

LPA/LPS

DIAGNOSTIC TEST 2
MD-11-DRLPI-A

EP-DRLPI-A-DL

COPYRIGHT © 1978

FICHE 1 OF 1

MAR 1978

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DRLPI-A-D
PRODUCT NAME: LPA/LPS DIAGNOSTIC TEST 2
DATE: JANUARY 1978
MAINTAINER: DIAGNOSTIC GROUP

COPYRIGHT (C)1978
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE
COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE
COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT
BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR
USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS.
TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN 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 IN EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

1. ABSTRACT

SEQ 0002

THIS DIAGNOSTIC TESTS AND EXERCISES THE "LPS". WHEN STARTED, IT WILL TYPE OUT THE PROGRAM TITLE. A SENTENCE IS THEN TYPED GIVING THE LETTER DESIGNATORS TO BE TYPED TO RUN ANY ONE OF THE FOUR (4) SEPERATE TESTS, OF WHICH THIS PROGRAM IS COMPRISED. THE PROGRAM THEN TYPES A 'CR.' AND THEN WAITS IN A KEYBOARD MONITOR MODE FOR A LETTER TO BE TYPED. ALTHOUGH THESE TESTS MAY BE RUN IN ANY ORDER IT IS IMPERATIVE THAT THE 'LOGIC' TESTS ARE RUN FIRST AND PROVED FULLY OPERATIONAL. OVER THE PROGRAM AS POSSIBLE VIA THE TELETYPE. TYPING A 'IC' (OBTAINED VIA TYPING THE 'CNTR' AND 'C' KEYS SIMULTANEOUSLY) WHILE RUNNING ANY TEST WILL ENABLE THE PROGRAM TO RETURN TO THE KEYBOARD MONITOR AND AWAIT A NEW LETTER DESIGNATOR TO BE TYPED. TYPING A 'IA' WHILE IN MONITOR MODE WILL ENABLE THE LETTER DESIGNATORS TO BE RETYPED. IF RUNNING ON A NON-SWITCH REGISTER CPU, TYPING A 'CTRL G' WILL ALLOW THE CHANGING OF A SOFTWARE SWITCH REGISTER.

THIS PROGRAM IS A MODIFIED VERSION OF "MD-11-DZLPD-C". IT WAS MODIFIED TO ENABLE THE OPERATOR TO CHECK OUT THE LPS11 OPTION WHEN IT IS ON THE LPA11-KX I/O BUS. NO RECABLING IS NEEDED. SOME TESTS DONE IN THE ORIGINAL DIAGNOSTIC SUCH AS ARBITRATION TEST, WERE DELETED AS THEY COULD NOT BE CHECKED. IF THIS DIAGNOSTIC DOESN'T FIND A SUSPECTED PROBLEM, YOU MAY HAVE TO RUN "MD-11-DZLPD-C". YOU SHOULD RUN "MD-11-DRLPA" BEFORE RUNNING THIS DIAGNOSTIC. PLEASE READ SECTION 16.

2. REQUIREMENTS (EQUIPMENT)

- A. PDP-11 COMPUTER WITH 16K OF MEMORY.
 - B. TELETYPE (OR EQUIVALENT)
 - C. LPS11 OPTION BOX WITH:
 - LPSKW REAL TIME CLOCK CONTROL AND/OR
 - LPSDR DIGITAL INPUT-OUTPUT CONTROL AND/OR
 - LPSVC POINT PLOT SCOPE CONTROL
- LPSDRA IS SUPPORTED BY A SEPARATE DIAGNOSTIC (MD-11-DRLPJ).

3. LOADING PROCEDURE

- A. USE STANDARD PROCEDURE FOR LOADING BINARY TAPES.

4. STARTING PROCEDURE

THE STARTING ADDRESS OF THIS DIAGNOSTIC IS 200.

5. CONSOLE SWITCH SETTINGS

- A. ALL SWITCHES SHOULD BE DOWN (0) WHEN THE PROGRAM IS STARTED.
- B. REFER TO THE INDIVIDUAL TEST DESCRIPTIONS FOR APPLICABLE CONSOLE SWITCH SETTINGS
- C. REFER TO 15. FOR SOFTWARE SWITCH REGISTER OPERATION.

* TYPE 'CARRIAGE RETURN' (CR) TO TERMINATE ALL INPUT DATA.

6. CLOCK LOGIC TEST

- A. THE "CLOCK LOGIC TEST" IS DESIGNED TO TEST INDIVIDUAL BITS IN THE CONTROL AND STATUS REGISTERS, COUNT PRESET BUFFER AND COUNTER ALONG WITH PROPER OPERATION UNDER INTERRUPT CONTROL MODE CONTROL.

B. STARTING SEQUENCE

1. TYPE 'A' TO RUN THE CLOCK LOGIC TEST.
2. THE PROGRAM WILL THEN EXECUTE THE CLOCK LOGIC TEST.

C. CONTROL SWITCHES

1. TYPING 'C' AT ANY TIME WILL ENABLE THE PROGRAM TO EXIT 'CLOCK LOGIC' TEST AND RETURN TO THE MONITOR.

2. CONTROL SWITCHFUNCTIONS

CONSOLE SW11=0	NORMAL RUN (2048 PASSES/TEST)
CONSOLE SW11=1	SUPPRESS SUBPROGRAM ITERATIONS
CONSOLE SW13=0	PRINT ERROR MESSAGE
CONSOLE SW13=1	INHIBIT ERROR MESSAGE
CONSOLE SW14=0	INHIBIT SCOPE MODE
CONSOLE SW14=1	RUN SCOPE MODE
CONSOLE SW15=0	CONTINUE AFTER TYPING ERROR
CONSOLE SW15=1	HALT ON ERROR

D. ERRORS

ON ENCOUNTERING AN ERROR (DATA SWITCHES DOWN) THE ERROR ADDRESS AND THE CONTENTS OF THE CLOCK STATUS AND CLOCK PRESET BUFFER ARE TYPED OUT.

E. RESTRICTIONS

NO CONNECTIONS SHOULD BE MADE TO THE SCHMITT TRIGGER.

F. TEST TIME

IT TAKES APPROXIMATELY 200 SECONDS TO RUN THE CLOCK LOGIC TEST AND TYPE "END PASS".

7. DIGITAL INPUT-OUTPUT

- A. THIS TEST IS DESIGNED TO TEST THE LPSDR <DIGITAL INPUT-OUTPUT> LOGIC. FOR THIS TEST THE EXTERNAL JUMPER CABLE MUST BE INSTALLED TO TEST THE DATA INPUT/OUTPUT REGISTERS AND THE CONTROL SIGNALS. IF THIS EXTERNAL JUMPER CABLE IS NOT INSTALLED, ONLY A MINIMAL LOGIC TEST CAN BE PERFORMED.

B. STARTING SEQUENCE

1. TYPE 'B' TO RUN THE 'DIGITAL I/O LOGIC' TEST.
2. THE PROGRAM WILL THEN EXECUTE THE DIGITAL I/O LOGIC TEST.

C. CONTROL SWITCHES

1. TYPING ↑C WILL CAUSE THE PROGRAM TO EXIT THE
'DIGITAL I/O LOGIC' TEST AND RETURN TO THE MONITOR.

E01

2. CONSOLE SWITCH FUNCTION

CONSOLE SW06=0*	EXTERNAL JUMPER CABLE CONNECTED
CONSOLE SW06=1*	EXTERNAL JUMPER CABLE NOT CONNECTED
	* = ACTIVATE SWITCH BEFORE SELECTING TEST
CONSOLE SW11=0	NORMAL RUN
CONSOLE SW11=1	SUPPRESS SUBPROGRAM ITERATIONS
CONSOLE SW13=0	PRINT ERROR MESSAGES
CONSOLE SW13=1	INHIBIT ERROR MESSAGES
CONSOLE SW14=0	INHIBIT SCOPE MODE
CONSOLE SW14=1	SCOPE MODE
CONSOLE SW15=0	CONTINUE AFTER TYPING ERROR
CONSOLE SW15=1	HALT ON ERROR

D. LOGIC ERRORS

ON ENCOUNTERING AN ERROR (DATA SWITCHES DOWN) THE ERROR
ADDRESS AND THE CONTENTS OF THE DIGITAL I/O STATUS, OUTPUT
AND INPUT REGISTERS ARE TYPED OUT.

E. RESTRICTIONS

NONE.

F. TEST TIME

IT TAKES APPROXIMATELY 5 SECONDS TO RUN THE DIGITAL I/O
LOGIC TEST AND RING THE TELETYPE BELL.

B. POINT PLOT SCOPE LOGIC TEST

A. THIS TEST IS DESIGNED TO TEST THE LPSVC SCOPE CONTROL LOGIC.
ALL USEABLE BITS OF THE STATUS REGISTER ARE TESTED.

B. STARTING SEQUENCE

1. TYPE 'C' TO RUN THE SCOPE LOGIC TEST.
2. THE PROGRAM WILL THEN EXECUTE THE SCOPE LOGIC TEST.

C. CONTROL SWITCHES

1. TYPING 'C' AT ANY TIME WILL ENABLE THE PROGRAM TO EXIT
AND RETURN TO THE MONITOR.

2. CONSOLE SWITCHES FUNCTION

CONSOLE SW05=0*
CONSOLE SW05=1*

611/613 NOT CONNECTED
611/613 CONNECTED
* = ACTIVATE SWITCH BEFORE SELECTING TEST

D. LOGIC ERRORS

ON ENCOUNTERING AN ERROR (DATA SWITCHES DOWN) THE ERROR ADDRESS
AND THE CONTENTS OF THE VC STATUS, X AXIS AND Y AXIS
REGISTERS ARE TYPED OUT ON THE TELETYPE.

E. RESTRICTIONS

IF 611/613 STORAGE SCOPE IS CONNECTED, IT MUST HAVE POWER ON.

F. TEST TIME

IT TAKES APPROXIMATELY 7 SECONDS TO RUN THE SCOPE LOGIC TEST.

9. POINT PLOT VISUAL DISPLAY TEST

A. THIS TEST IS DESIGNED TO AID IN THE ADJUSTING AND ALIGNMENT OF THE VR14/20 OR 611/613 SCOPE ON THE LPSVC DISPLAY CONTROL.

B. STARTING SEQUENCE

1. TYPE 'D' TO RUN THE VISUAL DISPLAY TEST.
2. THE PROGRAM WILL THEN EXECUTE THE VISUAL DISPLAY TEST.

C. CONTROL SWITCHES

1. TYPING 'C' AT ANY TIME WILL ENABLE THE PROGRAM TO EXIT AND RETURN TO THE MONITOR.

2. CONSOLE SWITCHES FUNCTION

CONSOLE SW08=0	LOOP THRU DISPLAY TEST
CONSOLE SW08=1	SELECT TEST IN SW 00-02
CONSOLE SW04=0*	PLOT CHARACTERS IN FAST INTENSIFY MODE (VR14/20)
CONSOLE SW04=1*	PLOT CHARACTERS IN NORMAL INTENSIFY MODE (611/613)
	* = ACTIVATE SWITCH BEFORE SELECTING TEST
CONSOLE SW00-02=0	DISPLAY A HORIZONTAL LINE
CONSOLE SW00-02=1	DISPLAY A VERTICAL LINE
CONSOLE SW00-02=2	DISPLAY A SQUARE
CONSOLE SW00-02=3	DISPLAY A "X"
CONSOLE SW00-02=4	DISPLAY CHARACTER SET
CONSOLE SW00-02=5	DISPLAY CHANNEL TEST (VR14/VR20)
CONSOLE SW00-02=6	DISPLAY COLOR PATTERN (VR20)
CONSOLE SW00-02=7	DISPLAY ERASE AND PHOSPOR (611/613)

D. ERRORS

NO PROVISIONS ARE MADE FOR LOGIC ERRORS. THE ONLY ERRORS IN THIS TEST ARE CHECKED VISUALLY.

E. RESTRICTIONS

IF VR14/VR20, CHANNEL SWITCH MUST BE SET TO "1 & 2" POSITION.
 IF VR20, COLOR SWITCH MUST BE SET IN THE REMOTE POSITION.
 IF 611/613, POWER MUST BE APPLIED.

F. EXECUTION TIME

IT TAKES APPROXIMATELY 90 SECONDS TO THIS TEST.

10. VISUAL DISPLAY TEST DESCRIPTIONS

SEQ 0007

DISPLAY HORIZONTAL LINE

A HORIZONTAL LINE IS DISPLAYED ON THE SCOPE BY INITIALLY SETTING THE X AND Y DAC'S TO ZERO AND THEN INCREMENTING THE X VALUE WHILE HOLDING THE Y VALUE AT ZERO. THE POINTS ARE DISPLAYED USING THE DISPLAY INTERRUPT ENABLED.

DISPLAY VERTICAL LINE

A VERTICAL LINE IS DISPLAYED ON THE SCOPE IN THE SAME MANNER AS FOR A HORIZONTAL LINE EXCEPT NOW THE Y VALUE IS INCREMENTED WHILE HOLDING THE X VALUE AT ZERO.

DISPLAY SQUARE

A SQUARE IS DISPLAYED BY INITIALLY SETTING THE X AND Y VALUES TO NEGATIVE FULL SCALE, THEN X IS INCREMENTED TO POSITIVE FULL SCALE (BOTTOM LINE) THEN Y IS INCREMENTED TO POSITIVE FULL SCALE (RIGHT LINE) THEN X IS DECREMENTED TO NEGATIVE FULL SCALE (TOP LINE) AND FINALLY Y IS DECREMENTED TO NEGATIVE FULL SCALE (LEFT LINE). MODE 01 (INTENSIFY ON LOADING X) AND MODE 10 (INTENSIFY ON LOADING Y) ARE USED.

