

RP11-C/RP02

DISKLESS
MD-11-DZRPE-A

EP-DZRPE-A DL-A

OCT 1976

COPYRIGHT ©1976

digital

FICHE 1 OF 1

Made in U.S.A.

137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192

4.2 STARTING ADDRESS

1. LOAD STARTING ADDRESS 600 TO RUN THE DISKLESS DIAGNOSTIC. THE RP11 MUST HAVE THE MAINTENANCE SWITCH ENABLED. IF A DRIVE IS ON THE SYSTEM, IT MUST BE POWERED DOWN AND CIRCUIT BREAKER #1 LOCATED TO THE RIGHT OF THE DRIVE'S "SIGNAL IN" CABLE MUST BE TURNED OFF.

4.3 PROGRAMS AND/OR OPERATOR ACTION

1. LOAD THE PROGRAM INTO MEMORY USING THE ABS LOADER.
2. LOAD STARTING ADDRESS OF 600.
3. SET SWITCHES (SEE SECTION 5.1.1).
4. PRESS START.
5. WHEN RUNNING THE DISKLESS TEST THE PROGRAM WILL MAKE 50 PASSES THRU THE TEST AND THEN TYPE "DONE" AND CONTINUE LOOPING.

5.0 OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

AFTER LOADING THE STARTING ADDRESS, SELECT THE DESIRED SWITCHES.

5.1.1 SWITCH SETTINGS ARE:

- SW<15>=1DELETE ERROR HALT
- SW<14>=1DELETE PRINTOUT
- SW<13>=1LOOP ON FAILING TEST
- SW<12>=1.....LOOP ON TEST UNCONDITIONALLY

F01

.MAIN. MACY11 27(732) 08-SEP-76 08:58 PAGE 5
DZRPEA.P11

193

5.2 SUBROUTINE ABSTRACTS

194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248

5.2.1 TRAP CATCHER

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.

6.0 ERRORS

6.1 ERROR PRINTOUTS CONTAIN THE ERROR ADDRESS, CPU STATUS AT ERROR TIME, AND OTHER PERTINENT INFORMATION CONCERNING THE PARTICULAR FAILURE. THIS INFORMATION MAY BE THE CONTENTS OF THE RELEVANT RPI1 STATUS REGISTER OR GOOD/BAD COMPARE DATA. IF THE ERROR OCCURRED IN A SUBROUTINE, THE ADDRESS OF THE SUBROUTINE CALL IS ALSO GIVEN. REFER TO THE PROGRAM LISTING AT THE STATED ADDRESS TO DETERMINE THE CAUSE OF AN ERROR.

7.0 RESTRICTIONS

NONE.

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

THE PROGRAM WILL MAKE 50 ITERATIONS THRU THE DISKLESS TEST TO COMPLETE ONE PASS. AT THE END OF EACH PASS "DONE" IS TYPED OUT AND THE PROGRAM CONTINUES LOOPING.

8.2 STACK POINTER

STACK IS INITIALLY SET TO 600.

9.0 PROGRAM DESCRIPTION

249
250
251
252
253
254
255
256
257
258
259
260
261
262
263

9.1 DISKLESS TEST

THIS TEST UTILIZES THE RP11 MAINTENANCE REGISTERS TO CHECK THE LOGICAL RESPONSES OF THE RP11. THE MAINTENANCE REGISTERS SIMULATE THE SIGNALS PASSING BETWEEN THE RP11 AND THE RP02. TO RUN THIS TEST THE RP11 MAINTENANCE SWITCH MUST BE ENABLED. IF A DRIVE EXISTS, IT MUST BE POWERED DOWN AND CIRCUIT BREAKER #1 LOCATED JUST TO THE RIGHT OF THE DRIVES'S "SIGNAL IN" CABLE MUST BE TURNED OFF.

264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319

177570
177776
177560
177562
177564
177566
000060
000064

000015
000012
000006
000000
000001
000002
000003
000004
000005
000006
000007
000007
000000
177776
000240
000000
000040
000100
000140
000200
000240
000300
000340
000001
000002
000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000
104000
104400

.ABS %
;RP11 DISCLESS TEST
;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.
SR=177570 ;SWITCH REGISTER
CC=177776 ;CONDITION CODES
TKS=177560
TKB=177562
TPS=177564
TPB=177566
TKV=60
TPV=64

CR=15
LF=12 ;STACK POINTER
SP=%6
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
R6=%6
R7=%7
PC=%7
XX=HALT
PS=CC
NOP=240
P0=0
P1=40
P2=100
P3=140
P4=200
P5=240
P6=300
P7=340
B0=1
B1=2
B2=4
B3=10
B4=20
B5=40
B6=100
B7=200
B8=400
B9=1000
B10=2000
B11=4000
B12=10000
B13=20000
B14=40000
B15=100000
ERR=EMT
ERM=TRAP

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

000001
000002
000004
000010
000020
000040
005726
022626

GB=B0
DD=B1
DS=B2
ER=B3
CS=B4
DA=B5
POP=5726
POPPOP=22626

.MACR ANDI A,B
BIC #-A-1,B ;MASK A APPLIED TO B
.ENDM

.MACR AND A,B
COM B ;C(B) =MASK. INVERT IT
BIC B,A ;AND A WITH B
COM B ;RESTORE B
.ENDM

.MACR BITEST A,B
JSR RS,BITST ;TEST BITS IN REGISTER
.WORD A ;MASK=A
.WORD B ;REGISTER=B
ERR+GB ;LOAD/READ ERROR IN B
RTS PC ;EXIT FROM ERROR CALL
.ENDM

.MACR ROTR A
JSR PC,RTR ;ROTATE RO A PLACES RIGHT
.WORD A
.ENDM

.MACR ROTL A
JSR PC,RTL ;ROTATE RO A PLACES LEFT
.WORD A
.ENDM

K01

358					
359					
360			.MACR	LOOP A,B	
361			JSR	PC, @#LERCHK	;LOOP TO A ON ERROR IF SR13=1
362			JMP	A	
363		B:	JSR	PC, @#LUPCHK	;LOOP TO A UNCONDITIONALLY IF SR12=1
364			JMP	A	
365			.ENDM		
366					
367			.MACR	CLRSOT	
368			JSR	PC, CLRP	
369			MOV	#1, RD	
370			JSR	PC, INDEXP	
371			BIC	#88, @RPM3	
372			BIS	#88+814, @RPM3	
373			BIC	#88, @RPM3	
374			.ENDM		
375					
376					
377					
378	000000		.=0		
379			.REPT	140	
380			.+2		
381			HALT		;TRAPPED OR INTERRUPTED TO PREVIOUS ADDRESS
382			.ENDR		
383					
384			.=30		
385	000030	000030	.WORD	ERROR	
386	000032	000340	.WORD	340	;NO INTERRUPTS ALLOWED DURING ERRORS
387	000034	006744	.WORD	ERROR	
388	000036	000340	.WORD	340	
389					
390					

L01

.MAIN. MACY11 27(732) 08-SEP-76 08:58 PAGE 11
 DZRPER.P11

391								
392		000600				.=600		
393								
394	000600	012706	000600		START:	MOV	#START,%6	;SET STACK POINTER
395	000604	012767	000340	177164		MOV	#340,CC	;LOCK OUT OTHER DEVICES
396	000612	012767	000062	007470		MOV	#50.,BTOG	
397								
398	000620	000005			T1:	RESET		
399	000622	005777	007474			TST	QRPDS	
400	000626	001405				BEQ	T1P	
401	000630	104004				ERR+DS		;RPDS NOT=0 AFTER RESET
402	000632					LOOP	T1,T1P	
403								
404	000652	000005			T2:	RESET		
405	000654	005777	007446			TST	QRPER	
406	000660	001405				BEQ	T2P	
407	000662	104010				ERR+ER		;RPER NOT=0 AFTER RESET
408	000664					LOOP	T2,T2P	

NO1

.MAIN. MACY11 27(732) 08-SEP-76 08:58 PAGE 13
DZRPEA.P11

465	001302	000005		T11:	RESET		
466	001304	005777	007056		TST	@RPB1	
467	001310	001410			BEQ	T11P	
468	001312	017767	007050 006752		MOV	@RPB1,DATA	
469	001320	104002			ERR+DD		;RPB1 NOT=0 AFTER RESET
470	001322				LOOP	T11,T11P	

