

# RP04

DISK PACK FORMATTER  
 MD-11-DZRPL-B

EP-DZRPL-B-DL-A

MAR 1976

COPYRIGHT © 1976

FICHE 1 OF 1

Made in U.S.A.

DZRPLB  
 SEQ

SEQ	NO.	NAME	START	END	...
01	000	...	...	...	...
02	001	...	...	...	...
03	002	...	...	...	...
04	003	...	...	...	...
05	004	...	...	...	...
06	005	...	...	...	...
07	006	...	...	...	...
08	007	...	...	...	...
09	008	...	...	...	...
10	009	...	...	...	...
11	010	...	...	...	...
12	011	...	...	...	...
13	012	...	...	...	...
14	013	...	...	...	...
15	014	...	...	...	...
16	015	...	...	...	...
17	016	...	...	...	...
18	017	...	...	...	...
19	018	...	...	...	...
20	019	...	...	...	...
21	020	...	...	...	...
22	021	...	...	...	...
23	022	...	...	...	...
24	023	...	...	...	...
25	024	...	...	...	...
26	025	...	...	...	...
27	026	...	...	...	...
28	027	...	...	...	...
29	028	...	...	...	...
30	029	...	...	...	...
31	030	...	...	...	...
32	031	...	...	...	...
33	032	...	...	...	...
34	033	...	...	...	...
35	034	...	...	...	...
36	035	...	...	...	...
37	036	...	...	...	...
38	037	...	...	...	...
39	038	...	...	...	...
40	039	...	...	...	...
41	040	...	...	...	...
42	041	...	...	...	...
43	042	...	...	...	...
44	043	...	...	...	...
45	044	...	...	...	...
46	045	...	...	...	...
47	046	...	...	...	...
48	047	...	...	...	...
49	048	...	...	...	...
50	049	...	...	...	...
51	050	...	...	...	...
52	051	...	...	...	...
53	052	...	...	...	...
54	053	...	...	...	...
55	054	...	...	...	...
56	055	...	...	...	...
57	056	...	...	...	...
58	057	...	...	...	...
59	058	...	...	...	...
60	059	...	...	...	...
61	060	...	...	...	...
62	061	...	...	...	...
63	062	...	...	...	...
64	063	...	...	...	...
65	064	...	...	...	...
66	065	...	...	...	...
67	066	...	...	...	...
68	067	...	...	...	...
69	068	...	...	...	...
70	069	...	...	...	...
71	070	...	...	...	...
72	071	...	...	...	...
73	072	...	...	...	...
74	073	...	...	...	...
75	074	...	...	...	...
76	075	...	...	...	...
77	076	...	...	...	...
78	077	...	...	...	...
79	078	...	...	...	...
80	079	...	...	...	...
81	080	...	...	...	...
82	081	...	...	...	...
83	082	...	...	...	...
84	083	...	...	...	...
85	084	...	...	...	...
86	085	...	...	...	...
87	086	...	...	...	...
88	087	...	...	...	...
89	088	...	...	...	...
90	089	...	...	...	...
91	090	...	...	...	...
92	091	...	...	...	...
93	092	...	...	...	...
94	093	...	...	...	...
95	094	...	...	...	...
96	095	...	...	...	...
97	096	...	...	...	...
98	097	...	...	...	...
99	098	...	...	...	...
100	099	...	...	...	...
101	100	...	...	...	...





IDENTIFICATION  
-----

PRODUCT CODE: MAINDEC-11-DZRPL-B-D  
PRODUCT NAME: RPO4 DISK PACK FORMATTER  
DATE CREATED: FEBRUARY 21, 1975  
MAINTAINER: DIAGNOSTIC GROUP  
AUTHOR: R. MOORE/C. HESS

COPYRIGHT (C) 1973, 1974, 1975 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM. EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

- 1.0 ABSTRACT
- 2.0 REQUIRMENTS
  - 2.1 EQUIPMENT
  - 2.2 STORAGE
  - 2.3 PRELIMINARY PROGRAMS
- 3.0 LOADING PROCEDURE
- 4.0 STARTING PROCEDURE
  - 4.1 STARTING ADDRESS
  - 4.2 RESTARTING ADDRESS
  - 4.3 RH11/RPO4 ADDRESSES
  - 4.4 SWITCH SETTINGS
- 5.0 OPERATING PHOCEDURE
  - 5.1 OPERATOR ACTION
  - 5.2 PROGRAM ACTION
  - 5.3 SUBROUTINE ABSTRACTS
    - 5.3.1 'RPO4'
    - 5.3.2 'DRINIT'
- 6.0 ERRORS
  - 6.1 ENTRY ERRORS
  - 6.2 ERROR TYPEOUTS - REPORTED WITHIN THE DRIVER
  - 6.3 ERROR TYPEOUTS - REPORTED AT THE RETURN FROM THE DRIVER
- 7.0 MISCELLANEOUS
  - 7.1 EXECUTION TIME
  - 7.2 HALTING PROGRAM
  - 7.3 DATA PATTERN
- 8.0 PROGRAM DESCRIPTION
  - 8.1 FORMAT OPERATION
  - 8.2 FORMAT ERRORS
  - 8.3 POSITIONER ERRORS
- 9. PROGRAM LISTING

## 1.0 ABSTRACT

-----  
 THE RPO4 FORMATTER IS DESIGNED TO WRITE AND VERIFY  
 HEADER AND DATA INFORMATION ON ALL POSSIBLE DISK  
 PACK ADDRESSES WITH THE INTENTION OF TESTING THE  
 RETENTIVITY OF THE RECORDING SURFACES. THE  
 FORMAT IS MAINTAINED ON A BASIS OF 411 CYLINDERS,  
 19 TRACKS PER CYLINDER AND 22 SECTORS PER TRACK.

## 2.0 REQUIRMENTS

## 2.1 EQUIPMENT

PDP-11 SYSTEM (12K CORE)  
 RH11 MASSBUS CONTROLLER  
 ONE OR MORE RPO4 DISK DRIVES

## 2.3 PRELIMINARY PROGRAMS

ALL RPO4 DIAGNOSTICS SHOULD HAVE PREVIOUSLY  
 BEEN RUN SUCCESSFULLY.

## 3.0 LOADING PROCEDURE

-----  
 USE THE STANDARD ABSOLUTE LOADER FROM THE 12K OR  
 HIGHER BANK OF MEMORY.

## 4.0 STARTING PROCEDURE

## 4.1 STARTING ADDRESS

-----  
 LOAD AND START AT ADDRESS 200

## 4.2 RESTARTING ADDRESS

LOAD AND START AT ADDRESS 204

## 4.3 RH11/RPO4 ADDRESSES

THE PROGRAM HAS BEEN ASSEMBLED TO USE 176700 AS THE RH11/RPO4 ADDRESS;  
 THE VECTOR IS ASSUMED TO BE 254. TO CHANGE THESE ADDRESSES, ENTER  
 THE PROPER ADDRESS INTO THE LOCATION(S) GIVEN BELOW.

LOCN	VALUE	FUNCTION
----	-----	-----
1100	176700	RH11/RPO4 UNIBUS ADDRESS
1102	254	RH11 INTERRUPT VECTOR ADDRESS

## 4.4 SWITCH SETTINGS (SW=1)

SW15 HALT ON ERROR--- THE OPERATION IN PROGRESS WHEN THE ERROR OCCURED IS POINTED TO BY THE PC LOCATED ON THE STACK.

SW14 LOOP ON THE CURRENT CYLINDER AND TRACK--- ALLOWS THE USER TO LOCK ON A FAILING TRACK ETC..

SW13 INHIBIT ERROR TYPEOUT

## 5.0 SW0 FORMAT (OR CHECK) CONTINUOUSLY OPERATING PROCEDURE

## 5.1 OPERATOR ACTION

1. TURN THE SELECTED DRIVE ON LINE.
2. DISABLE THE WRITE PROTECT ON THE SELECTED DRIVE.
3. WAIT FOR THE READY LIGHT INDICATION.
4. THE OPERATOR WILL RESPOND TO THE TYPED REQUESTS BY INPUTTING THE FOLLOWING PARAMETERS:

PARAMETER	VALUES	DEFAULT
DRIVE #	0-7	0
22 SECTOR (18 BIT) MODE	Y OR N	Y
PROGRAM MODE	F OR C	F
STARTING CYL, TRK	0-410, 0-18	0, 0 *
ENDING CYL, TRK	0-410, 0-18	410, 18 *
SELECT PATTERN	ZERO'S ONE'S WORST CASE	WORST CASE *

\* NORMAL FULL PACK FORMAT OPERATIONS SHOULD DEFAULT TO THESE VALUES.

1. TYPING THE CHARACTER 'D' WILL ASSUME THE DEFAULT VALUES.
2. A 'CONTROL C' COMBINATION WILL RETURN USER BACK TO THE DRIVE NUMBER REQUEST DURING THE INPUT REQUESTS.
3. A RUBOUT WILL RETYPE THE CURRENT LINE.
4. THE INPUT FOR PARTIAL PACK FORMATS MUST HAVE THE CYL AND TRK SEPARATED BY A COMMA AND THE LINE TERMINATED WITH A CARRIAGE RETURN. THE ENDING CYL, TRK ADDRESS MUST BE EQUAL TO OR GREATER THAN THE STARTING CYL, TRK ADDRESS. ALL LIMITS ARE INCLUSIVE.
5. THE DATA PATTERN TO BE WRITTEN ON EACH SECTOR IS SELECTED BY TYPING THE DIGIT ASSOCIATED WITH THE PATTERN TYPED.
6. THE PROGRAM MAY BE RUN IN 'READ ONLY' MODE BY ENTERING A 'C' (CHECK) IN THE RESPONSE TO THE 'PROGRAM MODE' REQUEST MESSAGE. THE NORMAL OR DEFAULT MODE IS 'F' (FORMAT & VERIFY); THIS MODE IS SELECTED BY ENTERING A 'D' (DEFAULT) OR BY

7. ENTERING AN 'F' IN RESPONSE TO THE 'MODE' REQUEST. STARTING AND ENDING CYLINDER AND TRACK ADDRESS ENTRIES MUST BE IN FORMAT 'CCC,TT' WHERE 'CCC' IS THE CYLINDER ADDRESS AND 'TT' IS THE TRACK ADDRESS.
8. ALL ENTRIES EXCEPT THE 'SELECT PATTERN' ENTRY AND THE 'D' (DEFAULT) ENTRY MUST BE TERMINATED BY A CARRIAGE RETURN.
9. THE PACK MAY BE FORMATTED IN 22 SECTOR (16 BIT) MODE OR IN 20 SECTOR (18 BIT) MODE. IF THE PACK IS FORMATTED IN 20 SECTOR MODE, BITS 16 & 17 OF EACH WORD WILL BE ZERO. ALL OF THE POP-11 RPO4 DIAGNOSTICS REQUIRE 22 SECTOR FORMATS.

## 5.2 PROGRAM ACTION

AFTER THE PARAMETERS HAVE BEEN INPUTTED THE PROGRAM WILL FORMAT OR CHECK WITHIN THE LIMITS SPECIFIED. UPON COMPLETION OF THE FORMAT OR CHECK (SWO = 0), 'FORMAT COMPLETE' OR 'CHECK COMPLETE' WILL BE TYPED AND A NEW DRIVE NUMBER WILL BE REQUESTED.

## 5.3 SUBROUTINE ABSTRACTS

### 5.3.1 'RPO4'

THIS ROUTINE IS THE DRIVER FOR THE RPO4 DISK SYSTEM. IT IS ENTERED UPON ALL REQUESTS TO THE RPO4 DRIVE.

### 5.3.2 'DRINIT'

THIS ROUTINE IS USED TO INITIALIZE THE RPO4 DRIVER. AT THE EXIT OF THIS ROUTINE THE AVAILABILITY OF EACH DRIVE HAS BEEN RECORDED IN TABLE DRVSTA.

## 6.0 ERRORS

-----

### 6.1 ENTRY ERRORS

1. THE '??' MESSAGE IDENTIFIES AN ILLEGAL INPUT CHARACTER HAS BEEN TYPED.
2. THE '?? CYLINDER LIMIT EXCEEDED' MESSAGE INDICATES THE USER REQUESTED A CYLINDER GREATER THAN 410.
3. THE '?? TRACK LIMIT EXCEEDED' MESSAGE INDICATES THE USER REQUESTED A TRACK GREATER THAN 18.
4. THE '?? ENDING DSK ADRS MUST BE EQUAL TO OR GREATER THAN THE STARTING ADRS' MESSAGE IS SELF-EXPLANATORY.

### 6.2 ERROR TYPEOUTS- REPORTED WITHIN THE DRIVER ROUTINE

- 01 ILLEGAL SUBSYSTEM INTERRUPT

A SUBSYSTEM INTERRUPT OCCURED, BUT THE 'SC' BIT IN RHCSI IS NOT SET.

#2 ILLEGAL DRIVE INTERRUPT  
 DRIVE= DRIVE NUMBER  
 RHAS= CONTENTS OF THE ATTENTION SUMMARY REGISTER

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3  
 CONTENTS OF THE ABOVE REGISTERS

THE INDICATED DRIVE INTERRUPTED BUT THE DRIVE REGISTERS DO NOT SHOW A REASON FOR THE INTERRUPT.

#3 MASSBUS CONTROL BUS PARITY ERROR  
 RD.ADR= ADDRESS OF THE REGISTER READ  
 RD.WRD= CONTENTS OF THE WORD READ

A CONTROL PARITY ERROR WAS DETECTED BY THE CONTROL UNIT WHEN THE INDICATED REGISTER WAS READ.

#4 CONTROL BUS PARITY ERROR  
 WRT.ADR= ADDRESS OF REGISTER WRITTEN  
 WRT.WD= CONTENTS OF WORD WRITTEN  
 RD.WRD= WORD READ BACK

THE ADDRESSED DRIVE DETECTED A PARITY ERROR WHEN THE CONTROLLER ATTEMPTED TO WRITE INTO THE INDICATED DRIVE REGISTER.

#5 ATTENTION FROM AN UNAVAILABLE DRIVE  
 DRIVE= DRIVE NUMBER  
 RHAS= CONTENTS OF THE ATTENTION SUMMARY REGISTER

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3  
 CONTENTS OF THE ABOVE REGISTERS

THE ATTENTION REGISTER HAS AN ATTENTION BIT SET FOR A DRIVE WHICH IS NOT CURRENTLY ACTIVE IN AN OPERATION.

### 6.3 ERROR TYPEOUTS- REPORTED AT THE RETURN FROM THE DRIVER

#1 WRITE ERROR  
 DRIVE=X SN=X - DRIVE & SERIAL NUMBER  
 CYL, TRK=XXX, XX - CYL & TRK WHERE ERROR OCCURED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3  
 CONTENTS OF THE ABOVE REGISTERS

THIS ERROR INDICATES AN ERROR OCCURED WHEN TRYING TO DO A WRITE HEADER & DATA COMMAND. THE PROGRAM WILL NOT ADVANCE BEYOND THE CYL, TRK TYPED UNTIL THE ERROR DISOLVES.

#2 WRITE CHECK ERROR  
 DRIVE=X SN=X - DRIVE & SERIAL NUMBER  
 CYL, TRK=XXX, XX - DISK ADDRESS OF BAD SECTOR  
 SECTOR=XX - BAD SECTOR  
 GOOD DATA=XXXXXX - DATA IN CORE  
 BAD DATA=XXXXXX - DATA READ FROM DISK  
 RHEC1=XXXXXX - LOCATION OF ERROR BURST  
 RHEC2=XXXXXX - ERROR BURST



RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3  
CONTENTS OF THE ABOVE REGISTERS

RETRIES UNSUCCESSFUL - SECTOR NOT ACCEPTABLE

THIS IS THE TYPICAL ERROR TIMEOUT EXPECTED  
WHEN A FORMAT ERROR OCCURS. IT SUGGESTS THAT THE  
DISK SURFACE SPECIFIED IS QUESTIONABLE. IF  
THE MESSAGE 'RETRIES UNSUCCESSFUL' IS TYPED,  
THE PACK IS NOT ACCEPTABLE.

#3 HEADER COMPARE NOT MADE  
DRIVE=X SN=X - DRIVE & SERIAL NUMBER  
CYL,TRK=XXX,XX - CYL & TRK COMPARISON ATTEMPTED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3  
CONTENTS OF THE ABOVE REGISTERS

REFORMAT PACK

THIS ERROR INDICATES THE FAILURE TO READ A  
HEADER SUCCESSFULLY. THE POSITIONER COULD HAVE  
ADVANCED TOO FAR DURING THE FORMAT OPERATION.

#4 SOFTWARE HEADER COMPARE ERROR  
DRIVE=X SN=X - DRIVE & SERIAL NUMBER  
EXPECTED- CYL,TRK=XXX,XX  
READ - CYL,TRK=XXX,XX

REFORMAT PACK

THIS ERROR SUGGESTS THAT THE POSITIONER FAILED  
TO ADVANCE DURING THE FORMAT OPERATION.

#5 READ OR WRITE UNSAFE ERROR  
DRIVE=X SN=X - DRIVE & SERIAL NUMBER  
CYL,TRK=XXX,XX - CYL & TRK WHEN UNSAFE OCCURED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3  
CONTENTS OF THE ABOVE REGISTERS

IF AN UNSAFE OCCURS THE FORMAT IS TERMINATED. THE  
FORMAT SHOULD BE RESTARTED AT OR BEFORE THE CYLINDER  
WHERE THE UNSAFE DEVELOPED. THE TYPE OF UNSAFE  
IS INDICATED BY THE CONTENTS OF RHER1, RHER2 OR RHER3.

#6 DRIVE ERROR  
DRIVE=X SN=X - DRIVE & SERIAL NUMBER  
CYL,TRK=XXX,XX - CYL & TRK WHEN ERROR OCCURED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3  
CONTENTS OF THE ABOVE REGISTERS

TWO RETRIES ARE MADE ON THE CURRENT DISK ADDRESS  
IF THE DRIVE ERROR REMAINS THE FORMAT IS TERMINATED,  
IF NOT THE FORMAT CONTINUES.

#7 DRIVE UNSAFE ERROR  
DRIVE=X SN=X - DRIVE & SERIAL NUMBER  
CYL,TRK=XXX,XX - CYL & TRK WHEN UNSAFE OCCURED

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3  
 CONTENTS OF THE ABOVE REGISTERS

IF A DRIVE UNSAFE DEVELOPS THE FORMAT IS TERMINATED.  
 THE TYPE OF UNSAFE IS INDICATED BY THE CONTENTS  
 OF RHER1, RHER2 OR RHER3.

#8 UNCORRECTABLE MASSBUS PARITY ERROR  
 FORMAT TERMINATED  
 CYL,TRK=XXX,XX - CYL & TRK WHEN ERROR OCCURED

THE PROGRAM HAS TRIED 3 TIMES TO READ OR WRITE  
 A REGISTER AND A PARITY ERROR HAS OCCURED EACH TIME.

#9 FATAL MASSBUS PARITY ERROR  
 CYL,TRK=XXX,XX - CYL & TRK WHEN ERROR OCCURED

THE PROGRAM DETECTED A MASSBUS PARITY ERROR  
 WHILE ISSUING A 'DRIVE CLEAR' TO RESET A PREVIOUSLY  
 DETECTED MASSBUS PARITY ERROR. THE PROGRAM WILL  
 HALT AFTER THE ERROR IS REPORTED.

#10 SOFTWARE TIMEOUT - SYSTEM HUNG  
 DRIVE#X SN=X - DRIVE & SERIAL NUMBER  
 CYL,TRK=XXX,XX - CYL & TRK WHEN OPERATION TIMED OUT

RHCS1 RHCS2 RHDS1 RHER1 RHER2 RHER3  
 CONTENTS OF THE ABOVE REGISTERS

IF A DISK OPERATION FAILS TO COMPLETE WITHIN 1 SECOND  
 THE ABOVE MESSAGE IS TYPED. THIS INDICATION ONLY  
 OCCURES IF THE PDP11 SYSTEM IS EQUIPPED WITH EITHER  
 A KW11-L OR KW11-P CLOCK.

7.0 MISCELLANEOUS  
 -----

7.1 EXECUTION TIME

TO FORMAT THE ENTIRE DISK TAKES APPROXIMATELY 9  
 MINUTES. IT TAKES ABOUT 1.2 SECONDS TO FORMAT  
 ONE CYLINDER PLUS 7 MS FOR EACH CYLINDER SEEK.

7.2 HALTING THE PROGRAM

THE FORMAT SHOULD NOT BE INTERRUPTED ONCE UNDERWAY  
 UNLESS A REFORMAT IS TO FOLLOW.

7.3 DATA PATTERN

THE WORST CASE PATTERN OF 165555 & 133333 IS MORE  
 LIKELY TO CREATE DATA CHECK (DCK) ERRORS  
 THAN ONES OR ZEROS. IF A SUCCESSFUL PASS WITH ZEROS  
 IS ACHIEVED WHERE WORST CASE FAILS THEN THE  
 RPO4 DIAGNOSTICS SHOULD BE RUN TO VERIFY THE  
 DRIVES RELIABILITY.

8.0 PROGRAM DESCRIPTION  
 -----

## 8.1 FORMAT OPERATION

THE FORMAT PROGRAM FORMATS THE DISK PACK ON THE ASSIGNED DRIVE ONE TRACK AT A TIME. THE DATA FIELDS ARE WRITTEN WITH THE SELECTED PATTERN. KEY WORDS ARE WRITTEN WITH ZERO'S. EACH TRACK IS VERIFIED WITH A WRITE CHECK ORDER IMMEDIATELY AFTER IT IS WRITTEN.

THE PORTION OF THE PACK TO BE FORMATTED IS DETERMINED BY THE BEGINNING AND THE ENDING CYLINDER AND TRACK ADDRESSES; THESE LIMITS ARE INCLUSIVE. A SINGLE TRACK IS THE SMALLEST ELEMENT THAT MAY BE FORMATTED.

## 8.2 FORMAT ERRORS

WRITE CHECK ERRORS ARE REPORTED WHEN THEY ARE DETECTED. IF AN ERROR IS DETECTED, THE SECTOR MUST BE REWRITTEN AND VERIFIED CORRECTLY TWO SUCCESSIVE TIMES TO BE CONSIDERED USEABLE. SECTORS WHICH CANNOT BE WRITTEN CORRECTLY TWICE AFTER AN ERROR WILL BE DECLARED NOT ACCEPTABLE BY THE PROGRAM.

## 8.3 POSITIONER ERRORS

AFTER THE LAST TRACK HAS BEEN FORMATTED AND VERIFIED AN ADDITIONAL CHECK IS PERFORMED. THE HEADER OF TRACK ZERO AND SECTOR ZERO OF EACH CYLINDER IS READ AND COMPARED BY THE SOFTWARE. THIS CHECK IS PERFORMED TO ISOLATE A POSSIBLE POSITIONER ERROR THAT MAY HAVE OCCURED DURING THE FORMAT OPERATION. TWO SUCH CASES OF POSITIONER MALFUNCTION ARE: FAILURE OF THE POSITIONER TO ADVANCE TO THE NEXT CYLINDER, AND THE CASE WHERE THE POSITIONER ADVANCES PAST THE CYLINDER DESIRED.

## 9.0 PROGRAM LISTING

-----



```

(1) 000003 R3= %3 ;; GENERAL REGISTER
(1) 000004 R4= %4 ;; GENERAL REGISTER
(1) 000005 R5= %5 ;; GENERAL REGISTER
(1) 000006 R6= %6 ;; GENERAL REGISTER
(1) 000007 R7= %7 ;; GENERAL REGISTER
(1) .EQUIV R6,SP ;; STACK POINTER
(1) .EQUIV R7,PC ;; PROGRAM COUNTER

```

```

(1)
(1)
(1) 000000 : *PRIORITY LEVEL DEFINITIONS
(1) 000040 PR0= 0 ;; PRIORITY LEVEL 0
(1) 000100 PR1= 40 ;; PRIORITY LEVEL 1
(1) 000140 PR2= 100 ;; PRIORITY LEVEL 2
(1) 000200 PR3= 140 ;; PRIORITY LEVEL 3
(1) 000240 PR4= 200 ;; PRIORITY LEVEL 4
(1) 000300 PR5= 240 ;; PRIORITY LEVEL 5
(1) 000340 PR6= 300 ;; PRIORITY LEVEL 6
(1) PR7= 340 ;; PRIORITY LEVEL 7

```

```

(1)
(1)
(1)
(1) 100000 : *"SWITCH REGISTER" SWITCH DEFINITIONS
(1) 040000 SW15= 100000
(1) 020000 SW14= 40000
(1) 010000 SW13= 20000
(1) 004000 SW12= 10000
(1) 002000 SW11= 4000
(1) 001000 SW10= 2000
(1) 000400 SW09= 1000
(1) 000200 SW08= 400
(1) 000100 SW07= 200
(1) 000040 SW06= 100
(1) 000020 SW05= 40
(1) 000010 SW04= 20
(1) 000004 SW03= 10
(1) 000002 SW02= 4
(1) 000001 SW01= 2
(1) SW00= 1
(1) .EQUIV SW09,SW9
(1) .EQUIV SW08,SW8
(1) .EQUIV SW07,SW7
(1) .EQUIV SW06,SW6
(1) .EQUIV SW05,SW5
(1) .EQUIV SW04,SW4
(1) .EQUIV SW03,SW3
(1) .EQUIV SW02,SW2
(1) .EQUIV SW01,SW1
(1) .EQUIV SW00,SW0

```

```

(1)
(1)
(1) 100000 : *DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1) 040000 BIT15= 100000
(1) 020000 BIT14= 40000
(1) 010000 BIT13= 20000
(1) 004000 BIT12= 10000
(1) 002000 BIT11= 4000
(1) 001000 BIT10= 2000
(1) 000500 BIT09= 1000

```

000400  
000200  
000100  
000040  
000020  
000010  
000004  
000002  
000001

BIT08= 400  
BIT07= 200  
BIT06= 100  
BIT05= 40  
BIT04= 20  
BIT03= 10  
BIT02= 4  
BIT01= 2  
BIT00= 1  
.EQUIV BIT09,BIT9  
.EQUIV BIT08,BIT8  
.EQUIV BIT07,BIT7  
.EQUIV BIT06,BIT6  
.EQUIV BIT05,BIT5  
.EQUIV BIT04,BIT4  
.EQUIV BIT03,BIT3  
.EQUIV BIT02,BIT2  
.EQUIV BIT01,BIT1  
.EQUIV BIT00,BIT0

000004  
000010  
000014  
000014  
000014  
000020  
000024  
000030  
000034  
000060  
000064  
000240

:#BASIC "CPU" TRAP VECTOR ADDRESSES  
ERRVEC= 4 ;: TIME OUT AND OTHER ERRORS  
RESVEC= 10 ;: RESERVED AND ILLEGAL INSTRUCTIONS  
TRITVEC= 14 ;: "T" BIT  
TRIVEC= 14 ;: TRACE TRAP  
BPTVEC= 14 ;: BREAKPOINT TRAP (BPT)  
IOTVEC= 20 ;: INPUT/OUTPUT TRAP (IOT) \*\*SCOPE\*\*  
PWRVEC= 24 ;: POWER FAIL  
EMTVEC= 30 ;: EMULATOR TRAP (EMT) \*\*ERROR\*\*  
TRAPVEC= 34 ;: "TRAP" TRAP  
TKVEC= 60 ;: TTY KEYBOARD VECTOR  
TPVEC= 64 ;: TTY PRINTER VECTOR  
PIRQVEC= 240 ;: PROGRAM INTERRUPT REQUEST VECTOR

177560  
177562  
177564  
177566  
172540  
172542  
177546

:REGISTER ADDRESSES  
TKS=177560  
TKB=177562  
TPS=177564  
TPB=177566  
\$LKCSR=172540  
\$LKCSB=172542  
\$LKS=177546

:ADRS OF KW11-P STATUS REG  
:ADRS OF KW11-P COUNTER BUFFER  
:ADRS OF KW11-L STATUS REG

001100

.=1100

001100 176700  
001102 000254

.SBTTL RPO4 ADDRESSES  
\$RPADR: .WORD 176700  
\$RPVEC: .WORD 254

:RH11/RPO4 UNIBUS ADDRESS  
:RH11 INTERRUPT VECTOR

.SBTTL MAIN PROGRAM

1  
2  
3  
4  
5  
6  
7  
8  
9  
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

```

4968 001104 000005          BEGIN: RESET          ;CLEAR WORLD
4969 001106 012706 001100      MOV          #STACK,SP      ;SET UP STACK
4970 001112 012737 003116 000030  MOV          #EMTHNO,2#30    ;SET UP EMT ADRS
4971 001120 012737 000340 000032  MOV          #340,2#32      ;SET PS TO 7
4972 001126 013737 001100 013070  MOV          $RPAOR,RPADR    ;MOVE RH11/RPO4 ADDRESSES TO VECTOR
4973 001134 013737 001102 013072  MOV          $RPVEC,RPVEC    ;RH11 VECTOR ADDRESS
4974 001142 012737 000240 001156  MOV          #240,RESTRT    ;ALLOW RESTARTS AFTER INITIAL START
4975 001150 004537 005166          JSR          RS,TYPOUT      ;GO TYPE MESSAGE
4976 001154 007466          MTITLE          ;ADRS OF 'TITLE' MESSAGE
4977 001156 000752          RESTRT: BR          BEGIN    ;CHANGE TO 'NOP' AFTER INITIAL START
4978 001160 012706 001100      MOV          #STACK,SP      ;SET UP STACK
4979 001164 000005          RESET          ;CLEAR WORLD
4980
4981          ;GO FIND OUT WHAT DRIVE
4982
4983 001166 004537 005166          MO:          JSR          RS,TYPOUT    ;GO TYPE MESSAGE
4984 001172 007546          MUNIT          ;ADRS OF MESSAGE
4985 001174 004737 005242          JSR          PC,TTI        ;GO WAIT FOR DRIVE #
4986 001200 022700 000104          CMP          #104,RO      ;IS IT THE DEFAULT CHAR?
4987 001204 001006          BNE          IS          ;BRANCH IF NOT THE DEFAULT CHAR
4988 001206 005037 012540          CLR          USEL        ;SELECT DRIVE ZERO
4989 001212 004537 005166          JSR          RS,TYPOUT    ;GO TYPE DEFAULT DRIVE NO.
4990 001216 007561          MORVD          ;ADRS OF MSG
4991 001220 000413          BR          GSEL         ;GO SEE IF DRIVE ZERO IS THERE
4992
4993 001222 004737 005314          IS:          JSR          PC,NUMTST  ;IS IT A DIGIT?
4994 001226 000757          BR          MO          ;RETURN HERE IF DRIVE # NOT A DIGIT
4995 001230 022700 000007          CMP          #7,RO       ;IS IT LESS THAN 7?
4996 001234 103003          BCC          2$         ;YES - GO SELECT DRIVE
4997 001236 004737 005406          JSR          PC,QUES     ;TYPE ??
4998 001242 000751          BR          MO          ;RETYPE LINE
4999 001244 010037 012540          2$:          MOV          RO,USEL    ;SAVE IN USEL
5000
5001          ;THIS CODE CHECKS IF REQUIRED DRIVE IS AVAILABLE
5002
5003 001250 004737 013106          GSEL:        JSR          PC,RPINIT    ;GO SEE WHAT DRIVES ARE AVAILABLE
5004 001254 012737 177777 013032  MOV          #177777,SEEKFG ;SET TO -1 IF NO OPTIMIZATION IN DRIVER
5005 001262 013700 012540          MOV          USEL,RO      ;GET DRIVE NO.
5006 001266 105760 012752          TSTB         DRVSTA(RO)   ;LOOK AT DRIVE STATUS
5008 001272 003035          BGT          M1          ;BRANCH IF ONLINE
5012 001274 002404          BLT          IS          ;BRANCH IF UNSAFE OR NONEXISTENT
5013 001276 004537 005166          JSR          RS,TYPOUT    ;TYPE 'DRIVE OFFLINE'
5014 001302 007577          MOFFLN        ;ADRS OF MESSAGE
5015 001304 000730          BR          MO          ;GO GET DRIVE # AGAIN
5016 001306 006300          1$:          ASL          RO        ;RO INTO WORD INDEX
5017 001310 005760 012762          TST          DRVTP(RO)   ;A DRIVE PRESENT?
5018 001314 001004          BNE          2$         ;BR IF SO
5019 001316 004537 005166          JSR          RS,TYPOUT    ;TYPE 'DRIVE NOT PRESENT'
5020 001322 007621          MORNP        ;ADRS OF MSG
5021 001324 000720          BR          MO          ;GO GET DRIVE NO. AGAIN
5022 001326 022760 020020 012762  2$:          CMP          #20020,DRVTP(RO) ;SINGLE PORT?
5023 001334 001410          BEQ          3$         ;BR IF UNSAFE SINGLE PORT
5024 001336 022760 024020 012762          CMP          #24020,DRVTP(RO) ;DUAL PORT?
5025 001344 001404          BEQ          3$         ;BR IF UNSAFE DUAL PORT