DISPLAY X

AN X IS DISPLAYED BY INITIALLY SETTING THE X AND Y VALUES TO NEGATIVE FULL SCALE AND THEN INCREMENTING BOTH TO POSITIVE FULL SCALE (LOWER LEFT TO UPPER RIGHT DIAGONAL) THEN X IS RESET TO NEGATIVE FULL SCALE, Y REMAINS AT POSITIVE FULL SCALE AND THEN X IS INCREMENTED WHILE Y IS DECREMENTED UNTIL BOTH REACH FULL SCALE AGAIN (UPPER LEFT TO LOWER RIGHT DIAGONAL). MODE 01 (INTENSIFY ON LOADING X) IS USED.

11. ADDITIONAL STARTING ADDRESSES

SEQ 0008

INCLUDED IN THIS PROGRAM ARE SEVERAL 'MINI' TESTS TO AID IN THE CHECKING OF THE UNIQUE NON-PROGRAMABLE HARDWARE TO THE LPS11 OPTION BOX.

SA --	TEST DESCRIPTION -----
204	MANUAL SA OF THE CLOCK LOGIC TEST
210	MANUAL SA OF THE DIGITAL I/O LOGIC TEST
214	MANUAL SA OF THE SCOPE LOGIC TEST
220	MANUAL SA OF THE VISUAL SCOPE TEST
224	MINI-TEST OF SCHMITT TRIGGER #1
230	MINI-TEST OF SCHMITT TRIGGER #2
234	MINI-TEST OF CLOCK OVERFLOW
240	MINI-TEST OF RELAYS

12. OPERATOR VARIABLE LOCATIONS

LOCATION 1000 CONTAINS THE LPS STARTING DEVICE ADDRESS.
 LOCATION 1002 CONTAINS THE LPS STARTING DEVICE VECTOR.
 LOCATION 1004 CONTAINS THE LPS A TO D BR LEVEL.
 LOCATION 1006 CONTAINS THE LPS CLOCK BR LEVEL.
 LOCATION 1010 CONTAINS THE LPS DIGITAL I/O BR LEVEL.
 LOCATION 1012 CONTAINS THE LPS DISPLAY BR LEVEL.
 LOCATION 1014 CONTAINS THE DELAY CONSTANT FOR MEMORY
 AND/OR CPU SPEED (1 IF AN 11/20, 2 IF AN 11/45 ETC.).
 LOCATION 1016 CONTAINS THE TTY FILLER COUNT.
 LOCATION 1020 CONTAINS THE TTY FILLER CHARACTER.
 LOCATION 170 CONTAINS THE SOFTWARE SWITCH REGISTER VALUE.
 LOCATION 172 CONTAINS THE SOFTWARE DISPLAY REGISTER VALUE.

13. MISC. INFORMATION

IF THE PROGRAM WAS LOADED BY ACT-11 OR DDP, THE CLOCK LOGIC, SCOPE LOGIC AND SCOPE VISUAL WILL BE RUN.
 THIS PROGRAM DOES NOT SUPPORT THE "APT" HOOKS.

14. MINI-TEST DESCRIPTIONS

SCHMITT TRIGGER #1

THE PURPOSE OF THIS MINI-TEST IS TO QUICK VERIFY THE OPERATION OF SCHMITT TRIGGER #1 ON THE CLOCK LOGIC MODULE. THIS IS DONE BY THE OPERATOR ROTATING THE THRESHOLD KNOB FROM END TO END OR CHANGING THE SLOPE SWITCH BETWEEN + OR -. THE PROGRAM WILL RING THE TTY BELL AND UPDATE THE NUMBER IN THE LPS DISPLAY LEDS UPON EACH SCHMITT TRIGGER #1 FLAG. ROTATING SCHMITT TRIGGER #2 THRESHOLD KNOB OR SWITCH SHOULD NOT CAUSE SCHMITT TRIGGER #1 TO FIRE.

SCHMITT TRIGGER #2

THE PURPOSE OF THIS MINI-TEST IS TO QUICK VERIFY THE OPERATION OF SCHMITT TRIGGER #2 ON THE CLOCK LOGIC MODULE. THIS IS DONE BY THE OPERATOR ROTATING THE THRESHOLD KNOB FROM END TO END OR CHANGING THE SLOPE SWITCH BETWEEN + OR -. THE PROGRAM WILL RING THE TTY BELL AND UPDATE THE NUMBER IN THE LPS DISPLAY LEDS UPON EACH SCHMITT TRIGGER #2 FLAG. ROTATING SCHMITT TRIGGER #1 THRESHOLD KNOB OR SWITCH SHOULD NOT CAUSE SCHMITT TRIGGER #2 TO FIRE.

CLOCK OVERFLOW

THE PURPOSE OF THIS MINI-TEST IS TO VERIFY THE OUTPUT OF THE CLOCK OVERFLOW LOGIC TO THE FRONT PANNEL. THE CLOCK IS ENABLED TO RUN AND OVERFLOW AT A FAST RATE. THE OUTPUT MUST BE VERIFIED WITH THE USE OF AN OSCILLOSCOPE.

RELAY TEST

THE PURPOSE OF THIS MINI-TEST, IS TO ALLOW THE OPERATOR TO VERIFY THE PROPER OPERATION OF BOTH RELAYS. THIS IS ACCOMPLISHED BY SWITCHING THE RELAYS AT A SLOW RATE TO ALLOW THE OPERATOR TO CHECK THE CONTINUITY OF THE RELAY CONTACTS.

15. SOFTWARE SWITCH REGISTER OPERATION

THE PROGRAM SUPPORTS NON-SWITCH REGISTER CPU TYPES. THIS IS ACCOMPLISHED BY TYPING A "CTRL G". THE RESPONSE WILL REPORT THE OLD VALUE AND WAIT FOR A NEW VALUE. THE OPERATOR NOW INPUTS THE NEW VALUE AND TERMINATES IT WITH A "CR". IF THE OPERATOR TYPES A "CR" WITH NO INPUT, THE SOFTWARE SWITCH REGISTER IS SET TO 0. UPON TERMINATING, THE PROGRAM WILL RESUME THE APPROPATE TEST.

16. LPA11 (SYSTEM) DIAGNOSTIC SUMMARY

SEQ 0010

DIAGNOSTICS FOR THE LPA11 ARE WRITTEN AT THREE LEVELS: (1) TOTAL PDP-11 SYSTEM, (2) LPA11 SYSTEM; AND, (3) LPA11 OPTIONS.

LEVEL 1, IS DESIGNED TO ISOLATE A FAILURE TO THE LPA11 SYSTEM. ALL OPTIONS ON THE PDP-11 ARE EXERCISED.

LEVEL 2 DIAGNOSTICS ISOLATE A FAILURE TO THE INDIVIDUAL OPTION WITHIN THE LPA11. THE LEVEL 2 DIAGNOSTIC IS MD-11-DRLPA. WHEN THE USER RUNS DRLPA HE CAN GENERALLY TELL WHICH OPTION DIAGNOSTIC (LEVEL 3) TO RUN NEXT. M8254 AND M8200-YC ERRORS MAY "LOOK" ALIKE AND DRLPA MAY NOT BE ABLE TO DISTINGUISH BETWEEN THEM. ARBITRATION ERRORS WILL NOT BE DETECTED BY THIS DIAGNOSTIC.

LEVEL THREE DIAGNOSTICS AID IN DETERMINING IF THE ERROR WAS IN FACT ON THE OPTION THE DRLPA SPECIFIED. THE USER MAY "LOOP" ON THE ERROR. WITHIN LEVEL THREE, THERE ARE TWO GROUPS OF DIAGNOSTICS. THE FIRST GROUP REQUIRES NO "EXTRA" WORK BY THE USER IN ORDER TO RUN. GROUP "A" DIAGNOSTICS DO NOT CHECK ARBITRATION, AND REQUIRE EXTRA TIME FOR EXECUTION. THE SECOND GROUP (GROUP "B") REQUIRES THAT THE USER RECONFIGURE THE PDP-11 SYSTEM. THIS RECONFIGURATION INVOLVES CABLING THE UNIBUS TO THE LPA'S I/O BUS.

THE DIAGNOSTIC FOR THE M8254 FALLS INTO THE GROUP "B" CATEGORY.

THE LPA11-KX DIAGNOSTIC KIT WILL INCLUDE:

SEQ 0011

<u>OPTION</u>	<u>GROUP</u>	<u>DIAG. #</u>	<u>DIAG. TITLE</u>
LPA11-KX	LEVEL 2	MD-11-DRLPA	LPA11-K SYSTEM DIAG.
M8254	"B"	MD-11-DRLPN	M8254 (IPBM) DIAG.
AA11-K	A	MD-11-DRLPB	AA11-K DIAG.
	B	MD-11-DZAAC	AA11-K DIAG.
AR11	A	MD-11-DRLPC	LPA/AR11 DIAG. #1
	A	MD-11-DRLPD	LPA/AR11 DIAG. #2
	A	MD-11-DRLPE	LPA/AR11 DIAG. #3
	B	MD-11-DZARA	AR11 DIAG. #1
	B	MD-11-DZARB	AR11 DIAG. #2
	B	MD-11-DZARC	AR11 DIAG. #3
DR11-K	A	MD-11-DRLPF	LPA/DR11-K DIAG.
	B	MD-11-DZDRG	DR11-K DIAG.
KW11-K	A	MD-11-DRLPG	LPA/KW11-K DIAG.
	B	MD-11-DZKWK	KW11-K DIAG.
LPS11	A	MD-11-DRLPH	LPA/LPS11 DIAG. #1
	A	MD-11-DRLPI	LPA/LPS11 DIAG. #2
	A	MD-11-DRLPJ	LPA/LPS11 DIAG. #3
	B	MD-11-DZLPC	LPS11 DIAG. #1
	B	MD-11-DZLPD	LPS11 DIAG. #2
	B	MD-11-DZLPI	LPS11 DIAG. #3
AD11-K	A	MD-11-DRLPK	LPA/AD11-K DIAG.
	B	MD-11-DZADL	AD11-K DIAG.
M8200-YC	B	MD-11-DZLPL	LPA/M8200-YC BASIC MICRO-CPU R/W TEST
	B	MD-11-DZLPM	LPA/M8200-YC JMP+ROM READ TEST

MO1

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPA.MAC

MACY11 27(654) 15-DEC-77 08:35 PAGE 1

SEQ 0012

.REM [

LPA.MAC

WELCOME, THIS DIAGNOSTIC IS ONE IN A SERIES OF DIAGNOSTIC
DESIGNED IN ORDER TO AID YOU IN TESTING THE LPA-11XX OPTION.
I HOPE THAT YOU HAVE READ THE DOCUMENTATION SECTION OF THIS
DIAGNOSTIC. IF YOU HAVE, YOU KNOW ABOUT ALL OF THE DIAGNOSTICS
THAT ARE AVAILABLE FOR TESTING THE LPA SYSTEM.

GOOD LUCK !

[
.GLOBL DRLPX2

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

NO1

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 2

SEQ 0013

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

.TITLE LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
.ENABL AMA
.LIST ME
.NLIST MC,MD,CND

.REM !

THIS IS A LIST OF TESTS DELETED FROM THIS DIAGNOSTIC.
THESE TESTS COULD NOT BE DONE THROUGH THE LPA-11.

TEST THAT THE SCHMITT TRIGGER INTERRUPTS AT
LEVEL INDICATED -1
TEST THAT IF PRIORITY IS LOWERED AGAIN NO INTERRUPT
SHOULD OCCUR
TEST THAT CLOCK INTERRUPTS AT LEVEL INDICATED -1 (MODE 0)
TEST THAT THE CLOCK DOES NOT INTERRUPT AT LEVEL INDICATED
TEST THAT THE CLOCK DOES NOT INTERRUPT AT LEVEL INDICATED +1
TEST THAT THE DIGITAL I/O DOES NOT INTERRUPT
TEST THAT THE INPUT CAN INTERRUPT
TEST THAT THE OUTPUT DOES INTERRUPT
TEST FOR INTERRUPT FROM DIGITAL I/O ON LEVEL
INDICATED -1
TEST FOR NO INTERRUPT FROM DIGITAL I/O ON LEVEL INDICATED
TEST THAT THE LPSVC DOES NOT INTERRUPT
TEST THAT THE DISPLAY DOES INTERRUPT AT LEVEL
INDICATED -1
TEST THAT THE DISPLAY DOES NOT INTERRUPT AT LEVEL
INDICATED
TEST 1MHZ REPEATABILITY
TEST 100KHZ REPEATABILITY
TEST 10KHZ REPEATABILITY
TEST 1KHZ REPEATABILITY
TEST 100HZ REPEATABILITY
TEST LINE REPEATABILITY
TEST THAT THERE IS A TIME DIFFERENCE BETWEEN FAST
INTENSIFY AND NOT INTENSIFY
PHOSPHOR AND ERASE TEST

!