C02

.MAIN. MACY11 27(732) 09-SEP-76 08:58 PAGE 15
DZRPER.P11

001610

B3BT: BITEST 17,RPB3


```

530
531
532
533 001624 012777 000377 006524 RESDS: MOV #377,ARPM2 ;SET ATTN7-0
534 001632 012777 000002 006522 MOV #B1,ARPM3 ;FIRST INDEX
535 001640 112777 000374 006514 MOV# #374,ARPM3 ;SECOND INDEX & OTHER GOODIES
536 001646 004767 004322 JSR PC,CLRP ;CLEAR RP11
537 001652 042777 020000 006502 BIC #B13,ARPM3
538 001660 005777 006436 TST ARPDS ;RPDS=0?
539 001664 001405 BEQ RESDSP ;YES
540 001666 104004 ERR+DS ;NO-RESET FAILED TO CLEAR RPDS
541 001670 LOOP RESDS,RESDSP
542
543
544 001710 012777 177776 006410 RESER: MOV #177776,ARPER ;SET RPER
545 001716 004767 004252 JSR PC,CLRP ;CLEAR RP11
546 001722 005777 006400 TST ARPER ;RPER=0?
547 001726 001405 BEQ RESERP ;YES
548 001730 104010 ERR+ER ;NO-RESET FAILED TO CLEAR RPER
549 LOOP RESER,RESERP
550
551 001752 012777 177760 006352 RESCS: MOV #177760,ARPCS ;SET RPCS
552 001760 004767 004210 JSR PC,CLRP ;CLEAR RP11
553 001764 022777 000200 006340 CMP #B7,ARPCS ;RPCS=0 EXCEPT FOR READY?
554 001772 001405 BEQ RESCSP ;YES
555 001776 104020 ERR+CS ;NO-RESET FAILED TO CLEAR RPCS
556 LOOP RESCS,RESCSP
557
558 002016 012777 177777 006312 RESWC: MOV #-1,ARPWC ;SET RPWC
559 002024 004767 004144 JSR PC,CLRP ;CLEAR RP11
560 002030 005777 006302 TST ARPWC ;RPWC=0?
561 002034 001410 BEQ RESWCP ;YES
562 002036 017767 006274 006226 MOV ARPWC,DATA ;NO-PUT RPWC INTO DATA DISPLAY
563 002044 104002 ERR+DD ;RESET FAILED TO CLEAR RPWC
564 LOOP RESWC,RESWCP
565
566 002066 012777 177777 006244 RESBA: MOV #-1,ARPBA ;SET RPBA
567 002074 004767 004074 JSR PC,CLRP ;CLEAR RP11
568 002100 005777 006234 TST ARPBA ;RPBA=0?
569 002104 001410 BEQ RESBAP ;YES
570 002106 017767 006226 006156 MOV ARPBA,DATA ;NO
571 002114 104002 ERR+DD ;RESET FAILED TO CLEAR RPBA
572 LOOP RESBA,RESBAP
573
574 002136 012777 000377 006176 RESCA: MOV #377,ARPCA ;SET RPCA
575 002144 004767 004024 JSR PC,CLRP ;CLEAR RP11
576 002150 105777 006166 TSTB ARPCA ;CAR=0?
577 002154 001410 BEQ RESCAP ;YES
578 002156 017767 006160 006106 MOV ARPCA,DATA ;NO
579 002164 104002 ERR+DD ;RESET FAILED TO CLEAR CAR
580 LOOP RESCA,RESCAP
581
582 002206 012777 177777 006132 RESDA: MOV #-1,ARPDA ;SET RPDA
583 002214 004767 003754 JSR PC,CLRP ;CLEAR RP11
584 002220 032777 017417 006120 BIT #17417,ARPDA ;TAR & SAR=0
585 002226 001410 BEQ RESDAP ;YES
586 002230 017767 006112 006034 MOV ARPDA,DATA ;NO

```

E02

.MAIN. MACY11 27(732) 09-SEP-76 08:58 PAGE 17
DZRPEA.P11

586 002236 104002
587 002240

ENR+DD
LOOP RESDA,RESDAP

;RESET FAILED TO CLEAR TAR & SAR


```

588
589
590 ;RPDS FUNCTION TEST
591 002260 005077 006076 DSF1: CLR @RPM3
592 002264 005077 006032 CLR @RPDS
593 002270 052777 040000 006064 BIS #B14,@RPM3 ;SET MAINT. READY
594 002276 032777 100000 006016 BIT #B15,@RPDS ;TEST READY=1
595 002304 001005 BNE DSF1P ;PROCEED IF SET
596 002306 104004 ERR+DS ;READY NOT SET BY RPM3,B14=1
597 002310 LOOP DSF1,DSF1P
598
599 002330 005077 006026 DSF2: CLR @RPM3
600 002334 005077 005762 CLR @RPDS
601 002340 032777 040000 006014 BIT #B14,@RPM3 ;SET M. READY
602 002346 042777 040000 006006 BIC #B14,@RPM3 ;CLEAR M. READY
603 002354 032777 100000 005740 BIT #B15,@RPDS ;TEST READY=0
604 002362 001405 BEQ DSF2P ;PROCEED IF CLEAR
605 002364 104004 ERR+DS ;READY NOT CLEARED BY RPM3,B14=0
606 002366 LOOP DSF2,DSF2P
607
608 002406 DSF3: CLRSOT
609 002444 117700 005676 MOV @RPDA,RO ;FETCH CURRENT SECTOR
610 002450 ROTR 4
611 002456 005200 INC RO ;LOAD SAR WITH
612 002460 110077 005662 MOV @RPDA,RO ;CURRENT SECTOR+1
613 002464 005077 005632 CLR @RPDS
614 002470 012777 000001 005640 MOV #1,@RPWC
615 002476 052777 020000 005656 BIS #B13,@RPM3
616 002504 012777 000017 005620 MOV #17,@RPCS ;READ AND GO
617 002512 004767 003522 JSR PC,T3P ;3 TP
618 002516 042777 000400 005636 BIC #B8,@RPM3
619 002524 052777 040400 005630 BIS #B8+B14,@RPM3 ;1 SP
620 002532 042777 000400 005622 BIC #B8,@RPM3
621 002540 012700 000002 MOV #2,RO ;ISSUE 2
622 002544 004767 003500 JSR PC,INDEXP ;INDEX PULSES
623 002550 032777 010000 005544 BIT #B12,@RPDS ;TEST HEADER-NOT-FOUND=1
624 002556 001007 BNE DSF3P ;OK IF SET
625 002560 104004 ERR+DS ;HEADER-NOT-FOUND NOT SET BY PROPER SEQUENCE
626 002562 004767 003506 JSR PC,HRDR
627 002566 LOOP DSF3,DSF3P
628
629 002606 DSF4: CLR @RPM3
630 002612 005077 005550 CLR @RPDS
631 002616 052777 002000 005536 BIS #B10,@RPM3 ;SET M. SEEK INC.
632 002624 032777 004000 005470 BIT #B11,@RPDS ;TEST SEEK INC.=1
633 002632 001005 BNE DSF4P ;OK IF SET
634 002634 104004 ERR+DS ;SEEK-INCOMPLETE NOT SET BY RPM3, BIT10=1
635 002636 LOOP DSF4,DSF4P
636
637 002656 DSF5: CLR @RPM3
638 002662 005077 005434 CLR @RPDS
639 002666 052777 002000 005466 BIS #B10,@RPM3 ;SET M. SEEK INC.
640 002674 042777 002000 005460 BIC #B10,@RPM3 ;CLEAR M. SEEK INC.
641 002702 032777 004000 005412 BIT #B11,@RPDS ;TEST SEEK INC.=0
642 002710 001405 BEQ DSF5P ;OK IF CLEAR
643 002712 104004 ERR+DS ;SEEK-INCOMPLETE NOT CLEARED BY RPM3,BIT10=0