```

```

5026 001346 004537 005166 JSR RS,TYPOUT ;TYPE 'NOT AN RPO4'
5027 001352 007646 MNRPO4 ;ADDR OF MSG
5028 001354 000704 BR MO ;GO GET DRIVE NO. AGAIN
5029 001356 004537 005166 3$: JSR RS,TYPOUT ;TYPE 'RPO4 UNSAFE'
5030 001362 007673 MUSR ;ADDR OF MSG
5031 001364 000700 BR MO ;GO GET DRIVE NO. AGAIN
5032
5056 ;SEE WHICH MODE THE PROGRAM IS TO BE RUN IN.
5057 ;MODES ARE: 'FORMAT & VERIFY' OR 'CHECK FORMAT'
5058
5059 001366 005037 012544 M1: CLR MODE ;SET MODE TO 'CHECK FORMAT'
5060 001372 004537 005166 JSR RS,TYPOUT ;TYPE 'PROGRAM MODE'
5061 001376 007725 MMODE ;ADDR OF MSG
5062 001400 004737 005242 JSR PC,TTI ;WAIT FOR ENTRY
5063 001404 022700 000104 CMP #'D',RO ;'D' (DEFAULT) ENTERED ?
5064 001410 001415 BEQ 2$ ;BR IF DEFAULT
5065 001412 022700 000103 CMP #'C',RO ;CHECK FORMAT MODE ?
5066 001416 001406 BEQ 1$ ;BR IF CHECK ONLY
5067 001420 022700 000106 CMP #'F',RO ;FORMAT & VERIFY ?
5068 001424 001413 BEQ 3$ ;BR IF YES
5069 001426 004737 005406 JSR PC,QUES ;TYPE '??'
5070 001432 000755 BR M1 ;TRY AGAIN
5071 001434 004537 005166 1$: JSR RS,TYPOUT ;TYPE REST OF 'CHECK'
5072 001440 010004 MCHK ;ADDR OF MSG
5073 001442 000412 BR M1A ;GET STARTING ADDRESS
5074 001444 004537 005166 2$: JSR RS,TYPOUT ;TYPE DEFAULT MESSAGE
5075 001450 007761 MFORMT ;ADDR OF MSG
5076 001452 000403 BR 4$ ;SET UP MODE
5077 001454 004537 005166 3$: JSR RS,TYPOUT ;TYPE REST OF 'FORMAT'
5078 001460 010016 MORMAT ;ADDR OF MSG
5079 001462 012737 177777 012544 4$: MOV #-1,MODE ;SET MODE TO 'FORMAT & VERIFY'
5080
5081 ;FIND OUT IF FORMAT IS TO BE IN 20 OR 22 SECTOR MODE
5082
5083 001470 004537 005166 M1A: JSR RS,TYPOUT ;TYPE FORMAT MODE REQUEST
5084 001474 010035 MSIZE ;ADDRESS OF MESSAGE
5085 001476 004737 005242 JSR PC,TTI ;WAIT FOR INPUT
5086 001502 022700 000131 CMP #'Y',RO ;IS ENTRY A 'Y' ?
5087 001506 001411 BEQ 1$ ;BR IF IT IS
5088 001510 022700 000116 CMP #'N',RO ;IS ENTRY A 'N' ?
5089 001514 001431 BEQ 2$ ;BR IF IT IS
5090 001516 022700 000104 CMP #'D',RO ;IS ENTRY A 'D' ?
5091 001522 001403 BEQ 1$ ;BR IF IT IS
5092 001524 004737 005406 JSR PC,QUES ;TYPE '??' - INVALID INPUT
5093 001530 000757 BR M1A ;TRY AGAIN
5094 001532 004537 005166 1$: JSR RS,TYPOUT ;TYPEOUT MODE SELECTED
5095 001536 010106 MSEC22 ;ADDRESS OF MESSAGE
5096 001540 012737 013130 012602 MOV #(260.*22.),WC ;TRACK SIZE IN 22 SECTOR MODE
5097 001546 012737 164650 012604 MOV #- (260.*22.),MWC ;2'S COMPLEMENT WORD COUNT
5098 001554 012737 177777 012610 MOV #-1,SEC20 ;22 SECTOR INDICATOR
5099 001562 012737 000025 012612 MOV #21,MAXSEC ;MAX SECTOR ADDRESS IN 22 SECTOR MODE
5100 001570 112737 000020 012713 MOV#B #20,FMTDPB+1 ;LOAD FMT22 BIT
5101 001576 000420 BR 3$ ;LOAD FORMAT BIT
5102 001600 004537 005166 2$: JSR RS,TYPOUT ;TYPE OUT 20 SECTOR MODE SELECTED

```



```

5103 001604 010165 MSEC20 ;ADDRESS OF MESSAGE
5104 001606 012737 012120 012602 MOV #(<260.*20.>,WC ;TRACK SIZE IN 20 SECTOR MODE
5105 001614 012737 165660 012604 MOV #-(260.*20.),MWC ;2'S COMPLEMENT WORD COUNT
5106 001622 005037 012610 CLR SEC20 ;20 SECTOR INDICATOR
5107 001626 012737 000023 012612 MOV #19,MAXSEC ;MAX SECTOR ADDRESS IN 20 SECTOR MODE
5108 001634 105037 012713 CLRB FMTDPB+1 ;CLEAR THE FMT22 BIT
5109 001640 113737 012540 012712 3$: MOVB USEL,FMTDPB ;LOAD DRIVE NUMBER
5110 001646 112737 000143 012714 MOVB #143,FMTDPB+2 ;'SET FORMAT' COMMAND
5111 001654 004037 013476 JSR RO,RPO4 ;GO TO THE DRIVER TO EXECUTE THE COMMAND
5112 001660 012712 FMTDPB ;DPB ADDRESS
5113 001662 000766 BR 3$ ;QUEUE FULL RETURN - RETRY THE COMMAND
5114 001664 005737 012730 4$: TST FMTDPB+16 ;CHECK STATUS WORD
5115 001670 001775 BEQ 4$ ;BR IF NOT FINISHED
5116 001672 100762 BMI 3$ ;BR IF ERROR IN SET FORMAT
5117 001674 000400 BR M2
5118
5119 ;GO GET PARAMETERS - STARTING CYL & TRK
5120 ;AND ENDING CYL & TRK
5121
5122 001676 004537 005166 M2: JSR RS,TYPOUT ;GO TYPE MESSAGE 'STARTING CYL,TRK='
5123 001702 010244 MMINCT ;ADRS OF MESSAGE
5124 001704 012737 000001 012542 MOV #1,SOFSW ;SET UP SOFSW TO INDICATE STARTING INPUT
5125 001712 004737 005446 JSR PC,GETDAT ;GO GET STARTING CYL & TRK
5126 001716 000767 BR M2 ;RETURN HERE IF ERROR OR RUBOUT
5127 001720 005037 012542 CLR SOFSW ;SET UP SOFSW TO INDICATE ENDING INPUT
5128 001724 004537 005166 M3: JSR RS,TYPOUT ;GO TYPE MESSAGE 'ENDING CYL,TRK='
5129 001730 010277 MMAXCT ;ADRS OF MESSAGE
5130 001732 004737 005446 JSR PC,GETDAT ;GO GET ENDING CYL & TRK
5131 001736 000772 BR M3 ;RETURN HERE IF ERROR OR RUBOUT
5132
5133 ;THIS CODE CHECKS THE ORDER AND MAGNITUDE OF THE CYL & TRK ADRS
5134 ;INPUTTED BY THE TTY. ALSO ESTABLISHES THE TOTAL TRK COUNT
5135
5136 001740 013700 012546 CKADRS: MOV ECTL,RO ;SET RO WITH ECTL
5137 001744 163700 012550 SUB SCYL,RO ;SEE IF END CYL IS SMALLER THAN START CYL
5138 001750 103004 BCC 1$ ;BRANCH IF NOT
5139 001752 004537 005166 JSR RS,TYPOUT ;TYPE IMPROPER DSK ADRS
5140 001756 010423 MADRER ;ADRS OF MESSAGE
5141 001760 000746 BR M2 ;RETYPE REQUEST - INPUT CYL NOT IN PROPER FORM
5142 001762 001014 1$: BNE 3$ ;BRANCH IF SCYL NOT EQUAL TO ECTL
5143 001764 013700 012552 MOV ETRK,RO ;SET UP RO WITH ETRK
5144 001770 163700 012554 SUB STRK,RO ;SEE IF END TRK IS SMALLER THAN START TRK
5145 001774 103004 BCC 2$ ;BRANCH IF NOT
5146 001776 004537 005166 JSR RS,TYPOUT ;TYPE IMPROPER DSK ADRS
5147 002002 010423 MADRER ;ADRS OF MESSAGE
5148 002004 000734 BR M2 ;RETYPE REQUEST - INPUT TRK NOT IN PROPER FORM
5149 002006 010037 012556 2$: MOV RO,TTRKS ;SAVE # OF TRACKS
5150 002012 000417 BR M4 ;GO GET DATA PATTERN
5151 002014 005037 012556 3$: CLR TTRKS ;CLEAR TOTAL TRACK LOC
5152 002020 062737 000023 012556 4$: ADD #23,TTRKS ;ADD CYLINDER WORTH OF TRACKS
5153 002026 005300 DEC RO ;MAINTAIN CONTROL
5154 002030 001373 BNE 4$ ;BRANCH IF CYL DIFF NOT 0
5155 002032 013700 012554 MOV STRK,RO ;PUT STRK IN RO
5156 002036 005400 NEG RO ;MAKE TWO'S COMP

```

```

5157 002040 060037 012556      ADD    R0,TRKS      ;SUB PARTICAL TRK
5158 002044 063737 012552 012556  ADD    ETRK,TRKS   ;ADD IN END TRACKS
5159
5160                               ;GO GET DATA PATTERN FOR FORMAT
5161
5162 002052 004537 005165      M4:   JSR    R5,TYPOUT ;GO TYPE 'SELECT PATTERN'
5163 002056 010530                MSELP ;ADRS OF MESSAGE
5164 002060 004737 005242      JSR    PC,TTI      ;GO GET REPLY
5165 002064 022700 000104      CMP    #104,R0    ;IS IT THE D-FULT CHAR?
5166 002070 001006                BNE    1$         ;BRANCH IF NOT
5167 002072 004537 005165      JSR    R5,TYPOUT  ;TYPE DEFAULT PATTERN
5168 002076 010632                MPTD   ;ADRS OF MSG
5169 002100 012700 000002      MOV    #2,R0      ;SELECT PATTERN OF WORST CASE
5170 002104 000411                BR     2$         ;GO SAVE PATTERN
5171 002106 004737 005314      1$:   JSR    PC,NUMTST ;SEE IF A NUMBER
5172 002112 000757                BR     M4        ;BRANCH IF NOT
5173 002114 020027 000003      CMP    R0,#3      ;IS # LARGER THAN 2
5174 002120 103403                BCS   2$         ;BRANCH IF NOT
5175 002122 004737 005406      JSR    PC,QUES    ;TYPE ??
5176 002126 000751                BR     M4        ;RETYPE LINE
5177 002130 010037 012564      2$:   MOV    R0,PSEL   ;SAVE PATTERN SELECTED
5178 002134 004537 005165      JSR    R5,TYPOUT  ;TYPE 'SELECTED'
5179 002140 007713                MSEL   ;ADRS OF MESSAGE
5180
5181                               ;GO TYPE 'STARTING FORMAT ON DRIVE # X'
5182
5183 002142 005737 012544      M5:   TST    MODE     ;'FORMAT' OR 'CHECK' MODE ?
5184 002146 001404                BEQ    1$         ;BR IF 'CHECK' MODE
5185 002150 004537 005165      JSR    R5,TYPOUT  ;TYPE 'STARTING FORMAT ON DRIVE #'
5186 002154 010646                MSFOU  ;ADRS OF MESSAGE
5187 002156 000403                BR     2$
5188 002160 004537 005165      1$:   JSR    R5,TYPOUT  ;TYPE 'STARTING CHECK ON DRIVE #'
5189 002164 010704                MSCHK  ;ADDR OF MSG
5190 002166 004737 005350      2$:   JSR    PC,UASM    ;GO PRINT DRIVE # IN USEL
5191
5192
5193                               ;THIS CODE SETS UP THE SELECTED DATA PATTERN TO BE WRITTEN ON EACH SECTOR IMAGE
5194
5195 002172 005037 012566      STUF: CLR    PATA     ;CLEAR DATA PATTERN A
5196 002176 005037 012570      CLR    PATB     ;CLEAR DATA PATTERN B
5197 002202 005737 012564      TST    PSEL     ;SEE IF PATTERN OF ONES
5198 002206 001416                BEQ    STUF     ;BR IF SO
5199 002210 005137 012566      COM    PATA     ;SET PATA TO ONES
5200 002214 005137 012570      COM    PATB     ;SET PATB TO ONES
5201 002220 022737 000001 012564  CMP    #1,PSEL   ;SEE IF PATTERN OF ZEROS
5202 002226 001406                BEQ    STUF     ;BRANCH IF SO
5203 002230 012737 165555 012566  MOV    #165555,PATA ;SET UP WORST CASE
5204 002236 012737 133333 012570  MOV    #133333,PATB ;SET UP WORST CASE
5205
5206                               ;THIS CODE DUMP THE SELECT DATA PATTERN IN THE DATA BUFFER IN CORE
5207
5208 002244 013703 012602      STUF: MOV    WC,R3    ;SET UP COUNTER
5209 002250 012700 020300      MOV    #BUFF,R0  ;SET UP MEMORY POINTER
5210 002254 013701 012566      MOV    PATA,A1   ;SET UP PATTERN IN R1

```

```

5211 002260 013702 012570      MOV      PATB,R2      ;SET UP PATTERN IN R2
5212 002264 010120      1$:      MOV      R1,(R0)+    ;MOV 1ST PAT INTO MEM
5213 002266 010220      MOV      R2,(R0)+    ;MOV 2ND PAT INTO MEM
5214 002270 005303      DEC      R3          ;KEEP COUNT
5215 002272 005303      DEC      R3          ;KEEP COUNT
5216 002274 001373      BNE      1$          ;DO IT AGAIN IF R3 NOT 0
5217
5218
5219
5220 002276 004737 006016      WRHDR1: JSR      PC,SETTBL    ;GO SET UP DRIVER TABLE
5221 002302 004737 006320      JSR      PC,SETHDR    ;GO INITIALIZE HEADERS IN THE SECTOR BUFFER IN CORE
5222 002306 004737 007340      JSR      PC,CKCLK     ;GO START UP THE CLOCK IF AVAILABLE
5223 002312 005037 012542      WRHDR2: CLR      SOFSW    ;CLEAR ERROR COUNTER
5224 002316 005037 012574      CLR      RETRY        ;ZERO THE RETRY COUNTER
5225 002322 105037 012722      CLR      FMTDPB+10    ;RESTORE SECTOR
5226 002326 013737 012604      MOV      MMC,FMTDPB+4 ;RESTORE MC
5227 002334 012737 020300      MOV      #BUFP,FMTDPB+6 ;RESTORE CA
5228 002342 005737 012544      TST      MODE         ;'FORMAT' OR 'CHECK' MODE ?
5229 002346 001416      BEQ      CKHDR        ;BR IF 'CHECK' MODE
5230 002350 112737 000163      WRHDR:  MOVB     #163,FMTDPB+2 ;SET WRITE HEADER & DATA COMMAND IN TBL
5231 002356 004037 013476      1$:      JSR      RD,RPO4      ;GO FORMAT A TRACK
5232 002362 012712      FMTDPB    ;ADRS OF PARAMETERS - TBL
5233 002364 000774      BR       1$          ;WAIT FOR QUEUE IF FULL
5234 002366 005737 012730      2$:      TST      FMTDPB+16    ;WAIT FOR COMMAND TO COMPLETE
5235 002372 001775      BEQ      2$          ;BRANCH IF NOT DONE
5236 002374 100003      BPL      CKHDR        ;BRANCH IF NO ERROR
5237 002376 004737 003716      JSR      PC,FE        ;GO RESOLVE ERROR
5238 002402 000762      BR       WRHDR       ;GO TRY AGAIN
5239
5240
5241
5242
5243
5244
5245
5246
5247
5248
5249
5250
5251 002404 112737 000153      CKHDR:  MOVB     #153,FMTDPB+2 ;SET WRITE CHECK HEADER & DATA COMMAND IN TBL
5252 002412 004037 013476      1$:      JSR      RD,RPO4      ;GO CHECK THE TRK JUST FORMATTED
5253 002416 012712      FMTDPB    ;ADRS OF PARAMETERS - TBL
5254 002420 000774      BR       1$          ;WAIT FOR QUEUE IF FULL
5255 002422 005737 012730      2$:      TST      FMTDPB+16    ;WAIT FOR COMMAND TO COMPLETE
5256 002426 001775      BEQ      2$          ;BRANCH IF NOT DONE
5257 002430 100006      BPL      GOOD        ;BRANCH IF NO ERROR
5258 002432 004737 003716      JSR      PC,ER        ;GO RESOLVE ERROR
5259 002436 005737 012544      TST      MODE         ;'FORMAT' OR 'CHECK' MODE ?
5260 002442 001760      BEQ      CKHDR        ;BR IF 'CHECK' MODE
5261 002444 000741      BR       WRHDR       ;ER-REFORMAT SECTOR
5262 002446 005737 012542      GOOD:   TST      SOFSW    ;SEE IF RETRY
5263 002452 001421      BEQ      LOOPTS       ;BRANCH IF NOT
5264 002454 022737 000002      CMP      #2,SOFSW     ;SEE IF LAST RETRY
5265 002462 001012      BNE      1$          ;BRANCH IF NOT
5266 002464 123737 012612      CMPB     MAXSEC,FMTDPB+10 ;SEE IF LAST SECTOR IN TRK
5267 002472 001411      BEQ      LOOPTS       ;BRANCH IF SO
5268 002474 004737 006260      JSR      PC,SCANC     ;GO SET UP 3L FOR REMAINING SECTORS
5269 002500 005737 012544      TST      MODE         ;'FORMAT' OR 'CHECK' MODE ?
5270 002504 001737      BEQ      CKHDR        ;BR IF 'CHECK' MODE
5271 002506 000720      BR       WRHDR       ;GO FINISH FORMATTING TRK
5272 002510 005237 012542      1$:      INC      SOFSW        ;RECORD GOOD RETRY
5273 002514 000715      BR       WRHDR       ;TRY ONCE MORE

```

```

5274 002516 032737 040000 177570 LOOPTS: BIT      #BIT14,2#SWR ;SEE IF LOOP ON CURRENT TRK
5275 002524 001401          BEQ      NXTRK   ;BRANCH IF NOT
5276 002526 000671          BR       WRHOR2 ;REPEAT SAME TRACK
5277 002530 004737 006166          NXTRK: JSR      PC,TRKTST ;GOOD FORMAT-GO SEE IF DONE-IF NOT, SET UP NEXT
5279 002534 000401          BR       ENOCK   ;RETURN HERE IF DONE - GO DO QUICK CK
5283 002536 000665          BR       WRHOR2 ;NOT DONE - GO WRITE NEXT TRACK
5289
5290
5291 ;THIS CODE MAKES SURE EACH CYLINDER FORMATTED CONTAINS THE
5292 ;PROPER CYLINDER ADDRESS. THE PROGRAM IS LOOKING FOR
5293 ;POSSIBLE POSITIONER ERRORS THAT MAY HAVE OCCURED DURING THE FORMAT.
5294 002540 005037 012606          ENOCK: CLR      HEDERR ;CLEAR HEADER CHECK ERROR INDICATOR
5295 002544 004737 006016          JSR      PC,SETTBL ;GO SET UP DRIVER TABLE
5296 002550 004737 006320          JSR      PC,SETHOR ;GO SET UP HEADERS IN CORE
5297 002554 013737 012550 012572          MOV      SCYL,TEMP ;PUT SCYL IN TEMP
5298 002562 012737 177776 012716          MOV      #-2,FMTDPB+4 ;SET UP WORD COUNT
5299 002570 012737 012622 012720          MOV      #RBUF,FMTDPB+6 ;SET UP BUFFER ADRS
5300 002576 005037 012722          CLR      FMTDPB+10 ;CLEAR THE SECTOR & TRACK ADDRESS FIELD
5301 002602 013737 012550 012724          MOV      SCYL,FMTDPB+12 ;SETUP THE CYLINDER FIELD
5306 002610 112737 000173 012714          MOVB    #173,FMTDPB+2 ;SET UP READ HEADER & DATA COMMAND
5307 002616 004037 013476          HOREAD: JSR      RD,RPO4 ;GO READ HEADER
5308 002622 012712          FMTDPB ;ADRS OF PARAMETER TBL
5309 002624 000774          BR       HOREAD ;WAIT IF QUEUE IS FULL
5310 002626 005737 012730          IS:    TST     FMTDPB+16 ;WAIT FOR COMMAND TO COMPLETE
5311 002632 001775          BEQ     IS ;BRANCH IF NOT DONE
5312 002634 100003          BPL     HORCK ;BRANCH IF DONE
5313 002636 004737 003716          JSR      PC,ER ;GO RESOLVE ERROR
5314 002642 000406          BR       HORCK1 ;BYPASS THE COMPARSION
5315 002644 023737 012622 020300          HORCK: CMP     RBUF,BUFF ;SEE IF CYL READ EQUALS CYL EXPECTED
5316 002652 001402          BEQ     .+6 ;BR IF CYL CORRECT
5317 002654 004737 002706          JSR      PC,CMPER ;REPORT INCORRECT CYLINDER
5318 002660 023737 012546 012572          HORCK1: CMP     ECYL,TEMP ;SEE IF LAST CYL
5319 002666 001453          BEQ     DONE ;BRANCH IF ALL DONE
5320 002670 005237 012572          INC     TEMP ;RECORD CYL CHECKED
5321 002674 004737 006374          JSR      PC,UPDACY ;SET UP FOR NEXT CYL
5322 002700 005237 012724          INC     FMTDPB+12 ;ADVANCE TO NEXT CYL #
5323 002704 000744          BR       HOREAD ;GO READ NEXT HEADER
5324
5325 ;THIS CODE REPORTS THAT THE HEADER READ DOES NOT
5326 ;COMPARE WITH THE ONE EXPECTED.
5327
5328 002706 032737 020000 177570          CMPEM: BIT      #BIT13,2#SWR ;SKIP TYPEOUT?
5329 002714 001037          BNE     IS ;BRANCH IF SO
5330 002716 004537 005166          JSR      RS,TYPOUT ;GO TYPE SOFT HEADER COMPARE ERROR
5331 002722 012474          MHDCHP
5332 002724 004737 006662          JSR      PC,DRSN ;GO TYPE DRIVE# & SN
5333 002730 004537 005166          JSR      RS,TYPOUT ;GO TYPE EXPECTED
5334 002734 012444          MEXPED
5335 002736 004737 006744          JSR      PC,CYLTK ;GO TYPE CYL,TRK EXPECTED
5336 002742 004537 005166          JSR      RS,TYPOUT ;GO TYPE READ
5337 002746 012460          MREAD
5338 002750 042737 010000 012622          BIC     #10000,RBUF ;DUMP FMT22
5339 002756 113737 012625 012614          MOVB    RBUF+3,LOCTRK ;TRACK FROM HEADER
5340 002764 013737 012622 012616          MOV     RBUF,LOCCYL ;CYLINDER FROM HEADER

```

```

5341 002772 004737 006760 JSR PC,CYLTRK ;GO TYPE CYL-TRK READ
5342 002776 012737 177777 012606 MOV #1,HEDERR ;SET THE CYLINDER ERROR INDICATOR
5343 003004 005737 177570 TST #SWR ;HALT?
5344 003010 100001 BPL IS ;BRANCH IF NOT
5345 003012 000000 HALT ;ERROR- SOFTWARE HEADER COMPARE ERROR
5346 003014 000207 1S: RTS PC ;RETURN
5347
5348 ;END OF FORMAT (OR CHECK) ROUTINE
5349
5351 003016 005737 012606 DONE: TST HEDERR ;CYLINDER ERROR ?
5352 003022 001404 BEQ IS ;BR IF NOT
5353 003024 004537 005166 JSR RS,TYPOUT ;TYPEOUT 'REFORMAT MESSAGE'
5354 003030 012332 MBOFMT ;MESSAGE ADDRESS
5355 003032 000412 BR 3S
5356 003034 005737 012544 1S: TST MODE ;'FORMAT' OR 'CHECK' MODE ?
5357 003040 001404 BEQ 2S ;BR IF 'CHECK' MODE
5358 003042 004537 005166 JSR RS,TYPOUT ;GO TYPE MESSAGE 'FORMAT COMPLETE'
5359 003046 010752 MFCMPT ;ADRS OF MESSAGE
5360 003050 000403 BR 3S
5361 003052 004537 005166 2S: JSR RS,TYPOUT ;TYPE 'END OF CHECK' MESSAGE
5362 003056 010777 MCCMPT ;ADDR OF MSG
5363 003060 032737 000001 177570 3S: BIT #SWD,SWR ;SEE IF SWR 0 SET
5364 003066 001002 BNE +6 ;BR IF SET
5365 003070 000137 001156 JMP RESTR ;ASK FOR NEXT DRIVE
5366 003074 012706 001100 MOV #STACK,SP ;RESET STACK POINTER
5367 003100 000137 002142 JMP MS ;DO THE FORMAT OR CHECK AGAIN
5368
5369 ;THIS CODE SERVICES A CLOCK INTERRUPT EVERY 16MS
5370
5371
5372
5373
5374
5375
5376 003104 012746 000020 CLOCK: MOV #16,-(SP) ;PUT MILLISECONDS ON THE STACK
5377 003110 004037 016436 JSR RO,RPTMR ;GO REPORT TIME
5378 003114 000002 RTI ;RETURN AND CONTINUE
5379
5380 ;THIS IS THE EMT ERROR HANDLER-SELECTS ADRS OF ERROR HANDLER
5381
5382
5383 003116 010137 003162 EMTMND: MOV R1,SAVR ;SAVE THE REGISTERS
5384 003122 010337 003164 MOV R3,SAVR+2 ;DRIVE NUMBER
5385 003126 010437 003166 MOV R4,SAVR+4 ;ATA BIT
5386 003132 010537 003170 MOV R5,SAVR+6 ;RHCSI ADDRESS
5387 003136 010037 003172 MOV R0,SAVR+10 ;MOL STATE (BIT 4)
5388 003142 010237 003174 MOV R2,SAVR+12 ;SAVE UNKNOWN
5389 003146 011600 MOV (SP),R0 ;SAVE UNKNOWN
5390 003150 014000 MOV -(R0),R0 ;GET EMT PC
5391 003152 006300 ASL R0 ;GET EMT CALL INSTR
5392 003154 016000 173174 MOV EMTTBL-10002(R0),R0 ;TIMES 2
5393 003160 000110 JMP (R0) ;DEVELOP EMT ADDRESS
5394 ;GO TO ERROR REPORT ROUTINE
5395 ;SAVE REGS 1,3,4, & 5 HERE AFTER DRIVER ERRORS 1->5
5396
5397 003162 000000 SAVR: 0 ;R1
5398 003164 000000 0 ;R3
5399 003166 000000 0 ;R4
5400 003170 000000 0 ;R5

```

```

5401 003172 000000      0      :R0
5402 003174 000000      0      :R2
5403
5404      ;TABLE OF ERROR HANDLERS-DRIVER LEVEL
5405
5406      EMTTBL: ER1      ;ILLEGAL SUBSYSTEM INTERRUPT
5407      ER2      ;ILLEGAL DRIVE INTERRUPT
5408      ER3      ;MASSBUS CONTROL BUS PARITY ERROR
5409      ER4      ;CONTROL BUS PARITY ERROR
5410      ER5      ;ATTENTION FROM AN UNAVAILABLE DRIVE
5411
5412      ;HANDLES ILLEGAL SUBSYSTEM INTERRUPT
5413
5414 003210 032737 020000 177570 ER1:  BIT      #BIT13, @#SWR      ;SKIP TYPEOUT?
5415 003216 001003      BNE      1$      ;BRANCH IF SO
5416 003220 004537 005166      JSR      R5, TYPOUT      ;TYPE ILLEGAL SUBSYSTEM INTERRUPT
5417 003224 011023      MER1
5418 003226 005737 177570      1$:  TST      @#SWR      ;HALT?
5419 003232 100001      BPL      2$      ;BRANCH IF NOT
5420 003234 000000      HALT      ;ERROR- ILLEGAL SUBSYSTEM INTERRUPT
5421 003236 004737 007270      2$:  JSR      PC, RESREG      ;RESTORE R0-R5
5422 003242 000002      RTI      ;RETURN TO DRIVER
5423
5424      ;HANDLES ILLEGAL DRIVE INTERRUPT
5425
5426 003244 032737 020000 177570 ER2:  BIT      #BIT13, @#SWR      ;SKIP TYPEOUT?
5427 003252 001031      BNE      1$      ;BRANCH IF SO
5428 003254 004537 005166      JSR      R5, TYPOUT      ;TYPE ILLEGAL DRIVE INTERRUPT
5429 003260 011065      MER2
5430 003262 004537 005166      JSR      R5, TYPOUT      ;TYPE DRIVE
5431 003266 011762      MDRV
5432 003270 013703 003162      MOV      SAVR, R3      ;GET DRIVE #
5433 003274 004737 006724      JSR      PC, TORNUM      ;GO TYPE DRIVE #
5434 003300 004737 005376      JSR      PC, CRLF      ;SET MARGIN
5435 003304 004537 005166      JSR      R5, TYPOUT      ;TYPE RHAS
5436 003310 011771      MAS
5437 003312 013703 003170      MOV      SAVR+6, R3      ;GET RHAS
5438 003316 004737 006604      JSR      PC, OCTYP      ;GO TYPE IT
5439 003322 004737 005376      JSR      PC, CRLF      ;SET UP MARGIN
5440 003326 004737 007064      JSR      PC, SETREG      ;GO PREPARE FOR TYPE
5441 003332 004737 007150      JSR      PC, RHREGS      ;GO TYPE REGISTERS
5442 003336 005737 177570      1$:  TST      @#SWR      ;HALT?
5443 003342 100001      BPL      2$      ;BRANCH IF NOT
5444 003344 000000      HALT      ;ERROR- ILLEGAL DRIVE INTERRUPT
5445 003346 004737 007270      2$:  JSR      PC, RESREG      ;RESTORE R0-R5
5446 003352 000002      RTI      ;RETURN TO DRIVER
5447
5448      ;HANDLES MASSBUS CONTROL BUS PARITY ERRORS
5449
5450 003354 032737 020000 177570 ER3:  BIT      #BIT13, @#SWR      ;SKIP TYPEOUT?
5451 003362 001027      BNE      1$      ;BRANCH IF SO
5452 003364 004537 005166      JSR      R5, TYPOUT      ;TYPE MASSBUS CONTROL BUS PARITY ERROR
5453 003370 011123      MER3
5454 003372 004737 005376      JSR      PC, CRLF      ;SET MARGIN