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010

BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10

85	000004	BIT2=4	
86	000002	BIT1=2	
87	000001	BIT0=1	
88			
89		;SWITCH REGISTER DEFINITIONS AND FUNCTIONS:	
90			
91	100000	SW15=100000	;=1, HALT ON ERROR
92	040000	SW14=40000	;=1, LOOP ON CURRENT TEST
93	020000	SW13=20000	;=1, SUPPRESS ERROR TYPEOUT
94	010000	SW12=10000	
95	004000	SW11=4000	;=1, SUPPRESS 'SUBPROGRAM' ITERATIONS
96	002000	SW10=2000	;=1, FORCE TYPEOUT (REPEATIBILITY)
97	001000	SW09=1000	
98	000400	SW08=400	;LPSVC SELECT TEST IN SR 0-2
99	000200	SW07=200	
100	000100	SW06=100	;=1, LPSDR JUMPER NOT CONNECTED
101	000040	SW05=40	
102	000020	SW04=20	
103	000010	SW03=10	
104	000004	SW02=4	;LPSVC VISUAL PATTERN
105	000002	SW01=2	;LPSVC VISUAL PATTERN
106	000001	SW00=1	;LPSVC VISUAL PATTERN
107			
108		;REGISTER DEFINITIONS	
109			
110	000000	R0=%0	
111	000001	R1=%1	
112	000002	R2=%2	
113	000003	R3=%3	
114	000004	R4=%4	
115	000005	R5=%5	
116	000006	SP=%6	
117	000007	PC=%7	

```

118 ;LOAD TRAP CATCHER INTO LOC'S 0-1000
119 .=24
120 000024 014464 PWRFAL ;POWER FAIL HANDLER
121 000026 000340 340
122 .=60
123 000060 014046 XTYYIN ;TELEPRINTER KEYBOARD ROUTINE
124 000062 000340 340
125 .=30
126 000030 014566 EMTSRV ;EMT TRAP, EMT DISPATCH SERVICE
127 000032 000340 340
128 000034 015154 LOGERR ;TRAP TRAP, LOGIC ERROR TRAP
129 000036 000340 340
130 .=46
131 000046 002174 LOGICAL
132 .=52
133 000052 000000 0
134 .=170
135 000170 000000 SOFTSW: 0 ;SOFTWARE SWITCH REGISTER VALUE
136 000172 000000 SOFTDI: 0
137 000174 000137 001512 JMP MONITR ;PROGRAM 'RESTART' ADDRESS
138 000200 000137 001224 JMP INIT ;INITIALIZATION ADDRESS
139 000204 000137 002236 JMP CKTEST ;MANUAL SA OF CLOCK TEST
140 000210 000137 007642 JMP IOTEST ;MANUAL SA OF DIGITAL I/O LOGIC TEST
141 000214 000137 011562 JMP VCTEST ;MANUAL SA OF SCOPE LOGIC TEST
142 000220 000137 016722 JMP VISUAL ;MANUAL SA OF SCOPE VISUAL TEST
143 000224 000137 021074 JMP ST1 ;MINI-TEST SCHMITT TRIGGER #1
144 000230 000137 021110 JMP ST2 ;MINI-TEST SCHMITT TRIGGER #2
145 000234 000137 021010 JMP CKOVFL ;MINI-TEST CLOCK OVERFLOW
146 000240 000137 020660 JMP RELAY ;MINI-TEST RELAY CONTACT
147 ;TRAP EQUIVALENCE TABLE:
148 104400 ERROR=TRAP ;LOGIC TEST ERROR ROUTINE
149 104000 PRINT=EMT ;MESSAGE PRINTER ROUTINE
150 104001 SCOPE0=EMT+1 ;SCOPE SUBROUTINE (1)
151 104002 SCOPE=EMT+2 ;LOGIC TEST SCOPE SUBROUTINE (4000)
152 104003 SCOPE1=EMT+3 ;LOGIC TEST SCOPE SUBROUTINE (10)
153 104004 SPACE=EMT+4 ;TYPE 'N' SPACES
154 104005 PRTOUT=EMT+5 ;OCTAL PRINT ROUTINE
155 104006 TTYIN=EMT+6 ;TELETYPE INPUT ROUTINE
156 104007 TSTTKS=EMT+7 ;SUBROUTINE TO TEST FOR KEYBOARD FLAG
157
158 .=1000
159 001000 170400 LPSADD: 170400 ;LPS STARTING ADDRESS
160 001002 000340 LPSVCT: 340 ;LPS STARTING VECTOR
161 001004 000300 ADBRL: 300 ;A TO D BR LEVEL
162 001006 000300 CKBRL: 300 ;CLOCK BR LEVEL
163 001010 000200 DIOBRL: 200 ;DIGITAL I/O BR LEVEL
164 001012 000200 VCBRL: 200 ;SCOPE BR LEVEL
165 001014 000001 PDPOLY: 1 ;1 FOR 11/20 2 FOR 11/45
166 001016 000002 FILLS: 2 ;TTY FILLER COUNT
167 001020 000000 FILCHR: 0 ;TTY FILLER CHARACTER
168 001022 177776 PSW: 177776
169 001024 177776 PS: 177776
170 001026 177560 TKS: 177560
171 001030 177562 TKB: 177562

```


172	001032	177564	TPS:	177564	
173	001034	177566	TPB:	177566	
174	001036	177570	SWR:	177570	;OR LOC. 170 IF RUNNING WITH NO SWITCH REGISTER
175	001040	177570	DISPLA:	177570	
176	001042	000000	PASSCT:	0	
177	001044	000002	ICOUNT:	2	
178					
179					
180			;LPS DEVICE ADDRESSES		
181	001046	170400	ADCS:	170400	;A TO D STATUS/CONTROL REGISTER
182	001050	170402	ADDBR:	170402	;A TO D CONVERTED VALUE <READ ONLY>
183					;A TO d LED DISPLAY LIGHTS <WRITE ONLY>
184					
185	001052	170404	CSR:	170404	;CLOCK STATUS/CONTROL REGISTER
186	001054	170406	CSB:	170406	;CLOCK PRESET BUFFER
187					
188	001056	170410	GRSTAT:	170410	;DIGITAL I/O STATUS/COMMAND REGISTER
189	001060	170412	GRDAI:	170412	;DIGITAL I/O INPUT REGISTER <READ ONLY>
190	001062	170414	GRDIO:	170414	;DIGITAL I/O OUTPUT REGISTER
191					
192	001064	170416	VCSTAT:	170416	;POINT PLOT STATUS REGISTER
193	001066	170420	VCXREG:	170420	;POINT PLOT X AXIS
194	001070	170422	VCYREG:	170422	;POINT PLOT Y AXIS
195	001072	170424	VCEXT:	170424	;EXTERNAL DAC REGISTER
196					
197	001074	170415	GRBHIO:	170415	;DIGITAL I/O OUTPUT REGISTER <HIGH BYTE>
198					
199					
200			; ADDRESS OF KMC-11 OF LPA-11 THE ADDR FOR KMADO MAY BE		
201			CHANGED BY THE USER TO REFLECT		
202			A DIFFERENT KMC-11 ADDR. THE		
203			REST OF THE ADDRESSES WILL		
204			BE CHANGED BY THE PROGRAM.		
205					
206					
207	001076		LPCI:		
208	001076	170460	KMADO:	.WORD 170460	;BASE KMC ADDR. MAY BE PATCHED BY USER.
209					
210	001100		LPMR:		
211	001100	170461	KMAD1:	.WORD 170460+1	; >DO NOT <;KMC-CSR ADDR
212	001102		LPC0:		
213	001102	170462	KMAD2:	.WORD 170460+2	; >PATCH <;
214	001104		LPS0:		
215	001104	170463	KMAD3:	.WORD 170460+3	; >THIS AREA <
216	001106		LPA0L:		
217	001106	170464	KMAD4:	.WORD 170460+4	;
218	001110		LPA0H:		
219	001110	170465	KMAD5:	.WORD 170460+5	; >DO NOT <
220	001112		LPMS1:		
221	001112	170466	KMAD6:	.WORD 170460+6	; >PATCH <
222	001114		LPMS2:		
223	001114	170467	KMAD7:	.WORD 170460+7	; >THIS AREA <
224					
225	001116	000300	VECTOR:	.WORD AVECT1&777	;BASE VECTOR OF KMC

226	001120	000304	VECTPS: .WORD	4+AVECT1&777	; VECOTR ADDR.+2
227					
228	001122	000004	VERSN: .WORD	4	; CURRENT VERSION NUMBER OF MICROCODE.
229					
230	001124	000000	.DVLS: .WORD	0	; /DEVICE LIST OF I/O ADDR. DEFINED
231	001126	000020	.BLKW	16.	; /BY INIT.
232					
233	001166	000000	\$BDDAT:	0	
234	001170	000000	\$GDDAT:	0	
235	001172	000000	\$ZERO:	0	
236	001174	000000	\$TMDAT:	0	
237			; LPS DEVICE INTERRUPT VECTORS		
238					
239	001176	000340	ADINT:	340	; A TO D INTERRUPT VECTOR
240	001200	000342	ADINT1:	342	
241		000300	AVECT1=	300	
242	001202	000300	\$VECT1:	.WORD AVECT1	
243					
244	001204	000344	CKV:	344	; CLOCK INTERRUPT VECTOR
245	001206	000346	CKVS:	346	
246					
247	001210	000350	GRIVA:	350	; DIGITAL INPUT INTERRUPT VECTOR
248	001212	000352	GRIVSA:	352	
249					
250	001214	000354	GRIVB:	354	; DIGITAL OUTPUT INTERRUPT VECTOR
251	001216	000356	GRIVSB:	356	
252					
253	001220	000360	VCIV:	360	; DISPLAY INTERRUPT VECTOR
254	001222	000362	VCIVS:	362	
255					

;THIS ROUTINE IS EXECUTED ON LOADING THE PROGRAM

```

256
257
258
259
260 001224 013706 016632 INIT: MOV STACK,SP ;INIT STACK POINTER=1000
261 001230 012777 000340 177564 MOV #340,SPSW
262
263 ;THIS SECTION OF CODE HANDLES INITIALIZING LPA-11 FUNCTIONS
264 ;
265
266 001236 010046 MOV RO,-(SP)
267 001240 010146 MOV R1,-(SP)
268 001242 013700 001076 MOV KMADO,RO ;GET KMC-11 ADDRESS.
269 001246 012701 001100 MOV #KMAD1,R1 ;GET ADDR. OF ADDR. LIST.
270
271 001252 005200 64$: INC RO ;UPDATE ADDR.
272 001254 010021 MOV RO,(1)+ ;WRITE ADDR.
273 001256 020127 001116 CMP R1,#KMAD7+2 ;DONE ALL ADDRESSES?
274 001262 001373 64$ BNE 64$ ;NO - DO NEXT ADDR.
275 001264 005037 001124 CLR .DVL5 ;CLR ADDR. LIST.
276 001270 012601 MOV (SP)+,R1
277 001272 012600 MOV (SP)+,RO
278
279 001274 004737 023054 JSR PC,$RESET ;CLEAR THE WORLD
280 001300 012700 001046 MOV #ADCS,RO
281 001304 013701 001000 MOV LPSADD,R1
282 001310 012702 000013 MOV #13,R2
283 001314 010120 INIT1A: MOV R1,(0)+
284 001316 062701 000002 ADD #2,R1
285 001322 005302 DEC R2
286 001324 001373 BNE INIT1A
287 001326 013737 001062 001074 MOV GRDIO,GRBHIO
288 001334 052737 000001 001074 BIS #1,GRBHIO
289 001342 012700 001176 MOV #ADINT,RO
290 001346 013701 001002 MOV LPSVCT,R1
291 001352 012702 000012 MOV #12,R2
292 001356 010120 INIT1B: MOV R1,(0)+
293 001360 062701 000002 ADD #2,R1
294 001364 005302 DEC R2
295 001366 001373 BNE INIT1B
296 001370 012737 001404 000004 MOV #15,SWR ;LOAD TRAP RETURN
297 001376 005777 177434 TST SWR ;TEST IF SWITCH REGISTER
298 001402 000407 BR 2$ ;BR IF YES
299 001404 022626 1$: CMP (SP)+,(SP)+
300 001406 012737 000170 001036 MOV #170,SWR ;LOAD LOC. 170 INTO ADDRESS
301 001414 012737 000172 001040 MOV #172,DISPLA
302 001422 005037 016030 2$: CLR NOLEDS
303
304 ;* MOV $ZERO,ADDBR ;/ PUT DATA FROM $ZERO TO DEVICE REG ADDBR
305 001436 000240 NOP
306 001440 005737 022102 TST $AERR
307 001444 001403 4$ BEQ 4$
308 001446 012737 177777 016030 MOV #-1,NOLEDS
309 001454 005737 000042 4$: TST #42