```

G02

.MAIN. MACY11 27(732) 08-SEP-76 08:58 PAGE 19
DZRPEA.P11

644 002714

LOOP DSF5,DSF5P

645											
646	002734	000240				DSF6:	NOP				;DO LATER
647											
648	002736	005077	005420			DSF7:	CLR	DRPM3			
649	002742	005077	005354				CLR	DRPDS			
650	002746	052777	004000	005406			BIS	#B11,DRPM3			;SET M. FILE UNSAFE
651	002754	032777	001000	005340			BIT	#B9,DRPDS			;TEST FILE-UNSAFE=1
652	002762	001005					BNE	DSF7P			;OK IF SET
653	002764	104004					ERR+DS				;FILE-UNSAFE NOT SET BY RPM3, B11=1
654	002766						LOOP	DSF7,DSF7P			
655											
656	003006	005077	005350			DSF10:	CLR	DRPM3			
657	003012	005077	005304				CLR	DRPDS			
658	003016	052777	004000	005336			BIS	#B11,DRPM3			;SET M. FILE UNSAFE
659	003024	042777	004000	005330			BIC	#B11,DRPM3			;CLEAR M. FILE UNSAFE
660	003032	032777	001000	005262			BIT	#B9,DRPDS			;TEST FILE-UNSAFE=0
661	003040	001405					BEQ	DSF10P			;OK IF CLEAR
662	003042	104004					ERR+DS				;FILE-UNSAFE NOT CLEARED BY RPM3,B11=0
663	003044						LOOP	DSF10,DSF10P			
664											
665	003064	005077	005272			DSF11:	CLR	DRPM3			
666	003070	005077	005226				CLR	DRPDS			
667	003074	052777	100000	005260			BIS	#B15,DRPM3			;SET M. READ ONLY
668	003102	032777	000400	005212			BIT	#B8,DRPDS			;TEST WRITE-PROTECTED=1
669	003110	001005					BNE	DSF11P			;OK IF SET
670	003112	104004					ERR+DS				;WRITE-PROTECTED NOT SET BY RPM3,BIT15=1
671	003114						LOOP	DSF11,DSF11P			
672											
673	003134	005077	005222			DSF12:	CLR	DRPM3			
674	003140	005077	005156				CLR	DRPDS			
675	003144	052777	100000	005210			BIS	#B15,DRPM3			;SET M. READ ONLY
676	003152	042777	100000	005202			BIC	#B15,DRPM3			;CLEAR M. READ ONLY
677	003160	032777	000400	005134			BIT	#B8,DRPDS			;TEST WRITE-PROTECTED
678	003166	001405					BEQ	DSF12P			;OK IF CLEAR
679	003170	104004					ERR+DS				;WRITE-PROTECTED NOT CLEARED BY RPM3,BIT15=0
680	003172						LOOP	DSF12,DSF12P			
681											
682	003212	012767	000001	005046		DSF13:	MOV	#1,GOOD			;START WITH M. ATTN IN GOOD
683	003220	005077	005136			DSL13:	CLR	DRPM3			
684	003224	005077	005072				CLR	DRPDS			
685	003230	016777	005032	005120			MOV	GOOD,DRPM2			;SET RELEVANT M. ATTN BIT
686	003236	126777	005024	005056			CMPB	GOOD,DRPDS			;TEST RELEVANT ATTN BIT
687	003244	001410					BEQ	DSF13P			;PROCEEDS IF OK
688	003246	117767	005050	005014			MOV#	DRPDS,BAD			;PUT RPDS7-0 ON BAD BOR DISPLAY
689	003254	104001					ERR+GB				;M. ATTN DID NOT SET CORRECT ATTN BIT
690	003256						LOOP	DSL13,DSF13P			
691	003276	000241					CLC				;CLEAR LINK
692	003300	006167	004762				ROL	GOOD			;SHIFT ATTN TEST BIT
693	003304	032767	000400	004754			BIT	#B8,GOOD			;FINISHED 7-0?
694	003312	001742					BEQ	DSL13			;RECYCLE IF NO


```

695
696
697
698 003314 000005
699 003316 052777 100000 005036 ERF1: RESET ;CLEAR THE FREE WORLD
700 003324 112777 000003 005000 MOVB #3,ARPC ;SET M. READ ONLY
701 003332 004767 002702 JSR PC,T3P ;WRITE + GO TO CONTROL REGISTER
702 003336 032777 100000 004762 BIT #B15,ARPER ;3 TP
703 003344 001001 BNE ERN1 ;IS WRITE VIOLATION SET?
704 003346 104010 ERR+ER ;PROCEED IF YES
705 003350 004767 002720 ERN1: JSR PC,HRDER ;WRITE VIOLATION NOT SET BY WRITE FUNCTION
706 003354 LOOP ERF1,ERF1P ;TEST INCLUSIVE ERROR BITS
707
708 003374 005077 004726 ERF2: CLR ARPER
709 003400 052777 004000 004754 BIS #B11,ARPM3 ;SET M. FILE UNSAFE.
710 003406 112777 000003 004716 MOVB #3,ARPC ;TRY TO GO
711 003414 004767 002620 JSR PC,T3P
712 003420 032777 040000 004704 BIT #B14,ARPC ;IS FILE UNSAFE SET?
713 003426 001001 BNE ERN2 ;PROCEED
714 003430 104010 ERR+ER ;FILE UNSAFE ERROR BIT NOT SET
715 003432 004767 002636 ERN2: JSR PC,HRDER ;TEST INCLUSIVE ERROR BITS
716 003436 LOOP ERF2,ERF2P
717
718 003456 005077 004660 ERF3: CLR ARPCA
719 003462 005067 004604 CLR DATA ;START AT CAR=0
720 003466 005077 004634 ERL3: CLR ARPER ;LOAD CAR WITH DATA
721 003472 016777 004574 004642 MOV DATA,ARPCA
722 003500 012777 000003 004624 MOV #3,ARPC
723 003506 004767 002526 JSR PC,T3P
724 003512 032777 020000 004606 BIT #B13,ARPER ;IS NXT CYLINDER SET?
725 003520 001405 BEQ ERL3P ;IF NO - PROCEED
726 003522 104012 ERR+DD+ER ;NXT CYLINDER SET BY A LEGAL ADDRESS
727 003524 LOOP ERL3,ERL3P
728 003544 005267 004522 INC DATA ;ADVANCE TO NEXT ADDRESS
729 003550 022767 000313 004514 CMP #203.,DATA ;ENTERED INTO ILLEGAL AREA?
730 003556 001343 BNE ERL3 ;IF NOT - RECYCLE
731
732 003560 005077 004542 ERF4: CLR ARPER ;MOVE ILLEGAL ADDRESS INTO CAR
733 003564 016777 004502 004550 MOV DATA,ARPCA
734 003572 012777 000003 004532 MOV #3,ARPC
735 003600 004767 002434 JSR PC,T3P
736 003604 032777 020000 004514 BIT #B13,ARPER ;IS NXT CYLINDER SET?
737 003612 001001 BNE ERN4 ;IF YES - PROCEED
738 003614 104012 ERR+DD+ER ;NXT CYLINDER SET BY AN ILLEGAL ADDRESS
739 003616 004767 002452 ERN4: JSR PC,HRDER
740 003622 LOOP ERF4,ERF4P
741 003642 005267 004424 INC DATA ;ADVANCE TO NEXT ADDRESS
742 003646 032767 000400 004416 BIT #B8,DATA ;DATA=400?
743 003654 001741 BEQ ERF4 ;IF NOT RECYCLE
744
745 003656 005077 004464 ERF5: CLR ARPDA ;START WITH TAR=0
746 003662 005067 004404 CLR DATA
747 003666 005077 004434 ERL5: CLR ARPER ;LOAD TAR WITH DATA
748 003672 116777 004374 004450 MOVB DATA,ARPDA1
749 003700 012777 000003 004424 MOV #3,ARPC
750 003706 004767 002326 JSR PC,T3P

```


J02

.MAIN. MACY11 27(732) 08-SEP-76 08:58 PAGE 22
DZRPEA.P11

751	003712	032777	010000	004406
752	003720	001405		
753	003722	104012		
754	003724			
755	003744	005267	004322	
756	003750	022767	000024	004314
757	003756	001343		

