574 013737 026540 026530 MOV USEC,USECO ;SAVE LAST BUS ADDRESS
4269 026602 004737 013662 JSR PC,TESTN
4270
4271 026606 000224 A= AS223: STAGAS=STAGAS+1
4272 026606 017737 152564 026542 MOV @TRCR,USED ;STORE COMMAND REGISTER
4273 026614 042737 175577 026542 BIC #175577,USED ;CLEAR UNWANTED BITS
4274 026622 022737 002200 026542 CMP #002200,USED ;TEST THE COMMAND REGISTER
4275 026630 001413 BEQ AS224
4276 026632 012705 002200 MOV #02200,R5
4277 026636 013704 026542 MOV USED,R4
4278 026642 013737 001376 001244 MOV TRCR,REGIST
4279 026650 104001 ERROR 1 ;INCORRECT REGISTER MATCHUP
4280 026652 104402 007207 TYPE ,MUNRDY
4281
4282 026656 000753 BR A ;GET RID OF DERORS
4283 026660 000225 AS224: STAGAS=STAGAS+1
4284
4285 026660 042777 000001 152514 BIC #BIT0,@TRSR
4286 026666 052737 000001 026534 BIS #BIT0,USEA ;SET GO BIT IN COMMAND
4287
4288 026674 013777 026536 152504 MOV USEB,@TRWC ;SET WORD COUNT
4289 026702 013777 026540 152502 MOV USEC,@TRBA ;SET INITIAL BUS ADDRESS
4290 026710 000240 NOP ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
4291 ;EQUIPMENT STATUS OR PROGRAM OPERATION
4292 026712 013777 026534 152456 MOV USEA,@TRCR ;SET CONTROL REGISTER
4293
4294 026720 004737 011522 JSR PC,TESTA
4295 026724 000240 NOP ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
4296 ;EQUIPMENT STATUS OR PROGRAM OPERATION
4297
4298 026726 032777 000200 152446 BIT #BIT7,@TRSR
4299 026734 001424 BEQ AS225
    
```

```

4300 026736 005737 001264      TST      WRTFLG      ;TEST FOR RETURN FLAG
4301 026742 001417              BEQ      1$         ;NO FLAG
4302 026744 032737 000001 001264  BIT      #1,WRTFLG  ;WRITE RETURN?
4303 026752 001003              BNE      2$         ;YES
4304 026754 022626              CMP      (SP)+,(SP)+ ;OK READ
4305 026756 005726              TST      (SP)+
4306 026760 000403              BR       3$
4307 026762 022626              2$:      CMP      (SP)+,(SP)+
4309 026764 022626              CMP      (SP)+,(SP)+
4309 026766 022626              CMP      (SP)+,(SP)+
4310 026770 005037 001264      3$:      CLR      WRTFLG
4311 026774 004737 012624      JSR      PC,TESTI
4312 027000 000207              RTS      PC
4313 027002              1$:
4314 027002 004737 012624      JSR      PC,TESTI
4315 027006 000226      AS225:  STAGAS=STAGAS+1
4316 027006 000205      RTS      R5      ;RETURN TO CALLING PROCEDURE
    
```

-----*

; THIS ROUTINE PERFORMS TAPE FUNCTION IT IS CALLED
 ; BY THE EOTFUN MACRO THIS IS USED FOR EOT FUNCTIONS ONLY

```

4324 027010 005737 030324      EOTTST: TST      INTFLG      ;CHECK INTERRUPT FLAG
4325 027014 001403              BEQ      AS226
4326 027016 052737 000100 026534  BIS      #BIT6,USEA  ;ENABLE INTERRUPT
4327 027024 000227      AS226:  STAGAS=STAGAS+1
4328 027024 013737 026534 026524  MOV      USEA,USEAO  ;SAVE LAST FUNCTION
4329 027032 013737 026536 026526  MOV      USEB,USEBO  ;SAVE LAST WORD COUNT
4330 027040 013737 026540 026530  MOV      USEC,UCECO  ;SAVE LAST BUS ADDRESS
4331 027046 004737 013662      JSR      PC,TESTN
4332 027052 032777 100000 152316  BIT      #BIT15,@TRCR
4333 027060 001412              BEQ      AS227
4334 027062 027062              A=:
4335 027062 104000              ERROR
4336 027064 104403 007155      TYPEF      ,MUNERR      ;ERROR-BIT WAS NOT CORRECT
4337
4338 027070 052777 004000 152300  BIS      #PWRCLR,@TRCR ;DEVICE MASTER RESET
4339 027076 032777 004000 152272  BIT      #PWRCLR,@TRCR ;INSTRUCTION CLEAR ?
4340 027104 001374              BNE      64$         ;IF NC. WAIT FOR IT TO CLEAR
4341 027106 000230      AS227:  STAGAS=STAGAS+1
4342 027106 017737 152264 026542  MOV      @TRCR,USED  ;STORE COMMAND REGISTER
4343 027114 042737 175577 026542  BIC      #175577,USED ;CLEAR UNWANTED BITS
4344 027122 022737 002200 026542  CMP      #002200,USED ;TEST THE COMMAND REGISTER
4345 027130 001413              BEQ      AS230
4346 027132 012705 002200              MOV      #02200,R5
4347 027136 013704 026542              MOV      USED,R4
4348 027142 013737 001376 001244  MOV      TRCR,REGIST
4349 027150 104001              ERROR      1      ;INCORRECT REGISTER MATCHUP
4350 027152 104403 007207      TYPEF      ,MUNRDY
4351
4352 027156 000741              BR       A           ;GET RID OF DERORS
4353 027160 000231      AS230:  STAGAS=STAGAS+1
4354
4355 027160 042777 000001 152214  BIC      #BIT0,@TRSR
    
```

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 87
 DZTRA.HED 13-APR-76 00:00 SUB-TEST ROUTINES

```

4356 027166 052737 000001 026534      BIS      #BIT0,USEA      ;SET GO BIT IN COMMAND
4357
4358 027174 013777 026536 152204      MOV      USEB,ATRWC      ;SET WORD COUNT
4359 027202 013777 026540 152202      MOV      USEC,ATRBA      ;SET INITIAL BUS ADDRESS
4360 027210 000240                                NOP                                ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
4361                                ;EQUIPMENT STATUS OR PROGRAM OPERATION
4362 027212 013777 026534 152156      MOV      USEA,ATRCR      ;SET CONTROL REGISTER
4363
4364 027220 004737 011522                                JSR      PC,TESTA
4365 027224 000240                                NOP                                ;THIS LOCATION MAY BE CHANGED TO A HALT TO INSPECT
4366                                ;EQUIPMENT STATUS OR PROGRAM OPERATION
4367
4368 027226 000207                                RTS      PC
4369
4370 ;-----*
4371
4372                                ;THIS ROUTINE PUTS THE CORRECT PARITY ON ALL WORDS IN
4373                                ;THE OUTPUT BUFFER
4374
4375 027230 012700 030576      FIXBUF: MOV      #OUTPUT,RO      ;SET INDEX
4376                                A=.
4377                                MOV      (RO),R1      ;GET THE WORD
4378 027234 011001                                CLR      R2      ;CLEAR THE COUNTER
4379 027236 005002                                B=.
4380                                TSTB    R1      ;TEST R1 LOWER
4381 027240 105701                                BPL     AS231
4382 027242 100001                                INC     R2      ;INCREMENT COUNTER
4383 027244 005202                                AS231: STAGAS=STAGAS+1
4384 027246 000232                                TSTB    R1      ;TEST R1 LOWER
4385 027248 105701                                BEQ     AS232
4386 027250 001402                                ASL     R1      ;SHIFT R1
4387 027252 006301                                BR      B      ;GO BACK TO B
4388 027254 000771                                AS232: STAGAS=STAGAS+1
4389 027256 000233                                BIT     #BIT0,R2      ;ODD OR EVEN
4390 027258 032702                                BNE     AS233
4391 027260 001002                                BIS     #BIT8,(RO)      ;SET PARITY
4392 027262 052710 000400                                AS233: STAGAS=STAGAS+1
4393 027264 000234                                TST     (RO)+
4394 027266 022700 034636                                CMP     #INPUT,RO      ;ALL DONE
4395 027268 001356                                BNE     A      ;NO
4396 027270 000207                                RTS      PC
4397
4398 ;-----*
4399                                ;SBTTL DEROR HANDLING ROUTINE
4400
4401
4402 ;-----*
4403 027302 012737 177777 001266      WRONG: MOV      #-1,LOC      ;SET UP FOR ERROR
4404 027310 012737 177777 001252      MOV      #-1,ERR      ;SET DEROR FLAG
4405 027316 012600                                MOV      (SP)+,RO      ;GET PC
4406 027320 005740                                TST     -(RO)      ;FIX IT
4407 027322 010037 001254                                MOV      RO,XPC      ;STORE IT
4408 027326 011637 001256                                MOV      (SP),XSR      ;STORE SR
4409 027332 005746                                TST     -(SP)      ;FIX THE STACK
4410 027334 005737 030206                                TST     VECSTR      ;TEST VECTOR FLAG
4411 027340 001402                                BEQ     AS234

```