```

ADDRESS	HEX	ASCII	DISPATCH	OPERATOR	INSTR	OPERANDS	REMARKS
310	001460	001402			BEQ	3S	
311	001462	000137	001542		JMP	INIT2	
312	001466	012777	000100	177332	3S:	MOV	#100, JTKS ;ENABLE TTY INTERRUPTS
313	001474	005037	016630		CLR	PRINT1	
314	001500	104000			PRINT		;CALL MESSAGE PRINTER VIA 'EMT'
315	001502	016032			TITLE		;TYPE PROGRAM HEADER.
316	001504	104000			INITA: PRINT		
317	001506	016353			MES4		;PRINT THE TEST CALL LETTERS.
318	001510	000414			BR	INIT2	;GO AND AWAIT COMMAND.
319					MONITOR	SUBROUTINE. ENTER VIA 'IC' OR A RESTART AT LOCATION '200'.	
320	001512	004737	023054		MONITR: JSR	PC \$RESET	;INITIALIZE ON ENTRY
321	001516	013706	016632		MOV	STACK, SP	;RESET STACK POINTER
322	001522	012777	000340	177272	MOV	#340, JPSW	
323	001530	012777	000100	177270	MOV	#100, JTKS	;ENABLE TTY INTERRUPTS
324	001536	104000			PRINT		;CALL MESSAGE PRINTER
325	001540	016541			CNTRLC		;TYPE 'IC'
326							


```

327
328 001542 012737 014464 000024 INIT2: MOV      #PWRFAL, @#24      ;SET UP POWER FAIL
329 001550 012737 000006 000004      MOV      #6, @#4        ;SET UP BUSS ERROR
330 001556 005037 000006          CLR      @#6
331 001562 013777 001200 177406      MOV      @DINT1, @ADINT
332 001570 005077 177404          CLR      @ADINT1
333 001574 013777 001206 177402      MOV      CKVS, @CKV
334 001602 005077 177400          CLR      @CKVS
335 001606 013777 001212 177374      MOV      @GRIVSA, @GRIVA
336 001614 005077 177372          CLR      @GRIVSA
337 001620 013777 001216 177366      MOV      @GRIVSB, @GRIVB
338 001626 005077 177364          CLR      @GRIVSB
339 001632 005037 015570          CLR      SCOPEF
340 001636 012737 001504 016634      MOV      @INITA, @VECTR ;SET UP 'A' VECTOR ADDRESS.
341 001644 004537 015702          JSR      RS, LEDS
342 001650 000006 6
343 001652 005737 000042          TST      @#42
344 001656 001402          BEQ      .+6
345 001660 000137 001774          JMP      WHAT
346 001664 012777 000100 177134      MOV      #100, @TKS      ;ENABLE KEYBOARD INTERRUPT
347 001672 104000          PRINT
348 001674 016601          DOT          ;PRINT '.' TO INDICATE MONITOR READY
349 001676 104006          TTYIN        ;WAIT FOR TTY ENTRY
350 001700 042737 000040 014446      BIC      #BITS, INBUF    ;ENABLE LOWER CASE
351 001706 122737 000101 014446      CMPB     #'A', INBUF    ;TEST FOR 'A'
352 001714 001002          BNE      .+6
353 001716 000137 002236          JMP      CKTEST        ;YES RUN 'CLOCK LOGIC TEST'
354 001722 122737 000102 014446      CMPB     #'B', INBUF    ;TEST FOR 'B'
355 001730 001002          BNE      .+6          ;NOT 'B'
356 001732 000137 007642          JMP      IOTEST        ;YES RUN 'I/O LOGIC TEST'
357 001736 122737 000103 014446      CMPB     #'C', INBUF    ;TEST FOR 'C'
358 001744 001002          BNE      .+6          ;NOT 'C'
359 001746 000137 011562          JMP      VCTEST        ;YES RUN 'SCOPE LOGIC TEST'
360 001752 122737 000104 014446      CMPB     #'D', INBUF    ;TEST FOR 'D'
361 001760 001002          BNE      .+6          ;NOT 'D'
362 001762 000137 016722          JMP      VISUAL        ;YES RUN 'VISUAL DISPLAY TEST'
363 001766 104000          PRINT        ;ILLEGAL ENTRY
364 001770 016604          QMARK
365 001772 000663          BR      INIT2      ;TYPE '?'
                                     ;WAIT AGAIN

```

```

366                                     ;EXECUTE ONLY IF LOCATION 42 IS NON ZERO
367
368 001774 005037 002206      WHAT: CLR      NOCLK      ;CLEAR NO CLOCK
369 002000 005037 002210      CLR      NODIO      ;CLEAR NO DIGITAL I/O
370 002004 005037 002212      CLR      NOSCOP     ;CLEAR NO SCOPE
371 002010 012737 002214 000004  MOV      #WHAT1,#4
372
373 002026 012737 002222 000004  ;*      MOV      $ZERO,ACSR      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
374 002026 012737 002222 000004  MOV      #WHAT2,#4
375
376 002044 012737 002230 000004  ;*      MOV      $ZERO,AGRSTAT    ;/ PUT DATA FROM $ZERO TO DEVICE REG GRSTAT
377 002044 012737 002230 000004  MOV      #WHAT3,#4
378
379 002062 012737 000006 000004  ;*      MOV      $ZERO,AVCSTAT    ;/ PUT DATA FROM $ZERO TO DEVICE REG VCSTAT
380 002070 005737 002206      MOV      #6,#4
381 002074 001410      TST      NOCLK      ;TEST FOR NO CLOCK
382 002076 005737 002210      BEQ      WHATA    ;BRANCH IF CLOCK
383 002102 001405      TST      NODIO      ;TEST FOR NO DIGITAL I/O
384 002104 005737 002212      BEQ      WHATA    ;BRANCH IF DIGITAL I/O
385 002110 001402      TST      NOSCOP     ;TEST FOR NO SCOPE
386 002112 000000      BEQ      WHATA    ;BRANCH IF SCOPE
387 002114 000777      HALT
388 002114 000777      BR      .           ;FATAL ERROR, NO SLAVE SYNC FROM ANY LPS-11 DEVICE
389                                     ;HANG HERE
390 002116 005737 002206      WHATA: TST      NOCLK      ;TEST FOR CLOCK
391 002122 001002      BNE      WHATB      ;BRANCH IF NO CLOCK
392 002124 000137 002246      JMP      CTEST1   ;TEST CLOCK
393 002130 005737 002210      WHATB: TST      NODIO      ;TEST FOR DIGITAL I/O
394 002134 000402      BR      WHATC      ;DON'T RUN LPS-11-DR IN CHAIN MODE <NEWER LPS-11-DRA>
395 002136 000137 007652      JMP      ITEST1   ;RUN DIGITAL I/O TEST
396 002142 005737 002212      WHATC: TST      NOSCOP     ;TEST FOR SCOPE
397 002146 001004      BNE      WHATE      ;BRANCH IF NO SCOPE
398 002150 000137 011572      JMP      VTEST1   ;TEST SCOPE LOGIC
399 002154 000137 016732      WHATD: JMP      VTEST2   ;TEST VISUAL
400 002160 004737 023054      WHATE: JSR      PC,$RESET
401 002164 004737 023054      JSR      PC,$RESET
402 002170 013700 000042      MOV      #42,R0
403 002174 004710      LOGICAL: JSR      PC,(R0)
404 002176 000240      NOP
405 002200 000240      NOP
406 002202 000240      NOP
407 002204 000673      BR      WHAT
408
409 002206 000000      NOCLK: 0
410 002210 000000      NODIO: 0
411 002212 000000      NOSCOP: 0
412
413 002214 005137 002206      WHAT1: COM      NOCLK
414 002220 000002      RTI
415 002222 005137 002210      WHAT2: COM      NODIO
416 002226 000002      RTI
417 002230 005137 002212      WHAT3: COM      NOSCOP
418 002234 000002      RTI

```


K02

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 12

SEQ 0023

```

419 ;*****
420 ;CLOCK LOGIC TEST
421 ;*****
422
423 002236 004737 023054 CKTEST: JSR PC,$RESET
424 002242 104000 PRINT
425 002244 016216 MES2 ;IDENTIFY TEST
426 002246 005037 001042 CTEST1: CLR PASSCT
427 002252 013777 001042 176560 BEGIN: MOV PASSCT,$DISPLA
428 002260 013706 016632 MOV STACK,$P
429 002264 005077 176532 CLR $PSW
430 002270 012737 002326 015572 MOV $KWD+2,RETURN ;SET UP RESTART OF PROGRAM
431 002276 052777 000100 176522 BIS $BIT6,$TKS
432
433 ;TEST FOR NO BUSS ERRORS
434
435
436 ;* MOV $ZERO,$CSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
437 ;* MOV $ZERO,$CSB ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
438
439 ;TEST THE COUNTER PRESET BUFFER
440
441
442 002324 104002 KWD0: SCOPE
443 002326 012737 177777 001174 MOV #-1,$TMDAT ;LOAD PRESET BUFFER
444
445 ;* MOV $TMDAT,$CSB ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSB
446
447 ;* MOV $CSB,$BDDAT ;/READ DEVICE REG CSB,PUT DATA IN $BDDAT.
448 002354 023737 001174 001166 CMP $TMDAT,$BDDAT
449 002362 001401 BEQ .+4 ;BRANCH IF EQUAL
450 002364 104400 ERROR ;ERROR, COUNTER PRESET FAILED TO LOAD
451
452 002366 104002 KWD2: SCOPE
453 002370 012737 052525 001174 MOV #52525,$TMDAT ;LOAD PRESET BUFFER
454
455 ;* MOV $TMDAT,$CSB ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSB
456
457 ;* MOV $CSB,$BDDAT ;/READ DEVICE REG CSB,PUT DATA IN $BDDAT.
458 002416 023737 001174 001166 CMP $TMDAT,$BDDAT
459 002424 001401 BEQ .+4 ;BRANCH IF EQUAL
460 002426 104400 ERROR ;ERROR, COUNTER PRESET FAILED TO LOAD
461
462 002430 104002 KWD3: SCOPE
463 002432 012737 025252 001174 MOV #25252,$TMDAT ;LOAD PRESET
464
465 ;* MOV $TMDAT,$CSB ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSB
466
467 ;* MOV $CSB,$BDDAT ;/READ DEVICE REG CSB,PUT DATA IN $BDDAT.
468 002460 023737 001174 001166 CMP $TMDAT,$BDDAT
469 002466 001401 BEQ .+4 ;BRANCH IF EQUAL
470 002470 104400 ERROR ;ERROR, COUNTER PRESET FAILED TO LOAD
471
472 ;TEST INIT TO CLEAR COUNT PRESET BUFFER WHEN IT IS =-1

```

L02

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 13

SEQ 0024

473								
474	002472	104003			KWT4:	SCOPE1		
475	002474	012737	177777	001174		MOV	#-1,\$TMDAT	
476	002502	004737	023054			JSR	PC,\$RESET	
477								
478					;	MOV	@CSB,\$BDDAT	;/READ DEVICE REG CSB,PUT DATA IN \$BDDAT.
479	002516	005737	001166		*	TST	\$BDDAT	
480	002522	001401				BEQ	+.4	
481	002524	104400				ERROR		;ERROR, INIT FAILED TO CLEAR CSB
482								
483	002526	052777	000100	176272		BIS	#BIT6,@TKS	


```
484
485 ;TEST ENABLE COUNTER (BIT 0) CAN BE SET AND CLEARED
486
487 002534 104003 000001 001174 KWT5: SCOPE1
488 002536 012737 MOV #BIT0,$TMDAT
489
490 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
491
492 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
493 002564 023737 001174 001166 CMP $TMDAT,$BDDAT
494 002572 001401 BEQ .+4
495 002574 104400 ERROR ;ERROR COUNTER ENABLE FAILED TO SET
496
497 ;TEST RATE SELECT (BIT 1) MAY BE SET AND CLEARED
498
499 002576 104002 KWT6: SCOPE
500 002600 012737 MOV #2,$TMDAT
501
502 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
503
504 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
505 002626 023737 001174 001166 CMP $TMDAT,$BDDAT
506 002634 001401 BEQ .+4
507 002636 104400 ERROR ;ERROR, CSR NOT = 2
508
509 ;TEST THAT RATE SELECT (BIT 2) MAY BE SET AND CLEARED
510
511 002640 104002 KWT7: SCOPE
512 002642 012737 MOV #4,$TMDAT
513
514 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
515
516 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
517 002670 023737 001174 001166 CMP $TMDAT,$BDDAT
518 002676 001401 BEQ .+4
519 002700 104400 ERROR ;ERROR, CSR NOT = 4
520
521 ;TEST THAT RATE SELECT (BIT 3) MAY BE SET AND CLEARED
522
523 002702 104002 KWT8: SCOPE
524 002704 012737 MOV #10,$TMDAT
525
526 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
527
528 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
529 002732 023737 001174 001166 CMP $TMDAT,$BDDAT
530 002740 001401 BEQ .+4
531 002742 104400 ERROR ;ERROR, CSR NOT = 10
532
```

```
533
534 ;TEST MODE INTERRUPT ENABLE (BIT 6) CAN BE SET AND CLEARED
535
536 002744 104002 000100 001174 KWT10: SCOPE
537 002746 012737 000100 001174 MOV #100,$TMDAT
538
539 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
540
541 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
542 002774 023737 001174 001166 CMP $TMDAT,$BDDAT
543 003002 001401 BEQ .+4
544 003004 104400 ERROR ;ERROR, CSR NOT = 100
545
546 ;TEST MODE (BIT 8) CAN BE SET AND CLEARED
547
548 003006 104002 000400 001174 KWT11: SCOPE
549 003010 012737 000400 001174 MOV #400,$TMDAT
550
551 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
552
553 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
554 003036 023737 001174 001166 CMP $TMDAT,$BDDAT
555 003044 001401 BEQ .+4
556 003046 104400 ERROR ;ERROR, CSR NOT = 400
557
558 ;TEST MODE (BIT 9) CAN BE SET AND CLEARED
559
560 003050 104002 001000 001174 KWT12: SCOPE
561 003052 012737 001000 001174 MOV #1000,$TMDAT
562
563 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
564
565 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
566 003100 023737 001174 001166 CMP $TMDAT,$BDDAT
567 003106 001401 BEQ .+4
568 003110 104400 ERROR ;ERROR, CSR NOT = 1000
569
570 ;TEST ST#1 START ENABLE (BIT 13) CAN BE SET AND CLEARED
571
572 003112 104002 020000 001174 KWT14: SCOPE
573 003114 012737 020000 001174 MOV #20000,$TMDAT
574
575 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
576
577 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
578 003142 023737 001174 001166 CMP $TMDAT,$BDDAT
579 003150 001401 BEQ .+4
580 003152 104400 ERROR ;ERROR, CSR NOT = 200000
581
```



```

582
583 ;TEST ST#1 INTERRUPT ENABLE (BIT 14) CAN BE SET AND CLEARED
584
585 003154 104002 040000 001174 KWT15: SCOPE
586 003156 012737 040000 001174 MOV #40000,$TMDAT
587
588 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
589
590 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
591 003204 023737 001174 001166 CMP $TMDAT,$BDDAT
592 003212 001401 BEQ .+4
593 003214 104400 ERROR ;ERROR, CSR NOT = 40000
594
595 ;TEST THAT THE DONE (BIT 7) CAN BE SET AND CLEARED
596
597 003216 104002 KWT16: SCOPE
598 003220 012737 000200 001174 MOV #BIT7,$TMDAT
599
600 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
601
602 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
603 003246 105737 001166 TST $BDDAT
604 003252 100401 BMI .+4
605 003254 104400 ERROR ;ERROR, CSR NOT = 200
606
607 ;* MOV $ZERO,$CSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
608
609 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
610 003276 005737 001166 TST $BDDAT
611 003302 001401 BEQ .+4
612 003304 104400 ERROR ;ERROR CSR NOT = 0
613
614 ;TEST THAT THE ST FLAG (BIT 15) CAN BE SET AND CLEARED
615
616 003306 104002 KWT17: SCOPE
617 003310 012737 100000 001174 MOV #BIT15,$TMDAT ;SET BIT 15
618
619 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
620
621 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
622 003336 005737 001166 TST $BDDAT
623 003342 100401 BMI .+4
624 003344 104400 ERROR ;ERROR CSR NOT 100000
625
626 ;* MOV $ZERO,$CSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
627
628 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
629 003366 005737 001166 TST $BDDAT
630 003372 001401 BEQ .+4
631 003374 104400 ERROR ;ERROR CSR NOT = 0
632
633 ;TEST THAT THE ST FLAG DOES NOT SET FROM THE OUTSIDE SOURCE
634
635 003376 104002 KWT18: SCOPE