```

```

5455 003376 004537 005166 JSR RS, TYP0UT ;TYPE RD.ADR
5456 003402 012000 MRDADR
5457 003404 013703 017062 MOV RD.ADR, R3 ;GET ADRS
5458 003410 004737 006604 JSR PC, OCTYP ;TYPE IT
5459 003414 004737 005376 JSR PC, CRLF ;SET MARGIN
5460 003420 004537 005166 JSR RS, TYP0UT ;TYPE RD.WRD
5461 003424 012010 MRDWRD
5462 003426 013703 017064 MOV RD.WRD, R3 ;GET WRD
5463 003432 004737 006604 JSR PC, OCTYP ;TYPE IT
5464 003436 004737 005376 JSR PC, CRLF ;SET MARGIN
5465 003442 005737 177570 1$: TST @#SWR ;HALT?
5466 003446 100001 BPL 2$ ;BRANCH IF NOT
5467 003450 000000 HALT ;ERROR- MASSBUS CONTROL BUS PARITY ERROR
5468 003452 004737 007270 2$: JSR PC, RESREG ;RESTORE RD-R5
5469 003456 000002 RTI ;RETURN TO DRIVER
5470
5471 ;HANDLES CONTROL BUS PARITY ERRORS
5472
5473 003460 032737 020000 177570 ER4: BIT #BIT13, @#SWR ;SKIP TYPEOUT?
5474 003466 001040 BNE 1$ ;BRANCH IF SO
5475 003470 004537 005166 JSR RS, TYP0UT ;TYPE CONTROL BUS PARITY ERROR
5476 003474 011172 MER4
5477 003476 004737 005376 JSR PC, CRLF ;SET MARGIN
5478 003502 004537 005166 JSR RS, TYP0UT ;TYPE WRT.AD
5479 003506 012020 MRWAD
5480 003510 013703 017264 MOV WRT.AD, R3 ;GET ADRS
5481 003514 004737 006604 JSR PC, OCTYP ;TYPE IT
5482 003520 004737 005376 JSR PC, CRLF ;SET MARGIN
5483 003524 004537 005166 JSR RS, TYP0UT ;TYPE WRT.WD
5484 003530 012030 MRWWD
5485 003532 013703 017262 MOV WRT.WD, R3 ;GET WRD
5486 003536 004737 006604 JSR PC, OCTYP ;TYPE IT
5487 003542 004737 005376 JSR PC, CRLF ;SET MARGIN
5488 003546 004537 005166 JSR RS, TYP0UT ;TYPE RD.WRD
5489 003552 012010 MRDWRD
5490 003554 013703 017064 MOV RD.WRD, R3 ;GET WRD
5491 003560 004737 006604 JSR PC, OCTYP ;TYPE IT
5492 003564 004737 005376 JSR PC, CRLF ;SET MARGIN
5493 003570 005737 177570 1$: TST @#SWR ;HALT?
5494 003574 100001 BPL 2$ ;BRANCH IF NOT
5495 003576 000000 HALT ;ERROR- CONTROL BUS PARITY ERROR
5496 003600 004737 007270 2$: JSR PC, RESREG ;RESTORE RD-R5
5497 003604 000002 RTI ;RETURN TO DRIVER
5498
5499 ;HANDLES ATTENTION FROM AN UNAVAILABLE DRIVE
5500
5501 003606 032737 020000 177570 ER5: BIT #BIT13, @#SWR ;SKIP TYPEOUT?
5502 003614 001031 BNE 1$ ;BRANCH IF SO
5503 003616 004537 005166 JSR RS, TYP0UT ;TYPE ATTENTION FROM AN UNAVAILABLE DRIVE
5504 003622 011231 MERS
5505 003624 004537 005166 JSR RS, TYP0UT ;TYPE DRIVE
5506 003630 011762 MDRV
5507 003632 013703 003162 MOV SAVR, R3 ;GET DRIVE #
5508 003636 004737 006724 JSR PC, TORNUM ;TYPE IT

```

```

5509 003642 004737 005376 JSR PC,CRLF ;SET MARGIN
5510 003646 004537 005166 JSR R5,TYPOUT ;TYPE RHAS
5511 003652 011771 MAS
5512 003654 013703 003170 MOV SAVR+6,R3 ;GET RHAS
5513 003660 004737 006604 JSR PC,OCTYP ;TYPE IT
5514 003664 004737 005376 JSR PC,CRLF ;SET MARGIN
5515 003670 004737 007064 JSR PC,SETREG ;GO PREPARE FOR TYPE
5516 003674 004737 007150 JSR PC,RHREGS ;GO TYPE REGISTERS
5517 003700 005737 177570 1$: TST @#SWR ;HALT?
5518 003704 100001 BPL 2$ ;BRANCH IF NOT
5519 003706 000000 HALT ;ERROR- ATTENTION FROM AN UNAVAILABLE DRIVE
5520 003710 004737 007270 2$: JSR PC,RESREG ;RESTORE RO-R5
5521 003714 000002 RTI ;RETURN TO DRIVER
5522
5523
5524 ;THIS CODE DECIDES WHAT TO DO WITH AN ERROR AT THE USER LEVEL
5525
5526 003716 032737 000200 012730 ER: BIT #BIT7,FMTDPB+16 ;DID THE COMMAND TERMINATE
5527 003724 001402 BEQ 1$ ;BRANCH IF FATAL ERROR
5528 003726 000137 004002 JMP ER6 ;GO DETERMINE WHAT KIND OF DATA ERROR
5529 003732 032737 070000 012730 1$: BIT #70000,FMTDPB+16 ;SEE WHICH FATAL ERROR
5530 003740 001402 BEQ 2$ ;BRANCH IF A PARITY ERROR
5531 003742 000137 004776 JMP ER7 ;GO REPORT DRIVE UNSAFE
5532 003746 032737 004000 012730 2$: BIT #BIT11,FMTDPB+16 ;CHECK CONDITION OF PARITY ERROR
5533 003754 001402 BEQ 3$ ;BRANCH IF FATAL MASSBUS PARITY ERROR
5534 003756 000137 005042 JMP ER8 ;GO REPORT UNCORRECTABLE MASSBUS PARITY ER
5535 003762 032737 002000 012730 3$: BIT #BIT10,FMTDPB+16 ;CHECK FOR FATAL PARITY ERROR
5536 003770 001402 BEQ 4$ ;BR IF NOT
5537 003772 000137 005106 JMP ER9 ;GO REPORT FATAL MASSBUS PARITY ERROR
5538 003776 000137 005136 4$: JMP ER10 ;MUST BE SOFTWARE TIMEOUT
5539
5540 ;HANDLES THE DATA ERROR INDICATIONS
5541
5542 004002 032737 000070 012730 ER6: BIT #70,FMTDPB+16 ;SEE IF A DATA ERROR
5543 004010 001402 BEQ 1$ ;BRANCH IF SO
5544 004012 000137 004506 JMP ER6A ;GO REPORT ERROR OTHER THAN DATA
5545 004016 122737 000173 012714 1$: CMPB #173,FMTDPB+2 ;SEE IF DOING A READ HEADER
5546 004024 001002 BNE 2$ ;BRANCH IF NOT
5547 004026 000137 004742 JMP ER6D ;GO REPORT ERROR ON READ HEADER
5548 004032 122737 000163 012714 2$: CMPB #163,FMTDPB+2 ;SEE IF DOING A WRITE HEADER
5549 004040 001002 BNE 3$ ;BRANCH IF NOT
5550 004042 000137 004706 JMP ER6C ;GO REPORT ERROR ON WRITE HEADER
5551 004046 005737 012542 3$: TST SOFSW ;IS THIS A RETRY?
5552 004052 001002 BNE 4$ ;BRANCH IF SO
5553 004054 000137 004140 JMP ERROR6 ;GO REPORT ERROR ON WRITE CHECK HEADER
5554 004060 032737 020000 177570 4$: BIT #BIT13,@#SWR ;SKIP TYPEOUT?
5555 004066 001006 BNE 5$ ;BRANCH IF SO
5556 004070 004537 005166 JSR R5,TYPOUT ;GO TYPE RETRIES UNSUCCESSFUL
5557 004074 012415 MREUNS
5558 004076 004537 005166 JSR R5,TYPOUT ;GO TYPE SECTOR NOT ACCEPTABLE
5559 004102 012254 MUNCOR
5560 004104 005737 177570 5$: TST @#SWR ;HALT?
5561 004110 100001 BPL 6$ ;BRANCH IF NOT
5562 004112 000000 HALT ;ERROR- SECTOR FAILS TO FORMAT AFTER 1 OR 2 RETRIES

```



```

5563 004114 022737 000025 012576 6$: CMP #25, SAVSEC ; IS THIS THE LAST SECTOR IN THE TRK?
5564 004122 001403 BEQ 7$ ; BRANCH IF SO
5565 004124 004737 006260 JSR PC, SCAWC ; GO SET UP TBL FOR REMAINING SECTORS
5566 004130 000207 RTS PC ; GO TRY NEXT SECTOR
5567 004132 005726 7$: TST (SP)+ ; RESTORE SP
5568 004134 000137 002516 JMP LOOPTS ; TRY NEXT TRK
5569
5570 ; HANDLES A DATA ERROR ON A WRITE CHECK COMMAND
5571
5572 004140 005237 012542 ERROR6: INC SOFSW ; RECORD DATA ERROR
5573 004144 013704 012650 MOV SAVREG+6, R4 ; GO GET RHDA
5574 004150 042704 177740 BIC #177740, R4 ; SAVE ONLY THE BAD SECTOR + 1
5575 004154 001002 BNE 1$ ; BRANCH IF NOT ZERO
5576 004156 012704 000026 MOV #26, R4 ; RESTORE TO LAST SECTOR +1
5577 004162 005304 1$: DEC R4 ; ADJUST SECTOR TO THE ONE THAT FAILED
5578 004164 010437 012576 MOV R4, SAVSEC ; SAVE IT
5579 004170 032737 020000 177570 BIT #BIT13, @#SWR ; SKIP TYPEOUT?
5580 004176 001076 BNE SETCA ; BRANCH IF SO
5581 004200 004537 005166 JSR R5, TYP0UT ; GO TYPE WRITE CHECK ERROR
5582 004204 011303 MERG
5583 004206 004737 006662 JSR PC, DRSN ; GO TYPE DR & SN
5587 004212 004737 006744 JSR PC, CYLTK ; GO TYPE CYL & TRK
5588 004216 004537 005166 JSR R5, TYP0UT ; GO TYPE SECTOR=
5589 004222 012057 MSTORE
5590 004224 005000 CLR R0 ; CLR TENS DIGIT
5591 004226 013703 012576 MOV SAVSEC, R3 ; GET SECTOR #
5592 004232 162703 000012 2$: SUB #12, R3 ; SEE HOW MANY TENS
5593 004236 100402 BMI 3$ ; BRANCH IF TOO SMALL
5594 004240 005200 R0 ; RECORD TEN
5595 004242 000773 BR 2$ ; SUBTRACT AGAIN
5596 004244 062703 000012 3$: ADD #12, R3 ; MAKE SECTOR POSITIVE AGAIN
5597 004250 062700 000260 ADD #260, R0 ; SET UP FOR PRINT
5598 004254 004737 005204 JSR PC, T10 ; GO PRINT IT
5599 004260 010300 MOV R3, R0 ; MOVE LSD INTO R0
5600 004262 062700 000260 ADD #260, R0 ; SET UP FOR PRINT
5601 004266 004737 005204 JSR PC, T10 ; GO PRINT LSD
5602 004272 004537 005166 JSR R5, TYP0UT ; GO TYPE GOOD DATA
5603 004276 012134 MGDATA
5604 004300 013703 012646 MOV SAVREG+4, R3 ; GET ADRS OF FAILURE
5605 004304 162703 000002 SUB #2, R3 ; BACK IT OFF BY TWO
5606 004310 011303 MOV (R3), R3 ; GET THE DATA WORD
5607 004312 004737 006604 JSR PC, OCTYP ; GO TYPE IT
5608 004316 004537 005166 JSR R5, TYP0UT ; GO TYPE BAD DATA
5609 004322 012067 MBDATA
5610 004324 013703 012664 MOV SAVREG+22, R3 ; GET BAD DATA AT RHDB
5611 004330 004737 006604 JSR PC, OCTYP ; GO TYPE IT
5612 004334 004537 005166 JSR R5, TYP0UT ; GO TYPE RHEC1=
5613 004340 012121 MRHEC1
5614 004342 013703 012706 MOV SAVREG+44, R3 ; GO GET RHEC1
5615 004346 004737 006604 JSR PC, OCTYP ; GO TYPE IT
5616 004352 004537 005166 JSR R5, TYP0UT ; GO TYPE RHEC2
5617 004356 012132 MRHEC2
5618 004360 013703 012710 MOV SAVREG+46, R3 ; GET RHEC2
5619 004364 004737 006604 JSR PC, OCTYP ; GO TYPE IT

```

```

5620 004370 004737 007150      JSR    PC,RHREGS      ;GO TYPE RH REGS
5621
5622                          ;THIS CODE COMPUTES WC & CA FOR RETRIES ON WRITE CHECK ERRORS
5623
5624 004374 012700 020300      SETCA: MOV    #BUFF,RO      ;GET STARTING ADRS OF BUFFER
5625 004400 013701 012576      MOV    SAVSEC,R1      ;GET SECTOR
5626 004404 005701 1$:      TST    R1              ;SEE IF ZERO
5627 004406 001404          BEQ    2$              ;BRANCH IF SO
5628 004410 062700 001010      ADD    #520.,RO      ;ADVANCE TO NEXT SECTOR
5629 004414 005301          DEC    R1              ;KEEP COUNT
5630 004416 000772          BR     1$              ;GO CHECK NEXT SECTOR
5631 004420 010037 012720      2$:  MOV    RO,FMTDPB+6  ;SET UP CA OF BAD SECTOR
5632 004424 013700 012604      SETWC: MOV    MWC,RO     ;GET MAX WC
5633 004430 013701 012576      MOV    SAVSEC,R1     ;GET SECTOR
5634 004434 005701 1$:      TST    R1              ;SEE IF ZERO
5635 004436 001404          BEQ    2$              ;BRANCH IF SO
5636 004440 062700 000404      ADD    #260.,RO     ;ADVANCE TO NEXT SECTOR
5637 004444 005301          DEC    R1              ;KEEP COUNT
5638 004446 000772          BR     1$              ;GO CHECK NEXT SECTOR
5639 004450 062700 000404      2$:  ADD    #260.,RO     ;POINT WC TO NEXT SECTOR
5640 004454 010037 012600      MOV    RO,SAVWC      ;SAVE WC OF REMAINING SECTORS
5641 004460 012737 177374 012716  MOV    #-260.,FMTDPB+4 ;SET UP WC FOR BAD SECTOR
5642 004466 113737 012576 012722  MOVB   SAVSEC,FMTDPB+10 ;SET UP SECTOR ADRS IN TBL
5643 004474 005737 177570      TST    @#SWR          ;HALT?
5644 004500 100001          BPL    3$              ;BRANCH IF NOT
5645 004502 000000          HALT                    ;ERROR- DATA ERROR
5646 004504 000207      3$:  RTS    PC              ;GO REFORMAT FAILING SECTOR
5647
5648                          ;CHECK FOR READ OR WRITE UNSAFE ERROR
5649
5650 004506 032737 000020 012730  ER6A: BIT    #20,FMTDPB+16  ;CHECK FOR AN UNSAFE
5651 004514 001422          BEQ    ER6B            ;BRANCH IF NOT AN UNSAFE
5652 004516 032737 020000 177570  BIT    #BIT13,@#SWR   ;SKIP TYPEOUT?
5653 004524 001010          BNE    1$              ;BRANCH IF SO
5654 004526 004537 005166      JSR    RS,TYPOUT     ;GO TYPE READ OR WRITE UNSAFE ERROR
5655 004532 011332          MER6A
5656 004534 004737 007322      JSR    PC,TYPALL     ;GO TYPE DATA
5657 004540 004537 005166      JSR    RS,TYPOUT     ;GO TYPE REFORMAT PACK
5658 004544 012307          MREFMT
5659 004546 005737 177570      1$:  TST    @#SWR          ;HALT?
5660 004552 100001          BPL    2$              ;BRANCH IF NOT
5661 004554 000000          HALT                    ;ERROR- READ OR WRITE UNSAFE OCCURED
5662 004556 000137 001156      2$:  JMP    RESTRT        ;GO GET NEXT DRIVE
5663
5664                          ;INDICATE A DRIVE ERROR HAS OCCURED
5665
5666 004562 005737 012574          ER6B: TST    RETRY          ;IS THIS A RETRY?
5667 004566 001020          BNE    3$              ;BRANCH IF SO
5668 004570 032737 020000 177570  BIT    #BIT13,@#SWR   ;SKIP TYPEOUT?
5669 004576 001005          BNE    1$              ;BRANCH IF SO
5670 004600 004537 005166      JSR    RS,TYPOUT     ;GO TYPE DRIVE ERROR
5671 004604 011372          MER6B
5672 004606 004737 007322      JSR    PC,TYPALL     ;GO TYPE DATA
5673 004612 005237 012574      1$:  INC    RETRY          ;RECORD ERROR

```

```

5674 004616 005737 177570      TST      2#SWR      ;HALT?
5675 004622 100001                BPL      25        ;BRANCH IF NOT
5676 004624 000000                HALT                    ;ERROR- DRIVE ERROR
5677 004626 000207                RTS      PC        ;TRY AGAIN
5678 004630 022737 000002 012574 3$:  CMP      2,RETRY    ;IS THIS THE LAST RETRY
5679 004636 001403                BEQ      4$        ;BRANCH IF SO
5680 004640 005237 012574                INC      RETRY     ;RECORD RETRY
5681 004644 000207                RTS      PC        ;TRY AGAIN
5682 004646 032737 020000 177570 4$:  BIT      #BIT13,2#SWR ;SKIP TYPEOUT?
5683 004654 001006                BNE      5$        ;BRANCH IF SO
5684 004656 004537 005166                JSR      R5,TYPOUT ;GO TYPE RETRIES UNSUCCESSFUL
5685 004662 012415                MREUNS
5686 004664 004537 005166                JSR      R5,TYPOUT ;GO TYPE REFORMAT PACK
5687 004670 012307                MREFMT
5688 004672 005737 177570      5$:  TST      2#SWR      ;HALT?
5689 004676 100001                BPL      6$        ;BRANCH IF NOT
5690 004700 000000                HALT                    ;ERROR- DRIVE ERROR STILL THERE AFTER TWO RETRIES
5691 004702 000137 001156      6$:  JMP      RESTRT    ;GO ASK FOR DRIVE
5692
5693                                ;HANDLES A DATA ERROR ON THE WRITE HEADER COMMAND
5694
5695 004706 032737 020000 177570 ER6C: BIT      #BIT13,2#SWR ;SKIP TYPEOUT?
5696 004714 001005                BNE      1$        ;BRANCH IF SO
5697 004716 004537 005166                JSR      R5,TYPOUT ;GO TYPE WRITE ERROR
5698 004722 011413                MER6C
5699 004724 004737 007322                JSR      PC,TYPALL ;GO TYPE DATA
5700 004730 005737 177570      1$:  TST      2#SWR      ;HALT?
5701 004734 100001                BPL      2$        ;BRANCH IF NOT
5702 004736 000000                HALT                    ;ERROR- ERROR OCCURED ON WRITE COMMAND
5703 004740 000207                2$:  RTS      PC        ;TRY TO WRITE AGAIN
5704
5705                                ;HANDLES A DATA ERROR ON THE READ HEADER COMMAND
5706
5707 004742 032737 020000 177570 ER6D: BIT      #BIT13,2#SWR ;SKIP TYPEOUT?
5708 004750 001005                BNE      1$        ;BRANCH IF SO
5709 004752 004537 005166                JSR      R5,TYPOUT ;GO TYPE HEADER COMPARE NOT MADE
5710 004756 011434                MER6D
5711 004760 004737 007322                JSR      PC,TYPALL ;GO TYPE DATA
5712 004764 005737 177570      1$:  TST      2#SWR      ;HALT?
5713 004770 100001                BPL      2$        ;BRANCH IF NOT
5714 004772 000000                HALT                    ;ERROR- HEADER COMPARE NOT MAJE
5715 004774 000207                2$:  RTS      PC        ;RETURN
5716
5717                                ;HANDLES THE DRIVE UNSAFE ERRORS
5718
5719 004776 032737 020000 177570 ER7:  BIT      #BIT13,2#SWR ;SKIP TYPEOUT?
5720 005004 001010                BNE      1$        ;BRANCH IF SO
5721 005006 004537 005166                JSR      R5,TYPOUT ;GO TYPE DRIVE UNSAFE
5722 005012 011556                MER7
5723 005014 004737 007322                JSR      PC,TYPALL ;GO TYPE DATA
5724 005020 004537 005166                JSR      R5,TYPOUT ;GO TYPE FORMAT TERMINATED
5725 005024 012226                MFTERM
5726 005026 005737 177570      1$:  TST      2#SWR      ;HALT?
5727 005032 100001                BPL      2$        ;BRANCH IF NOT

```

```

5728 005034 000000          HALT          ;ERROR- DRIVE UNSAFE ERROR
5729 005036 000137 001156 2$: JMP          RESTRT ;GO GET NEXT DRIVE
5730
5731          ;HANDLES UNCORRECTABLE MASSBUS PARITY ERROR
5732
5733 005042 032737 020000 177570 ER9: BIT          #BIT13,#SWR ;SKIP TYPEOUT?
5734 005050 001010          BNE          1$      ;BRANCH IF SO
5735 005052 004537 005166          JSR          RS,TYPOUT ;GO TYPE UNCORRECTABLE MASSBUS PARITY ERROR
5736 005056 011606          MERB
5737 005060 004537 005166          JSR          RS,TYPOUT ;GO TYPE FORMAT TERMINATED
5738 005064 012226          MFTERM
5739 005066 004737 006744          JSR          PC,CYLTK ;GO GIVE THEM CYL, TRK=XXX,XX
5740 005072 005737 177570 1$: TST          #SWR   ;HALT?
5741 005076 100001          BPL          2$      ;BRANCH IF NOT
5742 005100 000000          HALT          ;ERROR- UNCORRECTABLE MASSBUS PARITY ERROR
5743 005102 000137 001156 2$: JMP          RESTRT ;GO GET NEXT DRIVE
5744
5745          ;HANDLES FATAL MASSBUS PARITY ERROR
5746
5747 005106 032737 020000 177570 ER9: BIT          #BIT13,#SWR ;SKIP TYPEOUT?
5748 005114 001005          BNE          1$      ;BRANCH IF SO
5749 005116 004537 005166          JSR          RS,TYPOUT ;GO TYPE FATAL MASSBUS PARITY ERROR
5750 005122 011656          MER9
5751 005124 004737 006744          JSR          PC,CYLTK ;GO GIVE THEM CYL TRK=XXX,XX
5752 005130 000000 1$: HALT          ;FATAL MASSBUS PARITY ERROR
5753 005132 000137 001156          JMP          RESTRT ;RESTART PROGRAM
5754
5755
5756
5757          ;HANDLES A SOFTWARE TIMEOUT (HUNG RPC4)
5758
5759 005136 032737 020000 177570 ER10: BIT         #BIT13,#SWR ;SKIP TYPEOUT?
5760 005144 001005          BNE          1$      ;BR IF SO
5761 005146 004537 005166          JSR          RS,TYPOUT ;GO TYPE 'SOFTWARE TIMEOUT'
5762 005152 011716          MER10          ;ADRS OF MSG
5763 005154 004737 007322          JSR          PC,TYPALL ;GO TYPE DATA
5764 005160 000000 1$: HALT          ;SYSTEM HUNG - OPERATION FAILED TO COMPLETE
5765 005162 000137 001156          JMP          RESTRT ;GO RESTART PROGRAM
5766
5767          .SBTTL SUPPORT SUBROUTINES
5768
5769
5770          ;TYPES OUT CHARACTERS POINTED TO BY THE ADRS IN R1
5771
5772 005166 012501          TYP0UT: MOV          (R5)+,R1 ;SET UP ADRS OF MESSAGE
5773 005170 112100 1$: MOVB         (R1)+,R0 ;GET CHARACTER
5774 005172 001403          BEQ          2$      ;ARE WE DONE TYPING?
5775 005174 004737 005204          JSR          PC,TTO  ;GO TYPE IT
5776 005200 000773          BR          1$      ;GET NEXT CHARACTER
5777 005202 000205 2$: RTS          RS    ;EXIT - DONE TYPING
5778
5779          ;DOES THE ACTUAL TYPING - TYPES THE NUMBER OF NULL CHARS SPECIFIED IN 'NULCNT'
5780
5781

```

```

5782 005204 010446      T10:  MOV    R4, -(SP)      ;SAVE R4
5783 005206 013704 012620  T10:  MOV    NULCNT, R4     ;SET UP NULL COUNT
5784 005212 105737 177564  T101: TSTB   TPS         ;IS PRINTER AVAILABLE?
5785 005216 100375      BPL    T101            ;NO - KEEP LOOPING
5786 005220 110037 177566  MOVB   RO, TPB        ;PRINT CHAR
5787 005224 022700 000015  CMP    #15, RO        ;IS IT A RETURN?
5788 005230 001002      BNE    T102           ;BRANCH IF NOT
5789 005232 005304      DEC    R4             ;KEEP TRACK OF NULL CHARS
5790 005234 001366      BNE    T101           ;BRANCH IF MORE NULL CHARS
5791 005236 012604      T102: MOV    (SP)+, R4   ;RESTORE R4
5792 005240 000207      RTS    PC             ;EXIT FROM PRINTING
5793
5794
5795
;READS THE KEYBOARD AND CHECKS FOR 'CONTROL C'
5796 005242 105737 177560  T11:  TSTB   TKS         ;CHECK FOR CHAR
5797 005246 100375      BPL    T11            ;WAIT FOR CHAR
5798 005250 013700 177562  MOV    TKB, RO        ;PUT CHAR IN RC
5799 005254 020027 000215  CMP    RO, #215      ;DON'T PRINT A CR
5800 005260 001405      BEQ    IS             ;SKIP OVER TYPE
5801 005262 020027 000212  CMP    RO, #212      ;DON'T PRINT A LF
5802 005266 001402      BEQ    IS             ;SKIP OVER TYPE
5803 005270 004737 005204  JSR    PC, T10        ;ECHO CHAR TYPED
5804 005274 042700 000200  IS:   BIC    #200, RO     ;GET RID OF BIT 7
5805 005300 022700 000003  CMP    #3, RO        ;IS CHAR IN RO A CNTRL C?
5806 005304 001002      BNE    CTRLR         ;NO - SKIP OVER JUMP
5807 005306 000137 001156  JMP    RESTR         ;YES - GO TO START OF PROGRAM
5808 005312 000207      CTRLR: RTS    PC     ;RETURN TO MAIN PROGRAM
5809
;TESTS THAT A INPUT CHARACTER WAS A NUMBER
5810
5811
5812 005314 020027 000060  NUMTST: CMP    RO, #60     ;IS ASCII SMALLER THAN 60?
5813 005320 100410      BMI    TYPER         ;YES - GO TO TYPER
5814 005322 020027 000072  CMP    RO, #72     ;IS ASCII LARGER THAN 71?
5815 005326 100005      BPL    TYPER         ;YES - GO TO TYPER
5816 005330 042700 177760  BIC    #177760, RO   ;GET RID OF ASCII
5817 005334 062716 000002  ADD    #2, (SP) ;ADD TWO TO RETURN ADRS
5818 005340 000207      RTS    PC           ;RETURN - GOOD NUMBER
5819 005342 004737 005406  TYPER: JSR    PC, QUES ;GO TYPE ??
5820 005346 000207      RTS    PC           ;RETURN TO MAIN PROGRAM
5821
;PRINTS OUT THE DRIVE SELECTED NUMBER
5822
5823
5824 005350 013700 012540  UASM: MOV    USEL, RO   ;GET DRIVE SELECTED
5825 005354 062700 000260  ADD    #260, RO     ;MAKE IT ASCII
5826 005360 004737 005204  JSR    PC, T10     ;GO PRINT IT
5827 005364 012700 000240  MOV    #240, RO    ;PROVIDE SPACE
5828 005370 004737 005204  JSR    PC, T10     ;GO PRINT IT
5829 005374 000207      RTS    PC           ;RETURN
5830
;THIS ROUTINE DOES A CARRIAGE RETURN AND LINE FEED
5831
5832
5833 005376 004537 005166  CRLF: JSR    R5, TYP0UT ;PRINT LF + CR
5834 005402 007574      MCRLF
5835 005404 000207      RTS    PC           ;RETURN

```

```

5836
5837 ;THIS ROUTINE PRINTS ?? ON IMPROPER KEYBOARD INPUTS
5838
5839 005406 004537 005166      QUES: JSR    R5,TYPOUT      ;GO TYPE MESSAGE
5840 005412 007567              MQUES  ;ADRS OF MESSAGE
5841 005414 000207              RTS     PC                ;RETURN TO MAIN PROGRAM
5842
5843 ;THIS ROUTINE SETS UP THE TTY INPUT BUFFER AND COUNTER
5844
5845 005416 012703 012634      SETPC: MOV    #CPTR,R3      ;PUT CHAR PTR IN R3
5846 005422 010204              MOV    R2,R4            ;PUT CHAR COUNT IN R4
5847 005424 000207              RTS     PC                ;RETURN
5848
5849 ;THIS ROUTINE CLEARS THE TTY INPUT BUFFER
5850
5851 005426 004737 005416      CLRBUF: JSR   PC,SETPC     ;GO SET UP POINTER + COUNT
5852 005432 005023              1$: CLR    (R3)+         ;MOVE 0'S THRU R3 R4 TIMES
5853 005434 005304              DEC    R4              ;KEEP COUNT
5854 005436 001375              BNE    1$              ;DONE IF R4 EQ 0
5855 005440 004737 005416      JSR   PC,SETPC     ;SET UP POINTER + COUNT AGAIN
5856 005444 000207              RTS     PC                ;EXIT
5857
5858 ;THIS ROUTINE READS & CHECKS THE CYL & TRK INPUT FROM THE TTY
5859
5860 005446 012702 000004      GETDAT: MOV   #4,R2      ;SET UP MAX # OF DIGITS EXPECTED
5861 005452 004737 005426      JSR   PC,CLRBUF     ;GO CLEAR DIGIT BUFFER
5862 005456 004737 005242      JSR   PC,TTI        ;GO GET CHAR
5863 005462 022700 000104      CMP   #104,R0       ;IS IT THE DEFAULT CHAR?
5864 005466 001031              BNE    4$            ;BRANCH IF NOT
5865 005470 005737 012542      TST   SOFSW         ;IS THIS STARTING OR ENDING CYL?
5866 005474 001410              BEQ   1$            ;BRANCH IF ENDING CYL
5867 005476 005037 012550      CLR   SCYL          ;SET STARTING CYL TO 0
5868 005502 005037 012554      CLR   STRK          ;SET STARTING TRK TO 0
5869 005506 004537 005166      JSR   R5,TYPOUT     ;INDICATE DEFAULT VALUES
5870 005512 010272              MCVTK0 ;ADRS OF MSG
5871 005514 000411              BR    2$            ;GO SET UP RETURN
5872 005516 012737 000632 012546 1$: MOV   #410,ECYL     ;SET ENDING CYL TO 410
5873 005524 012737 000022 012552  MOV   #18,ETRK      ;SET ENDING TRK TO 18
5874 005532 004537 005166      JSR   R5,TYPOUT     ;INDICATE DEFAULT VALUES
5875 005536 010324              MCVTK1 ;ADRS OF MSG
5876 005540 062716 000002      2$: ADD   #2,(SP) ;ADD TWO TO RETURN ADRS
5877 005544 000207              RTS   PC            ;RETURN FROM DEFAULT COND.
5878 005546 004737 005242      3$: JSR   PC,TTI        ;GO GET NEXT CHAR FOR CYLINDER
5879 005552 022700 000177      4$: CMP   #177,R0     ;IS IT A RUBOUT?
5880 005556 001001              BNE    5$            ;BRANCH IF NOT A RUBOUT
5881 005560 000207              RTS   PC            ;GO RETYPE LINE
5882 005562 022700 000054      5$: CMP   #54,R0      ;IS IT THE SEPERATOR?
5883 005566 001024              BNE    8$            ;BRANCH IF NOT A COMMA
5884 005570 004737 006120      JSR   PC,DECBIN     ;CONVERT CYL # TO BINARY
5885 005574 023727 012572 000633  CMP   TEMP,#411.    ;SEE IF CYL REQUEST IS GREATER THAN 410
5886 005602 103404              BCS   6$            ;BRANCH IF WITHIN LIMITS
5887 005604 004537 005166      JSR   R5,TYPOUT     ;TYPE LIMIT EXCEEDED
5888 005610 010334              MCLIMT ;ADRS OF MESSAGE
5889 005612 000207              RTS   PC            ;RETURN TO RETYPE LINE

