

PDP11

MEMORY I/O
MD-11-DZQMA-B

EP-DZQMA-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

This microfiche card contains a grid of frames. The first four columns contain data, while the fifth column is mostly blank. The data in the frames is organized into tables with multiple columns and rows. The text is small and difficult to read, but it appears to be technical data or code. The frames are arranged in a regular grid pattern, typical of microfiche storage.

CONTENTS

- 1. ABSTRACT
- 2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRELIMINARY PROGRAMS
- 3. LOADING PROCEDURE
- 4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS
 - 4.2 STARTING ADDRESS
 - 4.3 PROGRAM AND/OR OPERATOR ACTION
- 5. OPERATING PROCEDURE
 - 5.1 OPERATIONAL SWITCH SETTINGS
 - 5.2 SUBROUTINE ABSTRACT
- 6. ERRORS
- 7. RESTRICTIONS
- 8. MISCELLANEOUS
 - 8.1 EXECUTION TIME
 - 8.2 STACK POINTER
 - 8.3 POWER F IL
- 9. PROGRAM DESCRIPTION

88
89
90
91
92
93
94
95
96
97
98
99
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87

88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143

MAINDEC-11-DZQMA-B
DESCRIPTION

MEMORY I/O EXERCISER

PAGE 3

1. ABSTRACT

THIS PROGRAM CHECKS BANK SELECTION, EA BITS, AND MEMORY USING ANY NPR DEVICE WITH EA BITS. IT RUNS STAND ALONE OR WITH KT11C OR KT11D TO ACCESS EXTENDED MEMORY. WORST CASE NOISE PATTERNS ARE USED WITH THE NPR DEVICE TO TEST THE MEMORY. IF THE PROGRAM IS STARTED BY ACT11, IT WILL CYCLE THROUGH ALL THE EXISTING DEVICES. IF IT IS STARTED BY DDP2 IN CHAIN MODE, IT WILL CYCLE THROUGH ALL THE DEVICES EXCEPT THE DDP2 DEVICE. IN ALL CASES, DRIVE 1 ON THE TC11 WILL BE USED.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD COMPUTER WITH A MINIMUM OF 8K OF MEMORY
KT11C/D FOR MEMORY EXPANSION (OPTIONAL)

2.2 STORAGE

PROGRAM STORAGE - THE ROUTINES USE MEMORY 0 - 17776

2.3 PRELIMINARY PROGRAMS

MEMORY DIAGNOSTICS
DEVICE DIAGNOSTICS

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

START AT 200.

144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199

MAINDEC-11-DZQMA-B
DESCRIPTION

MEMORY I/O EXERCISER

PAGE 4

4.3 PROGRAM AND/OR OPERATOR ACTION

- 1) START AT 200
2) TYPE DEVICE (RF11, RK11, RP11, RC11, TC11, TM11, DM11, OR DR11B) AND RETURN.
3) TEST WILL START (↑C WILL RETURN TO STEP 2 AND RESTORE THE LOADER)
OR
1) START AT 200
2) TYPE A RETURN
3) TYPE THE LOWER BANK TO BE TESTED ... 1=20000-37777 ETC.
4) TYPE THE UPPER BANK TO BE TESTED
5) TYPE DEVICE (RF11, RK11, RP11, RC11, TC11, TM11, DM11, OR DR11B) AND RETURN.
6) TEST WILL START (↑C WILL RETURN TO STEP 2 AND RESTORE THE LOADER)

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AT SA 200 ALL SWITCHES DOWN IS WORST CASE TESTING. THE BELL WILL RING UPON COMPLETION OF A PASS.

5.1.1 SWITCH SETTINGS ARE:

SW<15> = 1 HALT ON ERROR
SW<14> = 1 HANG ON CURRENT BANK
SW<13> = 1 INHIBIT PRINTOUT
SW<10> = 1 INHIBIT BELL ON PASS COMPLETE
SW<09> = 1 INHIBIT USE OF MEMORY EXPANSION DEVICE
SW<08> = 1 TRACE BANK UNDER TEST

5.2 SUBROUTINE ABSTRACTS

5.2.1 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR + 2.

5.2.2 TRAP HANDLER

MOST OF THE SUBROUTINE CALLS ARE DONE VIA A TRAP+N CALL. TO FIND THE SUBROUTINE BEING CALLED, LOOK IN THE COMMENT

F01

MAINDEC-11-DZQMA-B
DZQMA8.CMB

MEMORY I/O EXERCISER

MACY11 27(732) 26-OCT-76 16:08 PAGE 6

200
201

SECTION OF THE TRAP DEFINATION TABLE FOR THE NAME OF THE
TRAP, THEN SCAN TO THE LEFT MARGIN FOR THE STARTING ADDRESS

20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

OF THE SUBROUTINE.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

RF11 ERROR: CS = 100305 ER = 001000

RF11 = DEVICE UNDER TEST
CS = CONTROL AND STATUS REGISTER
ER = ERROR REGISTER

DATA ERROR AT 401220 TRUE = 177777 RECEIVED = 177757

401220 = ADDRESS OF BAD DATA (18 BIT)
TRUE = DATA SENT
RECEIVED = DATA FOUND

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

TO RUN RPO2, CHANGE LOC=5456 TO 000002 (RTI).

8. MISCELLANEOUS

8.1 EXECUTION TIME

THE EXECUTION TIME IS DEPENDANT ON THE AMOUNT OF MEMORY AND THE DEVICE USED. THE BELL SHOULD RING WITHIN 20 MINUTES (USING DM11 AT 110 BAUD IN 124K.)

8.2 STACK POINTER

STACK IS INITALLY SET TO 500

8.3 POWER FAIL

AFTER THE PROGRAM IS STARTED, POWER DOWN THEN JP. THERE

HO1

MAINDEC-11-DZQMA-B
DZQMA8.CMB

MEMORY I/O EXERCISER

MACY11 27(732) 26-OCT-76 16:08 PAGE 8

258

SHOULD BE NO ERROR TYPEOUTS.

259
260
261
262
263
264
265
266
267
268
269

MAINDEC-11-DZQMA-B
DESCRIPTION

MEMORY I/O EXERCISER

PAGE 6

9. PROGRAM DESCRIPTION

THE PROGRAM SIZES CORE AND TEST SEQUENTIALLY ALL CORE OR ANY SECTION OF CORE IN 4K CHUNKS USING THE DEVICE SPECIFIED AND MEMORY EXPANSION DEVICE. WORST CASE NOISE PATTERNS ARE USED FOR EACH MEMORY TYPE ON EACH PASS.

270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400

.TITLE % MAINDEC-11-DZQMA-B MEMORY I/O EXERCISER
.REM*
Abstract

This program checks bank selection, EA bits, and memory using any NPR device with EA bits. It runs stand alone or with KT11C or KT11D to access extended memory. Worst case noise patterns are used with the NPR device to test the memory.

Requirements

PDP-11 with at least 8K of memory (KT11C or KT11D are optional.)

Storage - First 4K

Loading - Absolute Loader

Execution time

The time is dependent on the amount of memory and the device which is used. A bell will signify end of pass. Maximum run time is no greater than 5 minutes.

Starting procedure

- 1) Start at 200
 - 2) Type device (RF11, RK11, RP11, RC11, TC11, TM11, DM11, or DR11B) and RETURN.
 - 3) Test will start (↑C will return to step 2 and restore the loader)
- or
- 1) Start at 200
 - 2) Type a RETURN
 - 3) Type the lower bank to be tested .. 1 = 20000-37777 etc.
 - 4) Type the upper bank to be tested
 - 5) Type device (RF11, RK11, RP11, RC11, TC11, TM11, DM11, or DR11B) and RETURN.
 - 6) Test will start (↑C will return to step 2 and restore the loader)

Switch register options - yes

- SW15 - HALT ON ERROR
- SW14 - LOOP ON CURRENT BANK
- SW13 - INHIBIT TYPEOUTS
- SW10 - INHIBIT BELL
- SW9 - INHIBIT USE OF MEMORY EXPANTION DEVICE
- SW8 - TYPE BANK UNDER TEST*

K01

MAINDEC-11-DZQMA-B
DZQMA8.CMB

MEMORY I/O EXERCISER

MACY11 27(732) 26-OCT-76 16:08 PAGE 11

```
325 ;COPYRIGHT 1972,DIGITAL EQUIPMENT CORP., MAYNARD MASS.
326 ;PROGRAM BY BOB BRAIN
327
328 104400 SCOPE= TRAP
329 104000 HLT= EMT
330 000004 TYPE= IOT
331 177776 PS= 177776
332 177570 SWR= 177570
333 177570 DISPLAY=SWR
334 000007 BELL= 7
335 000000 R0= %0
336 000001 R1= %1
337 000002 R2= %2
338 000003 R3= %3
339 000004 R4= %4
340 000005 R5= %5
341 000005 TTY= %5
342 000006 SP= %6
343 000007 PC= %7
344 000000 N= 0
345 040000 SW14= 40000
346 020000 SW13= 20000
347 010000 SW12= 10000
348 004000 SW11= 4000
349 002000 SW10= 2000
350 001000 SW9= 1000
351 000400 SW8= 400
352 000000 .= 0 ;TRAP CATCHER FROM 0 - 776
353 000200 .= 200
354 000200 000137 001024 JMP @#BEGIN ;JUMP TO BEGINING ADDRESS OF PROGRAM
355
356 001000 000000
357 001002 000000
358 001004 000000 000000
359 001010 000000
360 001012 000000
361 001014 000000
362 001016 000000
363 001020 000000
364 001022 000000
365
366 ICNT: 0 ;LH = ITERATION COUNT ;RH = TEST NO.
367 ERRORS: 0 ;ERROR COUNT
368 PCNT: 0,0 ;2 WORD PASS COUNT
369 LAD: 0 ;LOOP ADDRESS FOR SCOPE
370 HLTADR: 0 ;LAST HLT INSTRUCTION ADD. EXECUTED
371 MX: 0 ;BANK UNDER TEST
372 MIN: 0 ;FIRST BANK
373 LIMIT: 0 ;LAST BANK
374 OPTION: 0 ;OPTION - MI = MX11 - GT = KT11
```

L01

MAINDEC-11-DZQMA-B
DZQMAB.CMB

MEMORY I/O EXERCISER MACY11 27(732) 26-OCT-76 16:08 PAGE 12
SETUP, CORE, AND OPTION SCAN

```

370 001024 000005          BEGIN: RESET          ;ZAP THE WORLD
371 001026 105737 177564  TSTB          2#177564      ;TTY READY
372 001032 100375          BPL          .-4          ;NO - WAIT
373 001034 012706 000500  MOV          #500,SP      ;SET STACK TO ** 500 **
374 001040 012737 012104 000020  MOV          #IOTS,2#20   ;LOAD IOT VECTOR
375 001046 012737 000340 000022  MOV          #340,2#22    ;LOCK UP
376 001054 012737 012612 000024  MOV          #PDOWNS,2#24 ;LOAD PF VECTOR
377 001062 012737 000340 000026  MOV          #340,2#26    ;LOCK UP
378 001070 012737 002716 000034  MOV          #TRAPS,2#34  ;SET FOR TRAP
379 001076 012737 000340 000036  MOV          #340,2#36    ;LOCK UP
380 001076 012777 000240 010430  MOV          #240,2#RFTRAP+2
381 001076 012777 000240 010426  MOV          #240,2#RCTRAP+2
382 001120 012777 000240 010424  MOV          #240,2#RPTRAP+2
383 001126 012777 000240 010422  MOV          #240,2#RKTRAP+2
384 001134 012777 000300 010420  MOV          #300,2#TCTRAP+2
385 001142 012777 000240 010422  MOV          #240,2#TMTRAP+2
386 001150 012777 000240 010410  MOV          #240,2#ORTRAP+2
387 001156 012777 004232 010354  MOV          #TRP.RF,2#RFTRAP
388 001164 012777 004614 010352  MOV          #TRP.RC,2#RCTRAP
389 001172 012777 005324 010350  MOV          #TRP.RP,2#RPTRAP
390 001200 012777 005624 010346  MOV          #TRP.RK,2#RKTRAP
391 001206 012777 006230 010344  MOV          #TRP.TC,2#TCTRAP
392 001214 012777 007136 010346  MOV          #TRP.TM,2#TMTRAP
393 001222 012777 010514 010334  MOV          #TRP.DR,2#ORTRAP
394 001230 012767 012752 176622  MOV          #READR,60    ;KEYBOARD VECTOR
395 001236 012767 140000 010474  MOV          #140000,LOLIM ;INIT LOLIM
396 001244 012767 160000 010470  MOV          #160000,HILIM ;INIT HILIM
397 001252 005067 177540  CLR          MIN
398 001256 005067 177532  CLR          MX
399 001262 012767 000003 010244  MOV          #3,PSCNT     ;INIT PASS COUNT
400 001270 005067 002420  CLR          MEMORY     ;WHICH MEMORY BEING CHECKED
401 001274 005067 177522  CLR          OPTION     ;TYPE THE OPTION
402 001300 032737 001000 177570  BIT          #SW9,2#SWR   ;INHIBIT USE OF MEMORY EXTENTION DEVICE
403 001306 001030  BNE          SIZEIT     ;SKIP
404 001310 012737 001330 000004  MOV          #15,2#4     ;SET FOR TIMEOUT
405 001316 005777 010376  TST          2SR0       ;CHECK FOR KT11
406 001322 105167 177474  COMB        OPTION     ;OPTION IS KT11
407 001326 000415  BR          2S
408
409 001330 012737 001362 000004 1S:  MOV          #2S,2#4     ;SET FOR TIMEOUT
410 001336 005777 010374  TST          2MXC3      ;CHECK FOR MX11
411 001342 105167 177455  COMB        OPTION+1   ;OPTION IS MX11
412 001346 012767 000003 177442  MOV          #3,MIN
413 001354 012767 000003 177432  MOV          #3,MX
414 001362 012737 000006 000004 2S:  MOV          #6,2#4     ;RESET FOR MEM
415
416 001370 005767 177426  SIZEIT: TST          OPTION ;WHICH OPTION AM I
417 001374 100417  BMI          DOMX      ;MUST HAVE MX11
418 001376 001037  BNE          DOSEG     ;MUST HAVE MEMORY MANAGEMENT

```

MO1

MAINDEC-11-DZQMA-B
DZQMA8.CMB

MEMORY I/O EXERCISER MACY11 27(732) 26-OCT-76 16:08 PAGE 13
SETUP, CORE, AND OPTION SCAN

419	001400	012737	001432	000004	DOCORE:	MOV	#25, R#4	;SET FOR MEM
420	001406	012701	017776			MOV	#17776, R1	;SET UP ADDRESS
421	001412	005000				CLR	R0	;SET UP BANK COUNT
422	001414	062701	020000		1\$:	ADD	#20000, R1	;MOVE TO NEXT BANK
423	001420	005200				INC	R0	;INC THE BANK COUNT
424	001422	005711				TST	(1)	;TIMEOUT?
425	001424	022701	177776			CMP	#177776, R1	;END?
426	001430	001371				BNE	1\$;LOOP IF NOT AT THE END
427	001432	000446			2\$:	BR	TYPEIT	;TYPE IT
428								
429	001434	012737	001466	000004	DOMX:	MOV	#25, R#4	;SET UP FOR MEM
430	001442	012700	000003			MOV	#3, R0	;SET BANK COUNT
431	001446	005200			1\$:	INC	R0	;MOVE TO NEXT BANK
432	001450	010077	010262			MOV	R0, RMXC3	;SET UP MXC3
433	001454	005737	157776			TST	R#157776	;TIMEOUT?
434	001460	022700	000036			CMP	#36, R0	;END?
435	001464	001370				BNE	1\$;LOOP IF NOT END
436	001466	012777	000006	010242	2\$:	MOV	#6, RMXC3	;MAP INTO SELF
437	001474	000425				BR	TYPEIT	;TYPE IT
438								
439	001476	104444			DOSEG:	MAPIT		;SETUP MEMORY MANAGEMENT REGISTERS
440	001500	005277	010214			INC	RSR0	;TURN ON MEMORY MANAGEMENT
441	001504	012737	001534	000004		MOV	#25, R#4	;SET TIMEOUT ADDRESS FOR CORE CALCULATIONS
442	001512	005737	157776		1\$:	TST	R#157776	;TRAP ON NON EX MEM
443	001516	062777	000200	010202		ADD	#200, R#KISAR6	;GO TO NEXT BANK
444	001524	022777	007600	010174		CMP	#7600, R#KISAR6	;LAST ONE?
445	001532	003367				BGT	1\$;TRY NEXT
446	001534	017700	010166		2\$:	MOV	R#KISAR6, R0	;SAVE ASR6 IN R0
447	001540	006300				ASL	R0	; *ASH #-7, R0*
448	001542	000300				SWAB	R0	;KLUDGY ISN'T IT
449	001544	042700	177740			BIC	#177740, R0	;CLEAR JUNK