```

C03

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 17

SEQ 0028

636									
637									
638	003410	005037	016674		;	*	MOV	\$ZERO, @CSR	;/ PUT DATA FROM \$ZERO TO DEVICE REG CSR
639	003414						CLR	TEMP	
640							KWT18A:		
641							;	*	
642	003424	005737	001166				MOV	@CSR, \$BDDAT	;/ READ DEVICE REG CSR, PUT DATA IN \$BDDAT.
643	003430	100001					TST	\$BDDAT	
644	003432	104400					BPL	.+4	
645	003434	105237	016674				ERROR		; ERROR ST1 SET IN ERROR
646	003440	001365					INCB	TEMP	
							BNE	KWT18A	


```

647
648
649
650
651 003442 104003
652
653
654
655
656 003464 052737 010000 001174
657
658
659
660
661 003512 005737 001166
662 003516 100401
663 003520 104400
664
665
666
667
668 003522 104002
669
670
671 003534 012737 020000 001174
672
673
674
675
676 003562 052737 010000 001174
677
678
679
680
681 003610 032737 000001 001166
682 003616 001001
683 003620 104400
684
685

;TEST MAINT ST1 CAN SET ST1 FLAG
KWT19: SCOPE1
;* MOV $ZERO,ACSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
;* MOV ACSR,STMDAT ;/READ DEVICE REG CSR,PUT DATA IN STMDAT.
   BIS #BIT12,STMDAT
;* MOV STMDAT,ACSR ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
;* MOV ACSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
   TST $BDDAT
   BMI .+4
   ERROR ;ERROR, ST1 FLAG FAILED TO SET

;TEST THAT WHEN ST1 FIRES AND ST1 START ENABLE =1
; THAT THE ENABLE COUNTER GETS SET
KWT20: SCOPE
;* MOV $ZERO,ACSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
   MOV #BIT13,STMDAT ;SET ST1 START ENABLE
;* MOV STMDAT,ACSR ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
;* MOV ACSR,STMDAT ;/READ DEVICE REG CSR,PUT DATA IN STMDAT.
   BIS #BIT12,STMDAT
;* MOV STMDAT,ACSR ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
;* MOV ACSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
   BIT #BIT0,$BDDAT
   BNE .+4
   ERROR ;ST1 START ENABLE AND ST1
         ;FAILED TO SET ENABLE COUNTER

```

```

686 ;TEST THAT MODE 1 AND BIT 10 (MAINT. ST 2) SET BIT 7
687
688 003622 104002 KWT22: SCOPE
689
690 ;* MOV $ZERO,ACSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
691
692 ;* MOV $ZERO,ACSB ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
693 003644 012737 001000 001174 MOV #BIT9,$TMDAT ;LOAD MODE
694
695 ;* MOV $TMDAT,ACSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
696
697 ;* MOV ACSR,$TMDAT ;/READ DEVICE REG CSR,PUT DATA IN $TMDAT.
698 003672 052737 002000 001174 BIS #BIT10,$TMDAT
699
700 ;* MOV $TMDAT,ACSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
701
702 ;* MOV ACSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
703 003720 105737 001166 TSTB $BDDAT
704 003724 100401 BMI .+4
705 003726 104400 ERROR ;ERROR, ST 2 FAILED TO SET BIT 7
706
707 ;TEST THAT THE COUNTER CAN BE LOADED
708
709 003730 104002 KWT23: SCOPE
710
711 ;* MOV $ZERO,ACSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
712 003742 012737 177777 001174 MOV #-1,$TMDAT ;LOAD PRESET AND COUNTER
713
714 ;* MOV $TMDAT,ACSB ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSB
715 003760 012737 001401 001174 MOV #1401,$TMDAT ;LOAD MODE AND ENABLE COUNT
716
717 ;* MOV $TMDAT,ACSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
718
719 ;* MOV $ZERO,ACSB ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
720
721 ;* MOV ACSR,$TMDAT ;/READ DEVICE REG CSR,PUT DATA IN $TMDAT.
722 004016 052737 002000 001174 BIS #BIT10,$TMDAT ;FIRE ST
723
724 ;* MOV $TMDAT,ACSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
725
726 ;* MOV ACSB,$BDDAT ;/READ DEVICE REG CSB,PUT DATA IN $BDDAT.
727 004044 022737 177777 001166 CMP #-1,$BDDAT ;TEST THE NUMBER
728 004052 001401 BEQ .+4
729 004054 104400 ERROR ;ERROR, COUNTER FAILED TO LOAD PROPERLY
730
731 ;TEST THAT THE COUNTER CAN BE LOADED
732
733 004056 104002 KWT24: SCOPE
734
735 ;* MOV $ZERO,ACSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
736 004070 012737 125252 001174 MOV #125252,$TMDAT ;LOAD PRESET AND COUNTER
737
738 ;* MOV $TMDAT,ACSB ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSB
739 004106 012737 001401 001174 MOV #1401,$TMDAT ;LOAD MODE AND ENABLE COUNT

```



```

740
741          ;*      MOV      $TMDAT,$CSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
742
743          ;*      MOV      $ZERO,$CSB        ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
744
745          ;*      MOV      $CSR,$TMDAT        ;/ READ DEVICE REG CSR,PUT DATA IN $TMDAT.
746 004144 052737 002000 001174      BIS      $BIT10,$TMDAT
747
748          ;*      MOV      $TMDAT,$CSR        ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
749
750          ;*      MOV      $CSB,$BDDAT        ;/ READ DEVICE REG CSB,PUT DATA IN $BDDAT.
751 004172 022737 125252 001166      CMP      $125252,$BDDAT
752 004200 001401                      BEQ      .+4
753 004202 104400                      ERROR      ;ERROR, COUNTER FAILED TO LOAD PROPERLY
754
755          ;TEST THAT THE COUNTER CAN BE LOADED
756
757 004204 104002      KWT25: SCOPE
758
759          ;*      MOV      $ZERO,$CSR        ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
760 004216 012737 052525 001174      MOV      $52525,$TMDAT      ;LOAD PRESET AND COUNTER
761                                  ;LOAD MODE AND ENABLE COUNT
762                                  ;SHOULD ONLY CLEAR PRESET
763
764          ;*      MOV      $TMDAT,$CSB        ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSB
765 004234 012737 001401 001174      MOV      $1401,$TMDAT
766
767          ;*      MOV      $TMDAT,$CSR        ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
768
769          ;*      MOV      $ZERO,$CSB        ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
770
771          ;*      MOV      $CSR,$TMDAT        ;/ READ DEVICE REG CSR,PUT DATA IN $TMDAT.
772 004272 052737 002000 001174      BIS      $BIT10,$TMDAT
773
774          ;*      MOV      $TMDAT,$CSR        ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
775
776          ;*      MOV      $CSB,$BDDAT        ;/ READ DEVICE REG CSB,PUT DATA IN $BDDAT.
777 004320 022737 052525 001166      CMP      $52525,$BDDAT      ;TEST THE NUMBER
778 004326 001401                      BEQ      .+4
779 004330 104400                      ERROR      ;ERROR, COUNTER FAILED TO LOAD PROPERLY
780

```

```

781
782      ;TEST THAT THE COUNTER COUNTS UP
783      ;USE MAINT COUNT
784      ;USE MAINT ST2
785      ;USE MODE 2
786      ;USE RATE 1 <1MHZ>
787
788      004332 005037 016674      KWT26: CLR      TEMP
789      004336 104002
790
791      ;*      MOV      $ZERO,ACSR      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
792
793      ;*      MOV      $ZERO,ACSB      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
794      004360 012737 001002 001174      MOV      #1002,STMDAT
795
796      ;*      MOV      STMDAT,ACSR      ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
797      004376      KWT26A:
798
799      ;*      MOV      ACSR,STMDAT      ;/READ DEVICE REG CSR,PUT DATA IN STMDAT.
800      004406 052737 002000 001174      BIS      #BIT10,STMDAT
801
802      ;*      MOV      STMDAT,ACSR      ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
803
804      ;*      MOV      ACSB,$BDDAT      ;/READ DEVICE REG CSB,PUT DATA IN $BDDAT.
805      004434 023737 001166 016674      CMP      $BDDAT,TEMP
806      004442 001402      BEQ      .+6
807      004444 104400      ERROR
808      004446 000426      BR      KWT30
809
810      ;*      MOV      ACSR,$BDDAT      ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
811      004460 105737 001166      TSTB     $BDDAT      ;TEST DONE FLAG
812      004464 100402      BMI      1$
813      004466 104400      ERROR      ;ERROR, CKBF TO BF FAILED TO SET DONE
814      004470 000415      BR      KWT30
815      004472      1$:      ;MAINT COUNT
816      004472 052737 004000 001166      BIS      #BIT11,$BDDAT
817      004500 042737 000200 001166      BIC      #BIT7,$BDDAT
818
819      ;*      MOV      $BDDAT,ACSR      ;/ PUT DATA FROM $BDDAT TO DEVICE REG CSR
820      004516 005237 016674      INC      TEMP
821      004522 001325      BNE      KWT26A      ;BRANCH IF NOT FULL COUNT
822

```



```

823      ;TEST THAT THE COUNTER COUNTS UP
824      ;MAKE SURE THAT THE CLOCK SELECTION LOGIC WORKS
825      ;USE MAINT COUNT <100KHZ>
826      ;USE MAINT ST2
827      ;USE MODE 2
828      ;RATE #100KHZ
829
830      004524 104001      KWT30: SCOPE0
831
832      ;*      MOV      $ZERO,ACSR      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
833
834      ;*      MOV      $ZERO,ACSB      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
835      004546 012737 001004 001174      MOV      #1004,STMDAT      ;LOAD STATUS
836
837      ;*      MOV      STMDAT,ACSR      ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
838
839      ;*      MOV      ACSR,STMDAT      ;/READ DEVICE REG CSR,PUT DATA IN STMDAT.
840      004574 052737 002000 001174      BIS      #BIT10,STMDAT
841
842      ;*      MOV      STMDAT,ACSR      ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
843
844      ;*      MOV      ACSB,SBDDAT      ;/READ DEVICE REG CSB,PUT DATA IN SBDDAT.
845      004622 005737 001166      TST      SBDDAT
846      004626 001401      BEQ      .+4
847      004630 104400      ERROR
848      004632 012737 000012 016642      MOV      #10.,COUNT      ;ERROR PRESET INCREMENTED IN ERROR
849      004640      KWT30A:      ;SET UP A COUNTER
850
851      ;*      MOV      ACSR,STMDAT      ;/READ DEVICE REG CSR,PUT DATA IN STMDAT.
852      004650 052737 004000 001174      BIS      #BIT11,STMDAT
853
854      ;*      MOV      STMDAT,ACSR      ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
855      004666 005337 016642      DEC      COUNT
856      004672 001362      BNE      KWT30A      ;BR
857
858      ;*      MOV      ACSR,STMDAT      ;/READ DEVICE REG CSR,PUT DATA IN STMDAT.
859      004704 052737 002000 001174      BIS      #BIT10,STMDAT
860
861      ;*      MOV      STMDAT,ACSR      ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
862
863      ;*      MOV      ACSB,SBDDAT      ;/READ DEVICE REG CSB,PUT DATA IN SBDDAT.
864      004732 022737 000001 001166      CMP      #1,SBDDAT
865      004740 001401      BEQ      .+4
866      004742 104400      ERROR      ;ERROR, CLOCK PRESET BUFFER
867
868      ;COUNTED IN ERROR, FAULT IS PROBABLY IN THE
869      ;CLOCK DOWN COUNT OR RATE SELECTION LOGIC
870
871      ;TEST THAT THE COUNTER COUNTS UP
872      ;MAKE SURE THAT THE CLOCK SELECTION LOGIC WORKS
873      ;USE MAINT COUNT <10KHZ>
874      ;USE MAINT ST2
875      ;USE MODE 2
876      ;RATE #10KHZ