```

```

5890 005614 005737 012542      6$:  TST      SOFSW      ; IS THIS THE STARTING OR ENDING CYL?
5891 005620 001403          BEQ      7$          ; BRANCH IF ENDING CYL
5892 005622 013737 012572 012550  MOV     TEMP,SCYL   ; SAVE TEMP AT STARTING CYL
5893 005630 013737 012572 012546  7$:  MOV     TEMP,ECYL   ; SAVE TEMP AT ENDING CYL
5894 005636 000412          BR       10$        ; GO GET TRK ADRS
5895 005640 004737 005314      8$:  JSR     PC,NUMTST   ; SEE IF A NUMBER
5896 005644 000207          RTS     PC          ; RETURN - BAD TYPE
5897 005646 005304          DEC     R4         ; KEEP COUNT OF DIGITS COMING IN
5898 005650 001003          BNE     9$         ; BRANCH IF NOT THE 4TH
5899 005652 004737 005406      JSR     PC,QUES     ; NO COMMA AFTER 3 DIGITS - TYPE ??
5900 005656 000207          RTS     PC          ; RETURN - NUMBER TOO BIG
5901 005660 010023          9$:  MOV     RD,(R3)+   ; SAVE DIGIT IN BUFFER
5902 005662 000731          BR       3$         ; GET NEXT CHAR
5903 005664 012702 000003      10$: MOV     #3,R2      ; SET UP MAX # OF DIGITS EXPECTED
5904 005670 004737 005426      JSR     PC,CLRBUF   ; GO CLEAR DIGIT BUFFER
5905 005674 004737 005242      11$: JSR     PC,ITI      ; GO GET NEXT CHAR
5906 005700 022700 000177      CMP     #177,R0     ; IS IT A RUBOUT?
5907 005704 001001          BNE     12$        ; BRANCH IF NOT A RUBOUT
5908 005706 000207          RTS     PC          ; GO RETYPE LINE
5909 005710 022700 000015      12$: CMP     #15,R0     ; IS IT A TERMINATOR?
5910 005714 001026          BNE     15$        ; BRANCH IF NOT A CR
5911 005716 004737 006120      JSR     PC,DECBIN   ; CONVERT TRK # TO BINARY
5912 005722 023727 012572 000023  CMP     TEMP,#19.   ; SEE IF TRK REQUEST IS GREATER THAN 19
5913 005730 103404          BCS     13$        ; BRANCH IF WITHIN LIMITS
5914 005732 004537 005166      JSR     RS,TYPEOUT  ; TYPE LIMIT EXCEEDED
5915 005736 010371          MTLIMT           ; ADRS OF MESSAGE
5916 005740 000207          RTS     PC          ; RETURN - TRK ADRS TO LARGE
5917 005742 005737 012542      13$: TST     SOFSW      ; IS THIS THE STARTING OR ENDING TRK?
5918 005746 001403          BEQ     14$        ; BRANCH IF ENDING TRK
5919 005750 013737 012572 012554  MOV     TEMP,STRK   ; SAVE TEMP AT STARTING TRK
5920 005756 013737 012572 012552  14$: MOV     TEMP,ETRK   ; SAVE TEMP AT ENDING TRK
5921 005764 062716 000002      ADD     #2,(SP) ; ADD TWO TO RETURN ADRS
5922 005770 000207          RTS     PC          ; RETURN - CYL & TRK ADRS ARE SAVED
5923 005772 004737 005314      15$: JSR     PC,NUMTST   ; SEE IF A NUMBER
5924 005776 000207          RTS     PC          ; RETURN - BAD TYPE
5925 006000 005304          DEC     R4         ; KEEP COUNT OF DIGITS COMING IN
5926 006002 001003          BNE     16$        ; BRANCH IF NOT THE 3RD
5927 006004 004737 005406      JSR     PC,QUES     ; NO CR AFTER 2 DIGITS - TYPE ??
5928 006010 000207          RTS     PC          ; RETURN - NUMBER TOO BIG
5929 006012 010023          16$: MOV     RD,(R3)+   ; SAVE DIGIT IN BUFFER
5930 006014 000727          BR       11$       ; GET NEXT CHAR
5931
5932 ; THIS ROUTINE SETS UP THE INPUT TABLE FOR THE DRIVER ROUTINE
5933
5934 006016 113737 012540 012712  SETTBL: MOV     USEL,FMTDPB ; SET UP DRIVE #
5935 006024 105037 012715          CLR     FMTDPB+3   ; CLEAR HIGH ORDER ADRS BITS
5936 006030 013737 012604 012716  MOV     MWC,FMTDPB+4 ; LOAD UP WORD COUNT
5937 006036 012737 020300 012720  MOV     #BUFF,FMTDPB+6 ; LOAD UP CURRENT ADRS
5938 006044 105037 012722          CLR     FMTDPB+10  ; SET SECTOR TO ZERO
5939 006050 113737 012554 012723  MOV     STRK,FMTDPB+11 ; SET UP STARTING TRK ADRS
5940 006056 013737 012550 012724  MOV     SCYL,FMTDPB+12 ; SET UP STARTING CYL
5941 006064 012737 012642 012726  MOV     #SAVREG,FMTDPB+14 ; LOCATE POINTER TO 'RH' REGS
5942 006072 005037 012730          CLR     FMTDPB+16  ; CLEAR RPO4 STATUS
5943 006076 000240          NOP

```

# F03

```

5944 006100 000240      NOP
5945 006102 013737 012554 012562  MOV      STRK,TRKCNT      ;SET UP PARTIAL CYL TRACK COUNT
5946 006110 013737 012556 012560  MOV      TTRKS,TTRKSC    ;SET UP TOTAL TRACKS COUNTER
5947 006116 000207      RTS      PC                ;RETURN FROM SETUP
5948
5949                      ;THIS ROUTINE CONVERTS THE CYL OR TRK ADRS TO OCTAL
5950
5951 006120 014337 012572      DECBIN: MOV      -(R3),TEMP    ;SET UP ONES DIGIT
5952 006124 014300      MOV      -(R3),RO        ;PUT TENS DIGIT IN RO
5953 006126 162700 000001      ADD10:  SUB      #1,RO      ;SEE HOW MANY 10'S
5954 006132 103404      BCS      MSD            ;GO TRY HUNDREDS
5955 006134 062737 000012 012572  ADD      #12,TEMP        ;RECORD 10
5956 006142 000771      BR      ADD10           ;TRY AGAIN
5957 006144 014300      MSD:    MOV      -(R3),RO    ;PUT HUNDREDS DIGIT IN RO
5958 006146 162700 000001      ADD100: SUB      #1,RO      ;SEE HOW MANY 100'S
5959 006152 103404      BCS      DTBONE         ;EXIT IF DONE
5960 006154 062737 000144 012572  ADD      #144,TEMP       ;RECORD 100
5961 006162 000771      BR      ADD100          ;TRY AGAIN
5962 006164 000207      DTBONE: RTS      PC        ;EXIT
5963
5964                      ;THIS ROUTINE CONTROLS THE DISK ADDRESSING AND TOTAL TRK COUNT
5965                      ;IT IS ENTERED AFTER EVERY TRK OPERATION
5966
5968 006166 005737 012560      TRKTST: TST      TTRKSC      ;SEE IF LAST TRK
5969 006172 001001      BNE     1$             ;BRANCH IF NOT LAST TRK
5970 006174 000207      RTS      PC            ;RETURN LAST TRK OF OPERATION
5971 006176 005337 012560      1$:    DEC      TTRKSC      ;COUNT TRK JUST COMPLETED
5972 006202 105237 012723      INCB    FMTDPB+11      ;ADVANCE TRK #
5973 006206 022737 000022 012562  CMP      #22,TRKCNT     ;IS THIS THE LAST TRK IN CYL?
5974 006214 001403      BEQ     2$             ;BRANCH IF SO
5975 006216 005237 012562      INC     TRKCNT         ;COUNT UP TRK WITHIN CYLS
5976 006222 000411      BR      3$             ;BRANCH TO SET UP RETURN
5977 006224 005037 012562      2$:    CLR      TRKCNT     ;INITIALIZE TRKCNT FOR NEXT CYL
5978 006230 105037 012723      CLRB   FMTDPB+11      ;RESET TRK ADRS TO 0
5979 006234 005237 012724      INC     FMTDPB+12      ;UPDATE CYLINDER #
5980 006240 004737 006374      JSR     PC,UPDACY      ;UPDATE CYLINDER # IN MEM TABLE
5981 006244 000402      BR      4$             ;GO TO RETURN SETUP
5982 006246 004737 006420      3$:    JSR     PC,UPDATK   ;UPDATE TRACK # IN MEM TABLE
5983 006252 062716 000002      4$:    ADD      #2,(SP) ;ADD TWO ; TO RETURN ADRS
5984 006256 000207      RTS      PC            ;EXITS HERE IF NOT LAST TRACK
6019
6020                      ;THIS ROUTINE SETS UP WC & CA FOR REMAINING SECTORS
6021                      ;AFTER A WRITE CHECK ERROR
6022
6023 006260 013737 012600 012716  SCARC:  MOV      SAVWC,FMTDPB+4 ;SET UP WC FOR REMAINING SECTORS
6024 006266 062737 001010 012720  ADD      #520,FMTDPB+6 ;SET UP CA FOR REMAINING SECTORS
6025 006274 005237 012722      INC     FMTDPB+10      ;ADVANCE TO NEXT SECTOR IN TBL
6026 006300 005037 012542      CLR     SOFSW          ;RESET RETRY COUNTER
6027 006304 000207      RTS      PC            ;RETURN TO COMPLETE TRK FORMAT
6028
6029
6030                      ;THIS ROUTINE INITIALIZES THE SECTOR COUNT AND HEADER POINTER
6031                      ;FOR THE HEADER & DATA BUFFER IN MEMORY
6032

```



```

6033 006306 013701 012612      SECCNT: MOV      MAXSEC,R1      ;SET UP SECTOR COUNT
6034 006312 012700 020300      MOV      #BUFF,R0      ;SET UP HEADER POINTER IN R0
6035 006316 000207              RTS      PC              ;RETURN
6036
6037                          ;THIS ROUTINE SETS UP THE CYLINDER ADRS, FORMAT BIT, TRACK AND
6038                          ;SECTOR ADRS IN MEMORY WITH THE STARTING CYL - TRK INFORMATION
6039
6040 006320 004737 006306      SETHDR: JSR      PC,SECCNT  ;SET UP SECTOR COUNT AND HEADER POINTER
6041 006324 013702 012550      MOV      SCYL,R2      ;PUT STARTING CYL # IN R2
6042 006330 013703 012554      MOV      STRK,R3      ;PUT STARTING TRK # IN R3
6043 006334 000303              SWAB     R3            ;JUSTIFY TRACK ADRS
6044 006336 005737 012610      TST     SEC20         ;SEE IF 20 OR 22 SECTOR MODE
6045 006342 001402              BEQ     1$            ;BR IF 20 SECTOR MODE
6046 006344 052702 010000      BIS     #10000,R2     ;SET THE 22 SECTOR FORMAT BIT
6047 006350 010220 1$:      MOV     R2,(R0)+      ;WRITE IN HEADER AREA OF CORE THE CYL ADRS
6048 006352 010320              MOV     R3,(R0)+      ;WRITE IN HEADER AREA OF CORE THE TRK ADRS
6049 006354 005020              CLR     (R0)+         ;CLR 1ST KEYWORD
6050 006356 005010              CLR     (R0)         ;CLR 2ND KEYWORD
6051 006360 062700 001002      ADD     #514.,R0      ;SET UP FOR NEXT HEADER
6052 006364 005203              INC     R3            ;UPDATE SECTOR ADRS FOR NEXT HEADER
6053 006366 005301              DEC     R1            ;MAINTAIN COUNT OF SECTORS
6054 006370 002367              BGE     1$            ;BRANCH IF NOT LAST SECTOR
6055 006372 000207              RTS     PC            ;EXIT - HEADERS ARE LOADED INTO CORE
6056
6057                          ;THIS ROUTINE UPDATES THE CYLINDER ADRS OF THE HEADER WORDS IN CORE
6058
6059 006374 004737 006306      UPDACY: JSR      PC,SECCNT  ;SET UP SECTOR COUNT AND HEADER POINTER
6060 006400 005220 1$:      INC     (R0)+         ;INCREMENT FOR NEXT CYLINDER
6061 006402 042710 177400      BIC     #177400,(R0)  ;RESET TRK ADRS TO 0
6062 006406 062700 001006      ADD     #518.,R0      ;SET UP FOR NEXT HEADER
6063 006412 005301              DEC     R1            ;COUNT SECTORS
6064 006414 002371              BGE     1$            ;BRANCH IF NOT LAST SECTOR
6065 006416 000207              RTS     PC            ;EXIT
6066
6067                          ;THIS ROUTINE UPDATES THE TRACK ADRS OF THE HEADER WORDS IN CORE
6068
6069 006420 004737 006306      UPDATK: JSR      PC,SECCNT  ;SET UP SECTOR COUNT AND HEADER POINTER
6070 006424 005720              TST     (R0)+         ;POINT HEADER POINTER TO TRK - SEC ADRS
6071 006426 062710 000400 1$:      ADD     #400,(R0)     ;INDEX TRK ADRS
6072 006432 062700 001010      ADD     #520.,R0      ;SET UP FOR NEXT HEADER
6073 006436 005301              DEC     R1            ;COUNT SECTORS
6074 006440 002372              BGE     1$            ;BRANCH IF NOT LAST SECTOR
6075 006442 000207              RTS     PC            ;EXIT
6076
6077                          ;THIS ROUTINE CONVERTS FAILING CYL & TRK ADRS TO DECIMAL FOR TYPE
6078
6079 006444 012702 010741      BINDEC: MOV      #MCYLTK,R2 ;ESTABLISH POINTER FOR CYL & TRK TYPE
6080 006450 013700 012616      MOV      LOCCYL,R0    ;GO GET BINARY CYL ADRS
6081 006454 005001              CLR     R1            ;RESET WEIGHT COUNT
6082 006456 012703 000144      MOV      #144,R3      ;SET UP WEIGHT
6083 006462 162700 000144 1$:      SUB     #144,R0        ;SUBTRACT 100 DEC
6084 006466 100402              BMI     2$            ;BRANCH IF # NOT LARGE ENOUGH
6085 006470 005201              INC     R1            ;RECORD EACH 100 DEC
6086 006472 000773              BR     1$            ;SUB AGAIN

```

```

6087 006474 004737 006572      2$: JSR    PC,MKDIG      ;GO MAKE ASCII DIGIT & STORE
6088 006500 005001              CLR    R1              ;RESET WEIGHT COUNT
6089 006502 C12703 000012      MOV    #12,R3         ;SET UP WEIGHT
6090 006506 162700 000012      3$: SUB    #12,R0         ;SUBTRACT 10 DEC
6091 006512 100402              BMI    4$             ;BRANCH IF # NOT LARGE ENOUGH
6092 006514 005201              INC    R1              ;RECORD EACH 10 DEC
6093 006516 000773              BR     3$             ;SUB AGAIN
6094 006520 004737 006572      4$: JSR    PC,MKDIG      ;GO MAKE ASCII DIGIT & STORE
6095 006524 062700 000260      ADD    #260,R0         ;MAKE LSD ASCII DIGIT
6096 006530 110022              MOVB   R0,(R2)+        ;AND STORE IT AWAY
6097 006532 112722 000254      MOVB   #254,(R2)+     ;STORE A COMMA AS A SEPARATOR
6098 006536 113700 012614      MOVB   LOCTRK,R0      ;LOAD R0 WITH CURRENT TRK ADRS
6099 006542 005001              CLR    R1              ;RESET WEIGHT COUNT
6100 006544 162700 000012      5$: SUB    #12,R0         ;SUBTRACT 10 DEC
6101 006550 100402              BMI    6$             ;BRANCH IF # NOT LARGE ENOUGH
6102 006552 005201              INC    R1              ;RECORD EACH 10 DEC
6103 006554 000773              BR     5$             ;SUB AGAIN
6104 006556 004737 006572      6$: JSR    PC,MKDIG      ;GO MAKE ASCII DIGIT & STORE
6105 006562 062700 000260      ADD    #260,R0         ;MAKE LSD ASCII DIGIT
6106 006566 110022              MOVB   R0,(R2)+        ;STORE LSD OF TRK
6107 006570 000207              RTS     PC             ;GO RETURN AND TYPE BAD CYL & TRK
6108
6109                               ;THIS ROUTINE STORES THE FAILING CYL & TRK ADRS IN MCYLTK
6110
6111 006572 062701 000260      MKDIG: ADD    #260,R1         ;MAKE ASCII DIGIT
6112 006576 110122              MOVB   R1,(R2)+        ;PUSH DIGIT INTO MCYLTK
6113 006600 060300              ADD    R3,R0           ;MAKE # POS AGAIN
6114 006602 000207              RTS     PC             ;RETURN TO SET UP NEXT DIGIT
6115
6116                               ;THIS ROUTINE CONVERTS BINARY TO OCTAL FOR TYPE-R3 HAS #
6117
6118 006604 012702 000006      OCTYP: MOV    #6,R2         ;SET UP DIGIT COUNTER
6119 006610 006103              ROL    R3              ;ADJUST FOR FIRST TYPE
6120 006612 006103              ROL    R3              ;
6121 006614 010300              MOV    R3,R0           ;SEND TO TYPE REG
6122 006616 042700 177776      BIC    #177776,R0      ;CLEAR OUT GARBAGE
6123 006622 000405              BR     CHOP            ;GO MAKE DIGIT
6124 006624 006103              NXD:  ROL    R3         ;ADJUST FOR NEXT TYPE
6125 006626 006103              ROL    R3              ;
6126 006630 006103              ROL    R3              ;
6127 006632 006103              ROL    R3              ;
6128 006634 010300              MOV    R3,R0           ;SEND TO TYPE REG
6129 006636 006003              CHOP: ROR    R3         ;SAVE CARRY
6130 006640 042700 177770      BIC    #177770,R0      ;CLEAR OUT GARBAGE
6131 006644 062700 000260      ADD    #260,R0         ;MAKE A DIGIT
6132 006650 004737 005204      TYOCT: JSR    PC,T10     ;TYPE IT
6133 006654 005302              DEC    R2              ;IS IT THE LAST?
6134 006656 001362              BNE    NXD            ;BRANCH IF SO
6135 006660 000207              RTS     PC             ;DONE TYPING-RETURN
6136
6137                               ;THIS ROUTINE TYPES THE DRIVE# AND S/N
6138
6139 006662 004537 005166      DRSN: JSR    R5,TYOUT    ;GO TYPE DRIVE=
6140 006666 011762              MDRV

```

```

6141 006670 013703 012652      MOV     SAVREG+10,R3      ;GET DRIVE # FROM RHCS2
6142 006674 004737 006724      JSR     PC,TDRNUM        ;GO TYPE DRIVE #
6143 006700 004537 005166      JSR     RS,TYPOUT        ;TYPE SN=
6144 006704 012040                MSN
6145 006706 013703 012672      MOV     SAVREG+30,R3      ;GET DRIVE SN
6146 006712 004737 007020      JSR     PC,DECTYP        ;GO TYPE IT
6147 006716 004737 005376      JSR     PC,CRLF          ;SET UP MARGIN
6148 006722 000207                RTS     PC                ;RETURN TO ER HANDLER
6149
6150                                ;THIS ROUTINE TYPES THE FAILING DRIVE #
6151
6152 006724 042703 177770      TDRNUM: BIC     #177770,R3      ;CLEAR JUNK
6153 006730 010300                MOV     R3,R0            ;PLACE DIGIT IN OUTPUT REG
6154 006732 062700 000260      ADD     #260,R0          ;MAKE ASCII
6155 006736 004737 005204      JSR     PC,TTO           ;GO TYPE IT
6156 006742 000207                RTS     PC                ;RETURN
6157
6158                                ;THIS ROUTINE TYPES THE CYL,TRK=XXX,XX
6159
6160 006744 113737 012723 012614  CYLTK:  MOVB    FMTDPB+11,LOCTRK ;TRACK ADDRESS
6161 006752 013737 012724 012616  MOV     FMTDPB+12,LOCCYL ;CYLINDER ADDRESS
6162 006760 004537 005166      CYLTRK: JSR     RS,TYPOUT        ;GO TYPE CYL,TRK=
6163 006764 012046                MHCYTK
6164 006766 004737 006774      JSR     PC,TCYLTK        ;GO CONVERT AND TYPE CYL+TRK
6165 006772 000207                RTS     PC                ;RETURN TO ER HANDLER
6166
6167                                ;THIS ROUTINE ASSEMBLES & TYPES THE FAILING CYL & TRK ADRS
6168
6169 006774 004737 006444      TCYLTK: JSR     PC,BINDEC        ;GO CONVERT BIN TO DEC
6170 007000 004537 005166      JSR     RS,TYPOUT        ;TYPE FAILING CYL,TRK
6171 007004 010741                MHCYTK                    ;ADRS OF MESSAGE WITH FAILING CYL-TRK
6172 007006 000207                RTS     PC                ;RETURN
6173
6174                                ;THIS ROUTINE TYPES TWO SPACES
6175
6176 007010 004537 005166      TSPACE: JSR     RS,TYPOUT        ;GO TYPE TWO SPACES
6177 007014 007564                MSPACE
6178 007016 000207                RTS     PC                ;RETURN
6179
6180                                ;THIS ROUTINE TYPES OUT DECIMAL # IN R3
6181
6182 007020 012701 000004      DECTYP: MOV     #4,R1        ;SET UP DIGIT COUNT
6183 007024 006103      NXDD:  ROL     R3            ;SET UP DIGIT
6184 007026 006103      ROL     R3
6185 007030 006103      ROL     R3
6186 007032 006103      ROL     R3
6187 007034 006103      ROL     R3
6188 007036 010300      MOV     R3,R0            ;SET UP TYPE REG
6189 007040 006003      ROR     R3            ;SAVE CARRY
6190 007042 042700 177760      BIC     #177760,R0        ;GET RID OF JUNK
6191 007046 062700 000260      ADD     #260,R0          ;MAKE ASCII
6192 007052 004737 005204      JSR     PC,TTO           ;GO TYPE IT
6193 007056 005301      DEC     R1            ;SEE IF LAST DIGIT
6194 007060 001361      BNE     NXDD            ;BRANCH IF NOT

```

```

6195 007062 000207      RTS      PC      ;RETURN
6196
6197                      ;SETS UP RH REGS FOR THE 'RHREG' ROUTINE
6198
6199 007064 013737 012732 012654 SETREG: MOV      RPERRS, SAVREG+12      ;SET UP RHDS1
6200 007072 013737 012734 012656      MOV      RPERRS+2, SAVREG+14      ;SET UP RHER1
6201 007100 013737 012736 012702      MOV      RPERRS+4, SAVREG+40      ;SET UP RHER2
6202 007106 013737 012740 012704      MOV      RPERRS+6, SAVREG+42      ;SET UP RHER3
6203 007114 017737 174046 012642      MOV      @SAVR+4, SAVREG          ;SET UP RHCS1
6204 007122 013737 013070 007146      MOV      RPADR, IRHCS2           ;GET RPO4 BASE ADRS
6205 007130 062737 000010 007146      ADD      #10, IRHCS2            ;POINT TO RHCS2 ADRS
6206 007136 017737 000004 012652      MOV      @IRHCS2, SAVREG+10      ;GET RHCS2
6207 007144 000207      RTS      PC      ;RETURN
6208 007146 000000      IRHCS2: 0                      ;CONTAINS RHCS2 ADRS
6209
6210                      ;THIS ROUTINE TYPES OUT THE CS1,CS2,DS1,ER1,ER2+ER3 REGISTERS
6211
6212 007150 004537 005166      RHREGS: JSR      R5, TYP0UT        ;GO TYPE THE RH HEADER
6213 007154 012143      MREGHD
6214 007156 013703 012642      MOV      SAVREG, R3             ;GET RHCS1
6215 007162 004737 006604      JSR      PC, OCTYP             ;TYPE IT
6216 007166 004737 007010      JSR      PC, TSPACE           ;SPACE NX TYPE
6217 007172 013703 012652      MOV      SAVREG+10, R3         ;GET RHCS2
6218 007176 004737 006604      JSR      PC, OCTYP             ;TYPE IT
6219 007202 004737 007010      JSR      PC, TSPACE           ;SPACE NX TYPE
6220 007206 013703 012654      MOV      SAVREG+12, R3         ;GET RHDS1
6221 007212 004737 006604      JSR      PC, OCTYP             ;TYPE IT
6222 007216 004737 007010      JSR      PC, TSPACE           ;SPACE NX TYPE
6223 007222 013703 012656      MOV      SAVREG+14, R3         ;GET RHER1
6224 007226 004737 006604      JSR      PC, OCTYP             ;TYPE IT
6225 007232 004737 007010      JSR      PC, TSPACE           ;SPACE NX TYPE
6226 007236 013703 012702      MOV      SAVREG+40, R3         ;GET RHER2
6227 007242 004737 006604      JSR      PC, OCTYP             ;TYPE IT
6228 007246 004737 007010      JSR      PC, TSPACE           ;SPACE NX TYPE
6229 007252 013703 012704      MOV      SAVREG+42, R3         ;GET RHER3
6230 007256 004737 006604      JSR      PC, OCTYP             ;TYPE IT
6231 007262 004737 005376      JSR      PC, CRLF             ;SET MARGIN
6232 007266 000207      RTS      PC      ;RETURN TO ER HANDLER
6233
6234                      ;THIS ROUTINE RESTORES R0-R5 WHEN RETURNING TO THE DRIVER
6235
6236 007270 013701 003162      RESREG: MOV      SAVR, R1        ;RESTORE R0
6237 007274 013703 003164      MOV      SAVR+2, R3           ;RESTORE R3
6238 007300 013704 003166      MOV      SAVR+4, R4           ;RESTORE R4
6239 007304 013705 003170      MOV      SAVR+6, R5           ;RESTORE R5
6240 007310 013700 003172      MOV      SAVR+10, R0          ;RESTORE R0
6241 007314 013702 003174      MOV      SAVR+12, R2          ;RESTORE R2
6242 007320 000207      RTS      PC      ;RETURN TO DRIVER
6243
6244                      ;THIS ROUTINE TYPES THE DRIVE #, SERIAL NUMBER AND
6245                      ;THE CONTENTS OF THE RH11-RPO4 REGISTERS
6246
6247 007322 004737 006662      TYPALL: JSR      PC, DRSN        ;GO TYPE DRIVE # AND SN
6251 007326 004737 006744      JSR      PC, CYLTK           ;GO TYPE CYL, TRK=XXX, XX

```

# K03

```

6252 007332 004737 007150 JSR PC,RHREGS ;GO TYPE RH REGS
6253 007336 000207 RTS PC ;EXIT
6254
6255 ;ROUTINE TO CHECK FOR KW11-L OR KW11-P CLOCKS
6256
6257 007340 012737 007410 000004 CKCLK: MOV #CKCLK1,#ERRVEC ;SET UP VECTOR FOR P CLK
6258 007346 005037 000006 CLR #ERRVEC+2 ;NEW PSW
6259 007352 005777 163162 TST #SLKCSR ;CHECK FOR KW11-P
6260 007356 012737 003104 000104 MOV #CLOCK,104 ;SET UP KW11-P VECTOR
6261 007364 012737 000300 000106 MOV #300,106 ;PSW - PRI 6
6262 007372 012777 177777 163142 MOV #-1,#SLKCSB ;LOAD COUNTER BUFFER
6263 007400 012777 000135 163132 MOV #135,#SLKCSR ;SET CLK - CNT UP
6264 007406 000423 BR CKCLK3
6265 007410 062706 000004 CKCLK1: ADD #4,SP ;RESTORE THE STACK POINTER
6266 007414 012737 007452 000004 MOV #CKCLK2,#ERRVEC ;CHANGE ERROR VECTOR
6267 007422 005777 170120 TST #SLKS ;LOOK FOR KW11-L
6268 007426 012737 003104 000100 MOV #CLOCK,100 ;SET UP KW11-L VECTOR
6269 007434 012737 000300 000102 MOV #300,102 ;PSW - PRI 6
6270 007442 012777 000100 170076 MOV #100,#SLKS ;SET KW11-L INTERRUPT ENABLE
6271 007450 000402 BR CKCLK3
6272 007452 062706 000004 CKCLK2: ADD #4,SP ;RESTORE THE STACK POINTER
6273 007456 012737 000006 000004 CKCLK3: MOV #6,#ERRVEC ;RESTORE THE ERROR VECTOR
6274 007464 000207 RTS PC
6275
6366 007466 005015 050122 032060 MTITLE: .ASCII <15><12>/RPO4 FORMATTER PROGRAM/<15><12>
007474 043040 051117 040515
007502 052124 051105 050040
007510 047522 051107 046501
007516 005015
6367 007520 040515 0471.1 042504 .ASCIZ /MAINDEC-11-DZRPL-B/<15><12><12>
007526 026503 030461 042055
007534 051132 046120 041055
007542 005015 000012
6368
6369 007546 005015 051104 053111 MUNIT: .ASCIZ <15><12>/DRIVE # /
007554 020105 020043 000
6370
6371 007561 075 000060 MDRVD: .ASCIZ /=0/
6372
6373 007564 020040 000 MSPACE: .ASCIZ / /
6374
6375 007567 077 006477 000012 MQUES: .ASCIZ /?/?<15><12>
6376 007574 005015 000 MCRLF: .ASCIZ <15><12>
6377
6378 007577 040 042040 044522 MOFFLN: .ASCIZ / DRIVE OFFLINE/<15><12>
007604 042526 047440 043106
007612 044514 042516 005015
007620 000
6379
6380 007621 040 051104 053111 MDRNP: .ASCIZ / DRIVE NOT PRESENT/<15><12>
007626 020105 047516 020124
007634 051120 051505 047105
007642 006524 000012
6381

```

DZRPLB.P11

SUPPORT SUBROUTINES

6382	007646	042040	044522	042526	MNRPO4: .ASCIZ / DRIVE NOT AN RPO4/<15><12>
	007654	047040	052117	040440	
	007662	020116	050122	032060	
	007670	005015	000		
6383					
6384	007673	040	051104	053111	MUSDR: .ASCIZ / DRIVE UNSAFE/<15><12>
	007700	020105	047125	040523	
	007706	042506	005015	000	
6385					
6386	007713	040	042523	042514	MSELD: .ASCIZ / SELECTED/
	007720	052103	042105	000	
6387					
6394	007725	015	005012	051120	MMODE: .ASCIZ <15><12><12>/PROGRAM MODE (C OR F) = /
	007732	043517	040522	020115	
	007740	047515	042504	024040	
	007746	020103	051117	043040	
	007754	020051	020075	000	
6395					
6396	007761	040	020075	047506	MFORMAT: .ASCIZ / = FORMAT & VERIFY/
	007766	046522	052101	023040	
	007774	053040	051105	043111	
	010002	000131			
6397					
6398	010004	042510	045503	047440	MHECK: .ASCIZ /HECK ONLY/
	010012	046116	000131		
6399					
6400	010016	051117	040515	020124	MORMAT: .ASCIZ /ORMAT & VERIFY/
	010024	020046	042526	044522	
	010032	054506	000		
6401					
6402	010035	015	005012	050117	MSIZE: .ASCIZ <15><12><12>/OPERATE IN 22 SECTOR MODE (Y OR N) ? /
	010042	051105	052101	020105	
	010050	047111	031040	020062	
	010056	042523	052103	051117	
	010064	046440	042117	020105	
	010072	054450	047440	020122	
	010100	024516	037440	000040	
6403					
6404	010106	005015	050117	051105	MSEC22: .ASCIZ <15><12>/OPERATION WILL BE IN 22 SECTOR (16 BIT) MODE/
	010114	052101	047511	020116	
	010122	044527	046114	041040	
	010130	020105	047111	031040	
	010136	020062	042523	052103	
	010144	051117	024040	033061	
	010152	041040	052111	020051	
	010160	047515	042504	000	
6405					
6406	010165	015	047412	042520	MSEC20: .ASCIZ <15><12>/OPERATION WILL BE IN 20 SECTOR (18 BIT) MODE/
	010172	040522	044524	047117	
	010200	053440	046111	020114	
	010206	042502	044440	020116	
	010214	030062	051440	041505	
	010222	047524	020122	030450	
	010230	020070	044502	024524	

6407	010236	046440	042117	000105	
6408	010244	005015	051412	040524	MMINCT: .ASCIZ <15><12><12>/STARTING CYL,TRK= /
	010252	052122	047111	020107	
	010260	054503	026114	051124	
	010266	036513	000040		
6409					
6410	010272	030075	030054	000	MCYTK0: .ASCIZ /=0,0/
6411					
6412	010277	015	042412	042116	MMAXCT: .ASCIZ <15><12>/ENDING CYL,TRK= /
	010304	047111	020107	020040	
	010312	054503	026114	051124	
	010320	036513	000040		
6413					
6414	010324	032075	030061	030454	MCYTK1: .ASCIZ /=410,18/
	010332	000070			
6415					
6416	010334	037477	041440	046131	MCLIMT: .ASCIZ /?? CYLINDER LIMIT EXCEEDED/<15><12>
	010342	047111	042504	020122	
	010350	044514	044515	020124	
	010356	054105	042503	042105	
	010364	042105	005015	000	
6417					
6418	010371	077	020077	051124	MTLIMT: .ASCIZ /?? TRACK LIMIT EXCEEDED/<15><12>
	010376	041501	020113	044514	
	010404	044515	020124	054105	
	010412	042503	042105	042105	
	010420	005015	000		
6419					
6420	010423	077	020077	047105	MADRER: .ASCII /?? ENDING DSK ADRS MUST/<15><12>
	010430	044504	043516	042040	
	010436	045523	040440	051104	
	010444	020123	052515	052123	
	010452	005015			
6421	010454	042502	042440	052521	.ASCIZ /BE EQUAL TO OR GREATER THAN STARTING ADRS/<15><12>
	010462	046101	052040	020117	
	010470	051117	043440	042522	
	010476	052101	051105	052040	
	010504	040510	020116	052123	
	010512	051101	044524	043516	
	010520	040440	051104	006523	
	010526	000012			
6422					
6423	010530	005015	051412	046105	MSELP: .ASCII <15><12><12>/SELECT DATA PATTERN:/<15><12>
	010536	041505	020124	040504	
	010544	040524	050040	052101	
	010552	042524	047122	006472	
	010560	012			
6424	010561	050	024460	055040	.ASCII /(0) ZERO'S/<15><12>
	010566	051105	023517	006523	
	010574	012			
6425	010575	050	024461	047440	.ASCII /(1) ONE'S/<15><12>
	010602	042516	051447	005015	
6426	010610	031050	020051	047527	.ASCIZ /(2) WORST CASE/<15><12><12>