```

4412 027342 000137 030150          JMP      VECTR      ;RETURN TO VECTOR
4413 027346 000235          AS$234: $TAGAS=$TAGAS+1
4414 027346 000002          RTI
4415
4416
4417 027350          WRONGX:
4418 027350 004537 030412          JSR      R5,SAVE
4419 027354 032777 000020 151620          BIT      #BIT4,DSWR ;TEST SWITCH BIT 04
4420 027362 001406          BEQ      AS$235
4421 027364 023727 001232 000062          CMP      ERRCNT,#MAXERR ;TEST DEROR COUNT
4422 027372 101024          BHI      WRONGY     ;BRANCH IF HIGHER
4423 027374 005237 001232          INC      ERRCNT     ;INCREMENT COUNT
4424 027400 000236          AS$235: $TAGAS=$TAGAS+1
4425 027400 104402 027476          TYPE    WHED1
4426 027404 022737 177777 001266          CMP      #-1,LOC    ;TEST LOCATION FOR ERROR
4427 027412 001402          BEQ      AS$236
4428 027414 104402 027535          TYPE    WHED2
4429 027420 000237          AS$236: $TAGAS=$TAGAS+1
4430 027420 104403 004352          TYPEF   ,MCRLF     ;TYPE A CARRIAGE RETURN
4431 027424 104413          CNVRT   ;OCTAL TO ASCII CONVERT ROUTINE
4432 027426 027452          WRDATA  ;POINTER TO DATA TO PRINT
4433 027430 022737 177777 001266          CMP      #-1,LOC
4434 027436 001402          BEQ      WRONGY
4435 027440 104413          CNVRT   ;YES-EXIT
4436 027442 027464          WRDAT2  ;OCTAL TO ASCII CONVERT ROUTINE
4437 027444 005237 001232          WRONGY: INC      ERRCNT ;DATA TO CONVERT AND PRINT
4438 027450 000207          RTS      PC        ;INCREMENT 1 OVER LIMIT
4439 027452 000003          WRDATA: .WORD    3   ;NUMBER OF DATA TO PRINT
4440 027454 006 002          .BYTE   6,2       ;NUMBER OF CHARACTERS AND SPACES TO PRINT
4441 027456 001250          TSTPTR  ;LOCATIONS OF DATA
4442 027460 001254          XPC
4443 027462 001256          XSR
4444 027464 000003          WRDAT2: .WORD    3   ;NUMBER OF WORDS TO PRINT
4445 027466 006 002          .BYTE   6,2       ;NUMBER OF CHARACTERS, SPACES
4446 027470 001260          COR     ;LOCATION OF DATA
4447 027472 001262          ACT
4448 027474 001266          LOC
4449 027476 051777 041125 042524          WHED1: .ASCIZ  <377>/SUBTEST PC STATUS /
4450 027504 052123 020040 020040
4451 027512 041520 020040 020040
4452 027520 020040 020040 051440
4453 027526 040524 052524 020123
4454 027534 000
4455 027535 040 020040 020040          WHED2: .ASCIZ  / CORRECT ACTUAL LOCATION /
4456 027542 047503 051122 041505
4457 027550 020124 020040 040440
4458 027556 052103 040525 020114
4459 027564 020040 020040 047514
4460 027572 040503 044524 047117
4461 027600 000040
4462
4463          .EVEN
4464          ;-----*
4465          ;SBTTL CORE SIZE ROUTINE
4466
4467          ;THIS ROUTINE TESTS THE CORE SIZE AND INSURES IT IS OK

```



```

4468
4469 027602 000414 BR CORE
4470 027604 X=.
4471 027604 162700 000002 SUB #2,RO
4472 027610 010037 027714 MOV RO, LASTAD ;SET LAST ADDRESS
4473 027614 022626 POP2SP
4474 027616 023727 027714 040000 CMP LASTAD, #CORSIZ ;TEST CORE SIZE
4475 027624 103001 BHIS AS237
4476 027626 104000 ERROR ;ERROR-BIT WAS NOT CORRECT
4477 027630 000240 AS237: STAGAS=STAGAS+1
4478 027630 000137 002514 JMP RESTRT
4479 027634 012737 027604 000004 CORE: MOV #X,4 ;SET VECTOR
4480 027642 012737 000000 000006 MOV #0,6 ;SET STATUS
4481 027650 C05000 CLR RO ;SET RO TO ZERO
4482 027652 027652 A=.
4483 027652 105720 TSTB (RO)+ ;TEST ADDRESS
4484 027654 000776 BR A ;CONTINUE
4485
4486 027656 X=.
4487 027656 012737 177777 027716 MOV #-1, ADDERR ;SET DEROR FLAG
4488 027664 000002 RTI
4489 027666 005037 027716 ADDTST: CLR ADDERR ;CLEAR DEROR FLAG
4490 027672 012737 027656 000004 MOV #X,4 ;SET VECTOR
4491 027700 012737 000340 000006 MOV #340,6 ;SET STATUS
4492 027706 005710 TST (RO) ;TEST ADDRESS
4493 027710 000240 NOP ;DELAY
4494 027712 000207 RTS PC
4495
4496
4497
4498 027714 000000 LASTAD: 0 ;LAST ADDRESS IN CORE
4499 027716 000000 ADDERR: 0 ;ADDRESS DEROR FLAG
4500 027720 VECADD:
4501 027720 VECSTA:
4502 027720 000000 PRIOR: 0 ;PRIORITY LEVEL OF INTERRUPT
4503
4504 027722 104402 027730 TTYTAB: TYPE MTAB
4505 027726 000207 RTS PC
4506 027730 000011 MTAB: .ASCIZ<11>
4507 .EVEN
4508
4509 -----*
4510 ;SBTTL TTY DECIMAL OUTPUT ROUTINE
4511
4512 ;THIS ROUTINE CONVERTS A OCTAL VALUE IN THE EXCHANGE REGISTER TO
4513 ;A DECIMAL VALUE AND PRINTS IT AT THE TTY.
4514
4515
4516 027732 X=.
4517 027732 000207 RTS PC
4518 027734 CTR=.
4519 027734 000000 0
4520 027736 A=.
4521 027736 162700 000012 SUB #12,RO ;SUBTRACT 12
4522 027742 005237 027734 INC CTR ;INCREMENT COUNTER
4523 027746 000407 BR AS240
    