```

```

877
878 004744 104001 KWT31: SCOPED
879
880 ;* MOV $ZERO, @CSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
881
882 ;* MOV $ZERO, @CSB ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
883 004766 012737 001006 001174 ;* MOV #1006, $TMDAT ;/ LOAD STATUS
884
885 ;* MOV $TMDAT, @CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
886
887 ;* MOV @CSR, $TMDAT ;/ READ DEVICE REG CSR, PUT DATA IN $TMDAT.
888 005014 052737 002000 001174 ;* BIS #BIT10, $TMDAT
889
890 ;* MOV $TMDAT, @CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
891
892 ;* MOV @CSB, $BDDAT ;/ READ DEVICE REG CSB, PUT DATA IN $BDDAT.
893 005042 005737 001166 ;* TST $BDDAT
894 005046 001401 ;* BEQ .+4
895 005050 104400 ;* ERROR
896 005052 012737 000144 016642 ;* MOV #100., COUNT ;/ ERROR PRESET INCREMENTED IN ERROR
897 005060 ;* ;/ SET UP A COUNTER
898 KWT31A:
899
900 ;* MOV @CSR, $TMDAT ;/ READ DEVICE REG CSR, PUT DATA IN $TMDAT.
901 005070 052737 004000 001174 ;* BIS #BIT11, $TMDAT
902
903 ;* MOV $TMDAT, @CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
904 005106 005337 016642 ;* DEC COUNT
905 005112 001362 ;* BNE KWT31A ;/ BR
906
907 ;* MOV @CSR, $TMDAT ;/ READ DEVICE REG CSR, PUT DATA IN $TMDAT.
908 005124 052737 002000 001174 ;* BIS #BIT10, $TMDAT
909
910 ;* MOV $TMDAT, @CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
911
912 ;* MOV @CSB, $BDDAT ;/ READ DEVICE REG CSB, PUT DATA IN $BDDAT.
913 005152 022737 000001 001166 ;* CMP #1, $BDDAT
914 005160 001401 ;* BEQ .+4
915 005162 104400 ;* ERROR
916 ;/ ERROR, CLOCK PRESET BUFFER
917 ;/ COUNTED IN ERROR, FAULT IS PROBABLY IN THE
;/ CLOCK DOWN COUNT OR RATE SELECTION LOGIC

```



```

918
919
920      ;TEST THAT THE COUNTER COUNTS UP
921      ;MAKE SURE THAT THE CLOCK SELECTION LOGIC WORKS
922      ;USE MAINT COUNT <1KHZ>
923      ;USE MAINT ST2
924      ;USE MODE 2
925      ;RATE #1KHZ
926 005164 104001      KWT32: SCOPED
927
928      ;*      MOV      $ZERO, @CSR      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
929
930      ;*      MOV      $ZERO, @CSB      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
931 005206 012737 001010 001174      MOV      #1010, $TMDAT      ;LOAD STATUS
932
933      ;*      MOV      $TMDAT, @CSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
934
935      ;*      MOV      @CSR, $TMDAT      ;/ READ DEVICE REG CSR, PUT DATA IN $TMDAT.
936 005234 052737 002000 001174      BIS      #BIT10, $TMDAT
937
938      ;*      MOV      $TMDAT, @CSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
939
940      ;*      MOV      @CSB, $BDDAT      ;/ READ DEVICE REG CSB, PUT DATA IN $BDDAT.
941 005262 005737 001166      TST      $BDDAT
942 005266 001401      BEQ      .+4
943 005270 104400      ERROR
944 005272 012737 001750 016642      MOV      #1000., COUNT      ;ERROR PRESET INCREMENTED IN ERROR
945 005300      KWT32A:
946
947      ;*      MOV      @CSR, $TMDAT      ;/ READ DEVICE REG CSR, PUT DATA IN $TMDAT.
948 005310 052737 004000 001174      BIS      #BIT11, $TMDAT
949
950      ;*      MOV      $TMDAT, @CSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
951 005326 005337 016642      DEC      COUNT
952 005332 001362      BNE      KWT32A      ;BR
953
954      ;*      MOV      @CSR, $TMDAT      ;/ READ DEVICE REG CSR, PUT DATA IN $TMDAT.
955 005344 052737 002000 001174      BIS      #BIT10, $TMDAT
956
957      ;*      MOV      $TMDAT, @CSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
958
959      ;*      MOV      @CSB, $BDDAT      ;/ READ DEVICE REG CSB, PUT DATA IN $BDDAT.
960 005372 022737 000001 001166      CMP      #1, $BDDAT
961 005400 001401      BEQ      .+4
962 005402 104400      ERROR
963
964      ;COUNTED IN ERROR, FAULT IS PROBABLY IN THE
965      ;CLOCK DOWN COUNT OR RATE SELECTION LOGIC
966
967      ;TEST THAT THE COUNTER COUNTS UP
968      ;MAKE SURE THAT THE CLOCK SELECTION LOGIC WORKS
969      ;USE MAINT COUNT <100HZ>
970      ;USE MAINT ST2
971      ;USE MODE 2
972      ;RATE #100HZ

```

K03

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 25

SEQ 0036

```

972      005404  104001      KWT33:  SCOPED
973
974      ;*      MOV      $ZERO,ACSR      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
975
976      ;*      MOV      $ZERO,ACSB      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
977      005426  012737  001012  001174      MOV      #1012,$TMDAT      ;LOAD STATUS
978
979      ;*      MOV      $TMDAT,ACSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
980
981      ;*      MOV      ACSR,$TMDAT      ;/READ DEVICE REG CSR,PUT DATA IN $TMDAT.
982      005454  052737  002000  001174      BIS      #BIT10,$TMDAT
983
984      ;*      MOV      $TMDAT,ACSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
985
986      ;*      MOV      ACSB,$BDDAT      ;/READ DEVICE REG CSB,PUT DATA IN $BDDAT.
987      005502  005737  001166      TST      $BDDAT
988      005506  001401      BEQ      .+4
989      005510  104400      ERROR
990      005512  012737  023420  016642      MOV      #10000.,COUNT      ;ERROR PRESET INCREMENTED IN ERROR
991      005520      KWT33A:      ;SET UP A COUNTER
992
993      ;*      MOV      ACSR,$TMDAT      ;/READ DEVICE REG CSR,PUT DATA IN $TMDAT.
994      005530  052737  004000  001174      BIS      #BIT11,$TMDAT
995
996      ;*      MOV      $TMDAT,ACSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
997      005546  005337  016642      DEC      COUNT
998      005552  001362      BNE      KWT33A      ;BR
999
1000      ;*      MOV      ACSR,$TMDAT      ;/READ DEVICE REG CSR,PUT DATA IN $TMDAT.
1001      005564  052737  002000  001174      BIS      #BIT10,$TMDAT
1002
1003      ;*      MOV      $TMDAT,ACSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1004
1005      ;*      MOV      ACSB,$BDDAT      ;/READ DEVICE REG CSB,PUT DATA IN $BDDAT.
1006      005612  022737  000001  001166      CMP      #1,$BDDAT
1007      005620  001401      BEQ      .+4
1008      005622  104400      ERROR
1009
1010      ;ERROR, CLOCK PRESET BUFFER
1011      ;COUNTED IN ERROR, FAULT IS PROBABLY IN THE
1012      ;CLOCK DOWN COUNT OR RATE SELECTION LOGIC

```



```
1012
1013      ; TEST THAT THE COUNTER COUNTS UP
1014      ; MAKE SURE THAT THE CLOCK SELECTION LOGIC WORKS
1015      ; USE MODE 2
1016      ; USE MAINT ST1
1017      ; RATE #EXTURNAL
1018
1019      005624 005037 016674      CLR      TEMP
1020      005630 104001      KWT34: SCOPE0
1021
1022      ;*      MOV      $ZERO,ACSR      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
1023
1024      ;*      MOV      $ZERO,ACSB      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
1025      005652 012737 001014 001174      MOV      #1014,$TMDAT      ;/ MODE 2 EXTURNAL CLOCK RATE
1026
1027      ;*      MOV      $TMDAT,ACSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1028      005670      KWT34A:
1029
1030      ;*      MOV      ACSR,$TMDAT      ;/ READ DEVICE REG CSR, PUT DATA IN $TMDAT.
1031      005700 052737 002000 001174      BIS      #BIT10,$TMDAT
1032
1033      ;*      MOV      $TMDAT,ACSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1034
1035      ;*      MOV      ACSR,$BDDAT      ;/ READ DEVICE REG CSB, PUT DATA IN $BDDAT.
1036      005726 023737 001166 016674      CMP      $BDDAT,TEMP
1037      005734 001402      BEQ      .+6
1038      005736 104400      ERROR
1039      005740 000416      BR      KWT35      ; EXTURNAL CLOCIDAILED TO COUNT
1040      ; THE COUNTER, CHECK THE RATE SELECTION
1041
1042      ;*      MOV      ACSR,$TMDAT      ;/ READ DEVICE REG CSR, PUT DATA IN $TMDAT.
1043      005752 052737 010000 001174      BIS      #BIT12,$TMDAT
1044
1045      ;*      MOV      $TMDAT,ACSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1046      005770 005237 016674      INC      TEMP
1047      005774 001335      BNE      KWT34A
1048
1049      ; TEST EXTURNAL INTERVAL FROM ZERO BASE (MODE 3)
1050      005776 104001      KWT35: SCOPE0
1051
1052      ;*      MOV      $ZERO,ACSR      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
1053      006010 012737 025252 001174      MOV      #25252,$TMDAT
1054
1055      ;*      MOV      $TMDAT,ACSB      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSB
1056      006026 012737 001402 001174      MOV      #1402,$TMDAT
1057
1058      ;*      MOV      $TMDAT,ACSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1059
1060      ;*      MOV      ACSR,$TMDAT      ;/ READ DEVICE REG CSR, PUT DATA IN $TMDAT.
1061      006054 052737 002000 001174      BIS      #BIT10,$TMDAT
1062
1063      ;*      MOV      $TMDAT,ACSR      ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1064
1065      ;*      MOV      ACSR,$BDDAT      ;/ READ DEVICE REG CSB, PUT DATA IN $BDDAT.
```

1066	006102	022737	025252	001166		CMP	#25252,\$BDDAT	
1067	006110	001401				BEQ	.+4	
1068	006112	104400				ERROR		;ERROR, MODE 3 FAILED TO LOAD CLOCK
1069								; PRESET BUFFER
1070								
1071					;*	MOV	@CSR,\$TMDAT	; /READ DEVICE REG CSR,PUT DATA IN \$TMDAT.
1072	006124	052737	002000	001174		BIS	#BIT10,\$TMDAT	
1073								
1074					;*	MOV	\$TMDAT,@CSR	; / PUT DATA FROM \$TMDAT TO DEVICE REG CSR
1075								
1076					;*	MOV	@CSB,\$BDDAT	; /READ DEVICE REG CSB,PUT DATA IN \$BDDAT.
1077	006152	022737	000000	001166		CMP	#0,\$BDDAT	
1078	006160	001401				BEQ	.+4	
1079	006162	104400				ERROR		;ERROR, MODE 3 FAILED TO ZERO CLOCK COUNTER


```

1080
1081 ;TEST THAT RESET CLEARS RATE SELECT BITS
1082
1083 006164 104002 000016 001174 KWT36: SCOPE
1084 006166 012737 000016 001174 MOV #BIT3!BIT2!BIT1,$TMDAT ;SET MODE BITS
1085
1086 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1087 006204 004737 023054 JSR PC,$RESET
1088
1089 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1090 006220 005737 001166 TST $BDDAT
1091 006224 001401 BEQ .+4
1092 006226 104400 ERROR ;ERROR, RESET FAILED TO CLEAR RATE BITS
1093
1094 ;TEST THAT RESET CLEARS MODE SELECT BITS
1095
1096 006230 104003 001400 001174 KWT37: SCOPE1
1097 006232 012737 001400 001174 MOV #BIT9!BIT8,$TMDAT ;SET MODE BITS
1098
1099 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1100 006250 004737 023054 JSR PC,$RESET
1101
1102 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1103 006264 005737 001166 TST $BDDAT
1104 006270 001401 BEQ .+4
1105 006272 104400 ERROR ;ERROR, RESET FAILED TO CLEAR MODE BITS
1106
1107 ;TEST THAT RESET CLEARS ST1 FLAG,ST1 INTERRUPT ENABLE AND ST1 START ENABLE
1108
1109 006274 104003 160000 001174 KWT38: SCOPE1
1110 006276 012737 160000 001174 MOV #BIT15!BIT14!BIT13,$TMDAT ;SET ST1 FLAG,INT ENABLE AND ST1 START ENABLE
1111
1112 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1113 006314 004737 023054 JSR PC,$RESET
1114
1115 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1116 006330 005737 001166 TST $BDDAT
1117 006334 001401 BEQ .+4
1118 006336 104400 ERROR ;ERROR, RESET FAILED TO CLEAR ST1 LOGIC
1119
1120 ;TEST THAT RESET CLEARS DONE FLAG AND DONE INTERRUPT ENABLE
1121
1122 006340 104003 000300 001174 KWT39: SCOPE1
1123 006342 012737 000300 001174 MOV #BIT7!BIT6,$TMDAT ;SET DONE FLAG AND INTERRUPT ENABLE
1124
1125 ;* MOV $TMDAT,$CSR ;/ PUT DATA FROM $TMDAT TO DEVICE REG CSR
1126 006360 004737 023054 JSR PC,$RESET
1127
1128 ;* MOV $CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1129 006374 005737 001166 TST $BDDAT
1130 006400 001401 BEQ .+4
1131 006402 104400 ERROR ;ERROR, RESET FAILED TO CLEAR DONE FLAG OR DONE
1132
1133 ;TEST THAT RESET CLEARS COUNTER ENABLE