BIT	#B12,ORPER
BEQ	ERLSP
ERR+DD+ER	
LOOP	ERL5,ERLSP
INC	DATA
CMP	#20.,DATA
BNE	ERL5

; IS NXT TRACK SET?
; NO
; YES - NXT TRACK SET BY LEGAL ADDRESS
; ADVANCE TRACK
; DONE ALL LEGAL?
; CONTINUE


```

799
800 004256 005077 004044          ERF11: CLR      JRPER
801 004262 004767 001706          JSR      PC,CLRP      ;WCOV=1
802 004266 012777 000003 004036  MOV      #3,JRPCS      ;GO
803 004274 004767 001740          JSR      PC,T3P
804 004300 032777 002000 004020  BIT      #B10,JRPER    ;IS PROGRAM ERROR SET?
805 004306 001001                BNE      ERN11         ;YES
806 004310 104010                ERR+ER    ;PRGM ERROR NOT SET BY WC=0 AND GO
807 004312 004767 001756          ERN11: JSR      PC,HRDER
808 004316                LOOP     ERF11,ERF11P
809
810 004336 000240          ERF12: NOP
811
812 004340 000240          ERF13: NOP      ;LATER, BABY. FORMAT ERR.
813
814 004342 005077 003760          ERF14: CLR      JRPER
815 004346 012777 004003 003756  MOV      #4003,JRPCS   ;WRITE HEADER IN PDP-11 MODE
816 004354 004767 001660          JSR      PC,T3P
817 004360 032777 000400 003740  BIT      #B8,JRPER     ;TEST MODE ERROR
818 004366 001001                BNE      ERN14         ;PROCEED IF SET
819 004370 104010                ERR+ER    ;MODE ERROR NOT SET BY HEADER OP IN PDP-11 MODE
820 004372 004767 001676          ERN14: JSR      PC,HRDER
821 004376                LOOP     ERF14,ERF14P
822
823 004416 000240          ERF15: NOP      ;DO LATER. L.P. ERR.
824
825 004420 000240          ERF16: NOP      ;DO LATER. WD. PAR. ERR.
826
827 004422 000240          ERF17: NOP      ;DO LATER. CHKSUM ERR.
828
829 004424 000240          ERF20: NOP      ;DO LATER. WR. CHK. ERR.
830
831 004426 000240          ERF21: NOP      ;DO LATER. NXM ERR.
832
833 004430 000240          ERF22: NOP      ;DO LATER. E.O.P. ERR.
834
835

```

```

836
837
838
839 004432 012767 000001 003632 CSF1:  MOV  #1,DATA      ;START WITH ATTN0
840 004440 012777 000340 003736      MOV  #P7,ARPSV   ;SETUP PRIORITY
841 004446 012777 004536 003726      MOV  #CSN1,ARPIV ;SETUP RETURN VECTOR
842 004454 004767 001514      JSR  PC,CLRP     ;CLEAR RP11
843 004460 005067 173312      CLR  PS         ;CLEAR PRIORITY
844 004464 052777 020000 003640      BIS  #B13,ARPCS ;ENABLE ATTN PI
845 004472 016777 003574 003622      MOV  DATA,ARPDS ;SET ATTN
846 004500 000240      NOP            ;AND
847 004502 000240      NOP            ;WAIT
848 004504 012767 000340 173264      MOV  #P7,PS     ;LOCKOUT PI
849 004512 104024      ERR+CS+DS     ;NO ATTN PI WITH ATTN & ATTN PI ENABLE SET
850 004514
851 004534 000401      LOOP CSL1,CSF1P
      BR    .+4

```


852										
853	004536	022626			CSN1:	POPPOP				;RESTORE STACK POINTER BECAUSE OF PI
854	004540	000241				CLC				;CLEAR LINK
855	004542	006167	003524			ROL	DATA			;SHIFT ATTN BIT
856	004546	032767	000400	003516		BIT	#B8,DATA			;FINISHED?
857	004554	001737				BEQ	CSL1			;RECYCLE IF NO
859										
859	004556	004767	001412		CSF2:	JSR	PC,CLRP			;CLEAR RP11
860	004562	012777	004620	003612		MOV	#CSE2,ARPIV			;ERROR RETURN ON PI
861	004570	005067	173202			CLR	PS			;ALLOW INTERRUPTS
862	004574	012777	000377	003520		MOV	#377,ARPDS			;SET ALL ATTNS
863	004602	000240				NOP				;AND
864	004604	000240				NOP				;WAIT
865	004606	000240				NOP				;AWHILE
866	004610	012767	000340	173160		MOV	#P7,PS			;LOCKOUT PI AND
867	004616	000411				BR	CSF2P			;SKIP ERROR
868	004620	012767	000340	173150	CSE2:	MOV	#P7,PS			;LOCKOUT PI
869	004626	022626				POPPOP				;RESTORE STACK POINTER
870	004630	104024				ERR+DS+CS				;ATTN PI WITH ATTN PI ENABLE=0
871	004632					LOOP	CSF2,CSF2P			
872										
873	004652	004767	001316		CSF3:	JSR	PC,CLRP			;CLEAR RP11
874	004656	012777	004734	003516		MOV	#CSN3,ARPIV			;RETURN VECTOR
875	004664	005067	173106			CLR	PS			;ALLOW PI
876	004670	052777	000100	003434		BIS	#B6,ARPCS			;ENABLE READY PI
877	004676	000240				NOP				;WAIT FOR
878	004700	000240				NOP				;PI
879	004702	012767	000340	173066		MOV	#P7,PS			;LOCKOUT PI
880	004710	104020				ERR+CS				;NO PI ON READY WITH RDY PI ENABLE=1
881	004712					LOOP	CSF3,CSF3P			
882	004732	000401				BR	CSF4			
883	004734	022626			CSN3:	POPPOP				;RESTORE SP
884	004736	004767	001232		CSF4:	JSR	PC,CLRP			;RESET RP11
885	004742	012777	004772	003432		MOV	#CSE4,ARPIV			;ERROR RETURN
886	004750	005067	173022			CLR	PS			;ALLOW PI
887	004754	000240				NOP				;AND
888	004756	000240				NOP				;WAIT
889	004760	000240				NOP				;FOR IT
890	004762	012767	000340	173006		MOV	#P7,PS			;LOCKOUT PI
891	004770	000406				BR	CSF4P			;SKIP ERROR
892	004772	022626			CSE4:	POPPOP				;RESTORE SP
893	004774	104020				ERR+CS				;READY PI WITH RDY PI ENABLE=0
894	004776					LOOP	CSF4,CSF4P			
895										
896	005016	004767	001152		CSF5:	JSR	PC,CLRP			;RESET RP11
897	005022	012777	005106	003352		MOV	#CSN5,ARPIV			;PI RETURN VECTOR
898	005030	052777	000100	003274		BIS	#B6,ARPCS			;ENABLE ERROR PI
899	005036	005067	172734			CLR	PS			;LOWER PRIORITY
900	005042	052777	100000	003262		BIS	#B15,ARPCS			;SET ERROR
901	005050	000240				NOP				;AND
902	005052	000240				NOP				;WAIT
903	005054	012767	000340	172714		MOV	#P7,PS			;LOCKOUT PI


```

953
954
955          ;RPCA FUNCTION TEST
956
957 005404 004767 000564 CAF1: JSR PC,CLRP          ;RESET
958 005410 005067 002652          CLR GOOD          ;START DATA AT CYLINDER 0
959 005414 116777 002646 002736 CAL1: MOVB GOOD,ARPM21 ;LOAD M. C.A.
960 005422 117767 002716 002640          MOVB ARPCA1,BAD ;READ CURRENT CYLINDER
961 005430 026767 002632 002632          CMP GOOD,BAD ;OK?
962 005436 001405          BEQ CAF1P          ;SKIP IF YES
963 005440 104001          ERR+GB          ;CURRENT CYLINDER NOT= TO MAINT. CYL. ADD.
964 005442          LOOP CAF1,CAF1P
965 005462 005267 002600          INC GOOD          ;ADVANCE CYL. COUNT
966 005466 026727 002574 000001          CMP GOOD,#1 ;FINISHED?
967 005474 001347          BNE CAL1          ;NO
968
969          ;RPDA FUNCTION TEST
970
971 005476 004767 000472 DAF1: JSR PC,CLRP          ;RESET
972 005502 005067 002560          CLR GOOD          ;START DATA AT SECTOR 0
973 005506 012700 000001          MOV #1,R0
974 005512 004767 000532          JSR PC,INDEXP ;1 INDEX PULSE
975 005516 052777 000400 002636 DAL1: BIS #88,ARPM3 ;1 SECTOR PULSE
976 005524 042777 000400 002630          BIC #88,ARPM3
977 005532 017767 002610 002530          MOV ARPDA,BAD ;READ SECTOR
978 005540 006067 002524          ROR BAD          ;RIGHT-
979 005544 006067 002520          ROR BAD          ;JUSTIFY
980 005550 006067 002514          ROR BAD          ;S.O.T.
981 005554 006067 002510          ROR BAD          ;REGISTER
982 005560          ANDI 17,BAD
983 005566 026767 002476 002472          CMP BAD,GOOD ;CHECK SOT
984 005574 001405          BEQ DAL1P          ;SKIP IF OK
985 005576 104001          ERR+GB ;SECTORS INDEX PULSES NOT COUNTING SOT
986 005600          LOOP DAL1,DAL1P
987 005620 005267 002442          INC GOOD          ;ADVANCE SECTOR
988 005624 022767 000012 002434          CMP #12,GOOD ;DONE?
989 005632 001331          BNE DAL1          ;NO
990
991 005634 012700 000001 DAF2: MOV #1,R0 ;1 INDEX PULSE
992 005640 004767 000404          JSR PC,INDEXP ;1 SECTOR PULSE
993 005644 052777 000400 002510          BIS #88,ARPM3
994 005652 042777 000400 002502          BIC #88,ARPM3
995 005660 005067 002402          CLR GOOD
996 005664 017767 002456 002376          MOV ARPDA,BAD
997 005672 006067 002372          ROR BAD
998 005676 006067 002366          ROR BAD
999 005702 006067 002362          ROR BAD
1000 005706 006067 002356          ROR BAD
1001 005712          ANDI 17,BAD
1002 005720 026767 002344 002340          CMP BAD,GOOD ;SOT=0?
1003 005726 001405          BEQ DAF2P          ;SKIP IF YES
1004 005730 104001          ERR+GB ;INDEX PULSE DID NOT CLEAR SOT
1005 005732          LOOP DAF1,DAF2P
1006
1007
1008 005752 005367 002332          DEC BTOG