```

```

4524 027750 032777 000001 151224 TYPTEN: BIT #BIT0,DSWR ;TEST INHIBIT PRINT BIT
4525 027756 001365 BNE X ;EXIT IF SET
4526 027760 005046 CLR -(SP) ;STORE STOP CODE
4527 027762 B=.
4528 027762 005037 027734 CLR CTR ;SET THE COUNTER TO 000
4529 027766 000241 AS240: $TAGAS=$TAGAS+1
4530 027766 020027 000012 CMP RO,#12 ;TEST EXCHANGE
4531 027772 103361 BHIS A ;BRANCH IF MORE THAN 12
4532 027774 062700 000260 ADD #260,RO ;CONVERT OT TTY CODE
4533 030000 010046 MOV RO,-(SP) ;STACK THE CODE
4534 030002 013700 027734 MOV CTR,RO ;SET RO
4535 030006 001365 BNE B ;BRANCH IF RO NOT 000
4536 ;THIS ROUTINE PRINTS TTY CODES PREVIOUSLY PLACED ON THE STACK.
4537 ;TYPING A 00 CODE WILL TERMINATE THIS ROUTINE.
4538
4539
4540
4541 030010 030010 A=.
4542 030010 030010 TYPSTK:
4543 030010 105777 151174 $AS=.
4544 030014 100375 TSTB @TPCSR
4545 030016 012677 151170 BPL $AS
4546 030022 001372 MOV (SP)+,@TPDBR ;PRINT
4547 030024 030024 BNE A ;NO-CONTINUE
4548 030024 105777 151160 $AS=.
4549 030030 100375 TSTB @TPCSR
4550 030032 000207 BPL $AS
4551 RTS PC
  
```

-----*

:SBTTL INTERRUPT VECTOR ROUTINE

:THIS ROUTINE SETS UP CORE ADDRESSES FROM 0 TO 1000 AND WATCHES
 :FOR ILLEGAL INTERRUPTS

```

4558 030034 030034 A=.
4559 030034 005001 CLR R1 ;PRESET R1
4560 030036 012702 000002 MOV #2,R2 ;PRESET R2
4561 030042 030042 B=.
4562 030042 010221 MOV R2,(R1)+ ;SET ADDRESS
4563 030044 022222 CMP (R2)+,(R2)+ ;INCREMENT R2
4564 030046 012721 020367 MOV #20367,(R1)+ ;SET STATUS
4565 030052 022701 001000 CMP #1000,R1 ;ALL ADDRESSES SET
4566 030056 003371 BGT B ;NO
4567 030060 012737 000340 000016 MOV #340,16 ;SET TRAP STATUS
4568 030066 012737 030112 000014 MOV #VECTOR,14 ;SET TRAP VECTOR
4569 030074 012737 000340 000022 MOV #340,22 ;SET UP FOR
4570 030102 012737 027302 000020 MOV #WRONG,20 ;DEROR VECTOR
4571 030110 000207 RTS PC
4572 030112 005737 000000 VECTOR: TST 0 ;TEST ADDRESS 000
4573 030116 001746 BEQ A ;BRANCH IF ZERO
4574 030120 021627 001000 CMP (SP),#1000 ;TEST THE ADDRESS
4575 030124 003005 BGT AS241
4576 030126 011637 030206 MOV (SP),VECSTR ;STORE VECTOR ADDRESS
4577 030132 022626 POP2SP
4578 030134 030134 A=.
4579 030134 000137 027302 JMP WRONG ;GET PC AND SR
  
```

INTERRUPT VECTOR ROUTINE