```

1134									
1135	006404	104003			KWT40:	SCOPE1			
1136	006406	012737	00CJ01	001174		MOV	#BIT0,\$TMDAT		;LOAD COUNTER ENABLE
1137									
1138					;*	MOV	\$TMDAT,\$CSR		; / PUT DATA FROM \$TMDAT TO DEVICE REG CSR
1139	006424	004737	023054			JSR	PC,\$RESET		
1140									
1141					;*	MOV	\$CSR,\$BDDAT		; / READ DEVICE REG CSR, PUT DATA IN \$BDDAT.
1142	006440	005737	001166			TST	\$BDDAT		
1143	006444	001401				BEQ	+.4		
1144	006446	104400				ERROR			;ERROR, RESET FAILED TO CLEAR COUNTER ENABLE
1145									
1146									
1147	006450	052777	000100	172350		BIS	#BIT6,\$TKS		


```

1148 ;TEST CLOCK TO COUNT UP AT ALL FREQUENCIES (MODE 0)
1149 ;TEST THAT CLOCK ENABLE DOES CLEAR ON DONE FLAG
1150 ;1MHZ
1151
1152 006456 104003 KWT41: SCOPE1
1153 006460 012737 000003 007632 MOV #3,RATE ;SELECT MODE 0, 1MHZ., GO
1154 006466 004737 007130 JSR PC,UPCNT
1155
1156 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1157 006502 105737 001166 TSTB $BDDAT
1158 006506 100401 BMI .+4
1159 006510 104400 ERROR ;ERROR, 1MHZ FAILED TO COUNT
1160
1161 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1162 006522 032737 000001 001166 BIT #BIT0,$BDDAT
1163 006530 001401 BEQ .+4
1164 006532 104400 ERROR ;ERROR, MODE 0 FAILED TO CLEAR COUNTER ENABLE
1165
1166 ;TEST CLOCK TO COUNT UP AT ALL FREQUENCIES
1167 ;100 KHZ
1168
1169 006534 104003 KWT42: SCOPE1
1170 006536 012737 000005 007632 MOV #5,RATE ;SELECT MODE 0, 100KHZ., GO
1171 006544 004737 007130 JSR PC,UPCNT
1172
1173 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1174 006560 105737 001166 TSTB $BDDAT
1175 006564 100401 BMI .+4
1176 006566 104400 ERROR ;ERROR, 100KHZ. FAILED TO COUNT
1177
1178 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1179 006600 032737 000001 001166 BIT #BIT0,$BDDAT
1180 006606 001401 BEQ .+4
1181 006610 104400 ERROR ;ERROR MODE 0 FAILED TO CLEAR COUNT ENABLE
1182
1183 ;10 KHZ
1184
1185 006612 104003 KWT43: SCOPE1
1186 006614 012737 000007 007632 MOV #7,RATE ;SELECT MODE 0, 10KHZ., GO
1187 006622 004737 007130 JSR PC,UPCNT
1188
1189 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1190 006636 105737 001166 TSTB $BDDAT
1191 006642 100401 BMI .+4
1192 006644 104400 ERROR ;ERROR, 10 KHZ. FAILED TO COUNT
1193
1194 ;TEST CLOCK TO COUNT UP AT ALL FREQUENCIES
1195 ;1KHZ
1196
1197 006646 104003 KWT44: SCOPE1
1198 006650 012737 000011 007632 MOV #11,RATE ;SELECT MODE 0, 1 KHZ, GO
1199 006656 004737 007130 JSR PC,UPCNT
1200
1201 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.

```

004

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 31

SEQ 0042

1202	006672	105737	001166
1203	006676	100401	
1204	006700	104400	
1205			

TSTB	SBDDAT
BMI	.+4
ERROR	

;ERROR, 1KHZ FAILED TO COUNT


```

1206 ;TEST CLOCK TO COUNT UP AT ALL FREQUENCIES
1207 ;TEST 100 HZ
1208
1209 006702 104003 KWT45: SCOPE1
1210 006704 012737 000013 007632 MOV #13,RATE ;SELECT MODE 0, 100 HZ, GO
1211 006712 004737 007130 JSR PC,UPCNT
1212
1213 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1214 006726 105737 001166 TSTB $BDDAT
1215 006732 100401 BMI .+4
1216 006734 104400 ERROR ;ERROR, 100HZ FAILED TO COUNT
1217
1218 ;TEST CLOCK TO COUNT UP AT ALL FREQUENCIES
1219 ;LINE FREQ.
1220
1221 006736 104003 KWT46: SCOPE1
1222 006740 012737 000017 007632 MOV #17,RATE ;SELECT MODE 0, LINE FREQ, GO
1223 006746 004737 007130 JSR PC,UPCNT
1224
1225 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1226 006762 105737 001166 TSTB $BDDAT
1227 006766 100401 BMI .+4
1228 006770 104400 ERROR ;ERROR, LINE FREQUENCY FAILED TO COUNT
1229
1230 ;REPEAT T12 WITH MODE 1
1231 ;TEST THAT CLOCK ENABLE DOES NOT CLEAR ON DONE
1232 ; 1 MHZ.
1233
1234 006772 104003 KWT47: SCOPE1
1235 006774 012737 000403 007632 MOV #403,RATE ;SELECT MODE 1, 1MHZ., GO
1236 007002 004737 007130 JSR PC,UPCNT
1237
1238 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1239 007016 105737 001166 TSTB $BDDAT
1240 007022 100401 BMI .+4
1241 007024 104400 ERROR ;ERROR, 1 MHZ. FAILED TO COUNT
1242 ; MODE 1
1243
1244 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1245 007036 032737 000001 001166 BIT #BIT0,$BDDAT
1246 007044 001001 BNE .+4
1247 007046 104400 ERROR ;MODE 1 CLEARED COUNTER ENABLE IN ERROR
1248
1249 ;100KHZ.
1250
1251 007050 104003 KWT48: SCOPE1
1252 007052 012737 000405 007632 MOV #405,RATE ;MODE 1 100KHZ GO
1253 007060 004737 007130 JSR PC,UPCNT
1254
1255 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1256 007074 105737 001166 TSTB $BDDAT
1257 007100 100401 BMI .+4
1258 007102 104400 ERROR ;ERROR, 100KHZ FAILED TO COUNT MODE 1
1259

```

F04

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 33

SEQ 0044

1260
1261 007114 032737 000001 001166 ;*
1262 007122 001001
1263 007124 104400
1264 007126 000453
1265

MOV @CSR,\$BDDAT
BIT #BIT0,\$BDDAT
BNE .+4
ERROR
BR KWT49

; /READ DEVICE REG CSR, PUT DATA IN \$BDDAT.

; MODE 1 CLEARED COUNTER ENABLE IN ERROR


```

1266
1267 ;SUBROUTINE TO LOAD -2 INTO THE CLOCK PRESET REGISTER
1268 ; AND LOAD CLOCK RATE INTO CLOCK STATUS REGISTER
1269 ; AND START THE CLOCK. WAIT A PERIOD OF TIME THEN EXIT
1270
1271 007130 UPCNT:
1272
1273 ;* MOV $ZERO,ACSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
1274 007140 012737 177776 001174 MOV #-2,STMDAT ;MOVE -2 INT PRESET
1275
1276 ;* MOV STMDAT,ACSB ;/ PUT DATA FROM STMDAT TO DEVICE REG CSB
1277 007156 013737 007632 001174 MOV RATE,STMDAT ;LOAD RATE AND ENABLE CLOCK
1278
1279 ;* MOV STMDAT,ACSR ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
1280 007174 005037 016644 CLR DELAY ;DELAY
1281 007200
1282
1283 ;* MOV ACSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1284 007210 105737 001166 TSTB $BDDAT
1285 007214 100417 BMI UPCNTB ;
1286
1287 ;* MOV ACSR,$GDDAT ;/READ DEVICE REG CSR,PUT DATA IN $GDDAT.
1288
1289 ;* MOV ACSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
1290 007236 023737 001170 001166 CMP $GDDAT,$BDDAT
1291 007244 062737 000001 016644 ADD #1,DELAY
1292 007252 001352 BNE UPCNTA ;
1293 007254 000207 UPCNTB: RTS ;EXIT
1294
1295
1296 007256 KWT49:
1297 007256 KWT52:
1298 007256 104003 KWT62: SCOPE1
1299 007260 004537 007336 JSR RS,REPEAT
1300 007264 000016 ;CLOCK RATE
1301 007266 000001 ;CLOCK DEV
1302 007270 000001 ;MIN. COUNT
1303 007272 000000 ;DELAY
1304 007274 104400 ;ERROR, FAILED TO REACH MIN COUNT
1305 007276 003401 ;ERROR, CLOCK REPEATABILITY >+1
1306 007300 104400
1307
1308 ;BELL ON PASS COMPLETE
1309
1310 007302 104003 TSTEND: SCOPE1 ;LOGICAL END OF THE TEST
1311 007304 004737 023054 JSR PC,$RESET
1312 007310 005737 000042 TST #42
1313 007314 001402 BEQ HERE
1314 007316 000137 002130 JMP WHATB
1315
1316 007322 004737 015650 HERE: JSR PC,BELL ;REPORT END OF PASS
1317 007326 005237 001042 INC PASSCT
1318 007332 000137 002252 JMP BEGIN

```

```

1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332 007336
1333
1334
1335
1336
1337 007356 012537 007632
1338 007362 012537 007634
1339 007366 012537 007636
1340 007372 012537 007640
1341 007376 004737 007466
1342 007402 004737 007466
1343 007406 010037 001170
1344 007412 004737 007466
1345 007416 010037 001166
1346 007422 013700 001170
1347 007426 163700 001166
1348 007432 100001
1349 007434 005400
1350 007436 023737 001166 007636 3$:
1351 007444 002004
1352 007446 013737 007636 016674
1353 007454 000205
1354 007456 005725 4$:
1355 007460 020037 007634
1356 007464 000205
1357
1358 007466 013700 007640 10$:
1359 007472 052737 003000 007632
1360 007500 013737 007632 001174
1361
1362
1363
1364
1365 007526 005237 001174
1366
1367
1368 007542 005300
1369 007544 001376
1370
1371
1372 007556 052737 002000 001174

```

; SUBROUTINE TO TEST THE CLOCK REPEATIBILITY
 ; FIRST CLEAR CLOCK STATUS AND PRESET BUFFER
 ; THEN ENABLE THE CLOCK TO COUNT AT A RATE.
 ; DECREMENT RO FOR SOME PERIOD OF TIME, WHEN RO = 0
 ; FIRE THE ST #2 AND CAUSE THE COUNTER TO LOAD THE PRESET REGISTER
 ; SAVE THE PRESET VALUE AND REPEAT THIS OPERATION AGAIN
 ; THEN COMPARE THE FIRST TIMED VALUE TO THE SECOND TIMED VALUE
 ; <MACHINE AND MEMORY TIMING NOT IMPORTANT>
 ; TO BE WITHIN THE VALUE SPECIFIED BY LOCATION CNTDEV
 ; IF GREATER THAN EXPECTED IT IS AN ERROR.
 ; ALSO TEST THAT THE COUNTER HAS REACHED A MIN. COUNT

REPEAT:

```

; *      MOV      $ZERO, @CSR      ; / PUT DATA FROM $ZERO TO DEVICE REG CSR
; *      MOV      $ZERO, @CSB      ; / PUT DATA FROM $ZERO TO DEVICE REG CSB
; *      MOV      (R5)+, RATE      ; SET UP RATE
; *      MOV      (R5)+, CNTDEV     ; SET UP CNT. DEV
; *      MOV      (R5)+, MINCNT     ; SET UP MIN COUNT
; *      MOV      (R5)+, CKDLY     ; SAVE DELAY
; *      JSR      PC, 10$          ; DUMMY - TO CHARGE THE "CACHE"
; *      JSR      PC, 10$          ; ENABLE THE CLOCK
; *      MOV      RO, $GDDAT       ; SAVE 1ST RESULTS
; *      JSR      PC, 10$          ; ENABLE THE CLOCK
; *      MOV      RO, $BDDAT       ; SAVE THE 2ND RESULT
; *      MOV      $GDDAT, RO       ; GET 1ST RESULT
; *      SUB      $BDDAT, RO       ; SUBTRACT SECOND RESULT
; *      BPL      3$              ;
; *      NEG      RO              ;
; *      CMP      $BDDAT, MINCNT   ; COMPARE TO MIN. COUNT
; *      BGE      4$              ; BRANCK IF GREATER
; *      MOV      MINCNT, TEMP     ; LOAD TEMP FOR TYPE-OUT
; *      RTS      R5              ;
; *      TST      (R5)+           ; UPDATE THE STACK
; *      CMP      RO, CNTDEV       ; COMPARE TO DEVEATION
; *      RTS      R5              ;
; *      MOV      CKDLY, RO        ; LOAD DELAY COUNT
; *      BIS      #BIT10:BIT9, RATE ; ENABLE ST MODE
; *      MOV      RATE, $TMDAT     ; LOAD RATE
; *      MOV      $TMDAT, @CSR     ; / PUT DATA FROM $TMDAT TO DEVICE REG CSR
; *      MOV      @CSR, $TMDAT     ; / READ DEVICE REG CSR, PUT DATA IN $TMDAT.
; *      INC      $TMDAT          ;
; *      MOV      $TMDAT, @CSR     ; / PUT DATA FROM $TMDAT TO DEVICE REG CSR
; *      DEC      RO              ; DELAY
; *      BNE      1$              ;
; *      MOV      @CSR, $TMDAT     ; / READ DEVICE REG CSR, PUT DATA IN $TMDAT.
; *      BIS      #BIT10, $TMDAT  ;