450	001550	005300			TYPEIT:	DEC	RO	; DROP BACK
451	001552	012737	000006	000004		MOV	#6,2#4	; SET FOR MEM
452	001560	012706	000500			MOV	#500,SP	; CLEAR STACK **500**
453	001564	005227	177777			INC	#-1	; TYPE THE OPTION ONLY ONCE
454	001570	001050				BNE	REDO	; FIRST TIME?
455	001572	004767	010150			JSR	7,LODGET	; GET THE LOADER ONLY THE FIRST TIME
456	001576	000004	001602			TYPE	+2	; .ASCIZ <15><12><12>"MAINDEC-11-DZQMA-B"<15><12><12>"BAN
457	001646	010005				MOV	RO,TTY	; TYPE RO IN OCTAL
458	001650	004767	010522			JSR	PC,PRINTS	; AND SUPPRESS LEADING ZERO'S
459	001654	000004	001660			TYPE	+2	; .ASCIZ "EXIST"
460	001670	005767	177126			TST	OPTION	; WHICH OPTION?
461	001674	001406				BEQ	REDO	; NOTHING
462	001676	100403				BMI	IS	; MX11
463	001700	000004	011441			TYPE	,WITHSG	; MEMORY MANAGEMENT
464	001704	000402				BR	REDO	; SKIP
465	001706	000004	011454		IS:	TYPE	,WITHMX	; MX11
466	001712	010067	177102		REDO:	MOV	RO,LIMIT	; SAVE UPPER BANK
467	001716	004767	010112			JSR	7,LODRES	; RESTORE LOADER
468	001722	005737	000042			TST	2#42	; UNDER A MONITOR?
469	001726	001505				BEQ	TO	; NO
470	001730	005037	177776			CLR	2#PS	; CLEAR PS
471	001734	012767	011466	007570	RSIT:	MOV	#DEVTAB-2,DEVADR	; SET FOR DEVICE SCAN
472	001742	062767	000002	007562	ABRT:	ADD	#2,DEVADR	; GO TO NEXT ONE
473	001750	005777	007556		NEWDEV:	TST	2#DEVADR	; IS IT THE END?
474	001754	001003				BNE	IS	; YES
475	001756	004767	001040			JSR	PC,GETMON	; GET THE MONITOR
476	001762	000764				BR	RSIT	; LOOP
477	001764	016703	007542		IS:	MOV	DEVADR,R3	; GET ADDRESS
478	001770	162703	011470			SUB	#DEVTAB,R3	; MAKE IT AN OFFSET
479	001774	126337	011512	000041		CMPB	TAB1(3),2#41	; SAME DEVICE?
480	002002	001757				BEQ	ABRT	; YES - ABORT DEVICE
481	002004	017767	007522	010304		MOV	2#DEVADR,INPUT	; SET INPUT
482	002012	012767	030451	010300		MOV	#11,INPUT+2	; FOR TYPING
483	002020	005067	010276			CLR	INPUT+4	; MAKE TERMINATOR
484	002024	000004	002030			TYPE	,+2	; .ASCIZ <15><12>
485	002034	000004	012316			TYPE	,INPUT	; TYPE THE DEVICE
486	002040	012737	002122	000004		MOV	#BADDEV,2#4	; SET FOR TIMEOUT
487	002046	012777	004232	007464		MOV	#TRP.RF,2#RFTRAP	
488	002054	012777	004614	007462		MOV	#TRP.RC,2#RCTRAP	
489	002062	012777	005324	007460		MOV	#TRP.RP,2#RPTRAP	
490	002070	012777	005624	007456		MOV	#TRP.RK,2#RKTRAP	
491	002076	012777	006230	007454		MOV	#TRP.TC,2#TCTRAP	
492	002104	012777	007136	007456		MOV	#TRP.TM,2#TMTRAP	
493	002112	012777	010514	007444		MOV	#TRP.DR,2#DRTRAP	
494	002120	000423				BR	RTPF	; START IT
495								
496	002122				BADDEV:	TYPE	,+2	; .ASCIZ " - NONE"
497	002122	000004	002126			JMP	ABRT	; GO TO NEXT DEVICE
498	002136	000167	177600					

```

499 002142 012737 002650 000004 TO:  MOV  #NODEV,2#4 ;SET UP TRAP IF NO DEVICE
500 002150 005037 177776 CLR 2#177776 ;DROP PRIORITY
501 002154 000004 002160 TYPE +2 ;ASCIZ (15)(12)"*"
502 002164 004767 010022 JSR PC,READS ;INPUT DATA FROM TTY
503 002170 012706 000500 RTPF: MOV #500,SP ;RESET THE STACK ** 500 **
504 002174 052737 000100 177560 BIS #100,2#177560 ;IE THE TTY
505 002202 022767 041522 010106 CMP #"RC,INPUT ;CHECK FOR RC11
506 002210 001004 BNE T1
507 002212 052777 000100 007376 BIS #100,2#RCCS ;TURN ON INTERRUPT
508 002220 000503 BR WAIT
509 002222 022767 050122 010066 T1:  CMP #"RP,INPUT ;CHECK FOR RP11
510 002230 001004 BNE T2
511 002232 052777 000100 007370 BIS #100,2#RPCS ;TURN ON INTERRUPT
512 002240 000473 BR WAIT
513 002242 022767 043122 010046 T2:  CMP #"RF,INPUT ;CHECK FOR RF11
514 002250 001004 BNE T3
515 002252 052777 000100 007324 BIS #100,2#RFCS ;TURN ON INTERRUPT
516 002260 000463 BR WAIT
517 002262 022767 045522 010026 T3:  CMP #"RK,INPUT ;CHECK FOR RK11
518 002270 001004 BNE T4
519 002272 052777 000100 007344 BIS #100,2#RKCS ;TURN ON INTERRUPT
520 002280 000453 BR WAIT
521 002282 022767 046524 010006 T4:  CMP #"TH,INPUT ;CHECK FOR TH11
522 002290 001004 BNE T5
523 002292 052777 000100 007350 BIS #100,2#TMCS ;TURN ON INTERRUPT
524 002294 000443 BR WAIT
525 002296 022767 041524 007766 T5:  CMP #"TC,INPUT ;CHECK FOR TC11
526 002298 001004 BNE T6
527 002300 052777 000100 007316 BIS #100,2#TCCS ;TURN ON INTERRUPT
528 002302 000433 BR WAIT
529 002304 022767 046504 007746 T6:  CMP #"DM,INPUT ;CHECK FOR DM11
530 002306 001007 BNE T7
531 002308 005777 007322 TST 2#DMCS ;IS IT THERE?
532 002310 012737 000006 000004 MOV #6,2#4 ;RESTORE 4
533 002312 000167 005126 JMP TRP.DM
534 002314 022767 051104 007720 T7:  CMP #"DR,INPUT ;CHECK FOR DR11
535 002316 001004 BNE T8
536 002318 052777 000100 007306 BIS #100,2#DRCS ;TURN ON INTERRUPT
537 002320 000410 BR WAIT
538 002322 105767 007702 T8:  TSTB INPUT ;CR ALONE
539 002324 001420 BEQ CRLF
540 002326 000004 010716 TYPE #1
541 002328 005037 177560 CLR 2#177560 ;ID THE TTY
542 002330 000645 BR TO
543
544 002430 012737 000006 000004 WAIT: MOV #6,2#4 ;SET NEM TRAP
545 002432 016704 007276 15:  MOV LO LIM,R4 ;SET LOW LIMIT *****
546 002434 026704 007274 25:  CMP HI LIM,R4 ;END OF RANGE *
547 002436 001773 BEQ 15 ;YES - RESET * BACKGROUND *
548 002438 000241 CLC ;CLEAR CARRY BIT * TEST *
549 002440 005524 ADC (4)+ ;NOP WITH READ/WRITE *
550 002442 000772 BR 25 ;LOOP *****

```

```

551 002456 012737 000006 000004 CRLF:  MOV      #6,204
552 002464 005037 177560          CLR      #177560          ;ID THE TTY
553 002470 000004 011103          TYPE     M11             ;ASK FOR LOWER LIMIT
554 002474 004767 007512          JSR      PC,READS       ;GET IT
555 002500 005067 176312          CLR      MIN
556 002504 012701 012316          MOV      #INPUT,R1
557 002510 105711          CR1:    TSTB     (1)          ;CR?
558 002512 001414          BEQ      CR2
559 002514 106011          RORB    (1)
560 002516 106011          RORB    (1)
561 002520 106011          RORB    (1)
562 002522 106167 176270          ROLB    MIN
563 002526 106111          ROLB    (1)
564 002530 106167 176262          ROLB    MIN
565 002534 106121          ROLB    (1)+
566 002536 106167 176254          ROLB    MIN
567 002542 000762          BR      CR1
568 002544 005767 176246          CR2:    TST      MIN          ;MIN MUST BE >0
569 002550 003742          BLE     CRLF           ;NOPE!
570 002552 005367 176240          DEC     MIN
571 002556 016767 176234 176230          MOV     MIN,MAX       ;SAVE LOWER LIMIT
572 002564 000004 011121          NXTCR:  TYPE     M12             ;ASK FOR UPPER LIMIT
573 002570 004767 007416          JSR      PC,READS       ;GET IT
574 002574 005000          CLR      RO
575 002576 012701 012316          MOV     #INPUT,R1
576 002602 105711          CR3:    TSTB     (1)          ;CR?
577 002604 001411          BEQ      CR4          ;TEST FOR END
578 002606 106011          RORB    (1)
579 002610 106011          RORB    (1)
580 002612 106011          RORB    (1)
581 002614 106100          ROLB    RO
582 002616 105111          ROLB    (1)
583 002620 106100          ROLB    RO
584 002622 106121          ROLB    (1)+
585 002624 106100          ROLB    RO
586 002626 000765          BR      CR3
587 002630 020067 176164          CR4:    CMP      RO,LIMIT     ;GREATER THAN THE MAX?
588 002634 003353          BGT     NXTCR         ;YES - TRY AGAIN
589 002636 020067 176154          CMP     RO,MIN        ;RO MUST BE > MIN
590 002642 002705          BLT     CRLF         ;IT'S NOT
591 002644 000167 177272          JMP     TO            ;GET ANOTHER
592
593 002650 000004 011137          NODEV:  TYPE     ,M13          ;TYPES NO DEVICE
594 002654 000004 011137          TYPE     ,M13
595 002660 000004 012316          TYPE     ,INPUT
596 002664 000004 002670          TYPE     ,+2
597 002712 000167 176106          JMP     BEGIN        ;.ASCIZ " DOES NOT EXIST"(15)(12)
;RETURN TO TTY INPUT MODE

```



```

598
599
600
601
602
603
604
605 002716 011605
606 002720 162705 000002
607 002724 111505
608 002726 062705 002750
609 002732 022705 003020
610 002736 100401
611 002740 000175 000000
612 002744 000300
613 002746 000776
614
615 002750
616 002750 003716
617 002752 003754
618 002754 004360
619 002756 004742
620 002760 005354
621 002762 005752
622 002764 006322
623 002766 007266
624 002770 010616
625 002772 006566
626 002774 007174
627 002776 003364
628 003000 006322
629 003002 003544
630 003004 003774
631 003006 003562
632 003010 004070
633 003012 003064
634 003014 004160
635 003016 013022
636 003020 013042
637
638
639
640 003022 004767 007006
641 003026 000004 000177
642 003032 000005
643 003034 010046
644 003036 013703 000042
645 003042 004713
646 003044 000240 000240 000240
647 003052 012600
648 003054 005003
649 003056 005203
650 003060 001376
651 003062 000207

```

```

; STRAP ;TRAP HANDLER
; THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APROPRATE
; SUBROUTINE. THE CALL IS A "TRAP+N" WHERE N IS A MULTIPLE OF 2.
; THE "SET" MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
; FOLLOW THIS MACRO.

```

```

TRAPS:  MOV (6),R5 ;GET TRAP INSTRUCTION
        SUB #2,R5 ;BACK UP BY 2
        MOVB (5),R5 ;GET THE RIGHT BYTE OF TRAP
        ADD #ADRTAB,R5 ;INDEX TO TABLE
        CMP #ENDTAB,R5 ;CHECK FOR END OF TABLE
        BMI +4 ;OUT OF BOUNDS
        JMP @2(5) ;GO TO ROUTINE
        HALT ;TRAP OUT OF BOUNDS
        BR -2 ;HANG UP

```

```

ADRTAB:  WRAOR ;CORE = TRAP+0 (104400)
        WRCADR ;COMCOR = TRAP+2 (104402)
        CH.RF ;CHK.RF = TRAP+4 (104404)
        CH.RC ;CHK.RC = TRAP+6 (104406)
        CH.RP ;CHK.RP = TRAP+10 (104410)
        CH.RK ;CHK.RK = TRAP+12 (104412)
        CH.TC ;CHK.TC = TRAP+14 (104414)
        CH.TH ;CHK.TH = TRAP+16 (104416)
        CH.DR ;CHK.DR = TRAP+20 (104420)
        S.TC ;SEARCH = TRAP+22 (104422)
        RW.TH ;REWIND = TRAP+24 (104424)
        CDATA ;CKDATA = TRAP+26 (104426)
        CH.TC ;ERR2 = TRAP+30 (104430)
        POTS ;STOP = TRAP+32 (104432)
        ER3 ;ERR3 = TRAP+34 (104434)
        .BEGIN ;.START = TRAP+36 (104436)
        SETBAK ;SETBAN = TRAP+40 (104440)
        NXT ;NEXT = TRAP+42 (104442)
        MAP ;MAPIT = TRAP+44 (104444)
        PATCH1 ;DUM1 = TRAP+46 (104446)
        PATCH2 ;DUM2 = TRAP+50 (104450)
ENDTAB=-2

```

```

GETMON: JSR PC,LODRES ;RESTORE THE LOADER
        TYPE ,177 ;TYPE A NULL
        RESET ;KILL THE WORLD
        MOV R0,-(6) ;PUSH R0 ON STACK
        MOV @#42,R3 ;GET THE ADDRESS
        JSR PC,(3) ;GO TO MONITOR
        240,240,240 ;SAVE ROOM FOR ACT11
        MOV (6)+,R0 ;POP STACK INTO R0
        CLR R3 ;INIT COUNT
17$: INC R3 ;COUNT
    BNE 17$ ;DOWN
    RTS PC ;RETURN