```

1009	005756	001007				BNE	TSTND
1010	005760	012700	007672			MOV	#PMS,RO
1011	005764	004767	002234			JSR	PC,TYP0UT
1012	005770	012767	000062	002312		MOV	#50,BTOG
1013	005776	012706	000600		TSTND:	MOV	#START,SP
1014	006002	012767	000340	171766		MOV	#P7,CC
1015	006010	000167	172604			JMP	T1
1016							
1017							
1018							


```

1019
1020           ;SUBROUTINES
1021
1022 006014 012567 002274 BITST: MOV (R5)+,MASK ;FETCH DATA MASK
1023 006020 012504          MOV (R5)+,R4 ;FETCH REGISTER UNDER TEST
1024 006022 010567 002272          MOV R5,LERR ;FETCH ERROR CALL POINTER
1025 006026 062705 000004          ADD #4,R5 ;ADVANCE RETURN VECTOR OVER ERROR CALL
1026 006032 012703 000001          MOV #1,R3 ;INITIALIZE DATA WORD
1027 006036 004767 000016 BTL1: JSR PC,BTL2 ;TEST WITH WALKING 0
1028 006042 004767 000012          JSR PC,BTL2 ;TEST WITH WALKING 1
1029 006046 000241          CLC ;CLEAR LINK
1030 006050 006103          ROL R3 ;SHIFT TEST BIT
1031 006052 005703          TST R3 ;SEE IF IT PASSED INTO C-BIT
1032 006054 001370          BNE BTL1 ;IF NOT TEST AGAIN
1033 006056 000205          RTS R5 ;IF SO - EXIT
1034 006060 005103 BTL2: COM R3 ;MAKE WALKING 1 INTO WALKING 0 AND VICE VERSA
1035 006062 010367 002200          MOV R3,GOOD ;PUT PATTERN INTO GOOD
1036 006066          AND GOOD,MASK
1037 006104 016774 002156 000000          MOV GOOD,2(R4) ;SEND DATA TO RELEVANT REGISTER
1038 006112 017467 000000 002150          MOV 2(R4),BAD ;READ DATA INTO BAD FROM RELEVANT REGISTER
1039 006120          AND BAD,MASK
1040 006136 026767 002124 002124          CMP GOOD,BAD ;COMPARE SEND VS. RECEIVE
1041 006144 001406          BEQ BTN2 ;SKIP OVER ERROR ON GOOD COMPARE
1042 006146 004777 002146          JSR PC,2LERR ;CALL RELEVANT ERROR
1043 006152          LOOP BTL2+2,BTN2
1044 006172 000207          RTS ;IF NOT- EXIT
1045
1046
1047
1048 006174 012700 000002 CLRP: MOV #2,R0
1049 006200 012777 000001 002124          MOV #1,2RPCS ;RESET RP11
1050 006206 004767 000010          JSR PC,TIMEP ;GENERATE 2 TIME PULSES
1051 006212 052777 020000 002142          BIS #B13,2RPM3 ;SET UNIT ON-LINE
1052 006220 000207          RTS ;EXIT
1053
1054 006222 052777 160001 002132 TIMEP: BIS #B15+B14+B13+B0,2RPM3 ;GENERATE A TIME PULSE
1055 006230 005300          DEC R0 ;DEFLATE COUNT
1056 006232 005700          TST R0 ;DONE?
1057 006234 001372          BNE TIMEP ;0=YES
1058 006236 000207          RTS PC
1059 006240 012700 000003 T3P: MOV #3,R0
1060 006244 000167 177752          JMP TIMEP ;3 TIME PULSES
1061
1062
1063

```



```

1099
1100           ;SEEK ROUTINE.   SR7=0=CYLINDER. SR12=8= TRACK
1101
1102 006414 012777 000001 001710 SS:   MOV   #1,ARPC
1103 006422 116777 171142 001712   MOV  SR,ARPCA
1104 006430 116777 171135 001712   MOV  SR+1,ARPD1
1105 006436 012777 000011 001666   MOV  #11,ARPCS
1106 006444 012767 100000 001624   MOV  #815,TOG1
1107 006452 032777 000001 001642 SL:   BIT   #80,ARPD5
1108 006460 001404           BEQ   SSG0
1109 006462 005267 001610           INC   TOG1
1110 006466 001371           BNE   SL
1111 006470 000751           BR    SS
1112 006472 017767 001644 001566 SSG0: MOV   ARPCA,GOOD
1113 006500 017767 001642 001562   MOV   ARPDA,BAD
1114 006506 104041           ERR+GB+DA
1115 006510 000240           NOP
1116 006512 000740           BR    SS
1117
1118           ;READ/WRITE ROUTINE
1119
1120 006514 016767 171050 001570 SRW:  MOV   SR,TEMP1           ;START WITH # OF WORDS IN SR.(1-5777)
1121 006522 005467 001564           NEG   TEMP1
1122 006526 000000           XX           ;PUT DATA PATTERN IN SR AND CONTINUE
1123 006530 016700 001556           MOV   TEMP1,RO
1124 006534 012701 012000           MOV   #12000,R1
1125 006540 016721 171024           DL1:  MOV   SR,(R1)+
1126 006544 005200           INC   RO
1127 006546 001374           BNE   DL1
1128 006550 012777 000001 001554 WGO:  MOV   #1,ARPCS
1129 006556 012777 012000 001554   MOV   #12000,ARPBA
1130 006564 016777 001522 001544   MOV   TEMP1,ARPC
1131 006572 012777 000003 001532   MOV   #3,ARPCS
1132 006600 000240           NOP
1133 006602 000240           NOP
1134 006604 000240           NOP
1135 006606 000240           NOP
1136 006610 000240           NOP
1137 006612 000240           NOP
1138 006614 032777 000200 001510 WL1:  BIT   #87,ARPCS
1139 006622 001774           BEQ   WL1
1140 006624 016777 001462 001504   MOV   TEMP1,ARPC
1141 006632 012777 020000 001500   MOV   #20000,ARPBA
1142 006640 012777 000005 001464   MOV   #5,ARPCS
1143 006646 000240           NOP
1144 006650 000240           NOP
1145 006652 000240           NOP
1146 006654 000240           NOP
1147 006656 000240           NOP
1148 006660 000240           NOP
1149 006662 032777 000200 001442 RL1:  BIT   #87,ARPCS
1150 006670 001774           BEQ   RL1
1151 006672 000240           NOP
1152 006674 000240           NOP
1153 006676 000240           NOP
1154 006700 012701 012000           MOV   #12000,R1