	C10616	051522	020124	040503	
	010624	042523	005015	000012	
6427					
6428	010632	053475	051117	052123	MPATD: .ASCIZ /=WORST CASE/
	010640	041440	051501	000105	
6429					
6430	010646	005015	051412	040524	MSFOU: .ASCIZ <15><12><12>/STARTING FORMAT ON DRIVE #/
	010654	052122	047111	020107	
	010662	047506	046522	052101	
	010670	047440	020116	051104	
	010676	053111	020105	000043	
6431					
6432	010704	005015	051412	040524	MSCHK: .ASCIZ <15><12><12>/STARTING CHECK ON DRIVE #/
	010712	052122	047111	020107	
	010720	044103	041505	020113	
	010726	047117	042040	044522	
	010734	042526	021440	000	
6433					
6434	010741	060	030060	030054	MCYLTK: .ASCIZ /000,00/<15><12>
	010746	006460	000012		
6435					
6436	010752	005015	043012	051117	MFCMPT: .ASCIZ <15><12><12>/FORMAT COMPLETE/<15><12>
	010760	040515	020124	047503	
	010766	050115	042514	042524	
	010774	005015	000		
6437					
6438	010777	015	005012	044103	MCCMPT: .ASCIZ <15><12><12>/CHECK COMPLETE/<15><12>
	011004	041505	020113	047503	
	011012	050115	042514	042524	
	011020	005015	000		
6439					
6440	011023	015	005012	046111	MER1: .ASCIZ <15><12><12>/ILLEGAL SUBSYSTEM INTERRUPT/<15><12><12>
	011030	042514	040507	020114	
	011036	052523	051502	051531	
	011044	042524	020115	047111	
	011052	042524	051122	050125	
	011060	006524	005012	000	
6441					
6442	011065	015	005012	046111	MER2: .ASCIZ <15><12><12>/ILLEGAL DRIVE INTERRUPT/<15><12><12>
	011072	042514	040507	020114	
	011100	051104	053111	020105	
	011106	047111	042524	051122	
	011114	050125	006524	005012	
	011122	000			
6443					
6444	011123	015	005012	040515	MER3: .ASCIZ <15><12><12>/MASSBUS CONTROL BUS PARITY ERROR/<15><12><12>
	011130	051523	052502	020123	
	011136	047503	052116	047522	
	011144	020114	052502	020123	
	011152	040520	044522	054524	
	011160	042440	051122	051117	
	011166	005015	000012		
6445					
6446	011172	005015	041412	047117	MER4: .ASCIZ <15><12><12>/CONTROL BUS PARITY ERROR/<15><12><12>



	011200	051124	046117	041040		
	011206	051525	050040	051101		
	011214	052111	020131	051105		
	011222	047522	006522	005012		
	011230	000				
6447						
6448	011231	015	005012	052101	MER5:	.ASCIZ <15><12><12>/ATTENTION FROM AN UNAVAILABLE DRIVE/<15><12><12>
	011236	042524	052116	047511		
	011244	020116	051106	046517		
	011252	040440	020116	047125		
	011260	053101	044501	040514		
	011266	046102	020105	051104		
	011274	053111	006505	005012		
	011302	000				
6449						
6450	011303	015	005012	051127	MER6:	.ASCIZ <15><12><12>/WRITE CHECK ERROR/<15><12>
	011310	052111	020105	044103		
	011316	041505	020113	051105		
	011324	047522	006522	000012		
6451						
6452	011332	005015	051012	040505	MER6A:	.ASCIZ <15><12><12>/READ OR WRITE UNSAFE ERROR/<15><12>
	011340	020104	051117	053440		
	011346	044522	042524	052440		
	011354	051516	043101	020105		
	011362	051105	047522	006522		
	011370	000012				
6453						
6454	011372	005015	042012	044522	MER6B:	.ASCIZ <15><12><12>/DRIVE ERROR/<15><12>
	011400	042526	042440	051122		
	011406	051117	005015	000		
6455						
6456	011413	015	005012	051127	MER6C:	.ASCIZ <15><12><12>/WRITE ERROR/<15><12>
	011420	052111	020105	051105		
	011426	047522	006522	000012		
6457						
6458	011434	005015	042012	051511	MER6D:	.ASCII <15><12><12>/DISK ERROR TRYING TO PERFORM HEADER COMPARE/<15><12>
	011442	020113	051105	047522		
	011450	020122	051124	044531		
	011456	043516	052040	020117		
	011464	042520	043122	051117		
	011472	020115	042510	042101		
	011500	051105	041440	046517		
	011506	040520	042522	005015		
6459	011514	047520	044523	044524		.ASCIZ /POSITIONING CHECK NOT PERFORMED/<15><12>
	011522	047117	047111	020107		
	011530	044103	041505	020113		
	011536	047516	020124	042520		
	011544	043122	051117	042515		
	011552	006504	000012			
6460						
6461	011556	005015	042012	044522	MER7:	.ASCIZ <15><12><12>/DRIVE UNSAFE ERROR/<15><12>
	011564	042526	052440	051516		
	011572	043101	020105	051105		
	011600	047522	006522	000012		

6462										
6463	011606	005015	052412	041516	MER8:	.ASCIZ	<15><12><12>/UNCORRECTABLE MASSBUS PARITY ERROR/<15><12>			
	011614	051117	042522	052103						
	011622	041101	042514	046440						
	011630	051501	041123	051525						
	011636	050040	051101	052111						
	011644	020131	051105	047522						
	011652	006522	000012							
6464										
6465	011656	005015	043012	052101	MER9:	.ASCIZ	<15><12><12>/FATAL MASSBUS PARITY ERROR/<15><12>			
	011664	046101	046440	051501						
	011672	041123	051525	050040						
	011700	051101	052111	020131						
	011706	051105	047522	006522						
	011714	000012								
6466										
6467	011716	006415	051412	043117	MER10:	.ASCIZ	<15><15><12>/SOFTWARE TIMEOUT - SYSTEM HUNG/<15><12>			
	011724	053524	051101	020105						
	011732	044524	042515	052517						
	011740	020124	020055	054523						
	011746	052123	046505	044040						
	011754	047125	006507	000012						
6468										
6469	011762	051104	053111	036505	MDRV:	.ASCIZ	/DRIVE= /			
	011770	000								
6470										
6471	011771	122	040510	036523	MAS:	.ASCIZ	/RHAS= /			
	011776	000040								
6472										
6473	012000	042122	040456	051104	MROADR:	.ASCIZ	/RD.ADR= /			
	012006	000075								
6474										
6475	012010	042122	053456	042122	MROWRD:	.ASCIZ	/RD.WRD= /			
	012016	000075								
6476										
6477	012020	051127	027124	042101	MWRTAD:	.ASCIZ	/WRT.AD= /			
	012026	000075								
6478										
6479	012030	051127	027124	042127	MWRTWD:	.ASCIZ	/WRT.WD= /			
	012036	000075								
6480										
6481	012040	020040	047123	000075	MSN:	.ASCIZ	/ SN= /			
6482										
6483	012046	054503	026114	051124	MHCYTK:	.ASCIZ	/CYL,TRK= /			
	012054	036513	000							
6484										
6485	012057	123	041505	047524	MSTOR:	.ASCIZ	/SECTOR= /			
	012064	036522	000							
6486										
6487	012067	015	041012	042101	MBDATA:	.ASCIZ	<15><12>/BAD DATA= /			
	012074	020040	040504	040524						
	012102	000075								
6488										
6489	012104	005015	047507	042117	MGOATA:	.ASCIZ	<15><12>/GOOD DATA= /			

	012112	042040	052101	036501	
	012120	000			
6490					
6491	012121	015	051012	042510	MRHEC1: .ASCIZ <15><12>/RHEC1=/ 012126 030503 000075
6492					
6493	012132	005015	044122	041505	MRHEC2: .ASCIZ <15><12>/RHEC2=/ 012140 036462 000
6494					
6495	012143	015	020012	044122	MREGHD: .ASCIZ <15><12>/ RHCS1 RHCS2 RHCS3 RHER1 RHER2 RHER3/<15><12> 012150 051503 020061 020040 012156 044122 051503 020062 012164 020040 044122 051504 012172 020061 020040 044122 012200 051105 020061 020040 012206 044122 051105 020062 012214 020040 044122 051105 012222 006463 000012
6496					
6497	012226	005015	047506	046522	MFTERM: .ASCIZ <15><12>/FORMAT TERMINATED/<15><12> 012234 052101 052040 051105 012242 044515 040516 042524 012250 006504 000012
6498					
5499	012254	026440	051440	041505	MUNCOR: .ASCIZ / - SECTOR NOT ACCEPTABLE/<15><12> 012262 047524 020122 047516 012270 020124 041501 042503 012276 052120 041101 042514 012304 005015 000
6500					
6501	012307	015	005012	042522	MREFMT: .ASCIZ <15><12><12>/REFORMAT PACK/<15><12> 012314 047506 046522 052101 012322 050040 041501 006513 012330 000012
6502					
6503	012332	005015	050012	051517	MBDFMT: .ASCIZ <15><12><12>/POSITIONING ERROR DURING FORMAT - REFORMAT PACK/ 012340 052111 047511 044516 012346 043516 042440 051122 012354 051117 042040 051125 012362 047111 020107 047506 012370 046522 052101 026440 012376 051040 043105 051117 012404 040515 020124 040520 012412 045503 000
6504					
6505	012415	015	051012	052105	MREUNS: .ASCIZ <15><12>/RETRIES UNSUCCESSFUL/ 012422 044522 051505 052440 012430 051516 041525 042503 012436 051523 052506 000114
6506					
6507	012444	054105	042520	052103	MEXPED: .ASCIZ /EXPECTED - / 012452 042105 026440 000040
6508					
6509	012460	042522	042101	020040	MREAD: .ASCIZ /READ - /

6510	012466	020040	026440	000040
6511	012474	005015	051412	043117
	012502	053524	051101	020105
	012510	042510	042101	051105
	012516	041440	046517	040520
	012524	042522	042440	051122
	012532	051117	005015	000

MHOCMP: .ASCIZ <15><12><12>/SOFTWARE HEADER COMPARE ERROR/<15><12>

6512  
6525  
6526  
6527  
6528  
6529  
6530  
6531  
6532  
6533  
6534  
6535  
6536  
6537  
6538  
6539  
6540  
6541  
6542  
6543  
6544  
6545  
6546  
6547  
6548  
6549  
6550  
6551  
6552  
6553  
6554  
6555  
6556  
6563  
6564  
6565  
6566  
6567  
6568  
6569  
6570  
6571  
6572  
6573  
6574  
6575

;LISTED BELOW ARE LOCATIONS CONTAINING  
;PARAMETERS USED BY THE FORMAT OPERATION

012540

.EVEN

USEL:	0	;CONTAINS # OF UNIT SELECTED
SOF5W:	0	;CONTENTS ARE FOR SOFTWARE DECISIONS
MODE:	0	; 'FORMAT & VERIFY' OR 'CHECK' MODE INDICATOR
ECYL:	0	;ENDING CILINDER
SCYL:	0	;STARTING CILINDER
ETRK:	0	;ENDING TRACK
STRK:	0	;STARTING TRACK
TTRKS:	0	;TOTAL # OF TRACKS TO BE FORMATTED
TTRKSC:	0	;TOTAL # OF TRACKS COUNTER
TRKCNT:	0	;COUNTS TRKS FROM 0-18 PER CYL
PSEL:	0	;CONTAINS PATTERN SELECTED
PATA:	0	;PATTERN TO BE WRITTEN ON DISK
PATB:	0	;DITTO
TEMP:	0	;UTILITY STORAGE
RETRY:	0	;MAINTAINS # OF RETRIES MADE
SAVSEC:	0	;CONTAINS LAST BAD SECTOR ON FORMAT
SAVWC:	0	;CONTAINS WC FOR REMAINING SECTORS ON ERROR
WC:	0	;20 OR 22 SECTOR TRACK SIZE (IN WORDS)
MWC:	0	;2'S COMPLEMENT OF 'WC'
HEDERR:	0	;POSITIONING ERROR DURING FORMAT INDICATOR
SEC20:	0	;20 OR 22 SECTOR MODE INDICATOR
		;0 = 20 SECTOR MODE
		;1'S = 22 SECTOR MODE
MAXSEC:	0	;MAXIMUM SECTOR ADDRESS (FOR EITHER 20 OR 22 SECTOR
		;FORMAT)
LOCTRK:	0	;TEMP STORAGE FOR TRACK CONVERSION FOR TYPEOUT
LOCCYL:	0	;TEMP STORAGE FOR CYL CONVERSION FOR TYPEOUT
NULCNT:	1	;NULL (FILLER) CHARACTER COUNT FOR TYPEOUTS
RBUF:	0	;THIS LOCATION CONTAINS CYL ADRS ON HEADER CK
	0	;THIS LOCATION CONTAINS TRK-SEC ADRS ON HEADER CK
CPTR3:	0	;LEAVE SPACE FOR LEADING ZEROS
CPTR2:	0	
CPTR1:	0	
CPTR:	0	;TTY INPUT BUFFER
	.=.+4	;MOVE POINTER TO END OF BUFFER

6576	012642	000000
6577	012644	000000
6578	012646	000000
6579	012650	000000
6580	012652	000000
6581	012654	000000
6582	012656	000000
6583	012660	000000
6584	012662	000000
6585	012664	000000
6586	012666	000000
6587	012670	000000
6588	012672	000000
6589	012674	000000
6590	012676	000000
6591	012700	000000
6592	012702	000000
6593	012704	000000
6594	012706	000000
6595	012710	000000

SAVREG:	0	:	RHCS1
	0	:	RHNC
	0	:	RHBA
	0	:	RHDA
	0	:	RHCS2
	0	:	RHDS1
	0	:	RHER1
	0	:	RHAS
	0	:	RHLA
	0	:	RHOB
	0	:	RHMR
	0	:	RHOT
	0	:	RHSN
	0	:	RHOF
	0	:	RHCA
	0	:	RHCC
	0	:	RHER2
	0	:	RHER3
	0	:	RHEC1
	0	:	RHEC2

6599	012712	000
6600	012713	000
6601	012714	000
6602	012715	000
6603	012716	000000
6604	012720	000000
6605	012722	000
6606	012723	000
6607	012724	000000
6608	012726	000000
6609	012730	000000
6610		
6611		
6612		
6613		
6614		
6615		
6616		
6617		
6618		
6619		
6620		

FMTDPB:	.BYTE	0	:	DRIVE NUMBER
	.BYTE	00	:	OFFSET VALUE OR FMT22,ECI, AND HCI
	.BYTE	00	:	COMMAND
	.BYTE	00	:	PSEL AND A17 AND A16
	.WORD	00	:	WORD COUNT (NEG)
	.WORD	00	:	BUFFER ADDRESS
	.BYTE	00	:	SECTOR ADDRESS
	.BYTE	00	:	TRACK ADDRESS
	.WORD	00	:	CYLINDER ADDRESS
	.WORD	00	:	ERROR TABLE POINTER
	.WORD	0	:	STATUS-ERROR INDICATOR
			:	BIT 15 = 1: ERROR OCCURED
			:	BIT 07 = 1: DONE
			:	BIT 14-10 AND BIT 06-03
			:	INDICATE TYPE OF ERROR

(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)

.SBTTL RH11/RP04 DRIVER (REV. 0.9)  
; \*COPYRIGHT (C) 1974  
; \*DIGITAL EQUIPMENT CORP.  
; \*MAYNARD, MA 01754  
; \*AUTHOR: JIM LACEY  
  
; \*STORAGE FOR RHDS1, RHER1, RHER2, AND RHER3 ON AN ERROR "2" OR "5"



```

(1)
(1) 013002 000000 TRNSWT: .WORD 0
(1)
(1) ;*SEARCH WAIT KEYS (SRCHWT=1 WORD)
(1) ;*THIS IS A ONE WORD QUEUE THAT WILL CONTAIN A KEY FOR EACH OF
(1) ;*THE DRIVES THAT ARE PERFORMING A SEARCH COMMAND FOR THE I/O
(1) ;*REQUEST THAT IS AT THE TOP OF THEIR REQUEST QUEUE.
(1) ;*EACH DRIVE IS ASSIGNED ONE BIT, STARTING AT BIT00 FOR DRIVE 0.
(1)
(1) 013004 000000 SRCHWT: .WORD 0
(1)
(1) ;*RPO4 DRIVER ACTIVE FLAG (ACTDRV=1 BYTE)
(1) ;*ACTDRV=0 IMPLIES DRIVER IS INACTIVE
(1) ;*ACTDRV>0 IMPLIES DRIVER IS ACTIVE
(1)
(1) 013006 000 ACTDRV: .BYTE 0
(1)
(1) ;*SOFTWARE TIMER ROUTINE ACTIVE FLAG (ACTSTR=1 BYTE)
(1) ;*ACTSTR=0 IMPLIES SOFTWARE TIMER ROUTINE IS INACTIVE
(1) ;*ACTSTR>0 IMPLIES SOFTWARE TIMER ROUTINE IS ACTIVE
(1)
(1) 013007 000 ACTSTR: .BYTE 0
(1)
(1) ;*UNLOAD FLAG (ULDFLG=8 BYTES)
(1) ;*ULDFLG=0 IMPLIES NO UNLOAD COMMAND
(1) ;*ULDFLG>0 IMPLIES UNLOAD COMMAND IN PROGRESS
(1) ;*ULDFLG<0 IMPLIES UNLOAD COMMAND IN WAIT QUEUE
(1)
(1) 013010 000 ULDFLG: .BYTE 0 ;DRIVE 0
(1) 013011 000 .BYTE 0 ;DRIVE 1
(1) 013012 000 .BYTE 0 ;DRIVE 2
(1) 013013 000 .BYTE 0 ;DRIVE 3
(1) 013014 000 .BYTE 0 ;DRIVE 4
(1) 013015 000 .BYTE 0 ;DRIVE 5
(1) 013016 000 .BYTE 0 ;DRIVE 6
(1) 013017 000 .BYTE 0 ;DRIVE 7
(1)
(1) ;*LOOK AHEAD COUNT (LACNT=8 BYTES)
(1) ;*LACNT WILL INDICATE THE NUMBER OF LOOK AHEADS PERFORMED
(1)
(1) 013020 000 LACNT: .BYTE 0 ;DRIVE 0
(1) 013021 000 .BYTE 0 ;DRIVE 1
(1) 013022 000 .BYTE 0 ;DRIVE 2
(1) 013023 000 .BYTE 0 ;DRIVE 3
(1) 013024 000 .BYTE 0 ;DRIVE 4
(1) 013025 000 .BYTE 0 ;DRIVE 5
(1) 013026 000 .BYTE 0 ;DRIVE 6
(1) 013027 000 .BYTE 0 ;DRIVE 7
(1)
(1) ;*SAVE REGISTERS FLAG (SAVEFG =1 WORD)
(1) ;*SAVEFG <0 IMPLIES SAVE THE RH11/RPO4 REGISTERS WHEN THE
(1) ;*OPERATION IS COMPLETED AS PER (DPB+14).
(1) ;*SAVEFG=0 IMPLIES SAVE THE RH11/RPO4 REGISTERS, AS PER
(1) ;*(DPB+14), AFTER AN ERROR.

```

```

(1)
(1) 013030 000000 SAVEFG: .WORD 0
(1)
(1) ;*SEEK FLAG (SEEKFG=1 WORD)
(1) ;*SEEKFG=0 IMPLIES WHEN THE DISK ADDRESS ISN'T IN THE WINDOW
(1) ;*FOR A DATA TRANSFER START A SEARCH COMMAND
(1) ;*SEEKFG<0 IMPLIES DATA TRANSFER WILL DO IMPLIED SEEKS,
(1) ;*DISREGARD THE WINDOW
(1)
(1) 013032 000000 SEEKFG: .WORD 0
(1)
(1) ;*TIMEOUT TABLE (TIMER=8 WORDS)
(1) ;*THIS TABLE CONTAINS THE TIME ALLOWED FOR AN OPERATION
(1)
(1) 013034 177777 TIMER: .WORD -1 ;DRIVE 0
(1) 013036 177777 .WORD -1 ;DRIVE 1
(1) 013040 177777 .WORD -1 ;DRIVE 2
(1) 013042 177777 .WORD -1 ;DRIVE 3
(1) 013044 177777 .WORD -1 ;DRIVE 4
(1) 013046 177777 .WORD -1 ;DRIVE 5
(1) 013050 177777 .WORD -1 ;DRIVE 6
(1) 013052 177777 .WORD -1 ;DRIVE 7
(1)
(1) ;*DATA TRANSFER UNDERWAY INDICATOR (DTUW=1 WORD)
(1) ;*DTUW<0 IMPLIES NO DATA TRANSFER UNDERWAY
(1) ;*DTUW=+N (WHERE N=0 TO 7) IMPLIES DATA TRANSFER UNDERWAY ON DRIVE N
(1)
(1) 013054 177777 DTUW: .WORD -1
(1)
(1) ;*ATTENTION BITS TABLE (ATABIT=8 BYTES)
(1) ;*THIS TABLE CONTAINS THE CORRESPONDING BIT TO EACH DRIVES
(1) ;*ATTENTION BIT
(1)
(1) 013056 001 ATABIT: .BYTE 1 ;DRIVE 0
(1) 013057 002 .BYTE 2 ;DRIVE 1
(1) 013060 004 .BYTE 4 ;DRIVE 2
(1) 013061 010 .BYTE 10 ;DRIVE 3
(1) 013062 020 .BYTE 20 ;DRIVE 4
(1) 013063 040 .BYTE 40 ;DRIVE 5
(1) 013064 100 .BYTE 100 ;DRIVE 6
(1) 013065 200 .BYTE 200 ;DRIVE 7
(1)
(1) ;*RPO4 TO RH11 "MASS CONTROL BUS PARITY ERRORS" (MCPE) ALLOWED BEFORE
(1) ;*CALLING IT FATAL (MCPMX=1 WORD)
(1)
(1) 013066 000003 MCPMX: .WORD 3
(1)
(1) ;*STORAGE FOR RPADR (THE FIRST ADDRESS (776700) OF THE RH11/RPO4),
(1) ;*RPVEC (THE VECTOR ADDRESS (254)), AND RPVEC+2 (THE BR LEVEL (5)).
(1) 013070 176700 RPADR: .WORD 176700
(1) 013072 000254 000240 RPVEC: .WORD 254,5*32.
(1)
(1) ;*MAXIMUM NUMBER OF LOOK AHEADS ALLOWED IS 4 (MXLACT=1 WORD)
(1) 013076 000004 MXLACT: .WORD 4

```



```

(1) ;*MAXIMUM DELTA DELAY IS 8 SECTORS (MXDLTA=1 WORD)
(1) 013100 001000 MXDLTA: .WORD 8.*64.
(1) ;*MINIMUM DELTA DELAY IS 2 SECTORS (MNDLTA=1 WORD)
(1) 013102 000200 MNDLTA: .WORD 2.*64.
(1) ;*MAXIMUM SEARCH FOR I/O WINDOW IS 5 SECTORS (MXWINDOW=1 WORD)
(1) 013104 000005 MXWINDOW: .WORD 5
(1)
(1) ;*DEFINITIONS OF THE RH11/RPO4 ADDRESS INDEXES
(1)
(1) 000000 RHCS1=0 ;CONTROL AND STATUS REGISTER #1 (DRIVE REG. 00)
(1) 000002 RHWG=2 ;WORD COUNT REGISTER (NOT A DRIVE REG)
(1) 000004 RHBA=4 ;UNIBUS ADDRESS REGISTER (NOT A DRIVE REG)
(1) 000006 RHDA=6 ;DESIRED SECTOR/TRACK ADDRESS REGISTER (DRIVE REG. 05)
(1) 000010 RHCS2=10 ;CONTROL AND STATUS REGISTER #2 (NOT A DRIVE REG)
(1) 000012 RHDS1=12 ;DRIVE STATUS REGISTER (DRIVE REG 01)
(1) 000014 RHER1=14 ;ERROR REGISTER #1 (DRIVE REG. 02)
(1) 000016 RHAS=16 ;ATTENTION SUMMARY PSEUDO REGISTER (DRIVE REG. 04)
(1) 000020 RHLA=20 ;LOOK AHEAD REGISTER (DRIVE REG. 07)
(1) 000022 RHDB=22 ;DATA BUFFER REGISTER (NOT A DRIVE REG.)
(1) 000024 RHMR=24 ;MAINTAINABILITY REGISTER (DRIVE REG. 03)
(1) 000026 RHDT=26 ;DRIVE TYPE REGISTER (DRIVE REG. 06)
(1) 000030 RHSN=30 ;SERIAL NUMBER REGISTER (DRIVE REG. 10)
(1) 000032 RHOF=32 ;OFFSET REGISTER (DRIVE REG. 11)
(1) 000034 RHCA=34 ;DESIRED CYLINDER ADDRESS REGISTER (DRIVE REG. 12)
(1) 000036 RHCC=36 ;CURRENT CYLINDER ADDRESS REGISTER (DRIVE REG. 13)
(1) 000040 RHER2=40 ;ERROR REGISTER #2 (DRIVE REG. 14)
(1) 000042 RHER3=42 ;ERROR REGISTER #3 (DRIVE REG. 15)
(1) 000044 RHEC1=44 ;ECC POSITION REGISTER (DRIVE REG. 16)
(1) 000046 RHEC2=46 ;ECC PATTERN REGISTER (DRIVE REG. 17)
(1)
(1) ;*RH11/RPO4 DRIVER INIT. CODE
(1) ;*THIS ROUTINE WILL DETERMINE WHICH RPO4 DRIVES ARE
(1) ;*AVAILABLE FOR TESTING AND SET THE DRVSTA INDICATOR
(1) ;*TO THE PROPER STATE FOR EACH DRIVE.
(1) ;*NOTE: THIS ROUTINE CALLS DRVINT
(1)
(1) ;*CALL
(1)
(1) ;*
(1) ;* JSR PC,RPINIT
(1) ;* RETURN
(1)
(1) RPINIT: MOV 2*PS, -(SP) ;SAVE PROCESSOR STATUS
(1) 013106 013746 177776 MOV RPVEC+2, 2*PS ;SET PROCESSOR STATUS TO RH11/RPO4 LEVEL
(1) 013112 013737 013074 177776 JSR RO,SAVR15 ;GO SAVE R1-R5
(1) 013120 004037 020236 JSR PC,CLIQUE ;CLEAR ALL REQUEST QUEUES
(1) 013124 004737 017774 MOV #RPERRS,R1 ;FIRST ADDRESS TO BE CLEARED
(1) 013130 012701 012732 MOV #SEEKFG,R2 ;LAST ADDRESS TO BE CLEARED
(1) 013134 012702 013032
(1) 1$: CLR (R1)+ ;CLEAR
(1) 013140 005021 CMP R1,R2 ;ARE WE DONE?
(1) 013142 020102 BLOS 1$ ;BRANCH IF NO
(1) 013144 101775 MOV #DTUW,R2 ;LAST ADDRESS
(1) 013146 012702 013054 2$: MOV #-1,(R1)+ ;INITIALIZE
(1) 013152 012721 177777 CMP R1,R2 ;DONE?
(1) 013156 020102 BLOS 2$ ;LOOP IF NO
(1) 013160 101774
    
```

```

(1) 013162 005001          CLR      R1          ;DRIVE NUMBER WILL BE IN R1
(1) 013164 013704 013070  MOV      RPADR,R4    ;FIRST ADDRESS OF RH11/RPO4
(1) 013170 012764 000040 000010  MOV      #BIT05,RHCS2(R4) ;MASSBUS INIT.
(1) 013176 004037 013250      JSR      RD,DRVINT    ;GO INIT. A DRIVE
(1) 013202 000417          BR       7$
(1) 013204 005201          4$:      INC      R1          ;MOVE TO THE NEXT DRIVE
(1) 013206 042701 177770      BIC      #1C7,R1     ;DON'T LET THE DRIVE NUMBER GET TO BIG
(1) 013212 001371          BNE      3$          ;BRANCH IF MORE DRIVES TO INIT.
(1) 013214 013703 013072      MOV      RPVEC,R3    ;SETUP THE RH11/RPO4 VECTOR
(1) 013220 012723 015260      MOV      #ISP,(R3)+
(1) 013224 013713 013074      MOV      RPVEC+2,(R3)
(1) 013230 004037 020256      JSR      RD,GETR15   ;RESTORE R1-R5
(1) 013234 012637 177776      MOV      (SP)+,2*PS  ;RESTORE THE PROCESSOR STATUS
(1) 013240 000207          RTS      PC          ;BYE-BYE
(1) 013242 105061 012752      7$:      CLRB     DRVSTA(R1) ;SET THE DRIVE STATUS TO OFFLINE
(1) 013246 000756          BR       4$          ;GO DO THE NEXT DRIVE
(1)
(1)
(1) ;*DRIVE INIT. ROUTINE
(1) ;*THIS ROUTINE DETERMINES IF A DRIVE EXIST AND IF IT IS
(1) ;*AN RPO4. IF IT IS, A "READ-IN PRESET" IS ISSUED AND FMT22
(1) ;*IS SET TO A "1". THEN MOL, DPR, DRY, AND VV ARE CHECKED TO
(1) ;*INSURE THEY ARE ALL ON A "1". AND DEPENDING ON THEIR STATE,
(1) ;*DRVSTA IS SET TO THE PROPER CONDITION.
(1)
(1) ;*CALL
(1) ;*
(1) ;*      MOV      #DRVNUM,R1    ;DRIVE NUMBER TO R1
(1) ;*      MOV      RPADR,R4    ;UNIBUS ADDRESS OF RH11/RPO4 (RHCS1)
(1) ;*      JSR      RD,DRVINT    ;CALLED BY A JSR
(1) ;*      RETURN1   ;ERROR OCCURRED (PARITY)
(1) ;*      RETURN2   ;NORMAL RETURN
(1)
(1)
(1) 013250 004037 020236      DRVINT: JSR      RD,SAVR15  ;SAVE R1-R5
(1) 013254 010164 000010      MOV      R1,RHCS2(R4) ;SELECT A DRIVE
(1) 013260 010103          MOV      R1,R3       ;COPY THE DRIVE NUMBER INTO
(1) 013262 006303          ASL      R3          ;R3 AND POSITION IT FOR TABLE INDEXING
(1) 013264 112761 177777 012752  MOVVB   #-1,DRVSTA(R1) ;START DRIVE STATUS AS NONEXISTENT
(1) 013272 005063 012762      CLR      DRV'TYP(R3) ;CLEAR THE DRIVE TYPE INDICATOR
(1) 013276 112777 000111 177564  MOVVB   #111,2*RPADR ;DO A "DRIVE CLEAR" COMMAND
(1) 013304 032764 010000 000010  BIT      #BIT12,RHCS2(R4) ;NONEXISTENT DRIVE?
(1) 013312 001403          BEQ      1$          ;NO---BRANCH
(1) 013314 004737 017430      JSR      PC,SET.IE   ;GO SET "IE" WITHOUT A "TRE"
(1) 013320 000462          BR       6$          ;LEAVE THIS ROUTINE
(1) 013322 004037 017036      1$:      JSR      RD,RD.RP    ;READ THE DRIVE TYPE REG.
(1) 013326 000026          RHD1    7$
(1) 013330 013470          7$:      ;ERROR RETURN ADDRESS
(1) 013332 012605          MOV      (SP)+,R5    ;PUT DRIVE TYPE IN R5
(1) 013334 010563 012762      MOV      R5,DRV'TYP(R3) ;SAVE THE DRIVE TYPE
(1) 013340 022705 020020      CMP      #20020,R5   ;IS IT A SINGLE PORT RPO4?
(1) 013344 001403          BEQ      2$          ;BRANCH IF YES
(1) 013346 022705 024020      CMP      #24020,R5   ;IS IT A DUAL PORT RPO4?
(1) 013352 001043          BNE      5$          ;BRANCH IF NO
(1) 013354 012746 000121      2$:      MOV      #121,-(SP)  ;DO A "READ-IN PRESET"
(1) 013360 004037 017174      JSR      RD,WRT.RP
(1) 013364 000000          RHCS1
(1) 013366 013470          7$