```

652	003064	005067	006446		NXT:	CLR	FLG	:SET A FLAG
653	003070	032737	040000	177570		BIT	#SW14,2#SWR	:LOOP ON BANK?
654	003076	001126				BNE	2\$:YES - GET OUT
655	003100	126700	175710			CMPB	MX,R0	:LAST PAGE?
656	003104	002450				BLT	1\$:NO - SKIP
657	003106	032737	002000	177570		BIT	#SW10,2#SWR	:INHIBIT BELL
658	003114	001002				BNE	5\$:YES!
659	003116	000004	000007			TYPE	BELL	:RING THE BELL
660	003122	005367	006406		5\$:	DEC	PSCNT	:3 YET
661	003126	001020				BNE	12\$:NO - SKIP
662	003130	005267	005402			INC	FLG	:RECALL MONITER
663	003134	013703	000042			MOV	2#42,R3	:GET MONITOR ADDRESS
664	003140	001410				BEQ	10\$:SKIP IF 0
665	003142	062767	000002	006362		ADD	#2,DEVADR	:GO TO NEXT ONE
666	003150	005777	006356			TST	2DEVADR	:END OF TABLE?
667	003154	001002				BNE	10\$:SKIP MONITOR CALL
668	003156	004767	177640			JSR	PC,GETMON	:GET THE MONITOR
669	003162	012767	000003	006344	10\$:	MOV	#3,PSCNT	:RESET COUNT
670	003170	016767	175622	175616	12\$:	MOV	MIN,MX	:RESET MX
671	003176	005767	000512			TST	MEMORY	:CHECK STATE OF FLAG
672	003202	100407				BMI	4\$:SWAB IT
673	003204	001403				BEQ	3\$:COM IT
674	003206	005067	000502			CLR	MEMORY	:INIT IT
675	003212	000405				BR	1\$:CONTINUE
676	003214	105167	000475		3\$:	COMB	MEMORY+1	:MAKE IT 177400
677	003220	000402				BR	1\$:SKIP
678	003222	000367	000466		4\$:	SWAB	MEMORY	:MAKE IT 377
679	003226	026767	175564	175560	1\$:	CMP	MIN,MX	:FIRST?
680	003234	001322				BNE	6\$:SKIP CRLF
681	003236	005737	000042			TST	2#42	:MONITOR?
682	003242	001407				BEQ	14\$:SKIP LOOPING
683	003244	005767	006266			TST	FLG	:UNDER MONITOR?
684	003250	001404				BEQ	14\$:SKIP EXIT
685	003252	005037	177776			CLR	2#PS	:CLEAR PS
686	003256	000167	176466			JMP	NEWDEV	:LOOP
687	003262	032737	000400	177570	14\$:	BIT	#SW8,2#SWR	:TRACE?
688	003270	001404				BEQ	6\$:NO - SKIP
689	003272	000004	003276			TYPE	.+2	:ASCIZ <15><12>
690	003302	005267	175506		6\$:	INC	MX	:GO TO NEXT BANK
691	003306	016737	175502	177570		MOV	MX,2#DISPLAY	:DISPLAY THE BANK IN USE
692	003314	032737	000400	177570		BIT	#SW8,2#SWR	:TYPE BANK NUMBER??
693	003322	001414				BEQ	2\$:NO!
694	003324	000004	003330			TYPE	.+2	:ASCIZ " "
695	003332	016705	175456			MOV	MX,TTY	:SET FOR TYPING
696	003336	032705	000007			BIT	#7,TTY	:MULTIPLE OF 10?
697	003342	001402				BEQ	7\$:YES
698	003344	042705	177770			BIC	#177770,TTY	:CLEAR OUT UPPER BITS
699	003350	004767	007022		7\$:	JSR	PC,PRINTS	:TYPE MX AND SUPPRESS LEADING ZEROES
700	003354	104440			2\$:	SETBANK		:SET MXC
701	003356	016704	006356			MOV	L0LIM,R4	:SET UP FOR BACKGROUND
702	003362	000002				RTI		:RETURN

```

703 003364 104440          CDATA: SETBANK                ;CHECK DATA
704 003366 005767 005322  TST      EX
705 003372 001401          BEQ      .+4
706 003374 000002          RTI
707 003376 005067 000306  CLR      DATA
708 003402 005067 000304  CLR      BIT
709 003406 016702 006326  MOV      LOLIM,R2          ;SET UP BEGINNING OF BUFFER
710 003412 004767 000022  JSR      PC,GDATA
711 003416 026712 000266  CMP      DATA,(2)
712 003422 001401          BEQ      .+4
713 003424 104434          ERR3
714 003426 005722          TST      (2)+
715 003430 026702 006306  CMP      HILIM,R2        ;CHECK FOR END
716 003434 001366          BNE      IS
717 003436 000002          RTI
718
719 003440 005767 000250  GDATA: TST      MEMORY      ;CHECK FLAG
720 003444 100430          BMI      T3X0R9          ;MM11G
721 003446 003016          BGT      TBX0R13        ;MM11F
722
723 003450 010203  T1X0R8: MOV      R2,R3          ;MM11E MEMORY
724 003452 000303          SWAB    R3
725 003454 006103          ROL      R3
726 003456 060203          ADD     R2,R3
727 003460 006003          ROR     R3
728 003462 065703 000224  ALLOR:  ADD     BIT,R3
729 003466 006003          ROR     R3
730 003470 103404          BCS     IS
731 003472 005167 000214  COM     BIT
732 003476 005167 000206  COM     DATA
733 003502 000207  IS:     RTS      PC
734
735 003504 010203  TBX0R13: MOV     R2,R3          ;MM11F MEMORY
736 003506 010205          MOV     R2,R5
737 003510 000303          SWAB    R3
738 003512 006105          ROL     R5
739 003514 006105          ROL     R5
740 003516 006105          ROL     R5
741 003520 006105          ROL     R5
742 003522 060503  T3X9:   ADD     R5,R3
743 003524 000756          BR      ALLOR
744
745 003526 010203  T3X0R9: MOV     R2,R3          ;MM11G MEMORY
746 003530 006003          ROR     R3
747 003532 010305          MOV     R3,R5
748 003534 000305          SWAB    R5
749 003536 006003          ROR     R3
750 003540 006003          ROR     R3
751 003542 000767          BR      T3X9

```

```

752 003544 005037 177776      POTS: CLR      @#177776      ;DROP PRIORITY
753 003550 005737 177570      TST      @#SWR      ;HALT ON ERROR?
754 003554 100001      BPL      .+4
755 003556 000000      HALT
756 003560 000002      RTI
757
758 003562 016705 175226      .BEGIN: MOV     MX,TTY
759 003566 000241      CLC
760 003570 006005      ROR     TTY      ; *ASH #13,TTY*
761 003572 006005      ROR     TTY      ; TO GET
762 003574 006005      ROR     TTY      ; THE THREE
763 003576 006005      ROR     TTY      ; UPPER BITS
764 003600 042705 017777      BIC     #17777,TTY ; TO MAKE
765 003604 017646 000000      MOV     @(@),-(@) ; THE ADDRESS
766 003610 017646 000000      MOV     @(@),-(@) ; GET ADDRESS
767 003614 010576 000000      MOV     TTY,@(@)  ; GET ADDRESS
768 003620 022626      CMP     (@)+,(@)+ ; LOAD MEMORY ADDRESS
769 003622 062716 000002      ADD     #2,(@)    ; RESTORE STACK
770 003626 016705 175162      MOV     MX,TTY    ; INCREMENT STACK
771 003632 006105      ROL     TTY      ; GET UPPER 2 BITS OF MXC
772 003634 042705 177717      BIC     #177717,TTY ; GET INTO POSITION
773 003640 022767 041524 006450      CMP     #TC,INPUT ; CLEAR JUNK
774 003646 001002      BNE     IS       ; CHECK FOR DTA
775 003650 052705 000400      BIS     #400,TTY  ; SKIP IF NOT
776 003654 057605 000000      BIS     @(@),TTY ; SET DTA1
777 003660 062716 000002      ADD     #2,(@)    ; GET REST OF COMMAND
778 003664 017646 000000      MOV     @(@),-(@) ; INCREMENT STACK
779 003670 017646 000000      MOV     @(@),-(@) ; GET ADDRESS
780 003674 010576 000000      MOV     TTY,@(@)  ; GET ADDRESS
781 003700 022626      CMP     (@)+,(@)+ ; LOAD MEMORY ADDRESS
782 003702 062716 000002      ADD     #2,(@)    ; RESTORE STACK
783 003706 000002      RTI             ; INCREMENT STACK
784 003710 000000      DATA: 0
785 003712 000000      BIT: 0
786 003714 000000      MEMORY: 0
787
788 003716 104442      WRADR: NEXT
789 003720 005067 177764      CLR     DATA
790 003724 005067 177762      CLR     BIT
791 003730 016702 006004      MOV     LOLIM,R2 ; SET THE STARTING ADDRESS
792 003734 004767 177500      JSR     PC,GDATA
793 003740 016722 177744      MOV     DATA,(2)+
794 003744 026702 005772      CMP     HILIM,R2 ; END?
795 003750 001371      BNE     IS
796 003752 000002      RTI
797
798 003754 104440      WRADR: SETBANK
799 003756 016702 005756      MOV     LOLIM,R2 ; SET UP BEGINNING OF BUFFER
800 003762 005122      IS:     COM     (2)+ ; COMPLIMENT IT
801 003764 026702 005752      CMP     HILIM,R2
802 003770 001374      BNE     IS
803 003772 000002      RTI             ; LOOP TIL END OF CORE

```

H02

MAINDEC-11-DZQMA-B
DZQMA8.CMB

MEMORY I/O EXERCISER
SUBROUTINES

MACY11 27(732) 26-OCT-76 16:08 PAGE 21

804	003774	032737	020000	177570	ER3:	BIT	#SW13,@SWR	;INHIBIT TYPEOUTS
805	004002	001401				BEQ	.+4	
806	004004	000002				RTI		
807	004006	022767	041524	006302		CMP	#TC,INPUT	
808	004014	001003				BNE	IS	
809	004016	012777	000001	005632		MOV	#1,@TCCS	
810	004024	000004	011032		IS:	TYPE	M7	
811	004030	010205				MOV	R2,TTY	;TYPE R2 WITH MX AS 18 BIT ADDRESS
812	004032	004767	006364			JSR	PC,PRINTA	;GO TO ADDRESS PRINTER
813	004036	000004	011053			TYPE	M8	
814	004042	016705	177642			MOV	DATA,TTY	;TYPE DATA IN OCTAL
815	004046	004767	006314			JSR	PC,PRINTR	;TYPE LEADING ZERO'S
816	004052	000004	011065			TYPE	M9	
817	004056	011205				MOV	(2),TTY	;TYPE (2) IN OCTAL
818	004060	004767	006302			JSR	PC,PRINTR	;TYPE LEADING ZERO'S
819	004064	104432				STOP		
820	004066	000002				RTI		
821								
822	004070	016705	174720		SETBAK:	MOV	MX,TTY	
823	004074	005767	174722			TST	OPTION	;WHAT AM I
824	004100	001411				BEQ	2\$;I'M NOTHING
825	004102	100405				BMI	IS	;I'M AN MX11
826	004104	000305				SWAB	TTY	; *ASH #7,TTY*
827	004106	006205				ASR	TTY	
828	004110	010577	005612			MOV	TTY,@KISAR6	
829	004114	000002				RTI		
830	004116	010577	005614		IS:	MOV	TTY,@MXC3	;LOAD THE MXC
831	004122	000002				RTI		
832	004124	000241			2\$:	CLC		; *ASH #13,TTY*
833	004126	006005				ROR	TTY	;ROTATE
834	004130	006005				ROR	TTY	;THE UPPER
835	004132	006005				ROR	TTY	;THREE BITS
836	004134	006005				ROR	TTY	;TO MAKE
837	004136	042705	017777			BIC	#17777,TTY	;CLEAR JUNK
838	004142	010567	005572			MOV	TTY,LOLIM	;ADDRESS
839	004146	062705	020000			ADD	#20000,TTY	;MAKE UPPER
840	004152	010567	005564			MOV	TTY,HILIM	;LIMIT
841	004156	000002				RTI		
842								
843								
844	004160	012777	000000	005534	MAP:	MOV	#0,@KISAR0	
845	004166	012777	077406	005530		MOV	#77406,@KISDR0	
846	004174	012777	000200	005524		MOV	#200,@KISAR6	
847	004202	012777	077406	005520		MOV	#77406,@KISDR6	
848	004210	012777	007600	005514		MOV	#7600,@KISAR7	
849	004216	012777	077406	005510		MOV	#77406,@KISDR7	
850	004224	000002				RTI		

MAINDEC-11-DZQMA-8
DZQMA8.CMBMEMORY I/O EXERCISER
RF11 DISK HANDLER

MACY11 27(732) 26-OCT-76 16:08 PAGE 22

851	004226	104404			W.RF:	CHK.RF		;CHECK FOR ERRORS
852	004230	104426				CKDATA		;CHECK CORE
853	004232	104400			TRP.RF:	CORE		;LOAD CORE WITH DATA
854	004234	012777	004266	005276		MOV	%C.RF,@RFTRAP	; 'CHECK' TRAP ADDRESS
855	004242	016777	005476	005340		MOV	WRDCNT,@RFWC	;WORD COUNT
856	004250	005077	005340			CLR	@RFDA	;DSK ADDRESS
857	004254	104436				.START		;LOAD CURRENT ADDRESS & IE!3 AND GO
858	004256	011612	000103	011604		RFCAL100!3,RFCS		
859	004264	000002				RTI		
861	004266	104404			C.RF:	CHK.RF		;CHECK FOR ERRORS
862	004270	012777	004322	005242		MOV	%R.RF,@RFTRAP	; 'READ' TRAP ADDRESS
863	004276	016777	005442	005304		MOV	WRDCNT,@RFWC	;WORD COUNT
864	004304	005077	005304			CLR	@RFDA	;DSK ADDRESS
865	004310	104436				.START		;LOAD CURRENT ADDRESS & IE!7 AND GO
866	004312	011612	000107	011604		RFCAL100!7,RFCS		
867	004320	000002				RTI		
868								
869	004322	104404			R.RF:	CHK.RF		;CHECK FOR ERRORS
870	004324	104402				COMCOR		;COMPLIMENT CORE
871	004326	012777	004226	005204		MOV	%W.RF,@RFTRAP	; 'WRITE' TRAP ADDRESS
872	004334	016777	005404	005246		MOV	WRDCNT,@RFWC	;WORD COUNT
873	004342	005077	005246			CLR	@RFDA	;DSK ADDRESS
874	004346	104436				.START		;LOAD CURRENT ADDRESS & IE!5 AND GO
875	004350	011612	000105	011604		RFCAL100!5,RFCS		
876	004356	000002				RTI		