```

H03

.MAIN. MACY11 27(732) 08-SEP-76 08:58 PAGE 33
DZRPEA.P11

1155	006704	012702	020000		MOV	#20000,R2
1156	006710	016703	001376		MOV	TEMP1,R3
1157	006714	012167	001346	CL1:	MOV	(R1)+,GOOD
1158	006720	012267	001344		MOV	(R2)+,BAD
1159	006724	026767	001336	001336	CMP	GOOD,BAD
1160	006732	001401			BEQ	.+4
1161	006734	104001			ERR+GB	
1162	006736	005203			INC	R3
1163	006740	001365			BNE	CL1
1164	006742	0C0702			BR	WGO
1165						


```

1209
1210
1211 007164 032767 020000 170376 LERCHK: BIT      #B13,SR      ;SR13(1)=LOOP ON ERROR
1212 007172 001002          BNE      .+6      ;EXIT IF 1
1213 007174 062716 000004          ADD      #4,(SP) ;ADVANCE RETURN VECTOR OVER LOOP JUMP
1214 007200 000207          LERXT:  RTS      PC      ;EXIT
1215
1216 007202 032767 010000 170360 LUPCHK: BIT      #B12,SR      ;SR12(1)=LOOP ALWAYS
1217 007210 001002          BNE      .+6      ;EXIT IF 1
1218 007212 062716 000004          ADD      #4,(SP) ;ADVANCE RETURN VECTOR OVER LOOP JUMP
1219 007216 000207          RTS      PC      ;EXIT
1220
1221 007220 012700 007761          GBTYPE: MOV      #GDMS,RO      ;POINTER TO "GOOD"
1222 007224 004767 000774          JSR      PC,TYPOUT ;PRINT MESSAGE
1223 007230 016700 001032          MOV      GOOD,RO      ;FETCH C(GOOD)
1224 007234 004767 000246          JSR      PC,PNT OCT ;PRINT OCTAL NUMBER
1225 007240 012700 007771          MOV      #BDMS,RO      ;POINTER TO "BAD"
1226 007244 004767 000754          JSR      PC,TYPOUT ;PRINT MESSAGE
1227 007250 016700 001014          MOV      BAD,RO      ;FETCH C(BAD)
1228 007254 004767 000226          JSR      PC,PNT OCT ;PRINT OCTAL NUMBER
1229 007260 000207          RTS      PC      ;EXIT
1230
1231 007262 012700 007702          CSTYPE: MOV      #CSMS,RO      ;POINTER TO "STATUS"
1232 007266 004767 000732          JSR      PC,TYPOUT ;PRINT TEXT
1233 007272 017700 001034          MOV      @RPCS,RO      ;FETCH C(RPCS)
1234 007276 004767 000204          JSR      PC,PNT OCT ;PRINT OCTAL NUMBER
1235 007302 000207          RTS      PC      ;EXIT
1236 007304 012700 010011          DATYPE: MOV      #DAMS,RO      ;POINTER TO "DATA"
1237 007310 004767 000710          JSR      PC,TYPOUT ;PRINT TEST
1238 007314 016700 000752          MOV      DATA,RO      ;FETCH C(DATA)
1239 007320 004767 000162          JSR      PC,PNT OCT ;PRINT OCTAL NUMBER
1240 007324 000207          RTS      PC      ;EXIT
1241 007326 012700 007712          ERTYPE: MOV      #ERMS,RO      ;PRINT
1242 007332 004767 000666          JSR      PC,TYPOUT ;"RPER="
1243 007336 017700 000764          MOV      @RPER,RO      ;PRINT
1244 007342 004767 000140          JSR      PC,PNT OCT ;RPER
1245 007346 000207          RTS      PC      ;EXIT
1246 007350 012700 007722          DSTYPE: MOV      #DSMS,RO      ;PRINT
1247 007354 004767 000644          JSR      PC,TYPOUT ;"RPDS="
1248 007360 017700 000736          MOV      @RPDS,RO      ;PRINT
1249 007364 004767 000116          JSR      PC,PNT OCT ;RPDS
1250 007370 000207          RTS      PC      ;EXIT
1251 007372 012700 007732          ADTYPE: MOV      #CYMS,RO      ;PRINT
1252 007376 004767 000622          JSR      PC,TYPOUT ;"CAR="
1253 007402 117700 000734          MOV      @RPCA,RO      ;PRINT
1254 007406 004767 000450          JSR      PC,OCTPNT ;CAR
1255 007412 012700 007741          MOV      #TAMS,RO      ;PRINT
1256 007416 004767 000602          JSR      PC,TYPOUT ;"TAR="
1257 007422 117700 000722          MOV      @RPDA1,RO      ;PRINT
1258 007426 004767 000430          JSR      PC,OCTPNT ;TAR
1259 007432 012700 007751          MOV      #SEMS,RO      ;PRINT
1260 007436 004767 000562          JSR      PC,TYPOUT ;"SAR="
1261 007442 117700 000700          MOV      @RPDA,RO      ;PRINT
1262 007446          ANDI      17,RO
1263 007452 004767 000404          JSR      PC,OCTPNT ;SAR
1264 007456 000207          RTS      PC      ;EXIT

```


1265							
1266	007460	012700	007636		SUBER:	MOV	#SUBRMS,RO
1267	007464	004767	000534			JSR	PC, TYP0UT
1268	007470	016600	000022			MOV	22(SP),RO
1269	007474	162700	000004			SUB	#4,RO
1270	007500	004767	000356			JSR	PC, OCTPNT
1271	007504	000207				RTS	PC
1272	007506	012767	000006	000570	PNT0CT:	MOV	#6, TOG4
1273	007514	010067	000566		OLUP:	MOV	RO, WORK
1274	007520					ANDI	7, WORK
1275	007526	062767	000060	000552		ADD	#60, WORK
1276	007534	016746	000546			MOV	WORK, -(SP)
1277	007540	006000				ROR	RO
1278	007542	006000				ROR	RO
1279	007544	006000				ROR	RO
1280	007546	005367	000532			DEC	TOG4
1281	007552	003360				BGT	OLUP
1282	007554	012767	000005	000522		MOV	#5, TOG4
1283	007562	012600				MOV	(SP)+, RO
1284	007564					ANDI	61, RO
1285	007570	004767	000316			JSR	PC, TTO
1286	007574	012600			TOLUPO:	MOV	(SP)+, RO
1287	007576	004767	000310			JSR	PC, TTO
1288	007602	005367	000476			DEC	TOG4
1289	007606	003372				BGT	TOLUPO
1290	007610	000207				RTS	PC

1337									
1338		010022							
1339									.EVEN
1340	010022	010446		SAV05:	MOV	R4,-(SP)			;SAVE R0-R5 ON THE STACK
1341	010024	010346			MOV	R3,-(SP)			;R5 WAS STACKED BY THE JSR
1342	010026	010246			MOV	R2,-(SP)			;R4-R3 ARE STACKED ABOVE IT
1343	010030	010146			MOV	R1,-(SP)			;WITH R0 ON TOP
1344	010032	010046			MOV	R0,-(SP)			;R5 HOLDS THE RETURN PC, BUT AN
1345	010034	000115			JMP	(R5)			;RTS WOULD POP THE STACK-SO JUMP OUT
1346									
1347	010036	005726		REST05:	TST	(SP)+			;MOVE SP OVER WORD SAVED BY JSR
1348	010040	012600			MOV	(SP)+,R0			;R0-4
1349	010042	012601			MOV	(SP)+,R1			;ARE POPPED
1350	010044	012602			MOV	(SP)+,R2			;IN LIFO
1351	010046	012603			MOV	(SP)+,R3			;SEQUENCE
1352	010050	012604			MOV	(SP)+,R4			;R5 IS POPPED BY THE RTS AND
1353	010052	000205			RTS	R5			;THE PC IS TAKEN FROM R5
1354									
1355	010054	012702	000012	DECPNT:	MOV	#10,R2			;DIVISOR OF 10 FOR DECIMAL PRINT
1356	010060	000402			BR	DECREM			;PROCESS AND PRINT NUMBER
1357	010062	012702	000010	OCTPNT:	MOV	#8,R2			;DIVISOR OF 8 FOR OCTAL PRINT
1358	010066	004767	000060	DECREM:	JSR	PC, IDIVR			;DIVIDE (R0) BY (R2) WITH REMAINDER IN R1
1359	010072	010146			MOV	R1,-(SP)			;STACK REMAINDER
1360	010074	005700			TST	R0			;HAS NUMBER DEFLATED BELOW RADIX?
1361	010076	001402			BEQ	POPTT			;YES - POP AND PRINT
1362	010100	004767	177762		JSR	PC, DECREM			;NO - DIVIDE NUMBER BY RADIX
1363	010104	012600		POPTT:	MOV	(SP)+,R0			;POP NUMBER FROM STACK
1364	010106	062700	000060		ADD	#60,R0			;MAKE ASCII
1365	010112	032767	040000	167450 TTO:	BIT	#B14,SR			;IF SR14=1,DELETE TYPEOUT
1366	010120	001010			BNE	TTOLF-2			;EXIT
1367	010122	010067	167440		MOV	R0,TPB			;PRINT CONTENTS OF R0
1368	010126	105767	167432	TTOLF:	TSTB	TPS			;DONE YET?
1369	010132	100375			BPL	TTOLF			;NO - KEEP LOOPING
1370	010134	022700	000015		CMP	#CR,R0			;WAS CHARACTER A CR?
1371	010140	001401			BEQ	TTOLF			;YES - PRINT LINE FEED
1372	010142	000207			RTS	PC			;RETURN TO POPTT OR MAIN PROGRAM
1373	010144	012700	000012	TTOLF:	MOV	#LF,R0			;PRINT LF
1374	010150	000760			BR	TTO			;EXECUTE PRINT
1375									
1376	010152	010067	000116	IDIVR:	MOV	R0,DIVID			;SAVE DIVIDEND
1377	010156	005000			CLR	R0			;CLEAR QUOTIENT AREA
1378	010160	005001			CLR	R1			;CLEAR ACCUM.
1379	010162	060201		DIVLP:	ADD	R2,R1			;ADD DIVISOR TO ACCUM.
1380	010164	020167	000104		CMP	R1,DIVID			;COMPARE TO DIVIDEND
1381	010170	100002			BPL	DVEND			;WHEN ACCUM PASSES DIVIDEND - EXIT
1382	010172	005200			INC	R0			;INCREMENT QUOTIENT THEN
1383	010174	000772			BR	DIVLP			;ADD AGAIN
1384	010176	001003		DVEND:	BNE	DIVN1			;JUMP TO GET REMAINDER
1385	010200	005200			INC	R0			;NO REMAINDER - INCREMENT QUOTIENT
1386	010202	005001			CLR	R1			;REMAINDER OF 0
1387	010204	000207			RTS	PC			;EXIT
1388	010206	160167	000062	DIVN1:	SUB	R1,DIVID			;FANCY FINAGLING TO
1389	010212	060267	000056		ADD	R2,DIVID			;DETERMINE THE REMAINDER
1390	010216	016701	000052		MOV	DIVID,R1			;REMAINDER TO R1
1391	010222	000207			RTS	PC			;EXIT WITH QUOTIENT IN R0