```

4580 030140 000242 AS$241: $TAGAS=$TAGAS+1
4581 030140 012737 177777 030206 MOV #-1,VECSTR ;SET VECSTR TO -1
4582 030146 000772 BR A
4583 030150 VECTR:
4584 030150 004737 027350 JSR PC,WRONGX
4585 030154 022737 177777 030206 CMP #-1,VECSTR ;TEST FOR BREAK POINT
4586 030162 001410 BEQ AS$242
4587 030164 013700 030206 MOV VECSTR,R0 ;GET ADDRESS
4588 030170 162700 000006 SUB #6,R0 ;FIX IT
4589 030174 000000 A=.
4590 030176 005037 030206 HALT
4591 030202 000002 CLR VECSTR ;CLEAR STORAGE
4592 030204 000243 RTI
4593 030204 000773 AS$242: $TAGAS=$TAGAS+1
4594 030206 000000 BR A
4595 030206 000000 VECSTR: 0
4596
4597 -----*
4598 ;SBTTL INTERRUPT HANDLING ROUTINE
4599
4600 ;THIS ROUTINE HANDLES INTERRUPTS FROM THE EQUIPMENT BEING
4601 ;TESTED.
4602
4603 030210 012737 177777 030324 ENAINT: MOV #-1,INTFLG ;SET INTERRUPT TEST FLAG
4604 030216 012737 000340 177776 MOV #340,PS ;SET PROCESSOR TO 7
4605 030224 012777 030326 151162 MOV #INTIN,@TRVCT ;SET UP VECTOR
4606 030232 012777 000340 151156 MOV #340,@TRRIS ;SET UP STATUS
4607 030240 052777 000100 151130 BIS #100,@TRCR ;ENABLE INTERRUPT
4608 030246 000207 RTS PC
4609
4610
4611 030250 005037 177776 CHKINT: CLR PS ;CLEAR PROCESSOR PRIORITY
4612 030254 000240 NOP ;DELAY
4613 030256 000240 NOP
4614 030260 005737 030324 TST INTFLG ;TEST INTERRUPT FLAG
4615 030264 001401 BEQ AS$243
4616 030266 104000 ERROR
4617
4618 030270 000244 AS$243: $TAGAS=$TAGAS+1
4619
4620
4621 030270 042737 000100 164000 DISINT: BIC #100,164000 ;DISABLE INTERRUPT
4622 030276 005037 030324 CLR INTFLG ;CLEAR INTERRUPT FLAG
4623 030302 005037 000000 CLR 0 ;CLEAR VECTOR FLAG
4624 030306 004537 030412 JSR R5,SAVE
4625 030312 004737 030112 JSR PC,VECTOR
4626 030316 004537 030444 JSR R5,SETALL
4627 030322 000207 RTS PC
4628
4629 030324 000000 INTFLG: 0 ;INTERRUPT FLAG = 0 WHEN INTERRUPT OCCURS
4630
4631 030326 042737 000100 164000 INTIN: BIC #100,164000 ;DISABLE INTERRUPT
4632 030334 005037 030324 CLR INTFLG ;CLEAR THE FLAG
4633 030340 000240 NOP
4634 030342 000002 RTI
4635

```

4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691

-----*
 ;SBTTL PASS NUMBER ROUTINE

;THIS ROUTINE KEEPS TRACK OF THE CURRENT PASS COUNT AND
 ;DISPLAYS IT WHEN REQUESTED BY THE OPERATOR OR OTHER ROUTINES.

		030344		PASSUB=.	
		000000		0	
	030344	030346		PASTOL=.	
		000000		0	
		030350		A=.	
	030350	013700	030346	PAST:	MOV PASTOL,RO ;SET EXCHANGE
	030354	063700	030344		ADD PASSUB,RO ;ADD SUB TOTAL
	030360	004537	030412		JSR R5,SAVE
	030364	004737	027750		JSR PC,TYPTEN
	030370	004537	030444		JSR R5,SETALL
		030374			B=.
	030374	000207			RTS PC

-----*
 ;SBTTL AUTO RESTART ROUTINE

		000240		AUTRES:	NOP
				;PRINT	TOTAL PASSES COMPLETED
	030400	004737	010052		JSR PC,PRETST
	030404	013746	001276		MOV XTST,-(SP) ;STACK THE TEST ADDRESS
	030410	000207			RTS PC

-----*
 ;SBTTL SAVE AND SET REGISTERS ROUTINE

		010046		SAVE:	MOV RO,-(SP)
	030412	010146			MOV R1,-(SP)
	030414	010246			MOV R2,-(SP)
	030416	010346			MOV R3,-(SP)
	030420	010446			MOV R4,-(SP)
	030422	000205			MOV R5,-(SP)
	030424				RTS R5
		005726		SET:	TST (SP)+
	030426	012604			MOV (SP)+,R4
	030430	012603			MOV (SP)+,R3
	030432	012602			MOV (SP)+,R2
	030434	012601			MOV (SP)+,R1
	030436	005726			TST (SP)+ ;DONT RESTORE RO
	030440	000205			RTS R5

4690
4691
4692
4693
4694
4695 030444 005726
4696 030446 012604
4697 030450 012603
4698 030452 012602
4699 030454 012601
4700 030456 000205

SETALL: TST (SP)+
MOV (SP)+,R4
MOV (SP)+,R3
MOV (SP)+,R2
MOV (SP)+,R1
RTS R5

-----*
:SBTTL PRESET ROUTINE

:THIS ROUTINE IS USED TO PRESET ALL VARIABLES, REGISTERS ECT. PRIOR TO
:PERFORMING A TEST.

4710 030460 012600
4711 030462 012706 001200
4712 030466 010046
4713 030470 005000
4714 030472 005001
4715 030474 005002
4716 030476 005003
4717 030500 005004
4718 030502 005005
4719 030504 005077 150474
4720 030510 005077 150474
4721 030514 005077 150466
4722 030520 005077 150466
4723 030524 005037 177776
4724 030530 005037 030344
4725 030534 005037 030346
4726 030540 005077 150436
4727 030544 005037 030206
4728 030550 005037 030324
4729 030554 005037 000000
4730 030560 005037 001252
4731 030564 005037 001232
4732 030570 000207

PRESET: MOV (SP)+,R0 ;GET THE RETURN
MOV #STACK,SP ;SET STACK
MOV R0,-(SP) ;RESTORE RETURN
CLR R0 ;CLEAR THE REGISTERS
CLR R1
CLR R2
CLR R3
CLR R4
CLR R5
CLR @TKCSR ;CLEAR STATUS WORDS
CLR @TPCSR
CLR @TKCBA ;CLEAR BUFFERS
CLR @TPDBA
CLR PS ;CLEAR PROCESSOR STATUS
CLR PASSUB ;CLEAR PASS SUBTO TOTAL
CLR PASTOL ;CLEAR PASS TOTAL
CLR @SWR ;CLEAR SWITCH REGISTER
CLR VECSTR
CLR INTFLG ;CLEAR INTERRUPT ENABLE
CLR C
CLR ERR ;CLEAR THE DEROR FLAG
CLR ERRCNT ;CLEAR THE DEROR COLNT
RTS PC

-----*
:SBTTL INPUT - OUTPUT BUFFERS

4733
4734
4735
4736
4737 030572 000000 000000
4738 030576
4739
(1) 030576 000652 000525 000652
(1) 030636 000652 000525 000652
(1) 030676 000652 000525 000652
(1) 030736 000652 000525 000652
(1)
(1)
(1) 030776 000001 000002 000403 001,002,403,004,405,406,007,010,411,412,013,414,015,016,417,020
(1) 031036 000421 000422 000023 421,422,023,424,025,026,427,430,031,032,433,034,435,436,037,040

BADOUT: 0,0
OUTPUT: ;OUTPUT BUFFER STARTS HERE
652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
;100
001,002,403,004,405,406,007,010,411,412,013,414,015,016,417,020
421,422,023,424,025,026,427,430,031,032,433,034,435,436,037,040

(1)	031076	000441	000442	000043	441,442,043,444,045,046,447,450,051,052,453,054,455,456,057,460
(1)	031136	000061	000062	000463	061,062,463,064,465,466,067,070,471,472,073,474,075,076,477,100
(1)					:200
(1)	031176	000501	000502	000103	501,502,103,504,105,106,507,510,111,112,513,114,515,516,117,520
(1)	031236	000121	000122	000523	121,122,523,124,525,526,127,130,531,532,133,534,135,136,537,540
(1)	031276	000141	000142	000543	141,142,543,144,545,546,147,150,551,552,153,554,155,156,557,160
(1)	031336	000561	000562	000163	561,562,163,564,165,166,567,570,171,172,573,174,575,576,177,200
(1)					:300
(1)	031376	000601	000602	000203	601,602,203,604,205,206,607,610,211,212,613,214,615,616,217,620
(1)	031436	000221	000222	000623	221,222,623,224,625,626,227,230,631,632,233,634,235,236,637,640
(1)	031476	000241	000242	000643	241,242,643,244,645,646,247,250,651,652,253,654,255,256,657,260
(1)	031536	000661	000662	000263	661,662,263,664,265,266,667,670,271,272,673,274,675,676,277,700
(1)					:400
(1)	031576	000301	000302	000703	301,302,703,304,705,706,307,310,711,712,313,714,315,316,717,320
(1)	031636	000721	000722	000323	721,722,323,724,325,326,727,730,331,332,733,334,735,736,337,340
(1)	031676	000741	000742	000343	741,742,343,744,345,346,747,750,351,352,753,354,755,756,357,760
(1)	031736	000361	000362	000763	361,362,763,364,765,766,367,370,771,772,373,774,375,376,777,400
(1)					:500
(1)	031776	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)	032036	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)	032076	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)	032136	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)					:600
(1)	032176	000777	000400	000777	777,400,777,400,777,400,777,400,777,400,777,400,777,400,777,400
(1)	032236	000001	000001	000001	001,001,001,001,001,001,001,001,001,001,001,001,001,001,001,001
(1)	032276	000010	000010	000010	010,010,010,010,010,010,010,010,010,010,010,010,010,010,010,010
(1)	032336	000100	000100	000100	100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
(1)					:700
(1)	032376	000777	000400	000777	777,400,777,400,777,400,777,400,777,400,777,400,777,400,777,400
(1)	032436	000002	000002	000002	002,002,002,002,002,002,002,002,002,002,002,002,002,002,002,002
(1)	032476	000020	000020	000020	020,020,020,020,020,020,020,020,020,020,020,020,020,020,020,020
(1)	032536	000200	000200	000200	200,200,200,200,200,200,200,200,200,200,200,200,200,200,200,200
(1)					:1000
(1)	032576	000777	000400	000777	777,400,777,400,777,400,777,400,777,400,777,400,777,400,777,400
(1)	032636	000004	000004	000004	004,004,004,004,004,004,004,004,004,004,004,004,004,004,004,004
(1)	032676	000040	000040	000040	040,040,040,040,040,040,040,040,040,040,040,040,040,040,040,040
(1)	032736	000400	000400	000400	400,400,400,400,400,400,400,400,400,400,400,400,400,400,400,400
(1)					:1100
(1)	032776	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)	033036	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)	033076	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)	033136	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)					:1200
(1)	033176	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)	033236	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)	033276	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)	033336	000652	000525	000652	652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1)					:1300

```

(1)
(1) 033376 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 033436 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 033476 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 033536 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) ;1400
(1)
(1) 033576 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 033636 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 033676 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 033736 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) ;1500
(1)
(1) 033776 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 034036 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 034076 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 034136 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) ;1600
(1)
(1) 034176 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 034236 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 034276 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 034336 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) ;1700
(1)
(1) 034376 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 034436 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 034476 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) 034536 000652 000525 000652 652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525,652,525
(1) ;2000
(1)
(1) 034576 000777 000777 000777 777,777,777,777,777,777,777,777,777,777,777,777,777,777,777,777,777,777
(1)
(1) 034636 002000 INPUT: .BLKW 2000
(1)

```

```

4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759

```

```

;THIS UTILITY CALCULATES PRIORITY LEVEL,SETS UP CSR'S,SETS UP VECTORS.
TRLEV: ASL TRPRT ;BUILD PRIORITY IN THIS LOCATION
ASL TRPRT ;USING ARITHMETIC SHIFTS, ROTATE
ASL TRPRT ; THE PRIORITY LEVEL PAST
ASL TRPRT ; THE BIT POSITIONS CORRE-
ASL TRPRT ;SPONDING TO THE CONDITION CODES
MOV TRPRT,LESS1 ;MOVE THIS TO LESS1
DEC LESS1 ;CREATE THE NEXT LOWEST PRIORITY
BIC #37,LESS1 ;INSURE THAT THE INZVC BITS ARE CLEAR
MOV TRVCT,RO ;PLACE THE BASE VECTOR ADDRESS IN RO
TST (RO)+ ;CALCULATE THE RECEIVER INTERRUPT STATUS ADDR.
MOV RO,TRRIS ;STORE IT HERE