```

```

(1) 013370 012746 010000      MOV      #BIT12,-(SP)      ;SET FMT22=1
(1) 013374 004037 017174      JSR      RO,WAT.RP
(1) 013400 000032              RHOF
(1) 013402 013470              7$
(1) 013404 004037 017036      JSR      RO,RO.RP        ;READ RHDS1
(1) 013410 000012              RHDS1
(1) 013412 013470              7$
(1) 013414 012605              MOV      (SP)+,R5        ;AND SAVE IT IN R5
(1) 013416 100011              BPL      4$              ;BRANCH IF ATA=0
(1) 013420 116164 013056 000016  MOVB     ATABIT(R1),RHAS(R4) ;CLEAR ATTENTION BIT
(1) 013426 004037 017036      JSR      RO,RO.RP        ;FIND OUT WHY ATA=1
(1) 013432 000014              RHER1
(1) 013434 013470              7$
(1) 013436 006126              ROL      (SP)+          ;IS IT UNSAFE?
(1) 013440 100412              BMI      3$              ;BRANCH IF YES
(1) 013442 005105              4$:      COM      R5        ;CHECK MOL, DPR, DRY, AND VV
(1) 013444 042705 167077      BIC      #1<BIT12!BIT08!BIT07!BIT06>,R5
(1) 013450 001004              BNE      5$              ;BRANCH IF MOL, DPR, DRY, OR VV IS CLEAR
(1) 013452 112761 000001 012752  MOVB     #1,DRVSTA(R1)    ;SET DRIVE STATUS TO ONLINE
(1) 013460 000402              BR
(1) 013462 105061 012752      5$:      CLRB     DRVSTA(R1)  ;DRIVE STATUS = OFFLINE
(1) 013466 005720              6$:      TST      (RO)+      ;STEP OVER THE ERROR RETURN
(1) 013470 004037 020256      7$:      JSR      RO,GETR15    ;RESTORE R1-R5
(1) 013474 000200              RTS      RO              ;RETURN

(1)
(1) ;*REQUEST PRE-PROCESSOR-HANDLES SUBSYSTEM REQUEST
(1)
(1) ;*CALL
(1)
(1) ;*
(1) ;* JSR      RO,@#RPO4    ;CALL THE RPO4 DRIVER
(1) ;* PNTADR    ;ADDRESS OF POINTER OF DRIVES PARAMETER BLOCK
(1) ;* RETURN1   ;RETURN HERE IF QUEUE IS FULL
(1) ;* RETURN2   ;RETURN HERE IF REQUEST IS IN QUEUE OR THERE
(1) ;*          ;IS AN ERROR CONDITION
(1)
(1) 013476 013746 177776      RPO4:   MOV      @#PS -(SP)    ;SAVE THE CALLING STATUS
(1) 013502 013737 013074 177776  MOV      RPVEC+2,@#PS    ;DON'T ALLOW ANY RPO4 INTERRUPTS
(1) 013510 112737 000001 013006  MOVB     #1,ACTDRV      ;SET "ACTIVE DRIVER" FLAG
(1) 013516 004037 020236      JSR      RO,SAVR15      ;SAVE R1-R5
(1) 013522 012002              MOV      (RO)+,R2      ;PICKUP THE DRIVE PARAMETER BLOCK POINTER
(1) 013524 005062 000016      CLR      16(R2)        ;CLEAR THE STATUS/ERROR INDICATOR
(1) 013530 111201              MOVB     (R2),R1        ;PICKUP THE DRIVE NUMBER
(1) 013532 105761 012752      TSTB    DRVSTA(R1)     ;CHECK DRIVES STATUS
(1) 013536 003017              BGT      1$            ;BRANCH IF ONLINE
(1) 013540 105761 013010      TSTB    ULDFLG(R1)     ;UNLOAD COMMAND IN QUEUE?
(1) 013544 001034              BNE      3$            ;BRANCH IF YES
(1) 013546 013704 013070      MOV      RPADR,R4      ;UNIBUS ADDRESS OF RHCS1
(1) 013552 004037 013250      JSR      RO,DRVINT     ;GO INIT. THE DRIVE
(1) 013556 000442              BR      6$            ;ERROR RETURN
(1) 013560 105761 012752      TSTB    DRVSTA(R1)     ;IS DRIVE STATUS ONLINE?
(1) 013564 003004              BGT      1$            ;BRANCH IF YES
(1) 013566 052762 140000 000016  BIS      #BIT15!BIT14,16(R2) ;ERROR--DRIVE IS OFFLINE OR NONEXISTENT
(1) 013574 000423              BR      4$            ;GO TO EXIT
(1) 013576 004037 020076      1$:     JSR      RO,DRVQUE    ;PUT THIS REQUEST IN QUEUE

```

```

(1) 013602 000421 BR 5$ ;QUEUE IS FULL
(1) 013604 122762 000103 000002 CMPB #103,2(P2) ;IS THIS REQ. FOR AN UNLOAD?
(1) 013612 001003 BNE 2$ ;BR IF NO
(1) 013614 112761 177777 013010 MOVB #-1,ULDFLG(R1) ;SET THE "UNLOAD IN QUEUE" FLAG
(1) 013622 105761 012742 2$: TSTB DRVACT(R1) ;IS THIS DRIVE ACTIVE?
(1) 013626 001006 BNE 4$ ;BR IS YES
(1) 013630 004737 013672 JSR PC,OPT ;CALL THE OPTIMIZER
(1) 013634 000403 BR 4$
(1) 013636 052762 120000 000016 3$: BIS #BIT15:BIT13,16(P2) ;SET THE "UNLOAD IN QUEUE" ERROR FLAG
(1) 013644 005720 4$: TST (R0)+
(1) 013646 004037 020256 5$: JSR R0,GETR15 ;RESTORE R1-R5
(1) 013652 105037 013006 CLR B ACTDRV ;CLEAR "ACTIVE DRIVER" FLAG
(1) 013656 012637 177776 MOV (SP)+,2#PS ;RETURN "PS" TO USER LEVEL
(1) 013662 000200 RTS R0 ;RETURN TO CALLER
(1) 013664 004737 014674 6$: JSR PC,C17 ;GO HANDLE THE PARITY ERROR
(1) 013670 000765 BR 4$

(1) ;*OPTIMIZER-CALLED FOR A PARTICULAR DRIVE
(1) ;*CALL
(1) ;*
(1) ;* MOV #DRVNUM,R1 ;DRIVE NUMBER TO R1
(1) ;* JSR PC,OPT ;SETUP A COMMAND
(1) ;*
(1) OPT: JSR R0,SAVR15 ;SAVE R1-R5
(1) 013672 004037 020236 MOV 2#PS,-(SP) ;SAVE PROC. STATUS
(1) 013676 013746 177776 BIC B ATABIT(R1),SRCHWT ;CLEAR "SEARCH WAIT" KEY
(1) 013702 146137 013056 013004 JSR PC,GETREQ ;GET "DPB" POINTER OF REQUEST
(1) 013710 004737 020152 TST R2 ;IS THERE A REQUEST IN QUEUE?
(1) 013714 005702 BEQ 6$ ;NO--BRANCH TO EXIT
(1) 013718 001444 TSTB DRVSTA(R1) ;IS DRIVE ONLINE?
(1) 013722 105761 012752 BGT 1$ ;YES--BRANCH
(1) 013724 003006 JSR PC,POPQUE ;NO--REMOVE REQUEST FROM QUEUE
(1) 013726 004737 020174 BIS #BIT15:BIT14,16(R2) ;SET ERROR BIT OF STATUS/ERROR INDICATOR
(1) 013732 052762 140000 000016 BR 5$ ;BRANCH TO EXIT
(1) 013740 000426 CMPB #150,2(R2) ;IS THE REQUEST FOR I/O?
(1) 013742 122762 000150 000002 1$: BLT 2$ ;YES--BRANCH
(1) 013750 002403 JSR PC,C14 ;CALL THE COMMAND INITIATOR
(1) 013752 004737 014252 BR 5$ ;BRANCH TO EXIT
(1) 013756 000417 TST DTUW ;DATA TRANSFER UNDERWAY?
(1) 013760 005737 013054 2$: BGE 4$ ;YES--GO START A SEARCH
(1) 013764 002012 TST SEEKFG ;DO IMPLIED SEEKS?
(1) 013766 005737 013032 BMI 3$ ;YES---BRANCH
(1) 013772 100404 JSR R0,LA ;NO--DO LOOK AHEAD
(1) 013774 004037 015122 BR 5$ ;RETURN HERE ON A PARITY ERROR
(1) 014000 000406 BR 4$ ;GO START A SEARCH
(1) 014002 000403 JSR PC,C11 ;START A DATA TRANSFER
(1) 014004 004737 014036 3$: BR 5$
(1) 014010 000402 BR 5$
(1) 014012 004737 014144 4$: JSR PC,C13 ;START A SEARCH
(1) 014016 012637 177776 5$: MOV (SP)+,2#PS ;RESTORE PROC. STATUS
(1) 014022 004037 020256 JSR R0,GETR15 ;RESTORE R1-R5
(1) 014026 000207 RTS PC
(1) 014030 004737 017430 6$: JSR PC,SET.IE ;SET "IE" WITHOUT A "TRE"
(1) 014034 000770 BR 5$ ;EXIT THE ROUTINE

```

```

(1)          ;*COMMAND INITIATOR
(1)          ;*CALL
(1)          ;*
(1)          ;*   MOV   #DRVNUM,R1   ;DRIVE NUMBER
(1)          ;*   MOV   #DPB,R2    ;ADDRESS OF DPB
(1)          ;*   JSR   PC,C1?    ;C1?= C11,C13, OR C14
(1)          ;*          ;WHERE:
(1)          ;*          ;C11=DATA TRANSFER
(1)          ;*          ;C12=SEARCH REQUESTED BY DATA XFER
(1)          ;*          ;C14=NOT DATA TRANSFER
(1)          ;*
(1) 014036 004737 020174 C11: JSR   PC,POPQUE ;REMOVE REQUEST FROM "DRIVES WAIT" QUEUE
(1) 014042 010237 013002      MOV   R2,TRNSWT ;PUT REQ. IN TRANSFER WAIT QUEUE
(1) 014046 010203 C12:  MOV   R2,R3    ;DPB ADDRESS TO R3
(1) 014050 013704 013070      MOV   RPADR,R4   ;RHCSI ADDRESS
(1) 014054 010164 000010      MOV   R1,RHCS2(R4) ;SELECT DRIVE
(1) 014060 062703 000004      ADD   #4,R3      ;DESIRED WORD COUNT
(1) 014064 062704 000002      ADD   #2,R4      ;RHWI ADDRESS
(1) 014070 012324      MOV   (R3)+,(R4)+ ;LOAD WORD COUNT
(1) 014072 012324      MOV   (R3)+,(R4)+ ;LOAD BUFFER ADDRESS
(1) 014074 012346      MOV   (R3)+,-(SP) ;LOAD SECTOR AND TRACK
(1) 014076 004037 017174      JSR   RD,WRT.RP  ;CALL THE LOAD(WRITE) ROUTINE
(1) 014102 000006      RHDA          ;INDEX OF REGISTER TO LOAD
(1) 014104 014674      CI7          ;ERROR RETURN ADDRESS
(1) 014106 012346      MOV   (R3)+,-(SP) ;LOAD CYLINDER ADDRESS
(1) 014110 004037 017174      JSR   RD,WRT.RP
(1) 014114 000034      RHCA          ;
(1) 014116 014674      CI7          ;
(1) 014120 016246 000002      MOV   2(R2)-,(SP) ;LOAD "COMMAND+GO", "A17&A16", AND "PSEL"
(1) 014124 004037 017174      JSR   RD,WRT.RP
(1) 014130 000000      RHCS1        ;
(1) 014132 014674      CI7          ;
(1) 014134 010137 013054      MOV   R1,DTUW   ;SET "DATA TRANSFER UNDERWAY"
(1) 014140 000137 014630      JMP   CI5
(1) 014144 013704 013070 C13: MOV   RPADR,R4   ;RHCSI ADDRESS
(1) 014150 010164 000010      MOV   R1,RHCS2(R4) ;SELECT DRIVE
(1) 014154 016246 000012      MOV   12(R2)-,(SP) ;DESIRED CYLINDER ADDRESS
(1) 014160 004037 017174      JSR   RD,WRT.RP
(1) 014164 000034      RHCA          ;
(1) 014166 014674      CI7          ;
(1) 014170 116203 000010      MOVB  10(R2),R3  ;PICKUP SECTOR ADDRESS
(1) 014174 163703 013104      SUB   MXWINDW,R3 ;BACKUP BY MAX. SEARCH FOR I/O WINDOW
(1) 014200 002002      BGE   1$
(1) 014202 062703 000026      ADD   #22,R3
(1) 014206 010346      1$: MOV   R3,-(SP)  ;COMBINE THE ADJUSTED SECTOR WITH
(1) 014210 116266 000011 000001      MOVB  11(R2),1(SP) ;THE DESIRED TRACK
(1) 014216 004037 017174      JSR   RD,WRT.RP  ;LOAD DESIRED TRACK & SECTOR
(1) 014222 000006      RHDA          ;
(1) 014224 014674      CI7          ;
(1) 014226 012746 000131      MOV   #131,-(SP) ;START A SEARCH
(1) 014232 004037 017174      JSR   RD,WRT.RP
(1) 014236 000000      RHCS1        ;
(1) 014240 014674      CI7          ;
(1) 014242 156137 013056 013004      BISB  ATAEIT(R1),SPCHW ;SET "SEARCH WAIT" KEY

```

# B05

MAINDEC-11-DZRPL-B "RPO4 FORMATTER PROGRAM"  
DZRPLB.P11 RH11/RPO4 DRIVER (REV. 0.9)

MACY11 27(663) 23-JAN-76 10:26 PAGE 43-42

SEQ 0052

(1)	014250	000567			BR	C15	
(1)	014252	013704	013070	C14:	MOV	RPADR,R4	;RHCS1 ADDRESS
(1)	014256	010164	000010		MOV	R1,RHC52(R4)	;SELECT DRIVE
(1)	014262	116203	000002		MOV	2(R2),R3	;PICKUP THE REQUESTED COMMAND
(1)	014266	122703	000131		CMPB	#131,R3	;IS IT A SEARCH COMMAND?
(1)	014272	001007			BNE	1S	;BRANCH IF NO
(1)	014274	016246	000010		MOV	10(R2),-(SP)	;LOAD DESIRED TRACK & SECTOR
(1)	014300	004037	017174		JSR	RO,WRT.RP	
(1)	014304	000006			RHCA		
(1)	014306	014674			CI7		
(1)	014310	000403			BR	2S	;GO LOAD CYLINDER
(1)	014312	122703	000105	1S:	CMPB	#105,R3	;IS IT A SEEK COMMAND
(1)	014316	001007			BNE	3S	;BRANCH IF NO
(1)	014320	016246	000012	2S:	MOV	12(R2),-(SP)	;LOAD DESIRED CYLINDER
(1)	014324	004037	017174		JSR	RO,WRT.RP	
(1)	014330	000034			RHCA		
(1)	014332	014674			CI7		
(1)	014334	000546			BR	C16	
(1)	014336	122703	000115	3S:	CMPB	#115,R3	;IS IT AN "OFFSET" COMMAND?
(1)	014342	001013			BNE	4S	;BR IF NO
(1)	014344	004037	017036		JSR	RO,RO.RP	;MERGE THE OFFSET VALUE INTO RHOF
(1)	014350	000032			RHOF		;BUT DON'T CHANGE THE UPPER
(1)	014352	014674			CI7		
(1)	014354	116216	000001		MOV	1(R2),(SP)	;BYTE WHEN LOADING THE
(1)	014360	004037	017174		JSR	RO,WRT.RP	;REGISTER (RHOF)
(1)	014364	000032			RHOF		
(1)	014366	014674			CI7		
(1)	014370	000530			BR	C16	;GO START THE COMMAND
(1)	014372	122703	000107	4S:	CMPB	#107,R3	;IS IT A "RECALIBRATE" COMMAND?
(1)	014376	001525			BEQ	C16	;BRANCH IF YES
(1)	014400	122703	000117		CMPB	#117,R3	;IS IT A RETURN TO CENTER?
(1)	014404	001522			BEQ	C16	;BRANCH IF YES
(1)	014406	122703	000103		CMPB	#103,R3	;IS IT AN "UNLOAD" COMMAND?
(1)	014412	001016			BNE	5S	;BRANCH IF NO
(1)	014414	112761	000001	012742	MOV	#1,DRVACT(R1)	;SET THE DRIVE ACTIVE INDICATOR
(1)	014422	105061	012752		CLAB	DRVSTA(R1)	;PUT DRIVE STATUS TO OFFLINE
(1)	014426	112761	000001	013010	MOV	#1,ULDFLG(R1)	;SET "UNLOAD IN PROGRESS" FLAG
(1)	014434	010346			MOV	R3,-(SP)	;START THE "UNLOAD" COMMAND
(1)	014436	004037	017174		JSR	RO,WRT.RP	
(1)	014442	000000			RHCS1		
(1)	014444	014674			CI7		
(1)	014446	000207			RTS	PC	;RETURN TO USER
(1)	014450	122703	000143	5S:	CMPB	#143,R3	;IS IT A "SET FORMAT" COMMAND?
(1)	014454	001014			BNE	6S	;BRANCH IF NO
(1)	014456	004037	017036		JSR	RO,RO.RP	;READ THE OFFSET REGISTER
(1)	014462	000032			RHOF		
(1)	014464	014674			CI7		
(1)	014466	116266	000001	000001	MOV	1(R2),1(SP)	;COMBINE "FMT22", "ECI", AND "HCI"
(1)	014474	004037	017174		JSR	RO,WRT.RP	;LOAD "FMT22", "ECI", AND/OR "HCI".
(1)	014500	000032			RHOF		
(1)	014502	014674			CI7		
(1)	014504	000436			BR	12S	
(1)	014506	122703	000141	6S:	CMPB	#141,R3	;IS IT A "GET REGISTER" COMMAND?
(1)	014512	001023			BNE	10S	;BRANCH IF NO

## C05

MAINDEC-11-DZRPL-B "RPO4 FORMATTER PROGRAM"  
DZRPLB.P11 RH11/RPO4 DRIVER (REV. 0.9)

MACY11 27(663) 23-JAN-76 10:26 PAGE 43-43

SEQ 0053

(1)	014514	016203	000006		7\$:	MOV	6(R2),R3	;POINTS TO 1ST ADDRESS OF WHERE
(1)								;TO PUT THE REGISTER(S)
(1)	014520	116237	000010	014536		MOVB	10(R2),9\$	;INIT. THE INDEX FOR THE FIRST REG.
(1)	014526	116205	000011			MOVB	11(R2),R5	;INDEX OF LAST REG. TO MOVE
(1)	014532	004037	017036		8\$:	JSR	RD,RC.RP	;READ RPO4 REGISTER
(1)	014536	000000			9\$:	RHCSI		;INDEX OF REG. TO READ
(1)	014540	014674				CI7		
(1)	014542	012623				MOV	(SP)+,(R3)+	;GET THE CONTENTS OF RH11/RPO4 REG.
(1)	014544	023705	014536			CMP	9\$,R5	;LAST REG. BEEN READ?
(1)	014550	001414				BEQ	12\$	;GET OUT IF YES
(1)	014552	062737	000002	014536		ADD	#2,9\$	;INCREASE THE INDEX BY 2
(1)	014560	000764				BR	8\$	;LOOP--MORE TO READ
(1)	014562	122703	000145		10\$:	CMPB	#145,R3	;IS IT A "SELECT DRIVE" COMMAND?
(1)	014566	001405				BEQ	12\$	;BRANCH IF YES
(1)	014570	010346			11\$:	MOV	R3,-(SP)	;LOAD THE COMMAND
(1)	014572	004037	017174			JSR	RD,WRT.RP	
(1)	014576	000000				RHCSI		
(1)	014600	014674				CI7		
(1)	014602	004737	020174		12\$:	JSR	PC,POPQUE	;REMOVE REG. FROM QUEUE
(1)	014606	052762	000200	000016		BIS	#BIT07,16(R2)	;SET THE "DONE" BIT
(1)	014614	005737	013030			TST	SAVEFG	;SAVE THE RH11/RPO4 REGISTERS?
(1)	014620	100002				BPL	13\$	;BRANCH IF NO
(1)	014622	004737	017336			JSR	PC,SVRH11	;YES--GO SAVE THE REGISTERS
(1)	014626	000207			13\$:	RTS	PC	;RETURN TO USER
(1)	014630	006301			C15:	ASL	R1	
(1)	014632	012761	001750	013034		MOV	#1000.,TIMER(R1)	;SET A ONE SECOND TIMER
(1)	014640	006201				ASR	R1	
(1)	014642	112761	000001	012742		MOVB	#1,DRVACT(R1)	;SET THE DRIVE ACTIVE
(1)	014650	000207				RTS	PC	;RETURN TO THE USER
(1)	014652	010346			C16:	MOV	R3,-(SP)	;LOAD THE COMMAND
(1)	014654	004037	017174			JSR	RD,WRT.RP	
(1)	014660	000000				RHCSI		
(1)	014662	014674				CI7		
(1)	014664	000761				BR	C15	
(1)	014666	105761	012742		C17A:	TSTB	DRVACT(R1)	;IS THE DRIVE ACTIVE?
(1)	014672	001405				BEQ	C17B	;BRANCH IF NO
(1)	014674	012762	104000	000016	C17:	MOV	#BIT15:BIT11,16(R2)	;SET "PARITY" ERROR INDICATOR
(1)	014702	004737	017336			JSR	PC,SVRH11	;GO SAVE THE RH11/RPO4 REGISTERS
(1)	014706	012746	000111		C17B:	MOV	#111,-(SP)	;DO A "DRIVE CLEAR"
(1)	014712	004037	017174			JSR	RD,WRT.RP	
(1)	014716	000000				RHCSI		
(1)	014720	014760				CI8		
(1)	014722	004737	020056			JSR	PC,EMPTYQ	;EMPTY THE QUEUE
(1)	014726	105061	013010			CLRB	UNDFLG(R1)	;CLEAR THE UNLOAD IN QUEUE FLAG
(1)	014732	105061	012742			CLRB	DRVACT(R1)	;DRIVE IS IDLE
(1)	014736	020137	013054			CMP	R1,DTUW	;IF THIS DRIVE HAD AN I/O REQUEST
(1)	014742	001005				BNE	1\$	;IN PROGRESS CLEAR ALL OF THE FLAGS
(1)	014744	005037	013002			CLR	TRNSWT	
(1)	014750	012737	177777	013054		MOV	#-1,DTUW	
(1)	014756	000207			1\$:	RTS	PC	
(1)	014760	004037	020236		C18:	JSR	RD,SAVR15	;SAVE R1-R5
(1)	014764	013704	013070			MOV	RPAOR,R4	;PICKUP THE ADDRESS OF THE FIRST REGISTER
(1)	014770	005001				CLR	R1	
(1)	014772	005003				CLR	R3	

```

(1) 014774 105761 012742 1S: TSTB DRVACT(R1) ;DRIVE ACTIVE?
(1) 015000 001421 BEQ 3S ;BRANCH IF NO
(1) 015002 013702 013002 MOV TRNSWT,R2 ;GET THE "TRANSFER WAIT" QUEUE
(1) 015006 020137 013054 CMP R1,DTUW ;DID THIS DRIVE HAVE AN I/O IN PROGRESS?
(1) 015012 001402 BEQ 2S ;BRANCH IF YES
(1) 015014 004737 020152 JSR PC,GETREQ ;GET THE DPB POINTER
(1) 015020 004737 017336 2S: JSR PC,SVRH11 ;SAVE RH11/RPO4 REGISTERS
(1) 015024 052762 102000 000016 BIS #BIT15!BIT10,16(R2) ;SET "NON-CLEARABLE PARITY" ERROR INDICATOR
(1) 015032 012763 177777 013034 MOV #-1,TIMER(R3) ;STOP THE TIMER
(1) 015040 105061 012742 CLRB DRVACT(R1) ;SET "DRIVE ACTIVE" TO IDLE
(1) 015044 105061 013010 3S: CLRB ULDFLG(R1) ;CLEAR UNLOAD FLAG
(1) 015050 005201 INC R1 ;MOVE TO THE NEXT DRIVE
(1) 015052 062703 000002 ADD #2,R3
(1) 015056 042701 177770 BIC #C7,R1
(1) 015062 001344 BNE 1S ;BRANCH IF MORE DRIVES
(1) 015064 012737 177777 013054 MOV #-1,DTUW ;NO DATA TRANSFERS UNDERWAY
(1) 015072 005037 013002 CLR TRNSWT ;CLEAR THE "TRANSFER WAIT" QUEUE
(1) 015076 004737 017774 JSR PC,CLARUE ;CLEAR ALL OF THE REQUEST QUEUES
(1) 015102 012764 000040 000010 MOV #BIT05,RHCS2(R4) ;DO A MASSBUS INIT.
(1) 015110 004737 017430 JSR PC,SET.IE ;SET "IE" WITHOUT "TRE"
(1) 015114 004037 020256 JSR RO,GETR15 ;RESTORE THE REGISTERS
(1) 015120 000207 RTS PC ;RETURN

```

#LOOK AHEAD ROUTINE

#CALL

```

* MOV #DRVNUM,R1 ;DRIVE NUMBER
* MOV #DPB,R2 ;POINT TO DPB
* JSR RO,LA ;GO CHECK THE WINDOW
* RETURN1 ;ERROR RETURN
* RETURN2 ;START A SEARCH
* RETURN3 ;START A DATA TRANSFER

```

```

(1) 015122 013704 013070 LA: MOV RPADR,R4 ;GET RHCS1'S ADDRESS
(1) 015126 010164 000010 MOV R1,RHCS2(R4) ;SELECT DRIVE
(1) 015132 004037 017036 JSR RO,RO.RP ;READ CURRENT CYLINDER
(1) 015136 000036 RHCC
(1) 015140 015252 4S ;ERROR RETURN ADDRESS
(1) 015142 022662 000012 CMP (SP)+,12(R2) ;IS CURRENT CYLINDER=DESIRED
(1) ;CYLINDER?
(1) 015146 001037 BNE 3S ;EXIT IF NO
(1) 015150 105261 013020 INCB LACNT(R1) ;INCREMENT THE LOCK AHEAD COUNT
(1) 015154 126137 013020 013076 CMPB LACNT(R1),MXLACT ;EXCEED MAX?
(1) 015162 003026 BGT 2S ;BRANCH IF YES
(1) 015164 116203 000010 MOVB 10(R2),R3 ;GET DESIRED SECTOR ADDRESS AND
(1) 015170 000303 SHAB R3 ;MULT. BY 64--ALIGN WITH
(1) 015172 006203 ASR R3 ;LOOK AHEAD REGISTER
(1) 015174 006203 ASR R3
(1) 015176 012737 000340 177776 MOV #340,#PS ;PRIORITY LEVEL "7"
(1) 015204 004037 017036 JSR RO,RO.RP ;READ LOOK AHEAD REGISTER
(1) 015210 000020 RHLA
(1) 015212 015252 4S
(1) 015214 162603 SUB (SP)+,R3 ;CALCULATE THE DELTA
(1) 015216 002002 BGE 1S

```



EOS

MAINDEC-11-DZRPL-B "RPO4 FORMATTER PROGRAM"  
DZRPLB.P11 RH11/RPO4 DRIVER (REV. 0.9)

MACY11 27(663) 23-JAN-76 10:26 PAGE 43-45

SEG 0055

```

(1) 015220 062703 002600          ADD      #(<22.*64.>),R3   ; MAKE THE DELTA POSITIVE
(1) 015224 023703 013100      1S:    CMP      MXDLTA,R3      ; CHECK THE DELTA TO SEE
(1) 015230 002406              BLT      3S              ; IF IT IS WITHIN THE
(1) 015232 023703 013102      CMP      MNDLTA,R3      ; WINDOW---IF YES, ZERO
(1) 015236 002003              BGE      3S              ; THE LOOK AHEAD COUNT
(1) 015240 105061 013020      2S:    CLRB     LACNT(R1)     ; AND TAKE THE I/O EXIT
(1) 015244 005720              TST     (R0)+
(1) 015246 005720              3S:    TST     (R0)+      ; ADJUST THE RETURN ADDRESS
(1) 015250 000200              RTS     R0
(1) 015252 004737 014674      4S:    JSR     PC,C17      ; PROCESS THE ERROR
(1) 015256 000200              RTS     R0              ; TAKE ERROR RETURN

(1)
(1)
(1)          ;*INTERRUPT SERVICE ROUTINE
(1) 015260 112737 000001 013006  ISR:    MOVB     #1,ACTDRV      ; SET "ACTIVE DRIVER" FLAG
(1) 015266 004037 020236          JSR     R0,SAVR15      ; SAVE R1-R5
(1) 015272 013704 013070          MOV     RPAOR,R4      ; ADDRESS OF RHSCSI
(1) 015276 013701 013054          MOV     DTUW,R1       ; GET "DATA TRANSFER UNDERWAY" INDICATOR
(1) 015302 002403              BLT     1S              ; BRANCH IF NO DATA TRANSFER UNDERWAY
(1) 015304 004737 015330          JSR     PC,TD          ; CALL TRANSFER DONE
(1) 015310 000402              BR     2S              ; EXIT
(1) 015312 004737 015456      1S:    JSR     PC,SC          ; CALL SPECIAL CONDITIONS
(1) 015316 004037 020256      2S:    JSR     R0,GETR15  ; RESTORE R1-R5
(1) 015322 105037 013006          CLRB     ACTDRV      ; CLEAR "ACTIVE DRIVER" FLAG
(1) 015326 000002              RTI                    ; RETURN

(1)
(1)          ;*TRANSFER DONE ROUTINE
(1) 015330 105061 012742          TD:    CLRB     DRVACT(R1) ; SET DRIVE ACTIVE INDICATOR TO IDLE
(1) 015334 012737 177777 013054    MOV     #-1,DTUW      ; NO DATA TRANSFERS UNDERWAY
(1) 015342 006301              ASL     R1
(1) 015344 012761 177777 013034    MOV     #-1,TIMER(R1) ; CANCEL TIMEOUT
(1) 015352 006201              ASR     R1
(1) 015354 013702 013002          MOV     TRANST,R2     ; GET "DPB" ADDRESS FROM THE
(1) 015360 005037 013002          CLR     TRANST        ; TRANSFER WAIT QUEUE--CLEAR QUEUE
(1) 015364 052762 000200 000016    BIS     #BIT07,16(R2) ; SET DONE
(1) 015372 010164 000010          MOV     R1,RHCS2(R4) ; SELECT THE DRIVE
(1) 015376 004037 017036          JSR     R0,RD.RP      ; TRANSFER ERROR(TRE=1)?
(1) 015402 000000              RHCS1
(1) 015404 014674              C17
(1) 015406 003126              ROL     (SP)+
(1) 015410 100410              BMI     2S              ; BR IF YES
(1) 015412 005737 013030          TST     SAVEFG        ; SAVE THE RH11/RPO4 REGISTERS
(1) 015416 100002              BPL     1S              ; BRANCH IF NO
(1) 015420 004737 017336          JSR     PC,SVRH11     ; YES--SAVE THE REGISTERS
(1) 015424 004737 013672      1S:    JSR     PC,OPT        ; CALL OPTIMIZER
(1) 015430 000456              BR     SC2             ; SPECIAL CONDITION (ENTRY #2)
(1) 015432 052762 100100 000016  2S:    BIS     #BIT15!BIT06,16(R2) ; SET DATA ERROR FLAG
(1) 015440 004737 020056          JSR     PC,EMPTYQ     ; EMPTY THE "DRIVES WAIT" QUEUE
(1) 015444 004737 017336          JSR     PC,SVRH11     ; SAVE THE RH11/RPO4 REGISTERS
(1) 015450 012714 040111          MOV     #40111,(R4)  ; ISSUE A "DRIVE CLEAR"
(1) 015454 000444              BR     SC2             ; SPECIAL CONDITION (ENTRY #2)

(1)
(1)          ;*SPECIAL CONDITION ROUTINE

```

# F05

MAINDEC-11-DZRPL-B "RPO4 FORMATTER PROGRAM"  
DZRPLB.P11 RH11/RPO4 DRIVER (REV. 0.9)

MACY11 27(663) 23-JAN-76 10:26 PAGE 43-46

SEQ 0056

```

(1)
(1) 015456 005001          SC:  CLR      R1          ;START WITH DRIVE 0
(1) 015460 012702 000001    MOV      #1,R2
(1) 015464 105761 012752    1$:  TSTB   DRVSTA(R1) ;NONEXISTENT?
(1) 015470 002016          BGE     Z$          ;NO--BRANCH
(1) 015472 005201          INC     R1          ;YES--MOVE TO THE NEXT DRIVE
(1) 015474 106302          ASLB   R2          ;MORE DRIVES?
(1) 015476 103372          BCC    1$          ;YES--BRANCH
(1) 015500 013746 013032    MOV     SEEKFG,-(SP) ;SAVE THE "SEEK FLAG"
(1) 015504 013746 013030    MOV     SAVEFG,-(SP) ;SAVE THE "SAVE FLAG"
(1) 015510 004737 013106    JSR    PC,RPINIT   ;GO INIT. THE SUBSYSTEM
(1) 015514 012637 013030    MOV     (SP)+,SAVEFG ;RESTORE THE "SAVE FLAG"
(1) 015520 012637 013032    MOV     (SP)+,SEEKFG ;RESTORE THE "SEEK FLAG"
(1) 015524 000405          BR     SC1          ;TAKE ERROR EXIT
(1) 015526 010164 000010    2$:  MOV     R1,RHCS2(R4) ;SELECT DRIVE
(1) 015532 116405 000001    MOVB   1(R4),R5    ;IS "SC"=1?
(1) 015536 100413          BMI    SC2          ;BRANCH IF YES
(1) 015540 012701 000010    SC1:  MOV     #8.,R1   ;DO EIGHT DRIVES
(1) 015544 005301          1$:  DEC     R1          ;NEXT DRIVE
(1) 015546 002743          BLT    SC          ;BRANCH IF OUT OF DRIVES
(1) 015550 105761 012742    TSTB   DRVACT(R1) ;IS THIS DRIVE IDLE?
(1) 015554 001373          BNE    1$          ;BRANCH IF NO
(2) 015556 104001          ERROR  1           ;REPORT THE ERROR
(1) 015560 004737 017430    JSR    PC,SET.IE   ;GO SET INTERRUPT ENABLE
(1) 015564 000207          RTS    PC
(1) 015566 012701 000003    SC2:  MOV     #3,R1    ;READ RHAS UP TO THREE TIMES
(1) 015572 116403 000016    1$:  MOVB   RHAS(R4),R3 ;READ "RHAS"
(1) 015576 001011          BNE    Z$          ;BRANCH IF ANY ATA BITS = 1
(1) 015600 005301          DEC     R1          ;COUNT THIS READ
(1) 015602 003373          BGT    1$          ;LOOP IF MORE READS ALLOWED
(1) 015604 004037 017036    JSR    RO,RO.RP   ;READ CONTROL AND STATUS REGISTER
(1) 015610 000000          RHCS1
(1) 015612 014760          CIB
(1) 015614 106126          ROLB   (SP)+       ;IS "IE"=1?
(1) 015616 100350          BPL    SC1          ;NO--TAKE ERROR EXIT
(1) 015620 000207          RTS    PC          ;YES--RETURN
(1) 015622 005046          2$:  CLR     -(SP)      ;PROCESS ALL DRIVES THAT HAVE
(1) 015624 110316          MOVB   R3,(SP)     ;AN "ATA"=1
(1) 015626 012703 000001    MOV     #1,R3
(1) 015632 005001          CLR     R1
(1) 015634 030316          SC4:  BIT     R3,(SP)   ;ATA=1?
(1) 015636 001005          BNE    SC5          ;YES--BRANCH
(1) 015640 005201          SC3:  INC     R1          ;MOVE TO THE NEXT DRIVE
(1) 015642 106303          ASLB   R3
(1) 015644 001373          BNE    SC4          ;BRANCH IF MORE TO CHECK?
(1) 015646 005726          TST   (SP)+       ;CLEAN OFF THE STACK
(1) 015650 000207          RTS    PC          ;RETURN TO USER
(1) 015652 023701 013054    SC5:  CMP     DTUW,R1  ;IS THIS DRIVE SETUP FOR I/O?
(1) 015656 001002          BNE    1$          ;NO---BRANCH
(1) 015660 005726          TST   (SP)+       ;YES---CLEAN OFF THE STACK (RHAS)
(1) 015662 000622          BR     TD          ;BRANCH TO "TRANSFER DONE"
(1) 015664 105761 012752    1$:  TSTB   DRVSTA(R1) ;CHECK THE DRIVE STATUS
(1) 015670 003030          BGT    SC4          ;BRANCH IF ONLINE
(1) 015672 105761 013010    TSTB   ULDFLG(R1) ;UNLOAD IN PROGRESS?