1392									
1393	010224	010046		TYP0UT:	MOV	RO, -(SP)		; STACK ADDRESS POINTER FOR MESSAGE	
1394	010226	117600	000000	TPOFCH:	MOV B	3(SP), RO		; FETCH ASCII BYTE	
1395	010232	022700	000100		CMP	#100, RO		; IS IT 3(TERMINATOR)?	
1396	010236	001411			BEQ	TPOUTX		; YES - EXIT	
1397	010240	022700	000046		CMP	#46, RO		; IS IT CRLF FLAG?	
1398	010244	001002			BNE	.+6		; NO	
1399	010246	012700	000015		MOV	#CR, RO		; YES - CHANGE DATA TO CRLF	
1400	010252	004767	177634		JSR	PC, TPO		; PRINT	
1401	010256	005216			INC	(SP)		; MOVE POINTER TO NEXT BYTE	
1402	010260	000762			BR	TPOFCH		; FETCH NEXT CHARACTER	
1403	010262	005726		TPOUTX:	POP			; POP STACK TO REACH RETURN VECTOR	
1404	010264	000207			RTS	PC		; EXIT	
1405									
1406	010266	000000		GOOD:	XX				
1407	010270	000000		BAD:	XX				
1408	010272	000000		DATA:	XX				
1409	010274	000000		DIVID:	XX				
1410	010276	000000		TOG1:	XX				
1411	010300	000000		TOG2:	XX				
1412	010302	000000		TOG3:	XX				
1413	010304	000000		TOG4:	XX				
1414	010306	000000		WORK:	XX				
1415	010310	000000		BTOG:	XX				
1416	010312	000000		TEMP1:	XX				
1417	010314	000000		MASK:	XX				
1418	010316	000000		ROTOG:	XX				
1419	010320	000000		LERR:	XX				
1420	010322	176710		RPDS:	176710				
1421	010324	176711		RPDS1:	176711				
1422	010326	176712		RPER:	176712				
1423	010330	176713		RPER1:	176713				
1424	010332	176714		RPCS:	176714				
1425	010334	176715		RPCS1:	176715				
1426	010336	176716		RPWC:	176716				
1427	010340	176720		RPBA:	176720				
1428	010342	176722		RPCA:	176722				
1429	010344	176723		RPCA1:	176723				
1430	010346	176724		RPDA:	176724				
1431	010350	176725		RPDA1:	176725				
1432	010352	176726		RPM1:	176726				
1433	010354	176727		RPM11:	176727				
1434	010356	176730		RPM2:	176730				
1435	010360	176731		RPM21:	176731				
1436	010362	176732		RPM3:	176732				
1437	010364	176733		RPM31:	176733				
1438	010366	176734		RPB1:	176734				
1439	010370	176735		RPB11:	176735				
1440	010372	176736		RPB2:	176736				
1441	010374	176737		RPB21:	176737				
1442	010376	176740		RPB3:	176740				
1443	010400	176741		RPB31:	176741				
1444	010402	000254		RPIV:	.WORD	254			
1445	010404	000256		RPSV:	.WORD	256			
1446	010406	000000		DEBUG:	XX				
1447		010606							

B04

MAIN. MACY11 27(732) 09-SEP-76 09:58 PAGE 40
DZARPEA.P11

1448
1449
1450

000001

.END

;THAT'S ALL FOLKS!

DSF5	002656	637#	645																	
DSF5P	002724	642	645#																	
DSF6	002734	646#																		
DSF7	002736	648#	655																	
DSF7P	002776	652	655#																	
DSL13	003220	683#	691	694																
DSMS	007722	1246	1313#																	
DSTYPE	007350	1191	1246#																	
DVEND	010176	1381	1384#																	
EHALT1	007160	1206#																		
ER =	000010	323#	407	547	704	714	726	738	753	765	780	792	806	919						
ERBT	001450	495#																		
ERF1	003316	699#	707																	
ERF1P	003364	707#																		
ERF10	004160	786#	795	797																
ERF10P	004232	795#																		
ERF11	004256	800#	809																	
ERF11P	004326	809#																		
ERF12	004336	810#																		
ERF13	004340	812#																		
ERF14	004342	814#	822																	
ERF14P	004406	822#																		
ERF15	004416	823#																		
ERF16	004420	825#																		
ERF17	004422	827#																		
ERF2	003374	708#	717																	
ERF2P	003446	717#																		
ERF20	004424	829#																		
ERF21	004426	831#																		
ERF22	004430	833#																		
ERF3	003456	718#																		
ERF4	003560	732#	741	743																
ERF4P	003632	741#																		
ERF5	003656	745#																		
ERF6	003760	759#	768	770																
ERF6P	004032	768#																		
ERF7	004056	772#																		
ERL3	003466	720#	728	730																
ERL3P	003534	725	728#																	
ERL5	003666	747#	755	757																
ERL5P	003734	752	755#																	
ERL7	004066	774#	782	784																
ERL7P	004134	779	782#																	
ERM =	104400	319#	1074	1078	1082															
ERMS	007712	1241	1310#																	
ERMS1	007612	1170	1292#																	
ERN1	003350	703	705#																	
ERN10	004216	791	793#																	
ERN11	004312	805	807#																	
ERN14	004372	818	820#																	
ERN2	003432	713	715#																	
ERN4	003616	737	739#																	
ERN6	004016	764	766#																	
ERR =	104000	318#	401	407	413	420	427	434	441	448	455	462	469	476						
		482	492	496	500	504	508	512	516	520	524	528	540	547						
		554	562	570	578	586	596	605	625	634	643	653	662	670						