;THIS SEGMENT SETS UP POINTERS FOR THE GIVEN TR79. TRBASE IS THE BASE ADDRESS
;OF THE DEVICE
MOV TRBASE,RO ;COPY THE BASE ADDRESS INTO A SCRATCH REGISTER
MOV RO,TRCR ;XXXO

```

DZTRAA MACY11 27.1006) 22-SEP-76 12:30 PAGE 96
 DZTRAA.MED 13-APR-76 00:00 INPUT - OUTPUT BUFFERS

```

4760 040724 005200          INC      RO
4761 040726 010037 001400  MOV      RO,HTRCR      ;XXX1
4762 040732 005200          INC      RO
4763 040734 010037 001402  MOV      RO,TRSR      ;XXX2
4764 040740 005200          INC      RO
4765 040742 010037 001404  MOV      RO,HTRSR      ;XXX3
4766 040746 005200          INC      RO
4767 040750 010037 001406  MOV      RO,TRWC      ;XXX4
4768 040754 005720          TST      (RO)+
4769 040756 010037 001412  MOV      RO,TRBA      ;XXX6
4770 040762 000207          RTS      PC
4771 040764 000200          TRPRT:  LEVEL4
4772 040756 000140          LESS1:  LEVEL3      ;LEVEL TO ALLOW INTERRUPTS
4773
4774          ;THIS ROUTINE CALCULATES ODD PARITY FOR AN 8 BIT CHAR
4775 040770 013700 001304  ODD8:   MOV      TEMP1,RO      ;SAVE TEMP1
4776 040774 005001          CLR      RI      ;USE RI TO CREATE PARITY BIT
4777 040776 012727 000010  MOV      #8.,(PC)+      ;COUNT THE NUMBER OF BITS TO CALCULATE
4778 041002 000000          4$:     0      ;USE THIS WORD AS A LOOP COUNTER
4779 041004 006000          1$:     ROR      RO      ;GET A BIT INTO THE CARRY BIT
4780 041006 005501          ADC      RI      ;ADD THE CARRY BIT TO THE BIT COUNT
4781 041010 005337 041002  DEC      4$      ;REDUCE THE COUNT. ARE 8 BITS CHECKED?
4782 041014 001373          BNE      1$      ;IF NOT, GO CHECK THE NEXT BIT
4783 041016 006001          ROR      RI      ;IF SO, GET THE SCALE OF THE SUM INTO THE C BIT
4784 041020 103404          BCS      2$      ;IF IT IS ODD, GO SET UP ODD PARITY
4785 041022 052737 000400 001304  BIS      #BIT8,TEMP1      ;SET ODD PARITY
4786 041030 000403          BR      3$
4787 041032 042737 000400 001304  2$:     BIC      #BIT9,TEMP1      ;CLR EVEN PARITY
4788          ;TEMP1 NOW HAS ODD PARITY CHARACTER
4789 041040 000207          3$:     RTS      PC
4790
4791          ;THIS ROUTINE CALCULATES EVEN PARITY FOR AN 8 BIT CHARACTER
4792 041042 013700 001304  EVEN8:  MOV      TEMP1,RO      ;SAVE TEMP1
4793 041046 005001          CLR      RI      ;USE RI TO CREATE PARITY BIT
4794 041050 012727 000010  MOV      #8.,(PC)+      ;COUNT THE NUMBER OF BITS TO CALCULATE
4795 041054 000000          4$:     0      ;USE THIS WORD AS A LOOP COUNTER
4796 041056 006000          1$:     ROR      RO      ;GET A BIT INTO THE CARRY BIT
4797 041060 005501          ADC      RI      ;ADD THE CARRY BIT TO THE BIT COUNT
4798 041062 005337 041054  DEC      4$      ;REDUCE THE COUNT. ARE 8 BITS CHECKED?
4799 041066 001373          BNE      1$      ;IF NOT, GO CHECK THE NEXT BIT
4800 041070 006001          ROR      RI      ;IF SO, GET THE SCALE OF THE SUM INTO THE C BIT
4801 041072 103004          BCC      2$      ;IF IT IS ODD, GO SET UP ODD PARITY
4802 041074 052737 000400 001304  BIS      #BIT8,TEMP1      ;SET EVEN PARITY
4803 041102 000403          BR      3$
4804 041104 042737 000400 001304  2$:     BIC      #BIT8,TEMP1      ;CLR ODD PARITY
4805          ;TEMP1 NOW HAS EVEN PARITY CHARACTER
4806 041112 000207          3$:     RTS      PC
4807 041114 011637 001304  TRPREG: MOV      (SP),TEMP1      ;SAVE PC
4808 041120 104007          ERROR   7      ;DEVICE NON-EXISTENT
4809 041122 000002          RTI

```


F08

DJTTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 97
 DJTTRAA.MED 13-APR-76 00:00 INPUT - OUTPUT BUFFERS

			: ERROR TABLE		
4810			.ERRTAB:	0	;ERROR 0 BIT ERROR (GENERAL)
4811	041124	000000		0	
4812	041126	000000		0	
4813	041130	000000			
4814					
4815	041132	041220		EM1	;ERROR 1 REGISTER ERROR
4816	041134	041506		DH1	
4817	041136	041640		DT1	
4818					
4819	041140	041242		EM2	;ERROR 2 CONTROLLER ERROR
4820	041142	041506		DH1	
4821	041144	041640		DT1	
4822					
4823	041146	041266		EM3	;ERROR 3 UNIT ERROR
4824	041150	041506		DH1	
4825	041152	041640		DT1	
4826					
4827	041154	041304		MRECF	;ERROR 4 REC FAILED TO INTERRUPT
4828	041156	000000		0	
4829	041160	000000		0	
4830					
4831	041162	041345		MTRANF	;ERROR 5 TRANS FAILED TO INTERRUPT
4832	041164	000000		0	
4833	041166	000000		0	
4834					
4835	041170	041400		MDATAE	;ERROR 6 DATA ERROR
4836	041172	041543		DH2	
4837	041174	041656		DT2	
4838					
4839	041176	041415		MBUS	;ERROR 7 DEVICE NON-EXISTENT
4840	041200	041600		DH3	
4841	041202	041674		DT3	
4842					
4843	041204	041444		MUNITR	;ERROR 10 MISC. UNIT ERROR
4844	041206	041626		DH4	
4845	041210	041706		DT4	
4846					
4847	041212	041467		MWUNIT	;ERROR 11 INVALID UNIT
4848	041214	000000		0	
4849	041216	000000		0	
4850					
4851					

					;ERROR MESSAGES		
4852	041220	020377	042522	044507	EM1:	.ASCIZ	<377>/ REGISTER ERROR /
(1)	041242	020377	047503	052116	EM2:	.ASCIZ	<377>/ CONTROLLER ERROR /
(1)	041266	020377	047125	052111	EM3:	.ASCIZ	<377>/ UNIT ERROR /
(1)	041304	041777	047117	051124	MRECF:	.ASCIZ	<377>/CONTROLLER FAILED TO INTERRUPT /
(1)	041345	377	047125	052111	MTRANF:	.ASCIZ	<377>/UNIT FAILED TO INTERRUPT /
(1)	041400	042377	052101	020101	MDATAE:	.ASCIZ	<377>/DATA ERROR /
(1)	041415	377	042504	044526	MBUS:	.ASCIZ	<377>/DEVICE NON-EXISTENT /
(1)	041444	046777	051511	027103	MUNITR:	.ASCIZ	<377>/MISC. UNIT ERROR /
(1)	041467	377	047111	040526	MWUNIT:	.ASCIZ	<377>/INVALID UNIT /
(1)							
(1)	041506	051377	043505	051511	DH1:	.ASCIZ	<377>/REGISTER EXPECTED ACTUAL/
(1)	041543	377	047125	052111	DH2:	.ASCIZ	<377>/UNIT # EXPECTED ACTUAL/
(1)	041600	042777	051122	051117	DH3:	.ASCIZ	<377>/ERROR PC ADDRESS /
(1)	041626	052777	044516	020124	DH4:	.ASCIZ	<377>/UNIT # /