```

(1)	015676	003420			BLE	2\$			; BRANCH IF NO
(1)	015700	004737	020152		JSR	PC,GETREQ			; GET DPB POINTER
(1)	015704	004737	017336		JSR	PC,SVRH11			; SAVE THE RH11/RPO4 REGISTERS
(1)	015710	004737	016366		JSR	PC,SC12			; SAVE RHDS1, RHER1, RHER2, AND RHER3
(1)									; ALSO DO A DRIVE INIT (DRVINT)
(1)	015714	105761	012752		TSTB	DRVSTA(R1)			; DID DRIVE COME ONLINE?
(1)	015720	003411			BLE	3\$			; NO---BRANCH
(1)	015722	032737	040000	012732	BIT	#BIT14,RPERRS			; WAS THERE AN ERROR?
(1)	015730	001565			BEQ	SC11			; NO -- BRANCH
(1)	015732	013705	012734		MOV	RPERRS+2,R5			; YES -- PICKUP RHER1 AND
(1)	015736	000447			BR	SC6A			; GO PROCESS THE ERROR
(1)	015740	004737	016366		JSR	PC,SC12		2\$:	; SAVE RHDS1, RHER1, RHER2, AND RHER3
(1)									; ALSO DO A DRVINT
(1)	015744	011605			MOV	(SP),R5		3\$:	; PICKUP (RHAS) BEFORE THE ERROR CALL
(2)	015746	104005			ERROR	5			; REPORT THE ERROR
(1)	015750	000733			BR	SC3			; GO CHECK FOR MORE ATA'S
(1)	015752	006301			ASL	R1		SC6:	
(1)	015754	012761	177777	013034	MOV	#-1,TIMER(R1)			; STOP THE TIMER
(1)	015762	006201			ASR	R1			
(1)	015764	004737	020152		JSR	PC,GETREQ			; GET THE DPB POINTER FROM THE QUEUE
(1)	015770	010364	000016		MOV	R3,RHAS(R4)			; CLEAR ATTENTION
(1)	015774	010164	000010		MOV	R1,RHCS2(R4)			; SELECT DRIVE
(1)	016000	004037	017036		JSR	RO,RO.RP			; READ THE RPO4'S STATUS REG.
(1)	016004	000012			RHDS1				
(1)	016006	016224			SCB				
(1)	016010	011605			MOV	(SP),R5			; AND PUT IT IN R5
(1)	016012	006126			ROL	(SP)↓			; WAS THERE AN ERROR?
(1)	016014	100115			BPL	SC9			; BR IF NO
(1)	016016	105761	012742		TSTB	DRVACT(R1)			; CHECK THE DRIVE ACTIVE INDICATOR
(1)	016022	001522			BEQ	SC10			; BRANCH IF IDLE
(1)	016024	004037	017036		JSR	RO,RO.RP			; READ ERROR REGISTER #1
(1)	016030	000014			RHER1				
(1)	016032	016224			SCB				
(1)	016034	012605			MOV	(SP)+R5			; AND SAVE IT IN R5
(1)	016036	004737	017336		JSR	PC,SVRH11			; SAVE RH11/RPO4 REGISTERS
(1)	016042	012746	000111		MOV	#111,-(SP)			; ISSUE A DRIVE CLEAR
(1)	016046	004037	017174		JSR	RO,WAT.RP			
(1)	016052	000000			RHCS1				
(1)	016054	016224			SCB				
(1)	016056	006105			ROL	R5		SC6A:	; WAS "UNSAFE" CONDITION =1?
(1)	016060	100404			BIT	1\$			; BRANCH IF YES
(1)	016062	052762	100240	000016	BIS	#BIT15!BIT07!BIT05,16(R2)			; INFORM USER OF ERROR
(1)	016070	000443			BIT	SC7			
(1)	016072	004037	017036		JSR	RO,RO.RP		1\$:	; READ DRIVE STATUS REG. #1
(1)	016076	000012			RHDS1				
(1)	016100	016224			SCB				
(1)	016102	011605			MOV	(SP),R5			; SAVE RHDS1 IN R5
(1)	016104	006126			ROL	(SP)↓			; "ERR"=1?
(1)	016106	100013			BPL	2\$			; BR IF NO--UNSAFE CLEARED
(1)	016110	112761	177777	012752	MOVB	#-1,DRVSTA(R1)			; DRIVE IS UNSAFE
(1)	016116	004737	017336		JSR	PC,SVRH11			; SAVE RH11/RPO4 REGISTERS
(1)	016122	010364	000016		MOV	R3,RHAS(R4)			; CLEAR ATTENTION
(1)	016126	052762	110000	000016	BIS	#BIT15!BIT12,16(R2)			; INFORM USER OF UNSAFE ERROR
(1)	016134	000421			BR	SC7			

# H05

MAINDEC-11-DZRPL-B "RPO4 FORMATTER PROGRAM"  
 DZRPLB.P11 RH11/RPO4 DRIVER (REV. 0.9)

MACY11 27(663) 23-JAN-76 10:26 PAGE 43-48

SEQ 0058

```

(1) 016136 105705          2$:  TSTB   R5           ;"DRY" =1?
(1) 016140 100414          BMI    3$           ;BRANCH IF YES
(1) 016142 112761 177777 012742  MOVB  #-1,DRVACT(R1) ;ACTIVE ERROR RECOVER
(1) 016150 112761 000001 012752  MOVB  #1,DRVSTA(R1) ;ONLINE
(1) 016156 006301          ASL   R1
(1) 016160 012761 072460 013034  MOV   #30000.,TIMER(R1) ;START 30 SECOND TIMER
(1) 016166 006201          ASR   R1
(1) 016170 000623          BR    SC3
(1) 016172 052762 100220 000016 3$:  BIS   #BIT15!BIT07!BIT04,16(R2) ;INFORM USER OF ERROR
(1) 016200 105061 012742          SC7: CLRB  DRVACT(R1) ;DRIVE IS IDLE
(1) 016204 004737 020056          JSR   PC,EMPTYQ ;DUMP THE QUEUE
(1) 016210 105761 013010          TSTB  ULDFLG(R1) ;UNLOAD IN PROGRESS OR QUEUE?
(1) 016214 001611          BEQ   SC3 ;BR IF NO
(1) 016216 105061 013010          CLRB  ULDFLG(R1) ;CLEAR UNLOAD FLAG
(1) 016222 000606          BR    SC3
(1) 016224 005726          SC8: TST   (SP)+ ;REMOVE (RHAS) FROM THE STACK
(1) 016226 105761 012742          TSTB  DRVACT(R1) ;IS DRIVE IDLE?
(1) 016232 001404          BEQ   1$ ;YES--BRANCH
(1) 016234 004737 020152          JSR   PC,GETREQ ;GET DPB POINTER
(1) 016240 000137 014674          JMP   CI? ;PROCESS THE PARITY ERROR
(1) 016244 000137 014706          1$:  JMP   CI?B ;PROCESS THE PARITY ERROR
(1) 016250 105761 012742          SC9: TSTB  DRVACT(R1) ;TEST DRIVE ACTIVE
(1) 016254 003013          BGT   SC11 ;BRANCH IF DRIVE IS ACTIVE
(1) 016256 001404          BEQ   SC10 ;BRANCH IF DRIVE IS IDLE
(1) 016260 052762 100210 000016  BIS   #BIT15!BIT07!BIT03,16(R2) ;INFORM USER OF ERROR RECOVER COMPLETION
(1) 016266 000744          BR    SC7
(1) 016270 011605          SC10: MOV   (SP),R5 ;PUT (RHAS) IN R5
(1) 016272 004737 016366          JSR   PC,SC12 ;SAVE RH0S1, RHER1, RHER2, AND RHER3
(2) 016276 104002          ERROR 2 ;REPORT THE ERROR
(1) 016300 000137 015640          JMP   SC3 ;GO CHECK FOR MORE ATA'S
(1) 016304 105761 013010          SC11: TSTB  ULDFLG(R1) ;"UNLOAD IN PROGRESS"?
(1) 016310 003402          BLE   1$ ;BRANCH IF NO
(1) 016312 105061 013010          CLRB  ULDFLG(R1) ;CLEAR UNLOAD FLAG
(1) 016316 105061 012742          1$:  CLRB  DRVACT(R1) ;SET DRIVE IDLE
(1) 016322 136137 013056 013004  BITB  ATABIT(R1),SRCHWT ;DOING A SEARCH OPERATION FOR
(1)                                     ;AN I/O COMMAND?
(1) 016330 001012          BNE   2$ ;BRANCH IF YES
(1) 016332 004737 020174          JSR   PC,POPQUE ;REMOVE REQUEST FROM QUEUE
(1) 016336 052762 000200 000016  BIS   #BIT07,16(R2) ;SET "DONE" BIT
(1) 016344 005737 013030          TST   SAVEFG ;SAVE THE RH11/RPO4 REGISTERS?
(1) 016350 100002          BPL   2$ ;BRANCH IF NO
(1) 016352 004737 017336          JSR   PC,3VRH11 ;YES--SAVE ALL OF THE RH11/RPO4 REG'S
(1) 016356 004737 013672          2$:  JSR   PC,OPT ;START A REQUEST
(1) 016362 000137 015640          JMP   SC3
(1) 016366 010164 000010          SC12: MOV   R1,RHCS2(R4) ;SELECT DRIVE
(1) 016372 016437 000012 012732  MOV   RH0S1(R4),RPERRS ;SAVE THE FOUR REGISTERS THAT
(1) 016400 016437 000014 012734  MOV   RHER1(R4),RPERRS+2 ;WILL TELL US SOMETHING
(1) 016406 016437 000040 012736  MOV   RHER2(R4),RPERRS+4
(1) 016414 016437 000042 012740  MOV   RHER3(R4),RPERRS+6
(1) 016422 004037 013250          JSR   RD,DRVINT ;INIT. THE STATE OF THE DRIVE
(1) 016426 000401          BR    1$ ;TAKE ERROR EXIT
(1) 016430 000207          RTS   PC ;RETURN
(1) 016432 005726          1$:  TST   (SP)+ ;POP PC OFF OF THE STACK
(1) 016434 000673          BR    SC8 ;PROCESS THE PARITY ERROR

```

```

(1)
(1)
(1)
(1)
(1)
(1)
(1) 016436 005737 013006
(1) 016442 001032
(1) 016444 112737 000001 013007
(1) 016452 004037 020236
(1) 016456 005001
(1) 016460 005003
(1) 016462 005763 013034
(1) 016466 002407
(1) 016470 166663 000016 013034
(1) 016476 003003
(1) 016500 004037 016534
(1) 016504 000405
(1) 016506 005201
(1) 016510 005723
(1) 016512 022701 000010
(1) 016516 003361
(1) 016520 004037 020256
(1) 016524 105037 013007
(1) 016530 012616
(1) 016532 000200

; *RPO4 TIMER ROUTINE
; *CALL
; *
; *      MOV      #TIME, -(SP)      ; ELAPSED TIME IN MILLISECONDS ON THE STACK
; *      JSR      RO, RPTMR        ; CALL RPO4 TIME ROUTINE

RPTMR: TST      ACTDRV            ; CHECK "ACTDRV & ACTSTR"
        BNE      4$              ; IF NON ZERO EXIT
        MOV      #1, ACTSTR      ; SET "ACTSTR"
        JSR      RO, SAVR15      ; SAVE R1-R5
        CLR      R1              ; START WITH DRIVE 0
        CLR      R3
1$:     TST      TIMER(R3)        ; IS THE TIMER RUNNING?
        BLT      2$              ; BRANCH IF NO
        SUB      16(SP), TIMER(R3) ; COUNT THE INTERVAL
        BGT      2$              ; BR IF NO SOFTWARE TIMEOUT
        JSR      RO, STO          ; CALL SOFTWARE TIMEOUT ROUTINE
        BR       3$              ; GO TO THE EXIT
2$:     INC      R1              ; MOVE TO NEXT DRIVE
        TST      (R3)+
        CMP      #8., R1         ; OUT OF DRIVES?
        BGT      1$              ; BRANCH IF NO
        JSR      RO, GETR15      ; RESTORE R1-R5
        CLRB     ACTSTR          ; ZERO ACTIVE SOFTWARE TIMEOUT ROUTINE FLAG
3$:     MOV      (SP)+, (SP)     ; ADJUST THE STACK
        RTS      RO             ; RETURN

; *SOFTWARE TIMEOUT ROUTINE
; *CALL: STO
; *
; *      MOV      #DRVNUM, R1      ; DRIVE NUMBER
; *      JSR      RO, STO          ; CALL--DRVACT MUST BE NONZERO
; *      RETURN

STO:    MOV      2#PS, -(SP)      ; SAVE THE PROCESSOR STATUS
        MOV      RPVEC+2, 2#PS   ; SET THE "PS" TO RPO4 BR LEVEL
        JSR      RO, SAVR15      ; SAVE R1-R5
        MOV      RPADR, R4       ; GET ADDRESS OF "RHCS1"
        MOV      R1, RHCS2(R4)   ; SELECT THE DRIVE
        JSR      RO, RO.RP       ; READ "DRIVE STATUS REG"
        RHDS1
        STOS
        TSTB     (SP)+           ; IS "DRY"=1?
        BMI     ST02            ; BR IF YES
        MOV      TRNSWT, R2      ; PICKUP TRANSFER WAIT QUEUE
        CMP      R1, DTUM        ; TRANSFER UNDERWAY ON THIS DRIVE?
        BEQ     1$              ; BRANCH IF YES
        JSR      PC, GETREQ      ; GET DPG ADDRESS
1$:     BIS      #BIT15!BIT09, 16(R2) ; SET THE ERROR FLAGS
        JSR      PC, SVRH11      ; SAVE RH11/RPO4 REGISTERS
        MOV      #BIT05, RHCS2(R4) ; "INIT" THE MASS BUS
        CLRB     DRVACT(R1)      ; DRIVE IS IDLE
        CLRB     ULDFLG(R1)      ; CLEAR THE UNLOAD FLAG
        CLR      R1              ; START WITH DRIVE 0

ST01:   MOV      TRNSWT, R2
        CMP      R1, DTUM
        BEQ     1$
        JSR      PC, GETREQ
        BIS      #BIT15!BIT09, 16(R2)
        JSR      PC, SVRH11
        MOV      #BIT05, RHCS2(R4)
        CLRB     DRVACT(R1)
        CLRB     ULDFLG(R1)
        CLR      R1

```

```

(1) 016646 005003          CLR      R3
(1) 016650 004037 013250 2$: JSR      R0,DRVINT ;INIT. THIS DRIVE
(1) 016654 000465          BR       ST05 ;PARITY ERROR RETURN
(1) 016656 105761 012742  TSTB     DRVACT(R1) ;DRIVE IDLE BEFORE THE INIT.?
(1) 016662 001414          BEQ      4$ ;YES--BRANCH
(1) 016664 013702 013002  MOV      TRNSWT,R2 ;GET TRANSFER WAIT QUEUE
(1) 016670 027701 013054  CMP      DTUW,R1 ;WAS THERE I/O ON THIS DRIVE?
(1) 016674 001702          BEQ      3$ ;YES--BRANCH
(1) 016676 004737 020152  JSR      PC,GETREQ ;GET THE DPB POINTER FROM QUEUE
(1) 016702 052762 100400 000016 3$: BIS      #BIT15:BIT08,16(R2) ;INFORM USER OF INIT.
(1) 016710 105061 012742  CLRB     DRVACT(R1) ;SET DRIVE ACTIVE TO IDLE
(1) 016714 105061 013010 4$: CLRB     ULDFLG(R1) ;NO UNLOAD
(1) 016720 012763 177777 013034  MOV      #-1,TIMER(R3) ;STOP THE TIMER
(1) 016726 005723          TST      (R3)+ ;UPDATE THE INDEX
(1) 016730 005201          INC      R1 ;INCREMENT THE DRIVE NUMBER
(1) 016732 022701 000010  CMP      #8.,R1 ;LAST DRIVE BEEN CHECKED?
(1) 016736 003344          BGT      2$ ;NO--LOOP
(1) 016740 012737 177777 013054  MOV      #-1,DTUW ;NO DATA TRANSFERS UNDERWAY
(1) 016746 005037 013002  CLR      TRNSWT ;CLEAR TRANSFER WAIT QUEUE
(1) 016752 004737 017774  JSR      PC,CLRQUE ;CLEAR ALL REQUEST QUEUES
(1) 016756 000417          BR       ST04 ;EXIT
(1) 016760 016405 000016  ST02: MOV      RHAS(R4),R5 ;IS ATTENTION FOR THIS
(1) 016764 136105 013056  BITB     ATABIT(R1),R5 ;DRIVE UP?
(1) 016770 001011          BNE     ST03 ;YES--BRANCH
(1) 016772 020137 013054  CMP      R1,DTUW ;DATA TRANSFER UNDERWAY FOR THIS DRIVE
(1) 016776 001277          BNE     ST01 ;BR IF NO
(1) 017000 004037 017036  JSR      R0,RD.RP ;YES--CHECK "RDY"
(1) 017004 000000          RHCS1
(1) 017006 017030          ST05
(1) 017010 105726          TSTB     (SP)+
(1) 017012 100271          BPL     ST01 ;BR IF "RDY"=0
(1) 017014 005720  ST03: TST      (R0)+ ;ADJUST FOR THE PROPER RETURN
(1) 017016 004037 020256  ST04: JSR      R0,GETR15 ;RESTORE R1-R5
(1) 017022 012637 177776  MOV      (SP)+,2#PS ;RESTORE PROCESSOR STATUS
(1) 017026 000200          RTS     R0 ;RETURN
(1) 017030 004737 014760  ST05: JSR      PC,C18 ;GO HANDLE THE PARITY ERROR
(1) 017034 000770          BR       ST04

; *ROUTINE TO READ A RH11/RPO4 REGISTER
; *CALL
; * JSR      R0,RD.RP ;GO READ A REGISTER
; * INDEX ;REG. INDEX FROM BASE
; * ERRADR ;ERROR ADDRESS--PROCESS ERROR STARTING
; * ;AT THIS ADDRESS
; * RETURN ;CONTENTS OF REG. IS ON THE STACK

(1) 017036 013737 013066 017162 RD.RP: MOV      MCPMX,RD.RP2 ;MAX. RETRYS ALLOWED
(1) 017044 011646          MOV      (SP)-(SP) ;SAVE R0 FOR RETURN
(1) 017046 013737 013070 017062  MOV      RPAR,RD.ADR ;FORM THE DESIRED ADDRESS
(1) 017054 062037 017062  ADD      (R0)+,RC.ADR ;USING THE BASE AND THE INDEX
(1) 017060 013727  RD.RP1: MOV      2(PC)+,(PC)+ ;READ THE DESIRED REGISTER OF THE RPO4
(1) 017062 000000  RD.ADR: .WORD 0 ;ADDRESS IS FORMED HERE
(1) 017064 000000  RD.WRD: .WORD 0 ;REG. CONTENTS PUT HERE

```

# K05

MAINDEC-11-DZRPL-B "RPO4 FORMATTER PROGRAM"  
DZRPLB.P11 RH11/RPO4 DRIVER (REV. 0.9)

MACY11 27(663) 23-JAN-76 10:26 PAGE 43-51

SEQ 0061

```

(1) 017066 013766 017064 000002      MOV      RD.WRD,2(SP)      ;RETURN IT TO THE USER
(1) 017074 017746 173770              MOV      @RPADR,-(SP)     ;READ RHCS1
(1) 017100 032716 020000              BIT      #BIT13,(SP)     ;DID MCPE SET?
(1) 017104 001002                      BNE      1$              ;BRANCH IF YES
(1) 017106 022620                      CMP      (SP)+,(RO)+     ;ADJUST FOR RETURN
(1) 017110 000200                      RTS      RO              ;RETURN IF NO
(2) 017112                               1$:
(2) 017112 104003                      ERROR    3               ;REPORT THE ERROR
(1) 017114 005737 013054              TST      DTUW            ;DATA TRANSFER UNDERWAY?
(1) 017120 100405                      BMI      2$              ;NO--BRANCH
(1) 017122 032716 040000              BIT      #BIT14,(SP)     ;NO--"TRE"=1?
(1) 017126 001402                      BEQ      2$              ;NO--BRANCH
(1) 017130 005726                      TST      (SP)+           ;YES--CLEAN OFF THE STACK AND
(1) 017132 000415                      BR       RD.RP3          ;TAKE THE FATAL ERROR EXIT
(1) 017134 052716 040000              2$:  BIS      #BIT14,(SP) ;CLEAR "MCPE" BY SENDING A "1" TO "TRE"
(1) 017140 000316                      SWAB     (SP)            ;POSITION BEFORE WRITING
(1) 017142 013737 013070 017156      MOV      RPADR,3$       ;FORM ADDRESS OF HIGH BYTE
(1) 017150 005237 017156              INC      3$
(1) 017154 112637                      MOV      (SP)+,@(PC)+   ;WRITE THE HIGH BYTE OF RHCS1
(1) 017156 000000                      3$:  .WORD    0           ;ADDRESS STORAGE
(1) 017160 005327                      DEC      (PC)+          ;EXCEEDED MAX. RETRYS
(1) 017162 000003                      RD.RP2: .WORD    3
(1) 017164 002335                      BGE     RD.RP1          ;BRANCH IF NO
(1) 017166 011000                      RD.RP3: MOV      (RO),RO ;FATAL ERROR EXIT
(1) 017170 012616                      MOV      (SP)+,(SP)
(1) 017172 000200                      RTS      RO

(1)                                     ;*ROUTINE TO WRITE A RH11/RPO4 REGISTER
(1)                                     ;*CALL
(1)                                     ;*
(1)                                     ;*   MOV      DATA,-(SP)   ;DATA TO BE LOADED ON THE STACK
(1)                                     ;*   JSR      RO,WRT.RP    ;CALL THE ROUTINE TO LOAD(WRITE) THE REG.
(1)                                     ;*   INDEX   ;INDEX OF THE REGISTER TO BE LOADED
(1)                                     ;*   ERRADR  ;ADDRESS TO RETURN TO ON AN ERROR
(1)                                     ;*   RETURN  ;ERROR FREE RETURN
(1) 017174 016637 000002 017262 WRT.RP: MOV      2(SP),WRT.WD ;SAVE THE WORD TO WRITE
(1) 017202 012616                      MOV      (SP)+,(SP)     ;ADJUST THE STACK
(1) 017204 012037 017264              MOV      (RO)+,WRT.AD   ;GET INDEX OF REGISTER TO BE WRITTEN
(1) 017210 001020                      BNE      2$              ;BRANCH IF NOT RHCS1
(1) 017212 122737 000150 017262      CMPB    #150,WRT.WD     ;IS THE COMMAND FOR DATA TRANSFERS?
(1) 017220 002414                      BLT     2$              ;YES--DON'T GET THE OLD A16&A17. & PSEL
(1) 017222 004037 017036              JSR     RO,RD.RP        ;NO---COMBINE A16&A17. & PSEL WITH
(1) 017226 000000                      RHCS1   ;THE COMMAND BEFORE SENDING IT TO
(1) 017230 017246                      1$     ;THE RH11/RPO4
(1) 017232 000316                      SWAB     (SP)
(1) 017234 042716 177770              BIC     #1C7,(SP)
(1) 017240 112637 017263              MOV     (SP)+,WRT.WD+1
(1) 017244 000402                      BR      2$
(1) 017246 011000                      1$:  MOV      (RO),RO    ;TAKE THE ERROR EXIT
(1) 017250 000200                      RTS     RO
(1) 017252 063737 013070 017264      2$:  ADD      RPADR,WRT.AD ;FORM THE ADDRESS OF THE DISK REG.
(1) 017260 012737                      MOV     (PC)+,@(PC)+   ;LOAD THE DESIRED REG.
(1) 017262 000000                      WRT.WD: .WORD    0     ;WORD TO WRITE GOES HERE

```

```

(1) 017264 000000          WRT.AD:  WORD 0          ;ADDRESS IS FORMED HERE
(1) 017266 004037 017036 JSR      RO,RO.RP      ;CHECK FOR PARITY ERROR ON WRITE
(1) 017272 000014          RHER1
(1) 017274 017326          2$
(1) 017276 032726 000010 BIT      #BIT03,(SP)+
(1) 017302 001413          BEQ      3$          ;BRANCH IF "PAR=0"
(1) 017304 016037 177776 017316 MOV      -2(RO),1$   ;PICKUP THE INDEX
(1) 017312 004037 017036 JSR      RO,RO.RP   ;READ THE REG.
(1) 017316 000000          1$:  WORD 0          ;REG. INDEX
(1) 017320 017326          2$          ;RETURN TO THIS ADDRESS ON ERROR
(2) 017322 104004          ERROR 4          ;REPORT THE ERROR
(1) 017324 005726          TST      (SP)+      ;CLEAN OFF THE STACK
(1) 017326 011000          2$:  MOV      (RO),RO ;TAKE THE "PARITY ON WRITE" ERROR EXIT
(1) 017330 000200          RTS      RO
(1) 017332 005720          3$:  TST      (RO)+   ;ADJUST FOR ERROR FREE EXIT
(1) 017334 000200          RTS      RO

(1)
(1) ;*ROUTINE TO SAVE THE RH11/RPO4 REGISTERS AS PER DPB+14
(1) ;*CALL
(1) ;* MOV      #DPBNUM,R2 ;DPB POINTER TO R2
(1) ;* JSR      PC,SVRH11 ;SAVE THE DRIVES REG'S
(1)
(1) SVRH11: JSR      RO,SAVR15 ;SAVE REG.'S R1-R5
(1) 017342 013704 013070 MOV      RPAOR,R4
(1) 017346 111264 000010 MOV      (R2),RHCS2(R4) ;SELECT DRIVE
(1) 017352 016202 000014 MOV      14(R2),R2 ;GET THE ERROR TABLE POINTER
(1) 017356 001421          BEQ      4$          ;EXIT IF 0
(1) 017360 005003          CLR      R3          ;COUNTER & POINTER
(1) 017362 012705 000022 MOV      #RHDB,R5 ;PROBLEM REGISTER
(1) 017366 020305          1$:  CMP      R3,R5 ;REACHED RHDB?
(1) 017370 001005          BNE      2$          ;BR IF NO
(1) 017372 105764 000010 TSTB     RHCS2(R4) ;CHECK "OR"
(1) 017376 100402          BMI      2$          ;BRANCH IF RHDB CAN BE READ
(1) 017400 005022          CLR      (R2)+      ;ELSE SAVE IT AS 0'S
(1) 017402 000403          BR      3$
(1) 017404 010446          2$:  MOV      R4,-(SP) ;FORM RH11/RPO4 ADDRESS THAT IS
(1) 017406 060316          ADD      R3,(SP) ;TO BE READ
(1) 017410 013622          MOV      2(SP)+,(R2)+ ;AND READ IT
(1) 017412 005723          3$:  TST      (R3)+ ;MOVE TO NEXT REG INDEX
(1) 017414 020327 000046          CMP      R3,#RHEC2 ;DONE?
(1) 017420 003762          BLE      1$          ;BRANCH IF NO
(1) 017422 004037 020256          4$:  JSR      RO,GETR15 ;GET REGISTERS R1-R5
(1) 017426 000207          RTS      PC ;RETURN TO USER

(1)
(1) ;*ROUTINE TO SET THE INTERRUPT WITHOUT GETTING A "TRE"
(1) ;*CALL
(1) ;* MOV      #DRVNUM,R1 ;DRIVE NUMBER TO R1
(1) ;* JSR      PC,SET.IE ;SET "IE"
(1) ;* RETURN
(1)
(1) SET.IE: MOV      R4,-(SP) ;SAVE R4
(1) 017432 013704 013070 MOV      RPAOR,R4 ;PICKUP ADDRESS OF RHCS1
(1) 017436 010164 000010 MOV      R1,RHCS2(R4) ;SELECT DRIVE

```



```

(1) 017442 011446             MOV     (R4), -(SP)       ; READ RHCS1
(1) 017444 052716 040000      BIS     #BIT14, (SP)     ; SET THE "TRE" BIT OF THE WORD READ
(1) 017450 000316             SWAB   (SP)             ; ADJUST FOR DATO
(1) 017452 112714 000100     MOVB   #BIT06, (R4)     ; SET "IE"
(1) 017456 032764 010000 000010 BIT     #BIT12, RHCS2(R4) ; IS "NEO"=1?
(1) 017464 001002             BNE    1$              ; YES--CLEAR "TRE"
(1) 017466 005726             TST   (SP)+           ; CLEAN OFF THE STACK
(1) 017470 000402             BR     2$
(1) 017472 112664 000001 1$: MOVB  (SP)+, 1(R4)      ; CLEAR "TRE"
(1) 017476 012604 2$: MOV   (SP)+, R4       ; RESTORE R4
(1) 017500 000207             RTS    PC              ; RETURN TO CALLER
(1)
(1)
(1)
(1) 017502 000
(1) 017503 000
(1) 017504 000
(1) 017505 000
(1) 017506 000
(1) 017507 000
(1) 017510 000
(1) 017511 000
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 017512 017574
(1) 017514 017614
(1) 017516 017634
(1) 017520 017654
(1) 017522 017674
(1) 017524 017714
(1) 017526 017734
(1) 017530 017754
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 017532 017574
(1) 017534 017614
(1) 017536 017634
(1) 017540 017654
(1) 017542 017674
(1) 017544 017714
(1) 017546 017734
(1) 017550 017754
(1)
(1)
(1) 017552 017574
(1) 017554 017614
(1) 017556 017634
(1) 017560 017654
(1) 017562 017674
(1) 017564 017714
(1) 017566 017734
(1) 017570 017754
(1) 017572 017774
(1)
(1)