877	004360	005067	004330			CH.RF:	CLR	EX	:CLEAR ERROR FLA^
878	004364	005777	005214				TST	2RFCS	:ANY ERRORS?
879	004370	100026					BPL	1\$:BRANCH IF NO ERRORS
880	004372	005267	004316				INC	EX	:SET ERROR FLAG
881	004376	032737	020000	177570			BIT	#SW13,2#SWR	:INHIBIT ERROR TYPEOUT
882	004404	001020					BNE	1\$:INHIBIT TYPEOUTS
883	004406	017705	005172				MOV	2RFCS,TTY	:SET TTY FOR TYPING
884	004412	000004	011151				TYPE	,RF.M1	:TYPE DEVICE MESSAGE
885	004416	000004	011012				TYPE	M3	:TYPE CS
886	004422	004767	005740				JSR	PC,PRINTR	:TYPE RFCS IN OCTAL
887	004426	000004	011022				TYPE	M4	:TYPE ER
888	004432	017705	005150				MOV	2REFER,TTY	:TYPE 2REFER IN OCTAL
889	004436	004767	005724				JSR	PC,PRINTR	:TYPE LEADING ZERO'S
890	004442	104432					STOP		:HALT ON ERROR
891	004444	000002					RTI		:RETURN
892	004446	032737	020000	177570	1\$:		BIT	#SW13,2#SWR	:INHIBIT ERROR TYPEOUT
893	004454	001054					BNE	2\$:YES!
894	004456	117767	005122	004226			MOVB	2RFCS,CHK	:GET EA BITS
895	004464	116767	174324	004221			MOVB	MX,CHK+1	:GET MX BITS
896	004472	105267	004215				INCB	CHK+1	:INCREMENT INTO EA BITS
897	004476	106067	004210				RORB	CHK	:MOVE OVER BY 1
898	004502	006067	004204				ROR	CHK	:MOVE IT
899	004506	006067	004200				ROR	CHK	:INTO
900	004512	006067	004174				ROR	CHK	:POSITION
901	004516	042767	176374	004166			BIC	#176374,CHK	:CLEAR JUNK
902	004524	126767	004162	004161			CMPB	CHK,CHK+1	:MAKE SURE EA BITS INCREMENT
903	004532	001425					BEQ	2\$:RETURN IF EQUAL
904	004534	000004	011151				TYPE	,RF.M1	:TYPE DEVICE MESSAGE
905	004540	000004	011142				TYPE	M15	:TYPE "BANK"
906	004544	016705	174244				MOV	MX,TTY	:TYPE MX IN OCTAL
907	004550	004767	005622				JSR	PC,PRINTS	:AND SUPPRESS LEADING ZERO'S
908	004554	000004	011053				TYPE	M8	:TRUE
909	004550	116705	004127				MOVB	CHK+1,TTY	:GET BYTE
910	004564	004767	005606				JSR	7,PRINTS	:TYPE TRUE EA BITS
911	004570	000004	011065				TYPE	M9	:RECEIVED
912	004574	116705	004112				MOVB	CHK,TTY	:GET BYTE
913	004600	004767	005572				JSR	7,PRINTS	:TYPE RECEIVED EA BITS
914	004604	104432					STOP		:WAIT
915	004606	000002			2\$:		RTI		:RETURN

K02

MAINDEC-11-DZQMA-B
DZQMA8.CMB

MEMORY I/O EXERCISER
RC11 DISK HANDLER

MACY11 27(732) 26-OCT-76 16:08 PAGE 24

916	004610	104406			W.RC: CHK.RC		;CHECK FOR ERRORS
917	004612	104426			CKDATA		;CHECK CORE
918	004614	104400			TRP.RC: CORE		;LOAD CORE WITH DATA
919	004616	012777	004650	004720	MOV #C.RC,@RCTRAP		;CHECK TRAP ADDRESS
920	004624	016777	005114	004770	MOV WRDCNT,@RCWC		;WORD COUNT
921	004632	005077	004770		CLR @RCDA		;DSK ADDRESS
922	004636	104436			.START		;LOAD CURRENT ADDRESS & IE!3 AND GO
923	004640	011624	000103	011616	RCCA,100!3,RCCS		
924	004646	000002			RTI		
925							
926	004650	104406			C.RC: CHK.RC		;CHECK FOR ERRORS
927	004652	012777	004704	004664	MOV #R.RC,@RCTRAP		;READ TRAP ADDRESS
928	004660	016777	005060	004734	MOV WRDCNT,@RCWC		;WORD COUNT
929	004666	005077	004734		CLR @RCDA		;DSK ADDRESS
930	004672	104436			.START		;LOAD CURRENT ADDRESS & IE!7 AND GO
931	004674	011624	000107	011616	RCCA,100!7,RCCS		
932	004702	000002			RTI		
933							
934	004704	104406			R.RC: CHK.RC		;CHECK FOR ERRORS
935	004706	104402			COMCOR		;COMPLIMENT CORE
936	004710	012777	004610	004626	MOV #W.RC,@RCTRAP		;WRITE TRAP ADDRESS
937	004716	016777	005022	004676	MOV WRDCNT,@RCWC		;WORD COUNT
938	004724	005077	004676		CLR @RCDA		;DSK ADDRESS
939	004730	104436			.START		;LOAD CURRENT ADDRESS & IE!5 AND GO
940	004732	011624	000105	011616	RCCA,100!5,RCCS		
941	004740	000002			RTI		

942	004742	005067	003746		CH.RC:	CLR	EX	:CLEAR ERROR FLAG
943	004746	005777	004644			TST	2RCCS	:ANY ERRORS?
944	004752	100026				BPL	1\$:BRANCH IF NO ERRORS
945	004754	005267	003734			INC	EX	:SET ERROR FLAG
946	004760	032737	020000	177570		BIT	#SW13,2#SWR	:INHIBIT ERROR TYPEOUT
947	004766	001020				BNE	1\$:INHIBIT TYPEOUTS
948	004770	017705	004622			MOV	2RCCS, TTY	:SET TTY FOR TYPING
949	004774	000004	011174			TYPE	,RC.M1	:TYPE DEVICE MESSAGE
950	005000	000004	011012			TYPE	M3	:TYPE CS
951	005004	004767	005356			JSR	PC,PRINTR	:TYPE RCCS IN OCTAL
952	005010	000004	011022			TYPE	M4	:TYPE ER
953	005014	017705	004600			MOV	2RCER, TTY	:TYPE 2RCER IN OCTAL
954	005020	004767	005342			JSR	PC,PRINTR	:TYPE LEADING ZERO'S
955	005024	104432				STOP		:HALT ON ERROR
956	005026	000002				RTI		:RETURN
957	005030	032737	020000	177570	1\$:	BIT	#SW13,2#SWR	:INHIBIT ERROR TYPEOUT
958	005036	001054				BNE	2\$:YES!
959	005040	117767	004552	003644		MOVB	2RCCS,CHK	:GET EA BITS
960	005046	116767	173742	003637		MOVB	MX,CHK+1	:GET MX BITS
961	005054	105267	003633			INCB	CHK+1	:INCREMENT INTO EA BITS
962	005060	106067	003626			RORB	CHK	:MOVE OVER BY 1
963	005064	006067	003622			ROR	CHK	:MOVE IT
964	005070	006067	003616			ROR	CHK	:INTO
965	005074	006067	003612			ROR	CHK	:POSITION
966	005100	042767	176374	003604		BIC	#176374,CHK	:CLEAR JUNK
967	005106	126767	003600	003577		CMPB	CHK,CHK+1	:MAKE SURE EA BITS INCREMENT
968	005114	001425				BEQ	2\$:RETURN IF EQUAL
969	005116	000004	011174			TYPE	,RC.M1	:TYPE DEVICE MESSAGE
970	005122	000004	011142			TYPE	M15	:TYPE "BANK"
971	005126	016705	173662			MOV	MX,TTY	:TYPE MX IN OCTAL
972	005132	004767	005240			JSR	PC,PRINTS	:AND SUPPRESS LEADING ZERO'S
973	005136	000004	011053			TYPE	M8	:TRUE
974	005142	116705	003545			MOVB	CHK+1,TTY	:GET BYTE
975	005146	004767	005224			JSR	7,PRINTS	:TYPE TRUE EA BITS
976	005152	000004	011065			TYPE	M9	:RECEIVED
977	005156	116705	003530			MOVB	CHK,TTY	:GET BYTE
978	005162	004767	005210			JSR	7,PRINTS	:TYPE RECEIVED EA BITS
979	005166	104432				STOP		:WAIT
980	005170	000002			2\$:	RTI		:RETURN

MAINDEC-11-DZQMA-B
DZQMA8.CMB

MEMORY I/O EXERCISER
RP11 DISK HANDLER

MACY11 27(732) 26-OCT-76 16:08 PAGE 26

981	005172	104410			W.RP:	CHK.RP				:CHECK FOR ERRORS
982	005174	104425				CKDATA				:CHECK CORE
983	005176	104400			TP.RP:	CORE				:LOAD CORE WITH DATA
984	005200	012777	005232	004342		MOV	#C.RP	TRPTRAP		: 'CHECK' TRAP ADDRESS
985	005206	016777	004532	004416		MOV	WRDCNT,	TRPWC		:WORD COUNT
986	005214	005077	004416			CLR	TRPDA			:DSK ADDRESS
987	005220	104436				.START				:LOAD CURRENT ADDRESS & IE!3 AND GO
988	005222	011634	000103	011630		RPCA,100!3,RPCS				
989	005230	000002				RTI				
990										
991	005232	104410			C.RP:	CHK.RP				:CHECK FOR ERRORS
992	005234	012777	005266	004306		MOV	#R.RP	TRPTRAP		: 'READ' TRAP ADDRESS
993	005242	016777	004476	004362		MOV	WRDCNT,	TRPWC		:WORD COUNT
994	005250	005077	004362			CLR	TRPDA			:DSK ADDRESS
995	005254	104436				.START				:LOAD CURRENT ADDRESS & IE!7 AND GO
996	005256	011634	000107	011630		RPCA,100!7,RPCS				
997	005264	000002				RTI				
998										
999	005266	104410			R.RP:	CHK.RP				:CHECK FOR ERRORS
1000	005270	104402				CONCOR				:COMPLIMENT CORE
1001	005272	012777	005172	004250		MOV	#W.RP	TRPTRAP		: 'WRITE' TRAP ADDRESS
1002	005300	016777	004440	004324		MOV	WRDCNT,	TRPWC		:WORD COUNT
1003	005306	005077	004324			CLR	TRPDA			:DSK ADDRESS
1004	005312	104436				.START				:LOAD CURRENT ADDRESS & IE!5 AND GO
1005	005314	011634	000105	011630		RPCA,100!5,RPCS				
1006	005322	000002				RTI				
1007	005324	012777	000011	004276	TRP.RP:	MOV	#11,	TRPCS		
1008	005332	105777	004272			TSTB	TRPCS			
1009	005336	100375				BPL	.-4			
1010	005340	005777	004276			TST	TRPDS			
1011	005344	100375				BPL	.-4			
1012	005346	005077	004256			CLR	TRPCS			
1013	005352	000711				BR	TP.RP			

1014	005354	005067	003334		CH.RP:	CLR	EX	; CLEAR ERROR FLAG
1015	005360	005777	004244			TST	2RPCS	; ANY ERRORS?
1016	005364	100034				BPL	IS	; BRANCH IF NO ERRORS
1017	005366	005267	003322			INC	EX	; SET ERROR FLAG
1018	005372	032737	020000	177570		BIT	#SW13,2#SWR	; INHIBIT ERROR TYPEOUT
1019	005400	001026				BNE	IS	; INHIBIT TYPEOUTS
1020	005402	017705	004222			MOV	2RPCS,TTY	; SET TTY FOR TYPING
1021	005406	000004	011242			TYPE	,RP.M1	; TYPE DEVICE MESSAGE
1022	005412	000004	011012			TYPE	M3	; TYPE CS
1023	005416	004767	004744			JSR	PC,PRINTR	; TYPE RPCS IN OCTAL
1024	005422	000004	011022			TYPE	M4	; TYPE ER
1025	005426	017705	004206			MOV	2RPER,TTY	; TYPE 2RPER IN OCTAL
1026	005432	004767	004730			JSR	PC,PRINTR	; TYPE LEADING ZERO'S
1027	005436	104432				STOP		; HALT ON ERROR
1028	005440	012777	000001	004162		MOV	#1,2RPCS	; LOAD A CLEAR
1029	005446	105777	004156			TSTB	2RPCS	; WAIT FOR
1030	005452	100375				BPL	.-4	; DONE
1031	005454	000002				RTI		; RETURN
1032	005456	032737	020000	177570	15:	BIT	#SW13,2#SWR	; INHIBIT ERROR TYPEOUT
1033	005464	001054				BNE	25	; YES!
1034	005466	117767	004136	003216		MOVB	2RPCS,CHK	; GET EA BITS
1035	005474	116767	173314	003211		MOVB	MX,CHK+1	; GET MX BITS
1036	005502	105267	003205			INCB	CHK+1	; INCREMENT INTO EA BITS
1037	005506	106067	003200			RORB	CHK	; MOVE OVER BY 1
1038	005512	006067	003174			ROR	CHK	; MOVE IT
1039	005516	006067	003170			ROR	CHK	; INTO
1040	005522	006067	003164			ROR	CHK	; POSITION
1041	005526	042767	176374	003156		BIC	#176374,CHK	; CLEAR JUNK
1042	005534	126767	003152	003151		CPB	CHK,CHK+1	; MAKE SURE EA BITS INCREMENT
1043	005542	001425				BEQ	25	; RETURN IF EQUAL
1044	005544	000004	011242			TYPE	,RP.M1	; TYPE DEVICE MESSAGE
1045	005550	000004	011142			TYPE	M15	; TYPE "BANK"
1046	005554	016705	173234			MOV	MX,TTY	; TYPE MX IN OCTAL
1047	005560	004767	004612			JSR	PC,PRINTS	; AND SUPPRESS LEADING ZERO'S
1048	005564	000004	011053			TYPE	M8	; TRUE
1049	005570	116705	003117			MOVB	CHK+1,TTY	; GET BYTE
1050	005574	004767	004576			JSR	7,PRINTS	; TYPE TRUE EA BITS
1051	005600	000004	011065			TYPE	M9	; RECEIVED
1052	005604	116705	003102			MOVB	CHK,TTY	; GET BYTE
1053	005610	004767	004562			JSR	7,PRINTS	; TYPE RECEIVED EA BITS
1054	005614	104432				STOP		; WAIT
1055	005616	000002			25:	RTI		; RETURN

1056	005620	104412			W.RK:	CHK.RK		: CHECK FOR ERRORS
1057	005622	104426				CKDATA		: CHECK CORE
1058	005624	104400			TRP.RK:	CORE		: LOAD CORE WITH DATA
1059	005626	012777	005660	003720		MOV	#C.RK, @RKTRAP	: 'CHECK' TRAP ADDRESS
1060	005634	016777	004104	004006		MOV	WRDCNT, @RKWC	: WORD COUNT
1061	005642	005077	004006			CLR	@RKDA	: DSK ADDRESS
1062	005646	104436				.START		: LOAD CURRENT ADDRESS & IE!3 AND GO
1063	005650	011652	00103	011644		RKCA, 100!3, RKCS		
1064	005656	000002				RTI		
1065								
1066	005660	104412			C.RK:	CHK.RK		: CHECK FOR ERRORS
1067	005662	012777	005714	003664		MOV	#R.RK, @RKTRAP	: 'READ' TRAP ADDRESS
1068	005670	016777	004050	003752		MOV	WRDCNT, @RKWC	: WORD COUNT
1069	005676	005077	003752			CLR	@RKDA	: DSK ADDRESS
1070	005702	104436				.START		: LOAD CURRENT ADDRESS & IE!7 AND GO
1071	005704	011652	000107	011644		RKCA, 100!7, RKCS		
1072	005712	000002				RTI		
1073								
1074	005714	104412			R.RK:	CHK.RK		: CHECK FOR ERRORS
1075	005716	104402				COMCOR		: COMPLIMENT CORE
1076	005720	012777	005620	003626		MOV	#W.RK, @RKTRAP	: 'WRITE' TRAP ADDRESS
1077	005726	016777	004012	003714		MOV	WRDCNT, @RKWC	: WORD COUNT
1078	005734	005077	003714			CLR	@RKDA	: DSK ADDRESS
1079	005740	104436				.START		: LOAD CURRENT ADDRESS & IE!5 AND GO
1080	005742	011652	000105	011644		RKCA, 100!5, RKCS		
1081	005750	000002				RTI		