```

(1)          041640          .EVEN
(1)          041640 000003          DT1: 3 ;DATA TABLES FOR ERROR MESSAGES
(1) 041642 006 004          .BYTE 6,4
(1) 041644 001244          REGIST ;DEVICE REGISTER
(1) 041646 006 004          .BYTE 6,4
(1) 041650 001330          SAVRS ;EXPECTED DATA
(1) 041652 006 002          .BYTE 6,2
(1) 041654 001326          SAVR4 ;ACTUAL DATA
(1)
(1) 041656 000003          DT2: 3
(1) 041660 003 007          .BYTE 3,7
(1) 041662 001236          SAVLIN ;UNIT #
(1) 041664 003 007          .BYTE 3,7
(1) 041666 001330          SAVRS ;EXPECTED DATA
(1) 041670 003 002          .BYTE 3,2
(1) 041672 001326          SAVR4 ;ACTUAL DATA
(1)
(1) 041674 000002          DT3: 2
(1) 041676 006 004          .BYTE 6,4
(1) 041700 001304          TEMPI ;PC
(1) 041702 006 002          .BYTE 6,2
(1) 041704 001244          REGIST ;DEVICE REGISTER
(1)
(1) 041706 000001          DT4: 1
(1) 041710 003 002          .BYTE 3,2
(1) 041712 001236          SAVLIN ;UNIT #
(1) 041714
CORMAX:
.NLIST .SEQ,LOC,BIN
:.ERROR :YOU HAVE EXCEEDED BK CORE LIMITS.
.END
  
```


AS146	023146	3490	3497#
AS147	023206	3503	3510#
AS15	014066	2015	2022#
AS150	023240	3514	3521#
AS151	023272	3524	3531#
AS152	023364	3550	3557#
AS153	023416	3560	3567#
AS154	023446	3570	3577#
AS155	023500	3580	3587#
AS156	023552	3603	3609#
AS157	023632	3628	3635#
AS16	014120	2040	2044#
AS160	023664	3638	3645#
AS161	023716	3648	3655#
AS162	023750	3658	3665#
AS163	024046	3686	3693#
AS164	024100	3696	3703#
AS165	024132	3706	3713#
AS166	024164	3716	3723#
AS167	024214	3726	3732#
AS17	014136	2046	2050#
AS170	024330	3759	3766#
AS171	024362	3769	3776#
AS172	024426	3785	3792#
AS173	024456	3795	3802#
AS174	024572	3830	3837#
AS175	024624	3840	3847#
AS176	024656	3850	3857#
AS177	024710	3860	3867#
AS2	013200	1820	1828#
AS20	014442	2118	2125#
AS200	025024	3895	3902#
AS201	025056	3905	3912#
AS202	025106	3915	3922#
AS203	025140	3925	3932#
AS204	025200	3943	3950#
AS205	025262	3964	3978#
AS206	025336	3981	3995#
AS207	025370	3998	4005#
AS21	014474	2128	2135#
AS210	025444	4013	4026#
AS211	025516	4029	4042#
AS212	025550	4045	4052#
AS213	025722	4099	4106#
AS214	025756	4109	4116#
AS215	026012	4119	4126#
AS216	026050	4137	4144#
AS217	026104	4147	4154#
AS22	014524	2138	2145#
AS220	026152	4163	4170#
AS221	026204	4173	4180#
AS222	026560	4265#	
AS223	026606	4271#	
AS224	026660	4275	4283#
AS225	027006	4299	4315#
AS226	027024	4325	4327#

AS227	027106	4333	4341#
AS23	014554	2148	2155#
AS230	027160	4345	4353#
AS231	027246	4381	4383#
AS232	027256	4385	4388#
AS233	027270	4390	4392#
AS234	027346	4411	4413#
AS235	027400	4420	4424#
AS236	027420	4427	4429#
AS237	027630	4475	4477#
AS24	014660	2182	2191#
AS240	027766	4523	4529#
AS241	030140	4575	4580#
AS242	030204	4586	4593#
AS243	030270	4615	4618#
AS25	014676	2198#	
AS26	015000	2231	2238#
AS27	015032	2241	2248#
AS28	013232	1831	1839#
AS30	015062	2251	2258#
AS31	015112	2261	2268#
AS32	015210	2291	2298#
AS33	015240	2301	2307#
AS34	015270	2310	2316#
AS35	015316	2319	2324#
AS36	015500	2355	2362#
AS37	015540	2367	2374#
AS4	013340	1860	1867#
AS40	015572	2377	2384#
AS41	015622	2387	2394#
AS42	015762	2418	2425#
AS43	016014	2428	2435#
AS44	016046	2438	2445#
AS45	016076	2448	2455#
AS46	016212	2481	2488#
AS47	016244	2491	2498#
AS5	013372	1870	1877#
AS50	016276	2501	2508#
AS51	016330	2511	2518#
AS52	016452	2543	2550#
AS53	016504	2553	2560#
AS54	016536	2563	2570#
AS55	016570	2573	2580#
AS56	016730	2608	2615#
AS57	016762	2618	2625#
AS6	013512	1905	1912#
AS60	017014	2628	2635#
AS61	017046	2638	2645#
AS62	017210	2672	2678#
AS63	017242	2680	2687#
AS64	017274	2690	2697#
AS65	017326	2700	2707#
AS66	017452	2735	2742#
AS67	017504	2745	2752#
AS7	013544	1915	1922#
AS70	017536	2755	2762#

M08

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 105
DZTRA.HED 13-APR-76 00:00 CROSS REFERENCE TABLE -- USER SYMBOLS

LEVEL2=	000100	273#									
LEVEL3=	000140	272#	4772								
LEVEL4=	000200	271#	4771								
LEVEL5=	000240	270#									
LEVEL6=	000300	269#	488								
LEVEL7=	000340	268#	1596								
LF	001272	386#									
LIMITS	003304	782	794#								
LOBITS	003362	768#	79	815#							
LOC	001266	384#	4403#	4426	4433	4448					
LOCK	001220	361#	679#	691	693	949					
LOKFLG	001342	412#									
LOLIM	003354	765#	796	812#							
LPCNT	001224	363#	673#	674	677*						
LSTERR	001234	367#	495*	620*	661*	927	929*	1015*			
LSTMSG	003020	717#	720#								
MASTEK	004654	951	1020#								
MAXERR=	000062	1113#	4421								
MAXTST=	000006	1114#									
MBNXFL	006147	1020#	2548								
MBUS	041415	4839	4852#								
MCRLF	004352	706	846	947	948	956	1020#	1053	1071	4430	
MCSRX	004604	625	1020#								
MDATA	007476	869	879	1034#							
MDATAE	041400	4835	4852#								
MEOT	007235	1020#									
MEPASS	004413	624	1020#								
MEPSRF	007056	1020#	3835								
MERFL	004763	1020#	4142	4152	4168	4178					
MERRPC	004656	954	1020#								
MERRX	004631	631	1020#								
MERR2	004450	1020#									
MERR3	004517	1020#									
MESTAD	011454	1428	1452#								
METFL	006351	1020#	2981								
MFAIL	007344	724	1020#								
MIDSFL	005156	1020#	1616								
MIENFL	005132	1020#	1601								
MILFFL	005316	1020#	1825								
MINSRG	005054	1020#	1515								
MLOCK	004555	604	1020#								
MMRSFL	006413	1020#	3086								
MNTFLG	001344	414#									
MNXBFL	006211	1020#	2613								
MNXMFL	006261	1020#	2677								
MOFLEF	005671	1020#	2048								
MOLEFL	005552	1020#	1986								
MPARFL	007037	1020#	3764								
MPASSX	004620	629	1020#								
MPCFL	005203	1020#	1683								
MPFAIL	004355	1012	1020#								
MQM	004346	753	1020#	1076							
MR	004436	611	1020#								
MRCFL	006333	1020#	2740								
MRDERR	005736	1020#	2187								
MRDFL	004732	1020#	4024	4040	4050						

C09

D2TRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 108
 D2TRAA.HED 13-APR-76 00:00 CROSS REFERENCE TABLE -- USER SYMBOLS

SET	030426	4691#																
SETALL	030444	4626	4654	4695#														
SIFTRB	026354	4188	4221#	4226														
SPACER=	001410	287#	2227	2413	2852	2905	3207	3357	3544	3622	3816	4077	4221					
SPACNT	003635	990#																