```

; *QUEUE COUNT											
Qcnt:	. BYTE	0	:	DRIVE	0						
	. BYTE	0	:	DRIVE	1						
	. BYTE	0	:	DRIVE	2						
	. BYTE	0	:	DRIVE	3						
	. BYTE	0	:	DRIVE	4						
	. BYTE	0	:	DRIVE	5						
	. BYTE	0	:	DRIVE	6						
	. BYTE	0	:	DRIVE	7						

  

; QUEUE INPUT POINTERS											
QINPT:	. WORD	QDRV0	:	DRIVE	0						
	. WORD	QDRV1	:	DRIVE	1						
	. WORD	QDRV2	:	DRIVE	2						
	. WORD	QDRV3	:	DRIVE	3						
	. WORD	QDRV4	:	DRIVE	4						
	. WORD	QDRV5	:	DRIVE	5						
	. WORD	QDRV6	:	DRIVE	6						
	. WORD	QDRV7	:	DRIVE	7						

  

; QUEUE OUTPUT POINTERS											
QOUTPT:	. WORD	QDRV0	:	DRIVE	0						
	. WORD	QDRV1	:	DRIVE	1						
	. WORD	QDRV2	:	DRIVE	2						
	. WORD	QDRV3	:	DRIVE	3						
	. WORD	QDRV4	:	DRIVE	4						
	. WORD	QDRV5	:	DRIVE	5						
	. WORD	QDRV6	:	DRIVE	6						
	. WORD	QDRV7	:	DRIVE	7						

  

; DRIVE 0 START ADDRESS											
QSTART:	. WORD	QDRV0	:								
; DRIVE 0 STOP ADDRESS & DRIVE 1 START ADDRESS											
QSTOP:	. WORD	QDRV1	:								
	. WORD	QDRV2	:	STOP	DRIVE	1--	START	DRIVE	2		
	. WORD	QDRV3	:	STOP	DRIVE	2--	START	DRIVE	3		
	. WORD	QDRV4	:	STOP	DRIVE	3--	START	DRIVE	4		
	. WORD	QDRV5	:	STOP	DRIVE	4--	START	DRIVE	5		
	. WORD	QDRV6	:	STOP	DRIVE	5--	START	DRIVE	6		
	. WORD	QDRV7	:	STOP	DRIVE	6--	START	DRIVE	7		
	. WORD	QTERM	:	STOP	DRIVE	7					

```

(1) ;DRIVE REQUEST QUEUES
(1)
(1) 017574 000010 QDRV0: .BLKW 10
(1) 017614 000010 QDRV1: .BLKW 10
(1) 017634 000010 QDRV2: .BLKW 10
(1) 017654 000010 QDRV3: .BLKW 10
(1) 017674 000010 QDRV4: .BLKW 10
(1) 017714 000010 QDRV5: .BLKW 10
(1) 017734 000010 QDRV6: .BLKW 10
(1) 017754 000010 QDRV7: .BLKW 10
(1) 017774 017774 QTERM=.
(1)
(1) ;*ROUTINE TO CLEAR ALL OF THE REQUEST QUEUES
(1)
(1) ;*CALL
(1) ;* JSR PC,CLRQUE
(1)
(1) 017774 004037 020236 CLRQUE: JSR R0,SAVR15 ;SAVE R1-R5
(1) 020000 012702 017502 MOV #QCNT,R2 ;ZERO THE QUEUE COUNTS
(1) 020004 005022 CLR (R2)+ ;DRIVES 0 & 1
(1) 020006 005022 CLR (R2)+ ;DRIVES 2 & 3
(1) 020010 005022 CLR (R2)+ ;DRIVES 4 & 5
(1) 020012 005022 CLR (R2)+ ;DRIVES 6 & 7
(1) 020014 012703 000010 MOV #8,R3 ;MOVE THE STARTING
(1) 020020 012701 017552 MOV #QSTART,R1 ;ADDRESS OF THE QUEUE INTO
(1) 020024 012122 1$: MOV (R1)+,(R2)+ ;THE QUEUE INPUT POINTER
(1) 020026 005303 DEC R3
(1) 020030 001375 BNE 1$
(1) 020032 012703 000010 MOV #8,R3 ;MOVE THE STARTING ADDRESS
(1) 020036 012701 017552 MOV #QSTART,R1 ;OF THE QUEUE INTO THE
(1) 020042 012122 2$: MOV (R1)+,(R2)+ ;QUEUE OUTPUT POINTER
(1) 020044 005303 DEC R3
(1) 020046 001375 BNE 2$
(1) 020050 004037 020256 JSR R0,GETR15 ;RESTORE R1-R5
(1) 020054 000207 RTS PC
(1)
(1) ;*EMPTY THE QUEUE SPECIFIED BY R1
(1)
(1) ;*CALL
(1) ;* MOV DRVNUM,R1 ;DRIVE NUMBER TO R1
(1) ;* JSR PC,EMPTYQ
(1)
(1) 020056 105061 017502 EMPTYQ: CLRB QCNT(R1) ;CLEAR NUMBER OF ITEMS IN QUEUE
(1) 020062 006301 ASL R1
(1) 020064 016161 017512 017532 MOV QINPT(R1),QOUTPT(R1) ;SET OUTPUT QUEUE POINTER=INPUT POINTER
(1) 020072 006201 ASR R1
(1) 020074 000207 RTS PC
(1)
(1) ;*ROUTINE TO PUT A REQUEST IN QUEUE
(1)
(1) ;*CALL
(1) ;* MOV #DRVNUM,R1 ;DRIVE NUMBER
(1) ;* MOV #DPB,R2 ;ADDRESS OF PARAMETER BLOCK
(1) ;* JSR R0,DRVQUE ;GO PUT REQUEST IN QUEUE
    
```

```

(1)          ;*      RETURN1          ;RETURN HERE IF QUEUE IS FULL
(1)          ;*      RETURN2          ;RETURN HERE IF REQUEST 'S IN QUEUE
(1)
(1) 020076 122761 000010 017502 DRVQUE: CMPB   #10,QCNT(R1) ;IS QUEUE FULL?
(1) 020104 001421          BEQ     2$          ;BR IF YES-TAKE RETURN1
(1) 020106 105261 017502          INCB   QCNT(R1)      ;INCREMENT QUEUE COUNT
(1) 020112 006301          ASL    R1
(1) 020114 010271 017512          MOV    R2,QINPT(R1) ;PUT THIS REQUEST IN QUEUE
(1) 020120 062761 000002 017512          ADD    #2,QINPT(R1) ;UPDATE THE QUEUE POINTER
(1) 020126 026161 017512 017554          CMP    QINPT(R1),QSTOP(R1) ;TIME TO RESET THE POINTER
(1) 020134 001003          BNE    1$          ;BRANCH IF NO
(1) 020136 016161 017552 017512          MOV    QSTART(R1),QINPT(R1) ;YES--RESET POINTER
(1) 020144 006201          1$: ASR    R1
(1) 020146 005720          TST   (R0)+        ;TAKE RETURN 2
(1) 020150 000200          2$: RTS    R0          ;RETURN TO USER
(1)
(1)          ;*ROUTINE TO GET THE "DPB" ADDRESS OF NEXT REQUEST IN QUEUE
(1)
(1)          ;*CALL
(1)          ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
(1)          ;*      JSR    PC,GETREQ      ;GO GET THE REQUEST
(1)          ;*      RETURN   ;R2="DPB" ADDRESS OF THE REQUEST
(1)          ;*          ;R2=0 IF NO REQUEST IN QUEUE
(1)
(1) 020152 005002          GETREQ: CLR    R2
(1) 020154 105761 017502          TSTB  QCNT(R1)    ;IS THERE ANY REQUEST IN QUEUE?
(1) 020160 001404          BEQ    2$          ;NO---BRANCH
(1) 020162 006301          1$: ASL    R1
(1) 020164 017102 017532          MOV    #QOUTPT(R1),R2 ;PICKUP "DPB" POINTER FOR THIS DRIVE
(1) 020170 006201          ASR    R1
(1) 020172 000207          2$: RTS    PC          ;RETURN TO USER
(1)
(1)          ;*ROUTINE TO "POP" THE REQUEST FROM QUEUE
(1)
(1)          ;*CALL
(1)          ;*      MOV     #DRVNUM,R1      ;DRIVE NUMBER TO R1
(1)          ;*      JSR    PC,POPQUE     ;CALL TO REMOVE REQUEST
(1)          ;*      RETURN   ;R2=ADDRESS OF DPB REMOVED
(1)
(1) 020174 105361 017502          POPQUE: DECB  QCNT(R1) ;DECREMENT QUEUE COUNT
(1) 020200 006301          ASL    R1
(1) 020202 017102 017532          MOV    #QOUTPT(R1),R2 ;GET THE "DPB" POINTER
(1) 020206 062761 000002 017532          ADD    #2,QOUTPT(R1) ;UPDATE THE QUEUE POINTER
(1) 020214 026161 017532 017554          CMP    QOUTPT(R1),QSTOP(R1) ;TIME TO RESET THE POINTER
(1) 020222 001003          BNE    1$          ;NO--BRANCH TO EXIT
(1) 020224 016161 017552 017532          MOV    QSTART(R1),QOUTPT(R1) ;YES--RESET THE POINTER
(1) 020232 006201          1$: ASR    R1
(1) 020234 000207          RTS    PC          ;RETURN TO USER
(1)
(1)          ;*ROUTINES TO SAVE R0-R5 AND R1-R5
(1)
(1)          ;*CALL: SAVROS
(1)          ;*      JSR    R0,SAVROS
(1)          ;*      RETURN   ;R0-R5 IS ON THE STACK
(1)

```

```

(1)          ;*
(1)          ;*CALL: SAVR15
(1)          ;*      JSR      RO,SAVR15
(1)          ;*      RETURN                      ;R1-R5 IS ON THE STACK
(1)          ;*
(1)          ;*UPON RETURN FROM SAVROS AND SAVR15 THE STACK WILL LOOK LIKE:
(1)          ;*
(1)          ;*+12   RO
(1)          ;*+10   R1
(1)          ;*+06   R2
(1)          ;*+04   R3
(1)          ;*+02   R4
(1)          ;*TOP   R5
(1)          SAVROS:
(1)          SAVR15: MOV    R1,-(SP)      ;SAVE RO BY THE JSR INST.
(1)          MOV    R2,-(SP)      ;SAVE R1
(1)          MOV    R3,-(SP)      ;SAVE R2
(1)          MOV    R4,-(SP)      ;SAVE R3
(1)          MOV    R5,-(SP)      ;SAVE R4
(1)          MOV    12(SP),-(SP)   ;SAVE R5
(1)          RTS     RO           ;GET RO
(1)          020236 010146
(1)          020236 010246
(1)          020240 010346
(1)          020242 010446
(1)          020244 010546
(1)          020246 010546
(1)          020250 016646 000012
(1)          020254 000200
(1)          ;*ROUTINES TO RESTORE RO-R5 AND R1-R5
(1)          ;*
(1)          ;*CALL: GETROS
(1)          ;*      JSR      RO,GETROS
(1)          ;*      RETURN                      ;RO-R5 HAVE BEEN RESTORED
(1)          ;*
(1)          ;*CALL: GETR15
(1)          ;*      JSR      RO,GETR15
(1)          ;*      RETURN                      ;R1-R5 HAVE BEEN RESTORED
(1)          ;*
(1)          GETR15: MOV    (SP),14(SP) ;POSITION RO
(1)          GETROS: TST    (SP)+     ;POP RO OF THE JSR OFF OF THE STACK
(1)          MOV    (SP)+,R5        ;RESTORE R5
(1)          MOV    (SP)+,R4        ;RESTORE R4
(1)          MOV    (SP)+,R3        ;RESTORE R3
(1)          MOV    (SP)+,R2        ;RESTORE R2
(1)          MOV    (SP)+,R1        ;RESTORE R1
(1)          RTS     RO           ;RESTORE RO
(1)          020256 011666 000014
(1)          020262 005726
(1)          020264 012605
(1)          020266 012604
(1)          020270 012603
(1)          020272 012602
(1)          020274 012601
(1)          020276 000200
(1)          6621
(1)          6622
(1)          6623
(1)          6624 020300 000000
(1)          6625 000001
(1)          ;THIS BUFFER CONTAINS HEADER/DATA INFORMATION WRITTEN ON CURRENT TRK
(1)          BUFP: 0              ;START OF BUFFER FOR FORMAT (26260 BYTES DEEP)
(1)          .END
    
```





GETDAT	005446	5125	5130	5860#		
GETREQ	020152	6620#				
GETROS	020262	6620#				
GETRIS	020256	6620#				
GNS	= ***** U	4933				
GOOD	002446	5257	5262#			
GSEL	001250	4991	5003#			
HDRCK	002644	5312	5315#			
HDRCK1	002660	5314	5318#			
HDRAD	002616	5307#	5309	5323		
HEDERR	012606	5294#	5342*	5351	6551#	
IOTVEC=	000020	4942#				
IRHCS2	007146	6204#	6205*	6206	6208#	
ISR	015260	6620#				
LA	015122	6620#				
LACNT	013020	6620#*				
LOCCYL	012616	5340#	6080	6161*	6564#	
LOCTRK	012614	5339#	6098	6160*	6563#	
LOOPTS	002516	5263	5267	5274#	5568	
MADRER	010423	5140	5147	6420#		
MAS	011771	5436	5511	6471#		
MAXSEC	012612	5099*	5107*	5266	6033	6555#
MADATA	012067	5609	6487#			
MEDFMT	012332	5359	6503#			
MCCMPT	010777	5367	6438#			
MCLINT	010334	5888	6416#			
MCPENX	013066	6620#				
MCRLF	007574	5834	6376#			
MCYLTK	010741	6079	6171	6434#		
MCYTKO	010272	5870	6410#			
MCYTKI	010324	5875	6414#			
MORNP	007621	5020	6380#			
MORV	011762	5431	5506	6140	6469#	
MORVD	007561	4990	6371#			
MER1	011023	5417	6440#			
MER10	011716	5762	6467#			
MER2	011065	5429	6442#			
MER3	011123	5453	6444#			
MER4	011172	5476	6446#			
MER5	011231	5504	6448#			
MER6	011303	5582	6450#			
MER6A	011332	5655	6452#			
MER6B	011372	5671	6454#			
MER6C	011413	5698	6456#			
MER6D	011434	5710	6458#			
MER7	011556	5722	6461#			
MER8	011606	5736	6463#			
MER9	011656	5750	6465#			
MEXPED	012444	5334	6507#			
MFCMPT	010752	5364	6436#			
MFORMT	007761	5075	6396#			
MFTERM	012226	5725	5738	6497#		
MGDATA	012104	5603	6489#			
MHCYTK	012046	6163	6483#			

MHOCMP	012474	5331	6511#						
MHECK	010004	5072	6398#						
MKDIG	006572	6087	6094	6104	6111#				
MMAXCT	010277	5129	6412#						
MMINCT	010244	5123	6408#						
MMODE	007725	5061	6394#						
MNDLTA	013102	6620#							
MNRPO4	007646	5027	6382#						
MODE	012544	5059*	5079*	5183	5237	5259	5269	5361	6534#
MOFFLN	007577	5014	6378#						
MORMAT	010016	5078	6400#						
MPATD	010632	5168	6428#						
MGUES	007567	5840	6375#						
MRDADR	012000	5456	6473#						
MRDWRD	012010	5461	5489	6475#					
MREAD	012460	5337	6509#						
MREFMT	012307	5658	5687	6501#					
MREGHD	012143	6213	6495#						
MREUNS	012415	5557	5685	6505#					
MRHEC1	012121	5613	6491#						
MRHEC2	012132	5617	6493#						
MSCHK	010704	5189	6432#						
MSD	006144	5954	5957#						
MSEC20	010165	5103	6406#						
MSEC22	010106	5095	6404#						
MSELD	007713	5179	6386#						
MSELP	010530	5163	6423#						
MSFOU	010646	5186	6430#						
MSIZE	010035	5084	6402#						
MSN	012040	6144	6481#						
MSPACE	007564	6177	6373#						
MSTOR	012057	5589	6485#						
MTITLE	007466	4976	6366#						
MTLINT	010371	5915	6418#						
MUNCOR	012254	5559	6499#						
MUNIT	007546	4984	6369#						
MUSDR	007673	5030	6384#						
MWC	012604	5097*	5105*	5235	5632	5936	6550#		
MWRTAD	012020	5479	6477#						
MWRTWD	012030	5484	6479#						
MXDLTA	013100	6620#							
MXLACT	013076	6620#							
MXLNDW	013104	6620#							
M0	001166	4983#	4994	4998	5015	5021	5028	5031	
M1	001366	5008	5059#	5070					
M1A	001470	5073	5083#	5093					
M2	001676	5117	5122#	5126	5141	5148			
M3	001724	5128#	5131						
M4	002052	5150	5162#	5172	5176				
M5	002142	5183#	5372						
MULCNT	012620	5783	6565#						
NUMTST	005314	4993	5171	5812#	5895	5923			
NXD	006624	6124#	6134						
NXDD	007024	6183#	6194						





RD.RP1	017060	6620#												
RD.RP2	017162	6620##												
RD.RP3	017166	6620#												
RD.WRD	017064	5462	5490	6620#										
RESREG	007270	5421	5445	5468	5496	5520	6236#							
RESTR	001156	4940	4974*	4977#	5370	5662	5691	5729	5743	5753	5765	5807		
RESVEC=	000010	4942#												
RETRY	012574	5233*	5666	5673*	5678	5680*	6546#							
RHAS =	000016	6620##												
RHBA =	000004	6620#												
RHCA =	000034	6620#												
RHCC =	000036	6620#												
RHCS1 =	000000	6620#												
RHCS2 =	000010	6620##												
RHDA =	000006	6620#												
RHDB =	000022	6620#												
RHDS1 =	000012	6620#												
RHDT =	000026	6620#												
RHEC1 =	000044	6620#												
RHEC2 =	000046	6620#												
RHER1 =	000014	6620#												
RHER2 =	000040	6620#												
RHER3 =	000042	6620#												
RHLA =	000020	6620#												
RHMR =	000024	6620#												
RHOF =	000032	6620#												
RHREGS	007150	5441	5516	5620	6212#	6252								
RHSN =	000030	6620#												
RHWC =	000002	6620#												
RPADR	013070	4972*	6204	6620##										
RPERRS	012732	6199	6200	6201	6202	6620##								
RPINIT	013106	5003	6620#											
RPTMR	016436	5377	6620#											
RPVEC	013072	4973*	6620#											
RPO4	013476	5111	5240	5252	5307	6620#								
RP6 =	000300	4942#												
RO =%	000000	4942#	4986	4995	4999	5005*	5006	5016*	5017	5022	5024	5063	5065	5067
		5086	5088	5090	5111*	5136*	5137*	5143*	5144*	5149	5153*	5155*	5156*	5157
		5165	5169*	5173	5177	5209*	5212*	5213*	5240*	5252*	5307*	5377*	5387	5389*
		5390*	5391*	5392*	5393	5590*	5594*	5597*	5599*	5600*	5624*	5628*	5631	5632*
		5636*	5639*	5640	5773*	5786	5787	5798*	5799	5801	5804*	5805	5812	5814
		5816*	5824*	5825*	5827*	5863	5879	5882	5901	5906	5909	5929	5932*	5953*
		5957*	5958*	6034*	6047*	6048*	6049*	6050*	6051*	6060*	6061*	6062*	6070	6071*
		6072*	6080*	6083*	6090*	6095*	6096	6098*	6100*	6105*	6106	6113*	6121*	6122*
		6128*	6130*	6131*	6153*	6154*	6188*	6190*	6191*	6240*	6620*			
R1 =%	000001	4942#	5210*	5212	5383	5625*	5626	5629*	5633*	5634	5637*	5772*	5773	6033*
		6053*	6063*	6073*	6081*	6085*	6088*	6092*	6099*	6102*	6111*	6112	6182*	6193*
		6236*	6620*											
R2 =%	000002	4942#	5211*	5213	5388	5846	5860*	5903*	6041*	6046*	6047	6079*	6096*	6097*
		6106*	6112*	6118*	6133*	6241*	6620*							
R3 =%	000003	4942#	5208*	5214*	5215*	5384	5432*	5437*	5457*	5462*	5480*	5485*	5490*	5507*
		5512*	5591*	5592*	5596*	5599	5604*	5605*	5606*	5610*	5614*	5618*	5845*	5852*
		5901*	5929*	5951	5952	5957	6042*	6043*	6048	6052*	6082*	6089*	6113	6119*
		6120*	6121	6124*	6125*	6126*	6127*	6128	6129*	6141*	6145*	6152*	6153	6193*







COMMEN	904#	4942#	
ENDCOM	916#	4942#	
ERRCAL	6620#		
ERROR	4942#	6620	
ESCAPE	1018#	4942#	
MULT	3201#	4942#	
NEWTST	956#	4942#	
POP	1351#	4942#	
PUSH	1343#	4942#	
RPO4.D	3665#	6620	
SCOPE	4942#		
SETUP	738#	4942#	
SKIP	1052#	4942#	
SLASH	856#	4942#	
SPACE	4942#		
STARS	825#	4942#	
TYPBIN	1287#	4942#	
TYPDEC	1257#	4942#	
TYPNUM	1224#	4942#	
TYPOCS	1177#	4942#	
TYPOCT	1140#	4942#	
TYPTXT	1094#	4942#	
SSESCA	1031#	4942#	
SSNEWT	985#	4942#	
SSSKIP	1065#	4942#	
.EQUAT	164#	4917#	4942
.HEADE	39#	4917#	4921
.KT11	300#		
.SETUP	684#		
.SWAHI	81#	4917#	
.SCATC	478#	4917#	4933
.SCMTA	535#		
.SDB2D	3401#		
.SDB2D	3525#		
.SDIV	3303#		
.SEOP	1399#		
.SERRO	1773#		
.SERRT	1928#		
.SMULT	3239#		
.SPOWE	2964#		
.SRAND	3021#		
.SRDE	2660#		
.SRODC	2568#		
.SREAD	2354#		
.SSAVE	2736#		
.SSB2D	3486#		
.SSB2D	3588#		
.SSCOP	1575#		
.SSIZE	3083#		
.SSUPR	3626#		
.STRAP	2837#		
.STYPB	2269#		
.STYPD	2191#		
.STYPE	2016#		

N06

MAINDEC-11-DZRPL-B "RPO4 FORMATTER PROGRAM"  
DZRPLB.P11 CROSS REFERENCE TABLE

MACY11 27(663) 23-JAN-76 10:26 PAGE 43-67

SEQ 0077

.STYPO 2094\*

ADD	5152 5983 6272	5157 6024 6620	5158 6051 6620	5596 6062	5597 6071	5600 6072	5628 6095	5636 6105	5639 6111	5817 6113	5825 6131	5876 6154	5921 6191	5955 6205	5960 6265
ASL	5016	5391	6620												
ASLB	6620														
ASR	6620														
BCC	4996	5138	5145	6620											
BCCS	5174	5986	5913	5954	5959										
BEO	5023	5025	5064	5066	5068	5087	5089	5091	5115	5184	5198	5202	5238	5244	5256
	5260	5263	5267	5270	5275	5311	5316	5319	5357	5362	5527	5530	5533	5536	5543
	5564	5627	5635	5651	5679	5774	5800	5802	5866	5891	5918	5974	6045	6620	
BGE	6054	6064	6074	6620											
BGT	5008	6620													
BIC	5338	5574	5804	5816	6061	6122	6130	6152	6190	6620					
BICB	6620														
BIS	6046	6620													
BISB	6620														
BIT	5274	5328	5368	5414	5426	5450	5473	5501	5526	5529	5532	5535	5542	5554	5579
	5650	5652	5658	5682	5695	5707	5719	5733	5747	5759	6620				
BITB	6620														
BLE	6620														
BLOS	6620														
BUT	5012	6620													
BMI	5116	5593	5813	6084	6091	6101	6620								
BME	4987	5018	5142	5154	5166	5216	5265	5329	5369	5415	5427	5451	5474	5502	5546
	5549	5552	5555	5575	5580	5653	5667	5669	5683	5696	5708	5720	5734	5748	5760
	5788	5790	5806	5854	5864	5880	5883	5898	5907	5910	5926	5969	6134	6194	6620
BPL	5245	5257	5312	5344	5419	5443	5466	5494	5518	5561	5644	5660	5675	5689	5701
	5713	5727	5741	5785	5797	5815	6620								
BR	4977	4991	4994	4998	5015	5021	5028	5031	5070	5073	5076	5093	5101	5113	5117
	5126	5131	5141	5148	5150	5170	5172	5176	5187	5242	5247	5254	5261	5271	5273
	5276	5279	5283	5309	5314	5323	5360	5365	5595	5630	5638	5776	5871	5894	5902
CLR	5930	5956	5961	5976	5981	6086	6093	6103	6123	6264	6271	6620	6620	5852	5867
	4988	5059	5106	5127	5151	5195	5196	5232	5233	5294	5300	5590			
	5942	5977	6026	6049	6050	6081	6088	6099	6258	6620					
CLRB	5108	5234	5935	5938	5978	6620									
CMP	4986	4995	5022	5024	5063	5065	5067	5086	5088	5090	5165	5173	5201	5264	5315
	5318	5563	5678	5787	5799	5801	5805	5812	5814	5863	5879	5892	5885	5906	5909
	5912	5973	6620												
CMPB	5266	5545	5548	6620											
CJM	5199	5200	6620												
DEC	5153	5214	5215	5577	5629	5637	5789	5853	5897	5925	5971	6053	6063	6073	6133
	6193	6620													
DECB	6620														
EMT	4942														
HALT	4933	5345	5420	5444	5467	5495	5519	5562	5645	5661	5676	5690	5702	5714	5728
	5742	5752	5764												
INC	5272	5320	5322	5572	5594	5673	5680	5975	5979	6025	6052	6060	6085	6092	6102
	6620														
INCB	5972	6620													
IOT	4942														
JMP	4938	4940	5370	5372	5393	5528	5531	5534	5537	5538	5544	5547	5550	5553	5568
	5662	5691	5729	5743	5753	5765	5807	6620							
JSR	4975	4983	4995	4989	4993	4997	5003	5013	5019	5026	5029	5060	5062	5069	5071



	5074	5077	5083	5085	5092	5094	5102	5111	5122	5125	5128	5130	5139	5146	5162
	5164	5167	5171	5175	5178	5185	5188	5190	5220	5221	5222	5240	5246	5252	5258
	5268	5277	5295	5296	5307	5313	5317	5321	5330	5332	5333	5335	5336	5341	5358
	5362	5366	5377	5416	5421	5428	5430	5433	5434	5435	5438	5439	5440	5441	5445
	5452	5454	5455	5458	5459	5460	5463	5464	5468	5475	5477	5478	5481	5482	5483
	5486	5487	5488	5491	5492	5496	5503	5505	5508	5509	5510	5513	5514	5515	5516
	5520	5556	5558	5565	5581	5583	5587	5588	5598	5601	5602	5607	5608	5611	5612
	5615	5616	5619	5620	5654	5656	5657	5670	5672	5684	5686	5697	5699	5709	5711
	5721	5723	5724	5735	5737	5739	5749	5751	5761	5763	5775	5803	5819	5826	5828
	5833	5839	5851	5855	5861	5862	5869	5874	5878	5884	5887	5895	5899	5904	5905
	5911	5914	5923	5927	5980	5982	6040	6059	6069	6087	6094	6104	6132	6139	6142
	6143	6146	6147	6155	6162	6164	6169	6170	6176	6192	6212	6215	6216	6218	6219
MOV	6221	6222	6224	6225	6227	6228	6230	6231	6247	6251	6252	6260			
	4969	4970	4971	4972	4973	4974	4978	4999	5004	5005	5079	5096	5077	5098	5099
	5104	5105	5107	5124	5136	5143	5149	5155	5169	5177	5203	5204	5208	5209	5210
	5211	5212	5213	5235	5236	5297	5298	5299	5301	5340	5342	5371	5376	5383	5384
	5385	5386	5387	5388	5389	5390	5392	5432	5437	5457	5462	5480	5485	5490	5507
	5512	5573	5576	5578	5591	5599	5604	5606	5610	5614	5618	5624	5625	5631	5632
	5633	5640	5641	5772	5782	5783	5791	5798	5824	5827	5845	5846	5860	5872	5873
	5892	5893	5901	5903	5919	5920	5929	5936	5937	5940	5941	5945	5946	5951	5952
	5957	6023	6033	6034	6041	6042	6047	6048	6079	6080	6082	6089	6118	6121	6128
	6141	6145	6153	6161	6182	6188	6199	6200	6201	6202	6203	6204	6206	6214	6217
	6220	6223	6226	6229	6236	6237	6238	6239	6240	6241	6257	6260	6261	6262	6263
	6266	6268	6269	6270	6273	6620									
MOVB	5100	5109	5110	5239	5251	5306	5339	5642	5773	5786	5934	5939	6096	6097	6098
	6106	6112	6160	6620											
NEG	5156														
NOP	5943	5944													
RESET	4968	4979													
ROL	6119	6120	6124	6125	6126	6127	6183	6184	6185	6186	6187	6620			
ROLD	6620														
ROR	6129	6189													
RTI	5378	5422	5446	5469	5497	5521	6620								
RTS	5346	5566	5646	5677	5681	5703	5715	5777	5792	5808	5818	5820	5829	5835	5841
	5847	5856	5877	5881	5889	5896	5900	5908	5916	5922	5924	5928	5947	5962	5970
	5984	6027	6035	6055	6065	6075	6107	6114	6135	6148	6156	6165	6172	6178	6195
	6207	6232	6242	6253	6274	6620									
SUB	5137	5144	5592	5605	5953	5958	6083	6090	6100	6620					
SWAB	6043	6620													
TST	5017	5114	5183	5197	5237	5243	5255	5259	5262	5269	5310	5343	5351	5361	5418
	5442	5465	5493	5517	5551	5560	5567	5626	5634	5643	5659	5666	5674	5688	5700
	5712	5726	5740	5865	5890	5917	5968	6044	6070	6259	6267	6620			
TSTB	5006	5784	5796	6620											
.ASCII	6366	6420	6423	6424	6425	6458									
.ASCIZ	6367	6369	6371	6373	6375	6376	6378	6380	6382	6384	6386	6394	6396	6398	6400
	6402	6404	6406	6408	6410	6412	6414	6416	6418	6421	6426	6428	6430	6432	6434
	6436	6438	6440	6442	6444	6446	6448	6450	6452	6454	6456	6459	6461	6463	6465
	6467	6469	6471	6473	6475	6477	6479	6481	6483	6485	6487	6489	6491	6493	6495
	6497	6499	6501	6503	6505	6507	6509	6511							
.BLKW	6620														
.BYTE	6599	6600	6601	6602	6605	6606	6620								
.ENABL	4	4916													
.END	6625														
.ENDC	4921	4933	4942	4964	5011	5055	5231	5282	5288	5305	5356	5586	5985	6018	6250

.EQUIV	6365	6393	6524	6562	6620										
.EVEN	4942														
.IF	6530														
.IFF	4921	4933	4942	4961	5007	5033	5223	5278	5284	5302	5350	5584	5967	5986	6248
.IIF	5276	6388	6513	6557	6620										
.LIST	4942	5009	5280	5352											
.MACRO	4921	4933													
	2	3664	4920	4933	4942										
	39	81	164	300	478	535	684	738	825	856	904	916	956	985	1018
	1031	1052	1065	1094	1140	1177	1224	1257	1287	1343	1351	1399	1575	1773	1928
	2016	2094	2191	2269	2354	2568	2660	2736	2837	2964	3021	3083	3201	3239	3303
	3401	3486	3525	3588	3626	3665	6620								
.MCALL	4917	4942													
.MLIST	1	3	4911	4915	4933	4942									
.REPT	4933														
.SBTTL	4923	4933	4942	4957	4966	5767	6620								
.TITLE	4921														
.WORD	4933	4959	4960	6603	6604	6607	6608	6609	6620						

ERRORS DETECTED: 0

\*DZRPLB,DZRPLB/CRF=SYSMAC.SMA,RPO4.009,DZRPLB  
RUN-TIME: 42 44 4 SECONDS  
CORE USED: 32K