1082	005752	005067	002736		CH.RK:	CLR	EX	:CLEAR ERROR FLAG
1083	005756	005777	003662			TST	ARKCS	:ANY ERRORS?
1084	005762	100034				BPL	IS	:BRANCH IF NO ERRORS
1085	005764	005267	002724			INC	EX	:SET ERROR FLAG
1086	005770	032737	020000	177570		BIT	#SW13,2#SWR	:INHIBIT ERROR TYPEOUT
1087	005776	001026				BNE	IS	:INHIBIT TYPEOUTS
1088	005000	017705	003640			MOV	ARKCS,TTY	:SET TTY FOR TYPING
1089	006004	000004	011217			TYPE	,RK.M1	:TYPE DEVICE MESSAGE
1090	006010	000004	011012			TYPE	M3	:TYPE CS
1091	006014	004767	004346			JSR	PC,PRINTR	:TYPE RKCS IN OCTAL
1092	006020	000004	011022			TYPE	M4	:TYPE ER
1093	006024	017705	003616			MOV	ARKER,TTY	:TYPE ARKER IN OCTAL
1094	006030	004767	004332			JSR	PC,PRINTR	:TYPE LEADING ZERO'S
1095	006034	104432				STOP		:HALT ON ERROR
1096	006036	012777	000001	003600		MOV	#1,ARKCS	:LOAD A CLEAR
1097	006044	105777	003574			TSTB	ARKCS	:WAIT FOR
1098	006050	100375				BPL	-4	:DONE
1099	006052	000002				RTI		:RETURN
1100	006054	032737	020000	177570	15:	BIT	#SW13,2#SWR	:INHIBIT ERROR TYPEOUT
1101	006062	001054				BNE	25	:YES!
1102	006064	117767	003554	002620		MOVB	ARKCS,CHK	:GET EA BITS
1103	006072	116767	172716	002613		MOVB	MX,CHK+1	:GET MX BITS
1104	006100	105267	002607			INCB	CHK+1	:INCREMENT INTO EA BITS
1105	006104	106067	002602			RORB	CHK	:MOVE OVER BY 1
1106	006110	006067	002576			ROR	CHK	:MOVE IT
1107	006114	006067	002572			ROR	CHK	:INTO
1108	006120	006067	002566			ROR	CHK	:POSITION
1109	006124	042767	176374	002560		BIC	#176374,CHK	:CLEAR JUNK
1110	006132	126767	002554	002553		CMPB	CHK,CHK+1	:MAKE SURE EA BITS INCREMENT
1111	006140	001425				BEQ	25	:RETURN IF EQUAL
1112	006142	000004	011217			TYPE	,RK.M1	:TYPE DEVICE MESSAGE
1113	006146	000004	011142			TYPE	M15	:TYPE "BANK"
1114	006152	016705	172636			MOV	MX,TTY	:TYPE MX IN OCTAL
1115	006156	004767	004214			JSR	PC,PRINTS	:AND SUPPRESS LEADING ZERO'S
1116	006162	000004	011053			TYPE	M8	:TRUE
1117	006166	116705	002521			MOVB	CHK+1,TTY	:GET BYTE
1118	006172	004767	004200			JSR	7,PRINTS	:TYPE TRUE EA BITS
1119	006176	000004	011065			TYPE	M9	:RECEIVED
1120	006202	116705	002504			MOVB	CHK,TTY	:GET BYTE
1121	006206	004767	004164			JSR	7,PRINTS	:TYPE RECEIVED EA BITS
1122	006212	104432				STOP		:WAIT
1123	006214	000002			25:	RTI		:RETURN

1124	006216	104414			W.TC:	CHK.TC			:CHECK FOR ERRORS
1125	006220	104426				CKDATA			:CHECK CORE
1126	006222	012777	000001	003426		MOV	#1,ATCCS		:STOP ALL TRANSPORTS
1127	006230	104400			TRP.TC:	CORE			:LOAD WORST CASE NOISE
1128	006232	104422	000010			SEARCH	,10		:SEARCH FOR BLOCK 10
1129	006236	012777	006264	003314		MOV	#R.TC,ATCTRAP		:TRAP ADDRESS
1130	006244	016777	003474	003410		MOV	WROCNT,ATCWC		:WORD COUNT
1131	006252	104436				.START			:LOAD CURRENT ADDRESS & IE!15 AND GO
1132	006254	011664	000115	011656		TCCA,100!15,TCCS			
1133	006262	000002				RTI			
1134									
1135	006264	104414			R.TC:	CHK.TC			:CHECK FOR ERROR
1136	006266	104402				COMCOR			:COMPLIMENT CORE
1137	006270	104422	000010			SEARCH	,10		:SEARCH FOR BLOCK 10
1138	006274	012777	006216	003256		MOV	#W.TC,ATCTRAP		:TRAP ADDRESS
1139	006302	016777	003436	003352		MOV	WROCNT,ATCWC		:WORD COUNT
1140	006310	104436				.START			:LOAD CURRENT ADDRESS & IE!5 AND GO
1141	006312	011664	000105	011656		TCCA,100!5,TCCS			
1142	006320	000002				RTI			

E03

MAINDEC-11-DZQMA-B
DZQMA8.CMB

MEMORY I/O EXERCISER
ERROR ROUTINE FOR TC11

MACY11 27(732) 26-OCT-76 16:08 PAGE 31

1143	006322	005067	002366		CH.TC:	CLR	EX	; CLEAR ERROR FLAG
1144	006326	005777	003324			TST	@TCCS	; ANY ERRORS?
1145	006332	100034				BPL	IS	; BRANCH IF NO ERRORS
1146	006334	005267	002354			INC	EX	; SET ERROR FLAG
1147	006340	032737	020000	177570		BIT	#SW13,@SWR	; INHIBIT ERROR TYPEOUT
1148	006346	001026				BNE	IS	; INHIBIT TYPEOUTS
1149	006350	017705	003302			MOV	@TCCS,TTY	; SET TTY FOR TYPING
1150	006354	000004	011265			TYPE	,TC.M1	; TYPE DEVICE MESSAGE
1151	006360	000004	011012			TYPE	M3	; TYPE CS
1152	006364	004767	003776			JSR	PC,PRINTR	; TYPE TCCS IN OCTAL
1153	006370	000004	011022			TYPE	M4	; TYPE ER
1154	006374	017705	003260			MOV	@TCER,TTY	; TYPE @TCER IN OCTAL
1155	006400	004767	003762			JSR	PC,PRINTR	; TYPE LEADING ZERO'S
1156	006404	104432				STOP		; HALT ON ERROR
1157	006406	012777	000001	003242		MOV	#,@TCCS	; LOAD A CLEAR
1158	006414	105777	003236			TSTB	@TCCS	; WAIT FOR
1159	006420	100375				BPL	.-4	; DONE
1160	006422	000002				RTI		; RETURN
1161	006424	032737	020000	177570	15:	BIT	#SW13,@SWR	; INHIBIT ERROR TYPEOUT
1162	006432	001054				BNE	25	; YES!
1163	006434	117767	003216	002250		MOVB	@TCCS,CHK	; GET EA BITS
1164	006442	116767	172346	002243		MOVB	MX,CHK+1	; GET MX BITS
1165	006450	105267	002237			INCB	CHK+1	; INCREMENT INTO EA BITS
1166	006454	106067	002232			RORB	CHK	; MOVE OVER BY 1
1167	006460	006067	002226			ROR	CHK	; MOVE IT
1168	006464	006067	002222			ROR	CHK	; INTO
1169	006470	006067	002216			ROR	CHK	; POSITION
1170	006474	042767	176374	002210		BIC	#176374,CHK	; CLEAR JUNK
1171	006502	126767	002204	002203		CMPB	CHK,CHK+1	; MAKE SURE EA BITS INCREMENT
1172	006510	001425				BEQ	25	; RETURN IF EQUAL
1173	006512	000004	011265			TYPE	,TC.M1	; TYPE DEVICE MESSAGE
1174	006516	000004	011142			TYPE	M15	; TYPE "BANK"
1175	006522	016705	172266			MOV	MX,TTY	; TYPE MX IN OCTAL
1176	006526	004767	003644			JSR	PC,PRINTS	; AND SUPPRESS LEADING ZERO'S
1177	006532	000004	011053			TYPE	M8	; TRUE
1178	006536	116705	002151			MOVB	CHK+1,TTY	; GET BYTE
1179	006542	004767	003630			JSR	7,PRINTS	; TYPE TRUE EA BITS
1180	006546	000004	011065			TYPE	M9	; RECEIVED
1181	006552	116705	002134			MOVB	CHK,TTY	; GET BYTE
1182	006556	004767	003614			JSR	7,PRINTS	; TYPE RECEIVED EA BITS
1183	006562	104432				STOP		; WAIT
1184	006564	000002			25:	RTI		; RETURN

1185	006566	017667	000000	000330	S.TC:	MOV	@(6),TC.TA	
1186	006574	062716	000002			ADD	#2,(6)	
1187	006600	005767	000320			TST	TC.TA	; FOWARD OR REVERSE
1188	006604	100064				BPL	FOWSER	; LEAVE IT IN REVERSE
1189	006606	005467	000312			NEG	TC.TA	; GET PLUS BLOCK NUMBER
1190	006612	012777	006652	002740		MOV	#SR1,@TCTRAP	; TRAP ADDRESS
1191	006620	016767	000300	000274		MOV	TC.TA,WANT	
1192	006626	062767	000002	000266		ADD	#2,WANT	; 2 PAST BLOCK
1193	006634	012667	000270			MOV	(6)+,RETURN	; SAVE RETURN ADDRESS
1194	006640	005726				TST	(6)+	
1195	006642	012777	000503	003006	RETI:	MOV	#503,@TCCS	; SEARCH FOWARD
1196	006650	000002				RTI		
1197								
1198	006652	005777	003000		SR1:	TST	@TCCS	; CHECK FOR ERROR
1199	006656	100005				BPL	SR1A	; NO ERRORS
1200	006660	005777	002774			TST	@TCER	; CHECK FOR END OF TAPE
1201	006664	100401				BMI	.+4	
1202	006666	104434				ERR3		
1203	006670	000405				BR	SF1	
1204								
1205	006672	026777	000224	002766	SR1A:	CMP	WANT,@TCDB	; ON BLOCK?
1206	006700	002401				BLT	SF1	; GO FOWARD
1207	006702	000757				BR	RETI	; CONTINUE AS YOU WERE
1208								
1209	006704	012777	006722	002646	SF1:	MOV	#SF2,@TCTRAP	; TRAP ADDRESS
1210	006712	012777	004503	002736	RETR:	MOV	#4503,@TCCS	; SEARCH REVERSE
1211	006720	000002				RTI		
1212								
1213	006722	005777	002730		SF2:	TST	@TCCS	; ERROR?
1214	006726	100005				BPL	SF1A	; NO ERRORS
1215	006730	005777	002724			TST	@TCER	; END OF TAPE?
1216	006734	100401				BMI	.+4	
1217	006736	104430				ERR2		
1218	006740	000740				BR	RETI	; NO
1219								; SWITCH DIRECTIONS
1220	006742	026777	000156	002716	SF1A:	CMP	TC.TA,@TCDB	; RIGHT BLOCK?
1221	006750	001360				BNE	RETR	; NO
1222	006752	000177	000152			JMP	@RETURN	


```

1223 006756 012777 007016 002574 FOWSER: MOV #SC1,@TCRAMP ;TRAP ADDRESS
1224 006764 016767 000134 000130 MOV TC.TA,WANT
1225 006772 162767 000002 000122 SUB #2,WANT ;2 PAST BLOCK
1226 007000 012667 000124 MOV (6)+,RETURN ;SAVE RETURN ADDRESS
1227 007004 005726 TST (6)+
1228 007006 012777 004503 002642 RETI1: MOV #4503,@TCCS ;SEARCH REVERSE
1229 007014 000002 RTI
1230
1231 007016 005777 002634 SC1: TST @TCCS ;CHECK FOR ERROR
1232 007022 100005 BPL SC1A ;NO ERRORS
1233 007024 005777 002630 TST @TCER ;CHECK FOR END OF TAPE
1234 007030 100401 BMI .+4
1235 007032 104434 ERR3
1236 007034 000405 BR SCF1
1237
1238 007036 026777 000060 002622 SC1A: CMP WANT,@TCDB ;ON BLOCK?
1239 007044 003001 BGT SCF1 ;GO REVERSE
1240 007046 000757 BR RETI1 ;CONTINUE AS YOU WERE
1241
1242 007050 012777 007066 002502 SCF1: MOV #SCF2,@TCRAMP ;TRAP ADDRESS
1243 007056 012777 000503 002572 RETR1: MOV #503,@TCCS ;SEARCH FORWARD
1244 007064 000002 RTI
1245
1246 007066 005777 002564 SCF2: TST @TCCS ;ERROR?
1247 007072 100005 BPL SCF1A ;NO ERRORS
1248 007074 005777 002560 TST @TCER ;END OF TAPE?
1249 007100 100401 BMI .+4
1250 007102 104430 ERR2
1251 007104 000740 BR RETI1 ;NO
;SWITCH DIRECTIONS
1252
1253 007106 026777 000012 002552 SCF1A: CMP TC.TA,@TCDB ;RIGHT BLOCK?
1254 007114 001360 BNE RETR1 ;NO
1255 007116 000177 000006 JMP @RETURN
1256
1257 007122 000000 WANT: 0
1258 007124 000000 TC.TA: 0
1259 007126 000000 TC.WC: 0
1260 007130 000000 RETURN: 0

```

1261	007132	104416			W.TM:	CHK.TM			;CHECK FOR ERROR
1262	007134	104426				CKDATA			
1263	007136	104400			TRP.TM:	CORE			;SET UP CORE
1264	007140	104424				REWIND			
1265	007142	012777	007224	002420		MOV	#R.TM,@TMTRAP		;TRAP ADDRESS
1266	007150	016777	002570	002516		MOV	WRDCNT,@TMBC		;BYTE COUNT
1267	007156	006377	002512			ASL	@TMBC		;MAKE BYTE COUNT
1268	007162	104436				.START			;LOAD CURRENT ADDRESS & IE!60005 AND GO
1269	007164	011676	060105	011670		TMCA,100!60005,TMCS			
1270	007172	000002				RTI			
1271									
1272	007174	012777	007220	002366	RW.TM:	MOV	#X.TM,@TMTRAP		;TRAP ADDRESS
1273	007202	012667	000056			MOV	(6)+,TMRET		;SAVE ADDRESS
1274	007206	005726				TST	(6)+		;RESTORE STACK
1275	007210	012777	060117	002452		MOV	#60117,@TMCS		;REWIND
1276	007216	000002				RTI			
1277									
1278	007220	000177	000040		X.TM:	JMP	@TMRET		
1279									
1280	007224	104416			R.TM:	CHK.TM			;ANY ERRORS?
1281	007226	104402				COMCOR			;COMPLIMENT CORE
1282	007230	104424				REWIND			
1283	007232	012777	007132	002330		MOV	#W.TM,@TMTRAP		;TRAP ADDRESS
1294	007240	016777	002500	002426		MOV	WRDCNT,@TMBC		;BYTE COUNT
1285	007246	006377	002422			ASL	@TMBC		;MAKE BYTE COUNT
1286	007252	104436				.START			;LOAD CURRENT ADDRESS & IE!60003 AND GO
1287	007254	011676	060103	011670		TMCA,100!60003,TMCS			
1288	007262	000002				RTI			
1289									
1290	007264	000000			TMRET:	0			