STACK =	001200	232#	487	599	986	1008	1200	1249	1296	1347	1395	1425	4711					
STAT	001246	376#																
SUBTST	026426	4194	4221#															
SVDS	003372	825#																
SWR	001202	247#	504	506*	519	575	602	662	669	689	702	721	920	925				
		974	981	983	1051	1179	1196	1210	1259	1306	1358	1396	1402	1445				
		4419	4524	4726*														
SWREG	000176	338#	506															
SW00 =	000001	213#	518															
SW01 =	000002	212#	1051															
SW02 =	000004	211#																
SW03 =	000010	208#																
SW04 =	000020	207#																
SW05 =	000040	206#																
SW06 =	000100	205#																
SW07 =	000200	204#																
SW08 =	000400	202#	981															
SW09 =	001000	201#	689															
SW10 =	002000	200#	983															
SW11 =	004000	199#	669															
SW12 =	010000	198#	920															
SW13 =	020000	197#	702	721	925													
SW14 =	040000	196#																
SW15 =	100000	195#																
TABLE1	010466	1199	1221#															
TABLE2	010636	1248	1270#															
TABLE3	010776	1295	1317#															
TABLE4	011150	1346	1369#															
TABLE5	011304	1384	1409#															
TABLE6	011516	1424	1432	1435	1463#													
TBA =	164006	305#																
TC	001302	390#	1199*	1202	1203	1204*	1248*	1251	1252	1253*	1295*	1298	1299	1300*				
		1346#	1349	1350	1351*	1384*	1387	1388	1389*	1424*	1435*	1436	1437	1438*				
TCR =	164000	302#	1086															
TEMP	007434	1032#	1497*	1517*	1659*	1977*	2338*	2342*	2347*	2352*	3064*							
TEMP1	001304	391#	915*	1022	4775	4785*	4787*	4792	4802*	4804*	4807*	4852						
TEMP2	001306	392#	580*	1024														
TEMP3	001310	393#																
TEMP4	001312	394#																
TEMP5	001314	395#																
TESTA	011522	1473#	2415	2974	4294	4364												
TESTAB	016610	2590#																
TESTAB	017066	1279	2655#															
TESTAC	017346	1327	2717#															
TESTAD	017606	1328	2782#															
TESTAE	020046	1329	2848#															
TESTAF	020240	1330	2901#															
TESTAG	020424	2952#																
TESTAH	020710	1889	2332	2406	2467	2526	2591	2656	2718	2793	2849	2902	3021#	3743				
TESTAJ	020750	1224	3035#															
TESTAK	021022	1226	3059#															



TESTAL	021256	1228	3128*	3205	3301																	
TESTAM	021614	1229	3204*																			
TESTAN	022022	1230	3258*																			
TESTAO	022170	1231	3300*	3355																		
TESTAP	022376	1232	3354*	3408																		
TESTAQ	022604	1233	3407*																			
TESTAS	023040	3471*																				
TESTAT	023312	1278	3541*																			
TESTAU	023520	1280	3599*																			
TESTAV	023560	1321	3618*																			
TESTAW	023770	1322	1410	3675*																		
TESTAX	024234	1331	3742*	3813																		
TESTAY	024476	1332	3812*	3878																		
TESTAZ	024730	1333	3877*																			
TESTB	011562	1221	1490*																			
TESTBA	025142	1281	3939*																			
TESTC	011644	1223	1271	1317	1369	1512*																
TESTCA	025556	2407	4058*																			
TESTCK	025606	1707	2719	2784	2850	2903	4069*															
TESTCX	025206	3963*	4065	4073	4221																	
TESTD	011756	1272	1540*																			
TESTDX	025372	4012*	4221	4236																		
TESTE	012144	1273	1596*																			
TESTEK	025626	2720	2785	4075*																		
TESTEX	025666	4080	4098*	4221																		
TESTF	012200	1274	1611*																			
TESTFO	025646	2408	2409	3029	4085*																	
TESTFX	026014	4090	4136*																			
TESTG	012232	1626*	1958																			
TESTH	012476	1320	1705*																			
TESTI	012624	1227	1270	1318	1409	1742*	1793	1853	1933	1999	3060	3542	3676	4311								
		4314																				
TESTJ	013032	1275	1792*																			
TESTJ1	013102	1802	1807*																			
TESTK	013266	1276	1852*																			
TESTL	013444	1899*	4217																			
TESTM	013546	1277	1932*																			
TESTN	013562	1124	1529	1970*	2060	2162	4269	4331														
TESTO	013762	1319	1370	1706	1998*	2399	3027	3619														
TESTP	014070	1225	2033*																			
TESTQ	014212	1234	2071*																			
TESTS	014620	2177*	4054																			
TESTU	014732	2225*	4218																			
TESTV	015142	2285*	4219	4241																		
TESTW	015320	1326	2331*																			
TESTX	015646	2405*																				
TESTXX	026206	1371	1431	4187*	4220																	
TESTXY	026434	1411	4225*	4242																		
TESTY	016116	2466*																				
TESTZ	016332	2525*																				
TKCSR	001204	351*	665	738	4719*																	
TKOBR	001206	352*	667	740	746	4721*																
TLAST =	011314	1074	4852*																			
TPCSR	001210	353*	707	744	922	4543	4548	4720*														
TPOBR	001212	354*	709*	746*	924*	1499*	1519*	1979*	3066*	4545*	4722*											
TREA	001412	461*	2137	2140	2141	2250	2253	2254	2309	2312	2313	2335*	2341*	2346*								

E09

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 110
DZTRA.HED 13-APR-76 00:00 CROSS REFERENCE TABLE -- USER SYMBOLS

2351*	2376	2379	2380	2412*	2437	2440	2441	2500	2503	2504	2531*	2562
2565	2566	2596*	2627	2630	2631	2661*	2689	2692	2693	2754	2757	2758
2819	2822	2823	2876	2879	2880	2928	2931	2929	2958*	2995	2998	2999
3103	3106	3107	3232	3235	3236	3328	3331	3332	3382	3385	3386	3435
3438	3439	3513	3516	3517	3569	3572	3573	3601*	3602	3605	3606	3647
3650	3651	3705	3708	3709	3784	3787	3788	3849	3852	3853	3914	3917
3918	4162	4165	4166	4289*	4359*	4769*						
374#	1045*	4758										
455#	647	1133*	1134	1137	1140	1141	1156	1159	1160	1477	1479	1481
1503*	1504	1523	1526*	1527	1530*	1531	1541*	1542	1544	1545*	1546	1552*
1553	1560*	1561	1568*	1569	1576*	1577	1583*	1584	1597*	1598	1612*	1613
1627*	1628*	1634	1657	1664	1675	1681	1714	1717	1718	1733*	1734	1743*
1744	1755*	1756	1763	1766	1767	1783*	1784	1817	1823	1840*	1841	1859
1862	1863	1879*	1880	1904	1907	1908	1938	1941	1942	1983	1989*	1990
2004	2007	2008	2045	2051*	2061*	2062	2100*	2101	2107*	2117	2120	2121