P2	=	000100	296#															
P3	=	000140	297#															
P4	=	000200	298#	923														
P5	=	000240	299#	941														
P6	=	000300	300#															
P7	=	000340	301#	840	848	866	868	879	890	903	916	928	933	945	1014			
RESBA		002066	565#	572														
RESBAP		002126	568	572#														
RESCA		002136	573#	580														
RESCAP		002176	576	580#														
RESCS		001752	550#	556														
RESCSP		002006	553	556#														
RESDA		002206	581#	588														
RESDAP		002250	584	588#														
RESDS		001624	533#	542														
RESDSP		001700	539	542#														
RESER		001710	543#	549														
RESERP		001742	546	549#														
RESTOS		010036	1201	1347#														
RESWC		002016	557#	564														
RESWCP		002056	560	564#														
RLUP		006400	1093#	1096														
RLI		006662	1149#	1150														
ROTOG		010316	1084#	1088#	1091*	1095*	1418#											
RPBA		010340	424	426	508	565*	567	569	1129*	1141*	1427#							
RPBT		001500	503#															
RPB1		010366	466	468	520	1438#												
RPB11		010370	1439#															
RPB2		010372	473	475	524	1440#												
RPB21		010374	1441#															
RPB3		010376	480	528	1442#													
RPB31		010400	1443#															
RPCA		010342	431	433	512	573*	575	577	718*	721*	733*	1103*	1112	1253	1428#			
RPCA1		010344	960	1429#														
RPCS		010332	411	500	550*	552	616*	700*	710*	712	722*	734*	749*	761*	776*			
			788*	802*	815*	844*	876*	898*	900*	912*	925*	940*	942*	1049*	1072			
			1075	1080	1102*	1105*	1128*	1131*	1138	1142*	1149	1233	1424#					
RPCS1		010334	1425#															
RPDA		010346	438	440	516	581*	583	585	609	612*	745*	772*	775*	787*	977			
			996	1113	1261	1430#												
RPDA1		010350	748*	760*	1104*	1257	1431#											
RPDS		010322	399	492	538	592*	594	600*	603	613*	623	630*	632	638*	641			
			649*	651	657*	660	666*	668	674*	677	684*	686	688	845*	862*			
			938*	1107	1248	1420#												
RPDS1		010324	1421#															
RPER		010326	405	496	543*	545	702	708*	720*	724	732*	736	747*	751	759*			
			763	774*	778	786*	790	800*	804	814*	817	1243	1422#					
RPER1		010330	1423#															
RPIV		010402	841*	860*	874*	885*	897*	910*	924*	939*	1444#							
RPM1		010352	445	447	1432#													
RPM11		010354	1433#															
RPM2		010356	452	454	533*	685*	1434#											
RPM21		010360	959*	1435#														
RPM3		010362	459	461	534*	535*	537*	591*	593*	599*	601	602*	609*	615*	618*			
			619*	620*	629*	631*	637*	639*	640*	648*	650*	656*	658*	659*	665*			
			667*	673*	675*	676*	683*	699*	709*	975*	976*	993*	994*	1051*	1054*			

ADD	1025	1085	1092	1213	1218	1275	1364	1379	1389						
BEQ	400	406	412	418	425	432	439	446	453	460	467	474	481	539	546
	553	560	568	576	584	604	642	661	678	687	694	725	743	752	770
	779	797	857	962	984	1003	1041	1076	1081	1108	1139	1150	1160	1181	1184
	1127	1190	1193	1196	1199	1361	1371	1396							
BGT	1281	1289													
BIC	537	602	609	618	620	640	659	676	976	983	994	1002	1037	1040	1066
	1263	1275	1285												
BIS	593	609	615	619	631	639	650	658	667	675	699	709	844	876	898
	900	912	925	940	942	975	993	1051	1054	1065					
BIT	438	583	594	601	603	623	632	641	651	660	668	677	693	702	712
	724	736	742	751	763	769	778	790	796	804	817	856	1072	1075	1080
	1107	1138	1149	1167	1180	1183	1186	1189	1192	1195	1198	1204	1211	1216	1365
BNE	595	624	633	652	669	703	713	730	737	757	764	784	791	805	818
	967	989	1009	1032	1057	1069	1073	1089	1096	1110	1127	1163	1168	1205	1212
	1217	1366	1384	1398											
BPL	1369	1381													
BR	851	867	882	891	907	917	931	946	1083	1111	1116	1164	1356	1374	1383
	1402														
CLC	691	854	1029	1086	1093										
CLR	591	592	599	600	613	629	630	637	638	648	649	656	657	665	666
	673	674	683	684	708	718	719	720	732	745	746	747	759	772	773
	774	786	800	814	843	861	875	886	899	911	958	972	995	1377	1378
	1386														
CMP	411	552	729	756	783	961	966	983	988	1002	1040	1159	1370	1380	1395
	1397														
CMPB	686														
COM	1034	1037	1040												
DEC	1008	1055	1067	1088	1095	1280	1288								
EMT	318														
HALT	291	383													
INC	611	728	741	755	768	782	795	965	987	1109	1126	1162	1382	1385	1401
JMP	403	409	415	422	429	436	443	450	457	464	471	478	484	542	549
	556	564	572	580	588	598	607	628	636	645	655	664	672	681	691
	707	717	728	741	755	768	782	795	809	922	851	872	882	895	907
	921	931	935	950	965	987	1006	1015	1044	1060	1345				
JSR	403	409	415	422	429	436	443	450	457	464	471	478	484	492	496
	500	504	508	512	516	520	524	528	536	542	544	549	551	556	558
	564	566	572	574	580	582	588	598	607	609	611	617	622	626	628
	636	645	655	664	672	681	691	701	705	707	711	715	717	723	728
	735	739	741	750	755	762	766	768	777	782	789	793	795	801	803
	807	809	816	820	822	842	851	859	872	873	882	884	895	896	907
	909	921	922	931	934	937	950	957	965	971	974	987	992	1006	1011
	1027	1028	1042	1044	1050	1169	1171	1175	1177	1179	1182	1185	1188	1191	1194
	1197	1200	1201	1203	1222	1224	1226	1228	1232	1234	1237	1239	1242	1244	1247
	1249	1252	1254	1256	1258	1260	1263	1267	1270	1285	1287	1358	1362	1400	
MOV	394	395	396	419	426	433	440	447	454	461	468	475	533	534	543
	550	557	561	565	569	573	577	581	585	609	614	616	621	682	685
	721	722	733	734	749	761	775	776	787	788	802	815	839	840	841
	845	848	860	862	866	868	874	879	885	890	897	903	910	916	923
	924	928	933	938	939	941	945	973	977	991	996	1010	1012	1013	1014
	1022	1023	1024	1026	1035	1037	1038	1048	1049	1059	1084	1091	1102	1105	1106
	1112	1113	1120	1123	1124	1125	1128	1129	1130	1131	1140	1141	1142	1154	1155
	1156	1157	1158	1170	1172	1174	1176	1178	1202	1221	1223	1225	1227	1231	1233
	1236	1238	1241	1243	1246	1248	1251	1255	1259	1266	1268	1272	1273	1276	1282
	1283	1286	1340	1341	1342	1343	1344	1348	1349	1350	1351	1352	1355	1357	1359

.MAIN. MACY11 27(732) 08-SEP-76 08:58 PAGE 53
 DZRPEA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

MOV8	1363 535 1394	1367 609	1373 612	1376 688	1390 700	1393 710	1399 748	760	959	960	1103	1104	1253	1257	1261
NEG NOP	1121 646 878	810 887	812 888	823 889	825 901	827 902	829 913	831 914	833 915	846 926	847 927	863 543	864 944	865 1115	877 1132
RESET	1133 398	1134 404	1135 410	1136 416	1137 423	1143 430	1144 437	1145 444	1146 451	1147 458	1148 465	1151 472	1152 479	1153 698	
ROL ROR RTI RTS	692 978 1207 492	855 979	1030 980	1094 981	997	998	999	1000	1087	1277	1278	1279			
	1077 1372 1173	1079 1387 1269	1090 1391 1388	1097 1404	1214	1219	1229	1235	1240	1245	1250	1264	1271	1290	1353
SUB TRAP TST	319 399	405	417	424	431	445	452	459	466	473	480	538	545	559	567
TSTB .ABS .ASCII .BYTE .END .EVEN .MACR .REM .REPT .WORD	1031 575 265 1292 1303 1450 1338 329 1 379 385 1444	1056 1368 1297	1068 1304	1347 1307	1360 1310	1313 1316	1316 1319	1319 1322	1322 1325	1325 1328	1328 1331	1331 1334			
		333	339	348	353	359	367								
		386 1445	387	388	492	496	500	504	508	512	516	520	524	528	611

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*DZRPEA, DZRPEA, SEQ/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DZRPEA.P11
 RUN-TIME: 5 9 3 SECONDS
 RUN-TIME RATIO: 29/18=1.5
 CORE USED: 9K (18 PAGES)