1291	007266	005067	001422		CH.TM:	CLR	EX	:CLEAR ERROR FLAG
1292	007272	005777	002372			TST	@TMCS	:ANY ERRORS?
1293	007276	100026				BPL	IS	:BRANCH IF NO ERRORS
1294	007300	005267	001410			INC	EX	:SET ERROR FLAG
1295	007304	032737	020000	177570		BIT	#SW13,@#SWR	:INHIBIT ERROR TIMEOUT
1296	007312	001020				BNE	IS	:INHIBIT TIMEOUTS
1297	007314	017705	002350			MOV	@TMCS,TTY	:SET TTY FOR TYPING
1298	007320	000004	011306			TYPE	,TM.M1	:TYPE DEVICE MESSAGE
1299	007324	000004	011012			TYPE	M3	:TYPE CS
1300	007330	004767	003032			JSR	PC,PRINTR	:TYPE TMCS IN OCTAL
1301	007334	000004	011022			TYPE	M4	:TYPE ER
1302	007340	017705	002326			MOV	@TMER,TTY	:TYPE @TMER IN OCTAL
1303	007344	004767	003016			JSR	PC,PRINTR	:TYPE LEADING ZERO'S
1304	007350	104432				STOP		:HALT ON ERROR
1305	007352	000002				RTI		:RETURN
1306	007354	032737	020000	177570	IS:	BIT	#SW13,@#SWR	:INHIBIT ERROR TIMEOUT
1307	007362	001054				BNE	2\$:YES!
1308	007364	117767	002300	001320		MOVB	@TMCS,CHK	:GET EA BITS
1309	007372	116767	171416	001313		MOVB	MX,CHK+1	:GET MX BITS
1310	007400	105267	001307			INCB	CHK+1	:INCREMENT INTO EA BITS
1311	007404	106067	001302			RORB	CHK	:MOVE OVER BY 1
1312	007410	006067	001276			ROR	CHK	:MOVE IT
1313	007414	006067	001272			ROR	CHK	:INTO
1314	007420	006067	001266			ROR	CHK	:POSITION
1315	007424	042767	176374	001260		BIC	#176374,CHK	:CLEAR JUNK
1316	007432	126767	001254	001253		CMPB	CHK,CHK+1	:MAKE SURE EA BITS INCREMENT
1317	007440	001425				BEQ	2\$:RETURN IF EQUAL
1318	007442	000004	011306			TYPE	,TM.M1	:TYPE DEVICE MESSAGE
1319	007446	000004	011142			TYPE	,M15	:TYPE "BANK"
1320	007452	016705	171336			MOV	MX,TTY	:TYPE MX IN OCTAL
1321	007456	004767	002714			JSR	PC,PRINTS	:AND SUPPRESS LEADING ZERO'S
1322	007462	000004	011053			TYPE	M8	:TRUE
1323	007466	116705	001221			MOVB	CHK+1,TTY	:GET BYTE
1324	007472	004767	002700			JSR	7,PRINTS	:TYPE TRUE EA BITS
1325	007476	000004	011065			TYPE	M9	:RECEIVED
1326	007502	116705	001204			MOVB	CHK,TTY	:GET BYTE
1327	007506	004767	002664			JSR	7,PRINTS	:TYPE RECEIVED EA BITS
1328	007512	104432				STOP		:WAIT
1329	007514	000002			2\$:	RTI		:RETURN

```

1330                                     ;DM11 PRIMING ROUTINE. THIS ROUTINE SETS UP THE DM11 TO TRANSMIT
1331                                     ;A 400(8) BINARY COUNT PATTERN ON ALL 16 LINES.
1332
1333 007516 104442 TRP.DM: NEXT
1334 007520 016705 171270 MOV MX,TTY ;GET MX
1335 007524 006005 ROR TTY ;*ASH #13.,TTY*
1336 007526 006005 ROR TTY
1337 007530 006005 ROR TTY
1338 007532 006005 ROR TTY
1339 007534 042705 017777 BIC #17777,TTY ;CLEAR JUNK
1340 007540 010567 000410 MOV TTY,DMCAT ;ADDRESS OF CAT
1341 007544 016767 000404 000404 15: MOV DMCAT,DMWCT ;SET UP WORD COUNT
1342 007552 062767 000040 000376 ADD #32,DMWCT
1343 007560 016767 000372 000372 MOV DMWCT,DMBAT ;SET BASE ADDRESS
1344 007566 062767 000040 000364 ADD #32,DMBAT
1345 007574 016767 000360 000360 MOV DMBAT,TJMTAB ;SET UP TUMBLE TABLE
1346 007602 062767 000100 000352 ADD #64,TJMTAB
1347 007610 016767 000346 000346 MOV TJMTAB,BINCNT ;SET UP BINARY COUNT
1348 007616 062767 000200 000340 ADD #128,BINCNT
1349 007624 010246 MOV R2,-(6) ;SAVE R2 & R1
1350 007626 010146 MOV R1,-(6) ;ON THE STACK
1351 007630 016702 002104 MOV LO,IM,R2
1352 007634 012701 000020 MOV #16,R1
1353 007640 016722 000320 DMPRIA: MOV BINCNT,(2)+ ;LOAD CURRENT ADDRESS TABLE
1354 007644 005301 DEC R1
1355 007646 001374 BNE DMPRIA
1356 007650 012701 000020 MOV #16,R1
1357 007654 012722 177400 DMPRIB: MOV #-400,(2)+ ;LOAD WORD COUNT TABLE
1358 007660 005301 DEC R1
1359 007662 001374 BNE DMPRIB
1360 007664 062702 000076 ADD #62.,R2 ;BUMP TO TJMTAB-2
1361 007670 005022 CLR (2)+
1362 007672 012701 000100 DMPRIC: MOV #64.,R1
1363 007676 005022 CLR (2)+ ;CLEAR TUMBLE TABLE
1364 007700 005301 DEC R1
1365 007702 001375 BNE DMPRIC
1366 007704 005001 CLR R1

```

```

1367 007706 110122          DMPRID: MOVB    R1,(2)+      ;GENERATE BINARY COUNT
1368 007710 105201          INCB    R1                ;THIS WILL BE THE TRANSMITTED DATA
1369 007712 001375          BNE     DMPRID
1370 007714 005077 001760    CLR     @DMCS
1371 007720 005077 001756    CLR     @DMACT
1372 007724 005077 001754    CLR     @DMBRK
1373 007730 005067 000206    CLR     DMTDAT
1374 007734 016777 000214 001744    MOV     DMCAT,@DMADR
1375 007742 016767 001772 000174    MOV     LOLIM,DMPTR
1376 007750 062767 000200 000166    ADD     #200,DMPTR
1377 007756 016767 000162 000162    MOV     DMPTR,ENDPTR
1378 007764 062767 000200 000154    ADD     #200,ENDPTR
1379 007772 016701 001576    MOV     DMRVEC,R1
1380 007776 012721 010166    MOV     #DMRINT,(1)+
1381 010002 012721 000240    MOV     #240,(1)+
1382 010006 012721 010064    MOV     #DMTINT,(1)+
1383 010012 012711 000240    MOV     #240,(1)
1384 010016 016705 170772    MOV     MX,TTY
1385 010022 006305          ASL     TTY
1386 010024 042705 177717    BIC     #177717,TTY
1387 010030 010577 001644    MOV     TTY,@DMCS
1388 010034 052777 010105 001636    BIS     #10105,@DMCS
1389 010042 016777 000102 001632    MOV     DMBAIM,@DMACT ;START TRANSMITTING
1390 010050 012601          MOV     (6)+,R1
1391 010052 012602          MOV     (6)+,R2
1392 010054 005037 177776    CLR     @#177776      ;CLEAR PS
1393 010060 000167 172344    JMP     WAIT
1394
1395          ;DM11 TRANSMITTER INTERRUPT SERVICE ROUTINE
1396
1397 010064 032777 060000 001606  DMTINT: BIT     #60000,@DMCS ;TEST FOR ERROR FLAGS
1398 010072 001414          BEQ     DMTINC         ;BRANCH IF NO ERROR FLAGS
1399 010074 000004 011327          TYPE   ,DM.MI
1400 010100 000004 011012          TYPE   M3
1401 010104 017705 001570          MOV     @DMCS,TTY     ;TYPE @DMCS IN OCTAL
1402 010110 004767 002252          JSR     PC,PRINTR     ;TYPE LEADING ZERO'S
1403 010114 104432          STOP
1404 010116 042777 060000 001554  DMTINC: BIC     #60000,@DMCS ;CLEAR ERROR FLAGS
1405 010124 005777 001550          TST     @DMCS        ;TEST FOR READY FLAG
1406 010130 100003          BPL     IS
1407 010132 042777 100000 001540  IS:    BIC     #100000,@DMCS ;CLEAR READY
1408 010140 000002          RTI                ;EXIT
1409
1410
1411 010142 000000          DMTDAT: 0            ;STORAGE LOCATION FOR LAST TRANSMITTED CHAR
1412 010144 000000          DMPTR: 0            ;DM11 SOFTWARE TUMBLE TABLE POINTER
1413 010146 000000          ENDPTR: 0
1414 010150 177777          DMBAIM: 177777
1415 010152 177400          MASK: 177400
1416 010154 000000          DMCAT: 0
1417 010156 000000          DMWCT: 0
1418 010160 000000          DMBAT: 0
1419 010162 000000          TUMTAB: 0
1420 010164 000000          BINCNT: 0
;INITIALIZE TO TRANSMIT ON ALL LINES.

```

```

1421                                     ;DM11 RECEIVER INTERRUPT SERVICE ROUTINE
1422
1423 010166 010246 DMRINT: MOV      R2, -(6)
1424 010170 016702 177750      MOV      DMPTR, R2
1425 010174 104440      SETBANK
1426 010176 005712      TST      (2)                ;TEST FOR VALID DATA ENTRY
1427 010200 100446      BMI      DMRINC           ;BRANCH IF VALID DATA ENTRY
1428 010202 000004 011327      TYPE    ,DM.M1
1429 010206 000004 010212      TYPE    ,.+2              ;.ASCIZ " NO DATA RECEIVED"
1430 010236 104432      STOP
1431 010240 000457      BR       DMRINX           ;GO TO EXIT
1432
1433 010242 005742 DMRINB: TST      -(2)        ;CHECK LAST ENTRY
1434 010244 001421      BEQ     1$              ;OK IF 0
1435 010246 000004 011327      TYPE    ,DM.M1
1436 010252 000004 010256      TYPE    ,.+2              ;.ASCIZ " TABLE OVERFLOW"
1437 010300 012602      MOV     (6)+, R2        ;RESTORE R2
1438 010302 022626      CMP     (6)+, (6)+     ;CLEAR RETURN
1439 010304 000167 177206      JMP     TRP.DM         ;LOOP
1440 010310 005722      1$: TST      (2)+        ;MOD POINTER
1441 010312 005712      TST      (2)          ;VALID DATA ENTRY?
1442 010314 100031      BPL     DMRINX        ;EXIT IF NO MORE ENTRIES
1443 010316 032712 017000 DMRINC: BIT     #17000, (2) ;DATA ON LINE 0?
1444 010322 001013      BNE     DMRIND        ;BRANCH IF DATA RECEIVED ON OTHER THAN LINE 0
1445 010324 146712 177622      BICB   MASK, (2)     ;CLEAR NON TRANSMITTED BITS
1446 010330 016767 177606 173352      MOV     DMTDAT, DATA
1447 010336 121267 173346      CMPB   (2), DATA    ;COMPARE RECEIVED & TRANSMITTED DATA
1448 010342 001401      BEQ     1$
1449 010344 104434      ERR3
1450 010346 105267 177570      1$: INCB   DMTDAT      ;FORM WHAT NEXT RECEIVED CHAR. SHOULD BE
1451 010352 005022 DMRIND: CLR     (2)+     ;CLEAR TUMBLE TABLE ENTRY
1452 010354 020267 177566      CMP     R2, ENDPTR   ;IS POINTER AT THE END OF THE TABLE
1453 010360 001004      BNE     1$
1454 010362 016702 001352      MOV     LOLIM, R2
1455 010366 062702 000200      ADD     #200, R2
1456 010372 010267 177546      1$: MOV     R2, DMPTR   ;RESTORE POINTER
1457 010374 000721      BR      DMRINB      ;LOOK AT NEXT ENTRY
1458
1459 010400 012602 DMRINX: MOV     (6)+, R2   ;RESTORE R0
1460 010402 005777 001274      TST    @DMACT
1461 010406 001003      BNE    BICX
1462 010410 022626      CMP    (6)+, (6)+
1463 010412 000167 177100      JMP    TRP.DM
1464
1465 010416 042777 000200 001254 BICX: BIC     #200, @DMCS ;CLEAR DONE FLAG
1466 010424 000002      RTI

```

1467	010426	104440			SET.DR:	SETBANK		
1468	010430	104420				CRK.DR		
1469	010432	005767	000256			TST	EX	: ANY ERRORS
1470	010436	001026				BNE	TRP.DR	: YES
1471	010440	016702	001274			MOV	L0LIM,R2	: SET ADDRESS
1472	010444	012767	000001	173236		MOV	#1,DATA	: SET DATA
1473	010452	026712	173232		NX.DR:	CMP	DATA,(2)	: CHECK FIRST WORD
1474	010456	001401				BEQ	.+4	: SAME?
1475	010460	104434				ERR3		: NO
1476	010462	062702	000002			ADD	#2,R2	: INC ADDRESS
1477	010466	026712	173216			CMP	DATA,(2)	: SECOND WORD OK?
1478	010472	001401				BEQ	.+4	
1479	010474	104434				ERR3		: NO
1480	010476	062702	000002			ADD	#2,R2	: UP IT
1481	010502	005267	173202			INC	DATA	: GET NEXT DATA
1482	010506	026702	001230			CMP	HILIM,R2	: END?
1483	010512	001357				BNE	NX.DR	: NO?
1484	010514	104442			TRP.DR:	NEXT		
1485	010516	005037	177776			CLR	@#177776	: CLR PS
1486	010522	016702	001212			MOV	L0LIM,R2	: SET ADDRESS
1487	010526	012767	000001	173154		MOV	#1,DATA	: SET DATA
1488	010534	016722	173150		NX1.DR:	MOV	DATA,(2)+	: SET UP DATA
1489	010540	005167	173144			COM	DATA	: - NEXT DATA
1490	010544	016722	173140			MOV	DATA,(2)+	: LOAD IT
1491	010550	005467	173134			NEG	DATA	: MAKE IT POSITIVE
1492	010554	026702	001162			CMP	HILIM,R2	: END?
1493	010560	001365				BNE	NX1.DR	: NO?
1494	010562	012737	000240	177776		MOV	#240,@#177776	: LOCK AGAIN
1495	010570	012777	010426	000766		MOV	#SET.DR,@DRTRAP	: SET VECTOR
1496	010576	016777	001142	001104		MOV	WDRCNT,@DRWC	: SET WORD COUNT
1497	010604	104436				.START		: LOAD CURRENT ADDRESS & IE!10001 AND GO
1498	010606	011712	010101	011714		DRCA,100!10001,DRCS		
1499	010614	000002				RTI		

1500	010616	005067	000072		CH.DR:	CLR	EX	; CLEAR ERROR FLAG
1501	010622	005777	001066			TST	DRCS	; ANY ERRORS?
1502	010626	100030				BPL	IS	; BRANCH IF NO ERRORS
1503	010630	005777	001054			TST	DRWC	; IS IT BANK 7?
1504	010634	001425				BEQ	IS	; YES - GET OUT
1505	010636	005267	000052			INC	EX	; SET ERROR FLAG
1506	010642	032737	020000	177570		BIT	#SW13,@#SWR	; INHIBIT ERROR TYPEOUT
1507	010650	001017				BNE	IS	; INHIBIT TYPEOUTS
1508	010652	017705	001036			MOV	DRCS, TTY	; SET TTY FOR TYPING
1509	010656	000004	011345			TYPE	, DR.M1	; TYPE DEVICE MESSAGE
1510	010662	000004	011012			TYPE	M3	; TYPE CS
1511	010666	004767	001474			JSR	PC, PRINTR	; TYPE DRCS IN OCTAL
1512	010672	000004	011022			TYPE	M4	; TYPE ER
1513	010676	017705	001014			MOV	DRER, TTY	; TYPE DRER IN OCTAL
1514	010702	004767	001460			JSR	PC, PRINTR	; TYPE LEADING ZERO'S
1515	010706	104432				STOP		; HALT ON ERROR
1516	010710	000002			IS:	RTI		; RETURN