2163*	2164	2230	2233	2234	2290	2293	2294	2336*	2337*	2354	2357	2358
2396*	2397	2413*	2414*	2417	2420	2421	2457*	2458	2480	2483	2484	2527*
2528*	2532*	2542	2545	2546	2582*	2583	2592*	2593*	2597*	2607	2610	2611
2647*	2648	2657*	2658*	2662*	2671	2674	2675	2709*	2710	2734	2737	2738
2774*	2775	2799	2802	2803	2839*	2840	2856	2859	2860	2909	2912	2913
2953*	2954	2956*	2960*	2975	2978	2979	3022*	3023	3080	3083	3084	3139
3141	3143	3151*	3152	3168	3171	3172	3212	3215	3216	3264	3267	3268
3308	3311	3312	3362	3365	3366	3415	3418	3419	3472*	3473	3489	3492
3493	3533*	3534	3549	3552	3553	3589*	3590	3627	3630	3631	3667*	3668
3685	3688	3689	3733*	3734	3758	3761	3762	3804*	3805	3829	3832	3833
3869*	3870	3894	3897	3898	3963	3973	3974	4012	4021	4022	4098	4101
4102	4136	4139	4140	4272	4278	4292*	4332	4338*	4339	4342	4348	4362*
4607*	4759*											
473#	520*	540										
1048	4744#											
475#	522*	563										
903#												
4807#												
1047*	4744*	4745*	4746*	4747*	4748*	4749	4771*					
464#	4606*	4754*										
457#	1165	1171	1491	1495	1513	1521	1670	1687	1693	1724	1727	1728
1751	1753	1773	1776	1777	1829	1834	1869	1872	1873	1914	1917	1918
1948	1951	1952	1971	1975	2014	2017	2018	2038	2127	2130	2131	2158
2160	2240	2243	2244	2300	2303	2304	2333*	2364	2369	2370	2410*	2427
2430	2431	2490	2493	2494	2529*	2552	2555*	2556	2594*	2617	2620	2621
2659*	2679	2682	2683	2744	2747	2748	2809	2812	2813	2866	2869	2870
2919	2922	2923	2957*	2963	2985	2988	2989	3025	3068	3090	3096	3097
3129	3137	3178	3181	3182	3222	3225	3226	3274	3277	3278	3318	3321
3322	3372	3375	3376	3425	3428	3429	3499	3505	3506	3559	3562	3563
3637	3640	3641	3695	3698	3699	3768	3771	3772	3939	3842	3843	3904
3907	3908	3965	3980	3982	3990	3991	4014	4028	4030	4037	4039	4109
4111	4112	4146	4149	4150	4285*	4298	4355*	4763*				
463#	650	1046*	4605*	4752								
474#	521*	552										
459#	2147	2150	2151	2260	2263	2264	2318	2321	2322	2334*	2340*	2345*
2350*	2386	2389	2390	2411*	2447	2450	2451	2510	2513	2514	2530*	2572*
2575	2576	2595*	2637	2640	2641	2660*	2699	2702	2703	2764	2767	2768
2829	2832	2833	2886	2889	2890	2937	2940	2941	2959*	3005	3008	3009
3113	3116	3117	3189	3192	3193	3242	3245	3246	3285	3288	3289	3338
3341	3342	3392	3395	3396	3445	3448	3449	3523	3526	3527	3579	3582
3583	3657	3660	3661	3715	3718	3719	3794	3797	3798	3859	3862	3863

TRBASE 001242
TRCR 001376

TRCRO 001500
TRLEV 040636
TRLVO 001504
TRPOK 003652
TRPREG 041114
TRPRT 040764
TRRIS 001416
TRSR 001402

TRVCT 001414
TRVCO 001502
TRWC 001406

K09

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 117
 DZTRA.HED 13-APR-76 00:00 CROSS REFERENCE TABLE -- MACRO NAMES

TYPTAB	1#														
TYPTEN	1#	4652													
VECTOR	1#	4624													
WRONG	1#														
WRONGX	1#														
\$BUFFE	1#	1026													
\$CONFI	1#	4743													
\$CYCLE	1#	1036													
\$DELAY	1#	1125	1147	1640	1648	1808	2052	2076	2084	2092	2109	2472	2534	2599	2663
	2725	2790	2966	3044	3072	3480	3749	3819	3884						
\$DPOIN	1#														
\$ECP	1#	613													
\$ERTAB	1#	4910													
\$FINI	1#	4852													
\$GETPA	1#	534	546	558	1055	1427									
\$HEADE	1#	177													
\$HERES	1#	1771	1781	1828	1339	1967	1877	1912	1922	1946	1956	1982	1988	2012	2022
	2044	2050	2125	2135	2145	2155	2191	2198	2238	2248	2258	2268	2298	2307	2316
	2324	2362	2374	2384	2394	2425	2435	2445	2455	2488	2498	2508	2518	2550	2560
	2570	2580	2615	2625	2635	2645	2678	2687	2697	2707	2742	2752	2762	2772	2807
	2817	2827	2837	2864	2874	2884	2894	2917	2926	2935	2944	2983	2993	3003	3013
	3028	3088	3101	3111	3121	3146	3176	3186	3197	3220	3230	3240	3250	3272	3282
	3293	3316	3326	3336	3346	3370	3380	3390	3400	3423	3433	3443	3453	3497	3510
	3521	3531	3557	3567	3577	3587	3609	3635	3645	3655	3665	3693	3703	3713	3723
	3732	3766	3776	3792	3802	3837	3847	3857	3867	3902	3912	3922	3932	3950	3978
	3995	4005	4026	4042	4052	4106	4116	4126	4144	4154	4170	4180	4265	4271	4283
	4315	4327	4341	4353	4383	4388	4392	4413	4424	4429	4477	4529	4580	4593	4618
\$INTER	1#														
\$INTSE	1#														
\$JUNK	1#	472													
\$LINEU	1#														
\$LVLT5	1#														
\$MRESE	1#	1133	1503	1526	1530	1540	1733	1742	1755	1783	1840	1879	1989	2051	2100
	2163	2396	2457	2582	2647	2709	2774	2839	2952	3021	3151	3471	3533	3589	3657
	3733	3804	3869	4338											
\$MRR	1#														
\$MRRO	1#														
\$MRRW	1#														
\$MRRWR	1#														
\$MRWD	1#														
\$MSG	1#	1020													
\$NOSLA	1#	4807													
\$PAR8	1#	4774													
\$PFAIL	1#	995													
\$PRINT	1#														
\$QUEST	1#	499													
\$RESTO	1#														
\$SCOPE	1#	657													
\$SKIPS	1#	1764	1774	1820	1831	1860	1870	1905	1915	1939	1949	1976	1984	2005	2015
	2040	2046	2118	2128	2138	2148	2182	2231	2241	2251	2261	2291	2301	2310	2319
	2355	2367	2377	2387	2418	2428	2438	2448	2481	2491	2501	2511	2543	2553	2563
	2573	2608	2618	2628	2638	2672	2680	2690	2700	2735	2745	2755	2765	2800	2810
	2820	2830	2857	2867	2877	2887	2910	2920	2929	2938	2976	2986	2996	3006	3026
	3081	3094	3104	3114	3130	3169	3179	3190	3213	3223	3233	3243	3265	3275	3286
	3309	3319	3329	3339	3363	3373	3383	3393	3416	3426	3436	3446	3490	3503	3514
	3524	3550	3560	3570	3580	3603	3628	3638	3648	3658	3686	3696	3706	3716	3726

L09

DZTRAA MACY11 27(1006) 22-SEP-76 12:30 PAGE 118

DZTRA.HED 13-APR-76 00:00

CROSS REFERENCE TABLE -- MACRO NAMES

	3759	3769	3785	3795	3830	3840	3850	3860	3895	3905	3915	3925	3943	3964	3991
	3998	4013	4029	4045	4099	4109	4119	4137	4147	4163	4173	4275	4299	4325	4333
	4345	4381	4385	4390	4411	4420	4427	4475	4523	4575	4586	4615			
STRPDE	1#	425	427	429	431	433	435	437	439	441	443	445	447		
STR79	1#	1112													
STSTN	1#	1189	1241	1288	1339	1377	1417								
STUNIT	1#														
SUNIBU	1#														
SVARIA	1#	342													
SXZ	1#	1186	1188	1238	1240	1285	1287	1336	1338	1374	1376	1414	1416		
SSSSSB	1#														

. ABS. 041714 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZTRA.SEQ/SOL/CRF=DZTRA.P11,DZTRA.HED
RUN-TIME: 38 55 6 SECONDS
RUN-TIME RATIO: 160/100=1.5
CORE USED: 36K (71 PAGES)