1517	010712	000000			CHK:	0	
1518	010714	000000			EX:	0	
1519	010716	005015	041522	030461	M1:	.ASCII	<15><12>"RC11,RP11,RF11,RK11,TH11,TC11,DR11B, OR DM11 AND "
1520	010724	051054	030520	026061			
1521	010732	043122	030461	051054			
1522	010740	030513	026061	046524			
1523	010746	030461	052054	030503			
1524	010754	026061	051104	030461			
1525	010762	026102	047440	020122			
1526	010770	046504	030461	040440			
1527	010776	042116	040				
1528	011001	074	042522	052524		.ASCIZ	"<RETURN>"
1529	011006	047122	000076				
1530	011012	020040	051503	036440	M3:	.ASCIZ	" CS = "
1531	011020	000040					
1532	011022	020040	051105	036440	M4:	.ASCIZ	" ER = "
1533	011030	000040					
1534	011032	005015	040504	040524	M7:	.ASCIZ	<15><12>"DATA ERROR AT "
1535	011040	042440	051122	051117			
1536	011046	040440	020124	000			
1537	011053	040	052040	052522	M8:	.ASCIZ	" TRUE = "
1538	011060	020105	020075	000			
1539	011065	040	051040	041505	M9:	.ASCIZ	" RECEIVED = "
1540	011072	044505	042526	020104			
1541	011100	020075	000				
1542	011103	114	053517	051105	M11:	.ASCIZ	"LOWER BANK = "
1543	011110	041040	047101	020113			
1544	011116	020075	000				
1545	011121	125	050120	051105	M12:	.ASCIZ	"UPPER BANK = "
1546	011126	041040	047101	020113			
1547	011134	020075	000				
1548	011137	015	000012		M13:	.ASCIZ	<15><12>
1549	011142	041040	047101	020113	M15:	.ASCIZ	" BANK "
1550	011150	000					
1551							
1552	011151	015	051012	030506	RF.M1:	.ASCIZ	<15><12>"RF11 DISK ERROR:"
1553	011156	020061	044504	045523			
1554	011164	042440	051122	051117			
1555	011172	000072					
1556	011174	005015	041522	030461	RC.M1:	.ASCIZ	<15><12>"RC11 DISK ERROR:"
1557	011202	042040	051511	020113			
1558	011210	051105	047522	035122			
1559	011216	000					
1560	011217	015	051012	030513	RK.M1:	.ASCIZ	<15><12>"RK11 DISK ERROR:"
1561	011224	020061	044504	045523			
1562	011232	042440	051122	051117			
1563	011240	000072					
1564	011242	005015	050122	030461	RP.M1:	.ASCIZ	<15><12>"RP11 DISK ERROR:"
1565	011250	042040	051511	020113			
1566	011256	051105	047522	035122			
1567	011264	000					
1568	011265	015	042012	041505	TC.M1:	.ASCIZ	<15><12>"DECTAPE ERROR:"
1569	011272	040524	042520	042440			
1570	011300	051122	051117	000072			
1571	011306	005015	040515	052107	TH.M1:	.ASCIZ	<15><12>"MAGTAPE ERROR:"
1572	011314	050101	020105	051105			

1573	011322	047522	035122	000		
1574	011327	015	042012	030515	DM.M1:	.ASCIZ <15><12>"DM11 ERROR:"
1575	011334	020061	051105	047522		
1576	011342	035122	000			
1577	011345	015	042012	030522	DR.M1:	.ASCIZ <15><12>"DR11 ERROR:"
1578	011352	020061	051105	047522		
1579	011360	035122	000			
1580	011363	015	005012	052113	SEGA:	.ASCIZ <15><12><12>"KT11 ERROR:"
1581	011370	030461	042440	051122		
1582	011376	051117	000072			
1583	011402	005015	051123	020060	SEG1:	.ASCIZ <15><12>"SR0 = "
1584	011410	020075	000			
1585	011413	015	051412	030522	SEG2:	.ASCIZ <15><12>"SR1 = "
1586	011420	036440	000040			
1587	011424	005015	051123	020062	SEG3:	.ASCIZ <15><12>"SR2 = "
1588	011432	020075	000			
1589	011435	077	005015	000	QUES:	.ASCIZ "?"<15><12>
1590	011441	040	044527	044124	WITHSG:	.ASCIZ " WITH KT11"
1591	011446	045440	030524	000061		
1592	011454	053440	052111	020110	WITHMX:	.ASCIZ " WITH MX11"
1593	011462	054115	030461	000		
1594		011470				
1595	011470	041522	043122	045522	DEVTAB:	.ASCII "RCRFRKRPTCTMDRDM"
1596	011476	050122	041524	046524		
1597	011504	051104	046504			
1598	011510	000000				
1599	011512	177777	177777	000002	TAB1:	0 -1,-1,2,3,-1,4,-1,-1
1600	011520	000003	177777	000004		
1601	011526	177777	177777			
1602	011532	000000			DEVAOR:	0
1603	011534	000003			PSCNT:	3
1604	011536	000000			FLG:	0
1605						
1606	011540	000204	000206		RFTRAP:	204,206
1607	011544	000210	000212		RCTRAP:	210,212
1608	011550	000254	000256		RPTRAP:	254,256
1609	011554	000220	000222		RKTRAP:	220,222
1610	011560	000214	000216		TCTRAP:	214,216
1611	011564	000124	000126		DRTRAP:	124,126
1612	011570	000224	000226		TMTRAP:	224,226
1613	011574	000330	000332		DMRVEC:	330,332
1614	011600	000250	000252		SEGVEC:	250,252

1615	011604	177460	RFCS:	177460	;RF11 DISK (256K)
1616	011606	177470	RFER:	177470	
1617	011610	177462	RFCW:	177462	
1618	011612	177464	RFCA:	177464	
1619	011614	177466	RFDA:	177466	
1620	011616	177446	RCCS:	177446	;RC11 DISK (64K)
1621	011620	177444	RCER:	177444	
1622	011622	177450	RCWC:	177450	
1623	011624	177452	RCCA:	177452	
1624	011626	177442	RCDA:	177442	
1625	011630	176714	RPCS:	176714	;RP11 DISK (PACK)
1626	011632	176716	RPWC:	176716	
1627	011634	176720	RPCA:	176720	
1628	011636	176724	RPOA:	176724	
1629	011640	176712	RPER:	176712	
1630	011642	176710	RPOS:	176710	
1631	011644	177404	RKCS:	177404	;RK11 DISK (CARTRIDGE)
1632	011646	177402	RKER:	177402	
1633	011650	177406	RKWC:	177406	
1634	011652	177410	RKCA:	177410	
1635	011654	177412	RKDA:	177412	
1636	011656	177342	TCCS:	177342	;TC11 DECTAPE
1637	011660	177340	TCER:	177340	
1638	011662	177344	TCWC:	177344	
1639	011664	177346	TCCA:	177346	
1640	011666	177350	TCDB:	177350	
1641	011670	172522	TMCS:	172522	;TM11 MAGTAPE
1642	011672	172520	TMER:	172520	
1643	011674	172524	TMBC:	172524	
1644	011676	172526	TMCA:	172526	
1645	011700	175000	DMCS:	175000	;DM11 ASYNCHRONOUS MODEM
1646	011702	175002	DMACT:	175002	
1647	011704	175004	DMBRK:	175004	
1648	011706	175006	DMADR:	175006	
1649	011710	172410	DRWC:	172410	;DR11
1650	011712	172412	DRCA:	172412	
1651	011714	172414	DRCS:	172414	
1652	011716		DRER:		
1653	011716	172416	DRDB:	172416	
1654	011720	177572	SRO:	177572	;KT11 - MEMORY MANAGEMENT
1655	011722	172340	KISAR0:	172340	
1656	011724	172300	KISDR0:	172300	
1657	011726	172354	KISAR6:	172354	
1658	011730	172314	KISDR6:	172314	
1659	011732	172356	KISAR7:	172356	
1660	011734	172316	KISDR7:	172316	
1661	011736	177604	MXC3:	177604	;MX11
1662					
1663	011740	140000	LOLIM:	140000	;BEGINNING OF BANK
1664	011742	160000	HILIM:	160000	;END OF BANK
1665	011744	170000	WRDCNT:	-10000	;WORD COUNT

```

1666          ;          SLUADR          GET AND RESTORE THE LOADER
1667
1668          ; THESE ROUTINES FIRST FIND THE LOADER (TOP OF HIGHEST
1669          ; 4K BANK IN 28K) AND SAVE IT AT TAG "ENDP:". ENTRY FOR
1670          ; THIS ROUTINE IS "JSR PC,LOOGET".
1671
1672          ; "JSR PC,LOORES" WILL RESTORE THE LOADER IN ITS ORIGINAL
1673          ; LOCATION. ".LOO:" CONTAINS THE ADDRESS OF THE LOADER.
1674
1675 011746 005767 167050          LOOGET: TST          OPTION          ; WHICH OPTION?
1676 011752 003402                BLE          3$          ; NO MEMORY MANAGEMENT
1677 011754 005077 177740          CLR          JSRO          ; TURN OFF MM
1678 011760 010001          3$: MOV          R0,R1          ; GET THE BANK
1679 011762 022701 000006          CMP          #6,R1          ; IS IT > 6?
1680 011766 100002                BPL          1$          ; SKIP IF LES THAN 6
1681 011770 012701 000006          MOV          #6,R1          ; IF > 7 MAKE 7
1682 011774 006001          1$: ROR          R1          ; GET THE
1683 011776 006001                ROR          R1          ; UPPER
1684 012000 006001                ROR          R1          ; THREE
1685 012002 006001                ROR          R1          ; BITS
1686 012004 042701 017777          BIC          #17777,R1          ; CLEAR JUNK
1687 012010 012702 013062          MOV          #ENDP,R2          ; GET SAVE ADDRESS
1688 012014 060201                ADD          R2,R1          ; MAKE OTHER ONE
1689 012016 010167 000060          MOV          R1,.LOO          ; SAVE ADDRESS
1690 012022 012122          2$: MOV          (1)+,(2)+          ; MOVE WORD
1691 012024 022702 020000          CMP          #20000,R2          ; END?
1692 012030 001374                BNE          2$          ; NO!
1693 012032 000207                RTS          PC          ; RETURN
1694
1695 012034 005767 166762          LOORES: TST          OPTION          ; WHICH OPTION?
1696 012040 003402                BLE          2$          ; NO MEMORY MANAGEMENT
1697 012042 005077 177652          CLR          JSRO          ; TURN OFF MM
1698 012046 012702 013062          2$: MOV          #ENDP,R2          ; GET END OF PROGRAM
1699 012052 016701 000024          MOV          .LOO,R1          ; GET SAVE ADDRESS
1700 012056 012221          1$: MOV          (2)+,(1)+          ; RESTORE WORD
1701 012060 022702 020000          CMP          #20000,R2          ; END?
1702 012064 001374                BNE          1$          ; LOOP
1703 012066 005767 166730          TST          OPTION          ; WHICH OPTION?
1704 012072 003402                BLE          3$          ; NO MEMORY MANAGEMENT
1705 012074 005277 177620          INC          JSRO          ; TURN ON MM
1706 012100 000207          3$: RTS          PC          ; RETURN
1707
1708 012102 000000          .LOO: 0          ; STARTING ADDRESS OF LOADER

```

```

1709 ; STYPE MESSAGE TYPEOUT ROUTINE
1710 ; THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
1711 ; CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ADR" - TYPES THE
1712 ; MESSAGE STARTING IN LOCATION "ADR:" 2) "TYPE CHAR" - TYPES
1713 ; THE ASCII "CHAR", AND 3) "PRINT <(15)<(12)"MESSAGE"> - TYPES
1714 ; THE MESSAGE WHICH IS INLINE ASCII.
1715
1716
1717 012104 010546 10TS: MOV TTY, -(6) ;SAVE TTY RS=TTY
1718 012106 017605 000002 MOV #2(6), TTY ;GET ADDRESS TO BE TYPED
1719 012112 032705 177400 BIT #177400, TTY ;IS IT A TYPED?
1720 012116 001004 BNE 1$ ;NO
1721 012120 010567 000064 MOV TTY, .TYPE ;GET THE CHARACTER
1722 012124 012705 012210 MOV #.TYPE, TTY ;FUDGE THE ADDRESS
1723 012130 105715 1$: TSTB (TTY) ;TERMINATOR?
1724 012132 001406 BEQ 2$ ;GET OUT IF SO
1725 012134 112537 177566 MOVB (TTY)+, #177565 ;LOAD AND TYPE THE CHARACTER
1726 012140 105737 177564 TSTB #177564 ;IS THE PRINTER READY
1727 012144 100375 BPL .-4 ;WAIT UNTIL IT IS
1728 012146 000770 BR 1$ ;GET THE NEXT CHARACTER
1729 012150 017646 000002 2$: MOV #2(6), -(6) ;GET ADDRESS TO BE TYPED
1730 012154 062766 000004 ADD #2, 4(6) ;ADD 2 TO THE ADDRESS
1731 012162 022666 000002 CMP (6)+, 2(6) ;IS IT .+2?
1732 012166 001006 BNE 3$ ;NO
1733 012170 062705 000002 ADD #2, TTY ;ADD 2 TO THE ADDRESS
1734 012174 042705 000001 BIC #1, TTY ;BACK UP TO AN EVEN BYTE
1735 012200 010566 000002 MOV TTY, 2(6) ;RESTORE ADDRESS
1736 012204 012605 3$: MOV (6)+, TTY ;RESTORE TTY
1737 012206 000002 RTI ;RETURN
1738 012210 000000 .TYPE: 0 ;CHARACTER TYPE LOCATION
1739
1740 012212 010346 READS: MOV R3, -(6) ;SAVE R3
1741 012214 012703 012316 1$: MOV #INPUT, R3 ;GET ADDRESS
1742 012220 022703 012336 2$: CMP #INPUT+20, R3 ;BUFFER FULL?
1743 012224 001412 BEQ 4$ ;YES - TYPE ""
1744 012226 105737 177560 TSTB #177560 ;WAIT FOR
1745 012232 100375 BPL .-4 ;A CHARACTER
1746 012234 113713 177562 MOVB #177562, (3) ;GET CHARACTER
1747 012240 142713 000200 BICB #200, (3) ;GET RID OF JUNK
1748 012244 122713 000177 CMPB #177, (3) ;IS IT A RUBOUT
1749 012250 001005 BNE 3$ ;SKIP IF NOT
1750 012252
1751 012252 000004 012256 4$: TYPE .+2 ;. ASCII ""<(15)<(12)
1752 012262 000754 BR 1$ ;ZAP THE BUFFER AND LOOP
1753 012264 111367 177720 3$: MOVB (3), .TYPE ;SET UP FOR TYPING
1754 012270 000004 012210 TYPE .TYPE ;ECHO IT
1755 012274 122723 000015 CMPB #15, (3)+ ;CHECK FOR RETURN
1756 012300 001347 BNE 2$ ;LOOP IF NOT RETURN
1757 012302 105063 177777 CLRB -1(3) ;ZAP RETURN (THE 15)
1758 012306 000004 000012 TYPE .12 ;TYPE A LINE FEED
1759 012312 012603 MOV (6)+, R3 ;RESTORE R3
1760 012314 000207 RTS PC ;RETURN
1761 012316 000020 INPUT: .BLKW 20 ;TTY INPUT AREA

```

```

1762 ; SOCIAL OCTAL TYPEOUT ROUTINE
1763 ; THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
1764 ; ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
1765 ; THE 16 BITS. IT IS CALLED VIA THE DUMP, SDUMP, DUMP18, OR BITYPE MACRO'S.
1766
1767
1768 012356 012767 170101 000202 BITYPS: MOV #170101,.PR ;SET BIT FLAG ANS 16. CHARACTER COUNT
1769 012364 000411 BR .PTIT ;NOW TYPE IT IN BIT FORM
1770 012366 112767 000001 000172 PRINTR: MOVB #1,.PR ;SET ZERO FILL SWITCH
1771 012374 000402 BR .+6 ;SKIP
1772 012376 005067 000164 PRINTS: CLR .PR ;SUPPRESS LEADING ZERO'S
1773 012402 112767 177772 000157 MOVB #-6,.PR+1 ;SET COUNT
1774 012410 010446 .PTIT: MOV R4,-(6) ;SAVE R4
1775 012412 012704 012570 MOV #.PR+2,R4 ;SET POINTER TO FIRST ASCII CHAR.
1776 012416 105014 CLR# (4) ;CLEAR FIRST BYTE
1777 012420 000432 BR .PRF ;ROTATE FIRST BIT
1778 012422 010446 PRINTA: MOV R4,-(6) ;SAVE R4
1779 012424 012704 012570 MOV #.PR+2,R4 ;SET UP POINTER TO OUTPUT AREA
1780 012430 116714 166360 MOVB MX,(4) ;MX CONTAINS UPPER 5 BITS
1781 012434 006305 ASL TTY ;GET RID
1782 012436 006305 ASL TTY ;OF 3
1783 012440 006305 ASL TTY ;JUNK BITS
1784 012442 106214 ASRB (4) ;GET BIT13
1785 012444 006005 ROR TTY ;PACK IT
1786 012446 106214 ASRB (4) ;GET BIT14
1787 012450 006005 ROR TTY ;PACK IT
1788 012452 152724 000060 BISB #'0,(4)+ ;MAKE IT ASCII
1789 012456 012767 175401 000102 MOV #175401,.PR ;-5,1 - 5 BYTES AND FILL
1790 012464 105014 .PRL: CLR# (4) ;CLEAR BYTE OF CHARACTER
1791 012466 032767 000100 000072 BIT #100,.PR ;BIT TYPING MODE?
1792 012474 001004 BNE .PRF ;YES - SKIP 2 ROTATES
1793 012476 006105 ROL TTY ;ROTATE BIT INTO C
1794 012500 106114 ROLB (4) ;PACK IT
1795 012502 006105 ROL TTY ;ROTATE BIT INTO C
1796 012504 106114 ROLB (4) ;PACK IT
1797 012506 006105 .PRF: ROL TTY ;ROTATE BIT INTO C
1798 012510 106114 ROLB (4) ;PACK IT
1799 012512 105714 TSTB (4) ;IS IT ZERO?
1800 012514 001402 BEQ .+6 ;SKIP INC
1801 012516 105267 000044 INCB .PR ;SET FILL SWITCH
1802 012522 105767 000040 TSTB .PR ;CHECK FILL SWITCH
1803 012526 001402 BEQ .+6 ;SKIP BITSET
1804 012530 152724 000060 BISB #'0,(4)+ ;MAKE INTO ASCII CHAR
1805 012534 105267 000027 INCB .PR+1 ;INC COUNT
1806 012540 001351 BNE .PRL ;REPEAT
1807 012542 022704 012570 CMP #.PR+2,R4 ;EMPTY BUFFER?
1808 012546 001002 BNE .+6 ;SKIP IF NOT
1809 012550 112724 000060 MOVB #'0,(4)+ ;LOAD 1 ZERO
1810 012554 105014 CLR# (4) ;NULL TERMINATOR
1811 012556 000004 012570 TYPE .PR+2 ;TYPE IT
1812 012552 012604 MOV (6)+,R4 ;RESTORE R4
1813 012564 000207 RTS PC ;RETURN
1814 012566 000012 .PR: .BLKW 12 ;COUNT, SWITCH, AND OUTPUT BUFFER

```

```

1815 012612 012777 012740 000126 PDOWNS: MOV #ILLUP, @PUVECS ;SET FOR FAST UP
1816 012620 012777 J00340 000122 MOV #340, @PUVECS+2 ;PRIO:7
1817 012626 010046 MOV R0, -(6) ;PUSH R0 ON STACK
1818 012630 010146 MOV R1, -(6) ;PUSH R1 ON STACK
1819 012632 010246 MOV R2, -(6) ;PUSH R2 ON STACK
1820 012634 010346 MOV R3, -(6) ;PUSH R3 ON STACK
1821 012636 010446 MOV R4, -(6) ;PUSH R4 ON STACK
1822 012640 010546 MOV R5, -(6) ;PUSH R5 ON STACK
1823 012642 010667 000076 MOV SP, SAVR6 ;SAVE SP
1824 012646 012777 012656 000072 MOV #PUPS, @PUVECS ;SET UP VECTOR
1825 012654 000000 HALT ;WAIT FOR PF
1826
1827 012656 016706 000062 PUPS: MOV .SAVR6, SP ;GET SP
1828 012662 005001 CLR R1 ;WAIT LOOP FOR THE TTY
1829 012664 005201 IS: INC R1 ;WAIT FOR THE INC
1830 012666 001376 BNE IS ;OF WORD
1831 012670 012605 MOV (6)+, R5 ;POP STACK INTO R5
1832 012672 012604 MOV (6)+, R4 ;POP STACK INTO R4
1833 012674 012603 MOV (6)+, R3 ;POP STACK INTO R3
1834 012676 012602 MOV (6)+, R2 ;POP STACK INTO R2
1835 012700 012601 MOV (6)+, R1 ;POP STACK INTO R1
1836 012702 012600 MOV (6)+, R0 ;POP STACK INTO R0
1837 012704 012737 012612 000024 MOV #PDOWNS, @#24 ;SET UP THE POWER DOWN VECTOR
1838 012712 012737 000340 000026 MOV #340, @#26 ;PRIO:7
1839 012720 000004 012724 TYPE .+2 ;ASCIZ <15><12>"POWER"
1840 012734 000167 167230 JMP RTPF ;JMP TO USER ADDRESS
1841
1842 012740 000000 ILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
1843 012742 000776 BR .-2 ;BEFORE THE POWER DOWN WAS COMPLETE
1844
1845 012744 000000 .SAVR6: 0 ;PUT THE SP HERE
1846 012746 000024 000026 PUVECS: 24, 26 ;POWER UP VECTOR
1847
1848 012752 116767 164604 177336 READR: MOVB 177562, INPUT ;READ THE CHARACTER
1849 012760 142767 000200 177330 BICB #200, INPUT ;CLEAR JUNK
1850 012766 122767 000003 177322 CMPB #3, INPUT ;IS IT A ^C?
1851 012774 001401 BEQ .+4 ;SKIP IF ^C
1852 012776 J00002 RTI ;RETURN IF NOT
1853 013000 000004 013004 TYPE .+2 ;ASCIZ "^C" (<177>)
1854 013010 005037 000042 CLR @#42 ;DO NOT LOOP
1855 013014 000137 001024 JMP @#BEGIN ;RESTART
1856
1857 013020 000000 SAV6: 0
1858 013022 000010 PATCH1: .BLKW 10 ;PATCH AREA #1
1859 013042 000010 PATCH2: .BLKW 10 ;PATCH AREA #2
1860 013062 000000 ENDP: 0 ;LAST LOC OF PROGRAM
1861 000001 .END

```


K04

LODGET	011746	455	1675#												
LODRES	012034	467	640	1695#											
LOLIM	011740	395*	545	701	709	791	799	838*	1351	1375	1454	1471	1486	1663#	
MAP	004160	634	844#												
MAPIT =	104444	439	634#												
MASK	010152	1415#	1445												
MEMORY	003714	400*	671	674*	676*	678*	719	786#							
MIN	001016	367#	397*	412*	555*	562*	564*	566*	568	570*	571	589	670	679	
MX	001014	366#	398*	413*	571*	655	670*	679	690*	691	695	758	770	822	
		895	906	960	971	1035	1046	1103	1114	1164	1175	1309	1320	1334	
		1384	1780	1815											
MXC3	011736	410	432*	436*	830*	1661#									
M1	010716	540	1519#												
M11	011103	553	1542#												
M12	011121	572	1545#												
M13	011137	593	594	1548#											
M15	011142	905	970	1045	1113	1174	1319	1549#							
M3	011012	885	950	1022	1090	1151	1299	1400	1510	1530#					
M4	011022	887	952	1024	1092	1153	1301	1512	1532#						
M7	011032	810	1534#												
M8	011053	813	908	973	1048	1116	1177	1322	1537#						
M9	011065	816	911	976	1051	1119	1180	1325	1539#						
N =	000052	344#	616	617#	618#	619#	620#	621#	622#	623#	624#	625#	626#	627#	
		628#	629#	630#	631#	632#	633#	634#	635#	636#	637#				
NEWDEV	001750	473#	686												
NEXT =	104442	633#	788	1333	1484										
NODEV	002650	499	593#												
NXT	003064	633	652#												
NXTCR	002564	572#	588												
NX.DR	010452	1473#	1483												
NX1.DR	010534	1488#	1493												
OPTION	001022	369#	401*	406*	411*	416	460	823	1675	1695	1703				
PATCH1	013022	635	1858#												
PATCH2	013042	636	1859#												
PC =	%000007	343#	458*	475*	502*	554*	573*	640*	645*	651*	668*	699*	710*	733*	
		792*	812*	815*	818*	886*	889*	907*	951*	954*	972*	1023*	1026*	1047*	
		1091*	1094*	1115*	1152*	1155*	1176*	1300*	1303*	1321*	1402*	1511*	1514*	1693*	
		1706*	1760*	1813*											
PCNT	001004	363#													
PDOWNB	012612	376	1815#	1037											
POTS	003544	629	752#												
PRINTA	012422	812	1778#												
PRINTR	012366	815	818	886	889	951	954	1023	1026	1091	1094	1152	1155	1300	
		1303	1402	1511	1514	1770#									
PRINTS	012376	458	699	907	910	913	972	975	978	1047	1050	1053	1115	1118	
		1121	1176	1179	1182	1321	1324	1327	1772#						
PS =	177776	331#	470*	685*											
PSCNT	011534	399#	660*	669*	1603#										
PUPS	012656	1824	1827#												
PUVECS	012746	1815*	1816*	1824*	1846#										
QUES	011435	1589#													
RCCA	011624	923	931	940	1623#										
RCCS	011616	507*	923	931	940	943	948	959	1620#						
RCDR	011626	921*	929*	938*	1624#										
RCER	011620	953	1621#												
RCTRAP	011544	381*	388*	488*	919*	927*	936*	1607#							

		1030	1098	1159	1201	1216	1234	1249	1429	1430#	1436	1437#	1474	1478
		1594#	1727	1745	1751	1761#	1771	1800	1803	1808	1814#	1815	1839	1843
		1851	1853	1858#	1859#									
.BEGIN	003562	631	758#											
.LOD	012132	1689#	1699	1708#										
.PR	012566	1768#	1770*	1772*	1773*	1775	1779	1789*	1791	1801*	1802	1805*	1807	1811
		1814#												
.PRF	012506	1777	1792	1797#										
.PAL	012464	1790#	1806											
.PTIT	012410	1769	1774#											
.SAVR6	012744	1823*	1827	1845#										
.START=	104436	631#	857	865	874	922	930	952	987	995	1004	1062	1070	1079
		1131	1140	1268	1286	1497								
.TYPE	012210	1721*	1722	1738#	1753*	1754								

AOC	549															
AOO	422	443	472	608	665	726	728	742	769	777	782	839	1186	1192	1342	
	1344	1346	1348	1360	1376	1378	1455	1476	1480	1688	1730	1733				
ASL	447	1267	1285	1385	1781	1782	1783									
ASR	827															
ASRB	1784	1786														
BCS	730															
BEO	461	469	480	539	547	558	577	664	673	682	684	688	693	697	705	
	712	805	824	903	968	1043	1111	1172	1317	1398	1434	1448	1474	1478	1504	
	1727	1743	1800	1803	1851											
BGT	445	588	721	1239												
BIC	449	698	764	772	837	901	966	1041	1109	1170	1315	1339	1386	1404	1407	
	1465	1686	1734													
BICB	1445	1747	1849													
BIS	504	507	511	515	519	523	527	536	775	776	1388					
BISB	1788	1804														
BIT	402	653	657	687	692	696	804	881	892	946	957	1018	1032	1086	1100	
	1147	1161	1295	1306	1397	1443	1506	1719	1791							
BLE	569	1676	1696	1704												
BLT	590	656	1206													
BMI	417	462	610	672	720	825	1201	1216	1234	1249	1427					
BNE	403	418	426	435	454	474	506	510	514	518	522	526	530	535	650	
	654	658	661	667	680	716	774	795	802	808	882	893	947	958	1019	
	1033	1087	1101	1148	1162	1221	1254	1296	1307	1355	1359	1365	1369	1444	1453	
BPL	1461	1470	1483	1493	1507	1692	1702	1720	1732	1749	1756	1792	1806	1808	1830	
	372	754	879	944	1009	1011	1016	1030	1084	1098	1145	1159	1188	1199	1214	
	1232	1247	1293	1406	1442	1502	1680	1727	1745							
BR	407	427	437	464	476	494	508	512	516	520	524	528	537	542	550	
	567	586	613	675	677	743	751	1013	1203	1207	1218	1236	1240	1251	1431	
	1457	1728	1752	1769	1771	1777	1843									
CLC	548	759	832													
CLR	397	398	400	401	421	470	483	500	541	552	555	574	648	652	674	
	695	707	708	752	789	790	856	864	873	877	921	929	938	942	986	
	994	1003	1012	1014	1061	1069	1078	1082	1143	1291	1361	1363	1366	1370	1371	
	1372	1373	1392	1451	1485	1500	1677	1697	1772	1828	1854					
CLRB	1757	1776	1790	1810												
CMP	425	434	444	505	509	513	517	521	525	529	534	546	587	589	609	
	679	711	715	768	773	781	794	801	807	1205	1220	1238	1253	1438	1452	
	1462	1473	1477	1482	1492	1679	1691	1701	1731	1742	1807					
CMPB	479	655	902	967	1042	1110	1171	1316	1447	1748	1755	1850				
COM	731	732	800	1489												
COMB	406	411	676													
DEC	450	570	660	1354	1358	1364										
EMT	329															
HALT	354	612	755	1825	1842											
INC	423	431	440	453	649	662	690	880	945	1017	1085	1146	1294	1481	1505	
	1705	1829														
INCB	896	961	1036	1104	1165	1310	1368	1450	1801	1805						
ICT	330															
JMP	357	498	533	591	597	611	686	1222	1255	1278	1393	1439	1463	1840	1855	
JSR	455	458	467	475	502	554	573	640	645	668	699	710	792	812	815	
	818	886	889	907	910	913	951	954	972	975	978	1023	1026	1047	1050	
	1053	1091	1094	1115	1118	1121	1152	1155	1176	1179	1182	1300	1303	1321	1324	
	1227	1402	1511	1514												
MOV	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	
	388	389	390	391	392	393	394	395	396	399	404	409	412	413	414	

.LIST	1	271	344	354	370	457	460	485	498	499	502	597	598	616	617
	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632
	633	634	635	636	637	652	690	695	851	877	916	942	981	1014	1056
	1082	1124	1143	1261	1291	1330	1430	1437	1467	1500	1615	1666	1709	1740	1752
	1762	1815	1840	1854											
.MACRO	1	370	598												
.MCALL	271	344													
.MLIST	1	271	344	354	370	457	460	485	498	499	502	597	598	616	617
	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632
	633	634	635	636	637	652	690	695	851	877	916	942	981	1014	1056
	1082	1124	1143	1261	1291	1330	1430	1437	1467	1500	1615	1666	1709	1740	1752
	1762	1815	1840	1854											
.PAGE	598	877	942	1014	1082	1143	1291	1709	1762	1815					
.REM	1	272													
.REPT	354														
.SBTTL	370	499	598	652	851	877	916	942	981	1014	1056	1082	1124	1143	1261
	1291	1330	1467	1500	1615	1666	1709	1740	1762	1815					
.TITLE	271														

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*, DZOMAB.SEG=DZOMAB.SML, DZOMAB.CMB
 RUN-TIME: 17 23 3 SECONDS
 RUN-TIME RATIO: 286/44=6.4
 CORE USED: 22K (43 PAGES)