```

```
1373
1374      ;*      MOV      STMDAT, @CSR      ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
1375
1376      ;*      MOV      @CSB, STMDAT      ;/ READ DEVICE REG CSB, PUT DATA IN STMDAT.
1377 007604 013700 001174      MOV      STMDAT, R0
1378
1379      ;*      MOV      $ZERO, @CSR      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
1380
1381      ;*      MOV      $ZERO, @CSB      ;/ PUT DATA FROM $ZERO TO DEVICE REG CSB
1382 007630 000207      RTS      PC      ;EXIT
1383
1384 007632 000000      RATE:      0      ;CLOCK RATE
1385 007634 000000      CNTDEV:    0      ;CLOCK DEV.
1386 007636 000000      MINCNT:    0      ;MIN. COUNT
1387 007640 000000      CKDLY:     0
```



```

1388
1389
1390
1391
1392
1393 007642 004737 023054 IOTEST: JSR PC,$RESET
1394 007646 104000 PRINT
1395 007650 016243 MES3 ; IDENTIFY TEST
1396 007652 005037 001042 ITEST1: CLR PASSCT
1397 007656 012737 007742 015572 IOTSTA: MOV #DRT0+2, RETURN
1398 007664 013777 001042 171146 MOV PASSCT, @DISPLA
1399 007672 013706 016632 MOV STACK, $P
1400 007676 005077 171120 CLR @PSW
1401 007702 052777 000100 171116 BIS #BIT6, @TKS
1402
1403 ; TEST FOR NO BUSS ERRORS
1404
1405
1406 ;* MOV $ZERO, @GRSTAT ; / PUT DATA FROM $ZERO TO DEVICE REG GRSTAT
1407 ;* MOV $ZERO, @GRDAI ; / PUT DATA FROM $ZERO TO DEVICE REG GRDAI
1408 ;* MOV $ZERO, @GRDIO ; / PUT DATA FROM $ZERO TO DEVICE REG GRDIO
1409
1410
1411
1412 007740 104003 DRT0: SCOPE1
1413 007742 012737 177777 001174 MOV #-1, $TMDAT ; ALL ONES TO REGISTER
1414
1415 ;* MOV $TMDAT, @GRDIO ; / PUT DATA FROM $TMDAT TO DEVICE REG GRDIO
1416
1417 ;* MOV @GRDIO, $BDDAT ; / READ DEVICE REG GRDIO, PUT DATA IN $BDDAT.
1418 007770 022737 177777 001166 CMP #-1, $BDDAT
1419 007776 001401 BEQ .+4 ; REG WILL NOT HOLD ONES
1420 010000 104400 ERROR
1421
1422 010002 104002 DRT1: SCOPE1
1423 010004 012737 177777 001174 MOV #-1, $TMDAT
1424
1425 ;* MOV $TMDAT, @GRDIO ; / PUT DATA FROM $TMDAT TO DEVICE REG GRDIO
1426 010022 004737 023054 JSR PC,$RESET ; SET DATA TO ALL ONES
1427
1428 ;* MOV @GRDIO, $BDDAT ; / READ DEVICE REG GRDIO, PUT DATA IN $BDDAT.
1429 010036 005737 001166 TST $BDDAT
1430 010042 001401 BEQ .+4 ; REG FAILED TO CLEAR
1431 010044 104400 ERROR
1432
1433 010046 052777 000100 170752 DRT2: BIS #BIT6, @TKS
1434 010054 104003 SCOPE1
1435 010056 012737 052525 001174 MOV #52525, $TMDAT
1436
1437 ;* MOV $TMDAT, @GRDIO ; / PUT DATA FROM $TMDAT TO DEVICE REG GRDIO
1438
1439 ;* MOV @GRDIO, $BDDAT ; / READ DEVICE REG GRDIO, PUT DATA IN $BDDAT.
1440 010104 022737 052525 001166 CMP #52525, $BDDAT
1441 010112 001401 BEQ .+4

```

K04

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 38

SEQ 0049

1442	010114	104400			ERROR		;DATA NOT=52525
1443							
1444	010116	104002			SCOPE		
1445	010120	012737	125252	001174	MOV	#125252,\$TMDAT	
1446							
1447					;*	MOV	\$TMDAT,@GRDIO ;/ PUT DATA FROM \$TMDAT TO DEVICE REG GRDIO
1448							
1449					;*	MOV	@GRDIO,\$BDDAT ;/READ DEVICE REG GRDIO,PUT DATA IN \$BDDAT.
1450	010146	022737	125252	001166	CMP	#125252,\$BDDAT	
1451	010154	001401			BEQ	+.4	
1452	010156	104400			ERROR		;DATA NOT=125252
1453							

1454	010160	104003			DRT4:	SCOPE1		
1455	010162	012737	177777	001174		MOV	#-1,STMDAT	
1456								
1457					;	MOV	STMDAT,@GRDIO	;/ PUT DATA FROM STMDAT TO DEVICE REG GRDIO
1458	010200	105037	001174			CLRB	STMDAT	;CLEAR LOW BYTE
1459								
1460					;	MOV	STMDAT,@GRDIO	;/ PUT DATA FROM STMDAT TO DEVICE REG GRDIO
1461								
1462					;	MOV	@GRDIO,TEMP	;/READ DEVICE REG GRDIO,PUT DATA IN TEMP.
1463	010224	022737	177400	016674		CMP	#177400,TEMP	
1464	010232	001401				BEQ	.+4	
1465	010234	104400				ERROR		;BYTE LOW FAILED TO CLEAR
1466								
1467	010236	104002			DRT5:	SCOPE		
1468	010240	012737	177777	001174		MOV	#-1,STMDAT	
1469								
1470					;	MOV	STMDAT,@GRDIO	;/ PUT DATA FROM STMDAT TO DEVICE REG GRDIO
1471	010256	105037	001175			CLRB	STMDAT+1	;CLEAR HIGH BYTE
1472								
1473					;	MOV	STMDAT,@GRDIO	;/ PUT DATA FROM STMDAT TO DEVICE REG GRDIO
1474								
1475					;	MOV	@GRDIO,TEMP	;/READ DEVICE REG GRDIO,PUT DATA IN TEMP.
1476	010302	022737	000377	016674		CMP	#377,TEMP	
1477	010310	001401				BEQ	.+4	
1478	010312	104400				ERROR		;HIGH BYTE CLEAR FAILED
1479								
1480	010314	005037	016674			CLR	TEMP	
1481	010320	104002			DRT6:	SCOPE		
1482	010322	113737	016674	001174	DRT6A:	MOVB	TEMP,STMDAT	;LOAD THE OUTPUT
1483								
1484					;	MOV	STMDAT,@GRDIO	;/ PUT DATA FROM STMDAT TO DEVICE REG GRDIO
1485								
1486					;	MOV	@GRDIO,\$BDDAT	;/READ DEVICE REG GRDIO,PUT DATA IN \$BDDAT.
1487	010350	123737	016674	001166		CMPB	TEMP,\$BDDAT	
1488	010356	001402				BEQ	.+6	;BRANCH IF EQUAL
1489	010360	104400				ERROR		;ERROR, LOW BYTE HAS BAD DATA
1490	010362	000405				BR	DRT7	
1491	010364	105237	016674			INCB	TEMP	
1492	010370	001354				BNE	DRT6A	
1493								
1494	010372	005037	016674			CLR	TEMP	
1495	010376	104002			DRT7:	SCOPE		
1496	010400	113737	016674	001175	DRT7A:	MOVB	TEMP,STMDAT+1	;LOAD THE HIGH BYTE
1497								
1498					;	MOV	STMDAT,@GRDIO	;/ PUT DATA FROM STMDAT TO DEVICE REG GRDIO
1499								
1500					;	MOV	@GRDIO,\$BDDAT	;/READ DEVICE REG GRDIO,PUT DATA IN \$BDDAT.
1501	010426	123737	016674	001167		CMPB	TEMP,\$BDDAT+1	
1502	010434	001402				BEQ	.+6	;BRANCH IF EQUAL
1503	010436	104400				ERROR		;ERROR, HIGH BYTE IN ERROR
1504	010440	000403				BR	DRT8	
1505	010442	105237	016674			INCB	TEMP	
1506	010446	001354				BNE	DRT7A	


```
1507
1508 ;RELAY #1 TEST
1509
1510 010450 104003
1511 010452 012737 000001 001174 DRT8: SCOPE1
1512 ;* MOV #BIT0,$TMDAT ;SET BIT 0
1513 ;* MOV $TMDAT,$GRSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRSTAT
1514 ;* MOV $GRSTAT,$BDDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN $BDDAT.
1515 010500 022737 000001 001166 ;* CMP #BIT0,$BDDAT ;TEST IT
1516 010506 001401 BEQ .+4
1517 010510 104400 ERROR ;ERROR, RELAY 1 FAILED TO SET
1518
1519 ;RELAY #2 TEST
1520
1521 010512 104002
1522 010514 012737 000400 001174 DRT9: SCOPE
1523 ;* MOV #BIT8,$TMDAT ;SET RELAY 2
1524 ;* MOV $TMDAT,$GRSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRSTAT
1525 ;* MOV $GRSTAT,$BDDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN $BDDAT.
1526 010542 022737 000400 001166 ;* CMP #BIT8,$BDDAT
1527 010550 001401 BEQ .+4
1528 010552 104400 ERROR ;ERROR, RELAY 2 FAILED TO SET
1529
1530 ;TEST OUTPUT DATA ACCEPT FLAG
1531
1532 010554 104002
1533 010556 012737 100000 001174 DRT10: SCOPE
1534 ;* MOV #BIT15,$TMDAT ;SET BIT 15
1535 ;* MOV $TMDAT,$GRSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRSTAT
1536 ;* MOV $GRSTAT,$BDDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN $BDDAT.
1537 010604 022737 100000 001166 ;* CMP #BIT15,$BDDAT
1538 010612 001401 BEQ .+4
1539 010614 104400 ERROR ;ERROR, BIT 15 FAILED TO SET
1540
1541 ;TEST OUTPUT INTERRUPT ENABLE
1542
1543 010616 104002
1544 010620 012777 000340 170174 DRT11: SCOPE
1545 010626 012737 040000 001174 ;* MOV #340,$PSW ;RAISE PRIORITY
1546 ;* MOV #BIT14,$TMDAT ;LOAD BIT 14
1547 ;* MOV $TMDAT,$GRSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRSTAT
1548 ;* MOV $GRSTAT,$BDDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN $BDDAT.
1549 010654 022737 040000 001166 ;* CMP #BIT14,$BDDAT
1550 010662 001401 BEQ .+4
1551 010664 104400 ERROR ;ERROR BIT 14 FAILED TO SET
1552
1553
1554
1555
1556
```

```

1557
1558 ;TEST INPUT DATA READY FLAG
1559
1560 010666 104002 DRT12: SCOPE
1561 010670 012737 000200 001174 MOV #BIT7,$TMDAT ;SET BIT 7
1562
1563 ;* MOV $TMDAT,$GRSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRSTAT
1564
1565 ;* MOV $GRSTAT,$BDDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN $BDDAT.
1566 010716 022737 000200 001166 CMP #BIT7,$BDDAT
1567 010724 001401 BEQ .+4
1568 010726 104400 ERROR ;ERROR, BIT 7 FAILED TO SET
1569
1570 ;TEST INPUT INTERRUPT ENABLE
1571
1572 010730 104002 DRT13: SCOPE
1573 010732 012777 000340 170062 MOV #340,$PSW
1574 010740 012737 000100 001174 MOV #BIT6,$TMDAT ;SET BIT 6
1575
1576 ;* MOV $TMDAT,$GRSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRSTAT
1577
1578 ;* MOV $GRSTAT,$BDDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN $BDDAT.
1579 010766 022737 000100 001166 CMP #BIT6,$BDDAT
1580 010774 001401 BEQ .+4
1581 010776 104400 ERROR ;ERROR, BIT 6 FAILED TO SET
1582
1583 ;TEST THAT RESET CLEARS THE DIGITAL STATUS REGISTER
1584
1585 011000 104002 DRT14: SCOPE
1586 011002 012777 000340 170012 MOV #340,$PSW
1587 011010 012737 140701 001174 MOV #BIT15:BIT14:BIT8:BIT7:BIT6:BIT0,$TMDAT
1588
1589 ;* MOV $TMDAT,$GRSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRSTAT
1590 011026 004737 023054 JSR PC,$RESET
1591
1592 ;* MOV $GRSTAT,$BDDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN $BDDAT.
1593 011042 042737 100000 001166 BIC #BIT15,$BDDAT
1594 011050 005737 001166 TST $BDDAT
1595 011054 001401 BEQ .+4
1596 011056 104400 ERROR ;ERROR, RESET FAILED TO CLEAR DIGITAL STATUS REG
1597
1598 011060 005077 167736 CLR $PSW
1599 011064 052777 000100 167734 BIS #BIT6,$TKS

```



```

1600 ;TEST EXTERNAL TRANSFERS - CABLE MUST BE CONNECTED
1601 011072 104003 DRT15: SCOPE1
1602 011074 032777 000100 167734 BIT #BIT6,ASWR ;TEST SWITCH BIT
1603 011102 001402 BEQ .+6 ;BRANCH IF DOWN
1604 011104 000137 011444 JMP DRT28 ;BYPASS SOME TEST USING THE EXTERNAL CABLE
1605
1606 ;* MOV $ZERO,AGRDIO ;/ PUT DATA FROM $ZERO TO DEVICE REG GRDIO
1607
1608 ;* MOV AGRDAL,$BDDAT ;/READ DEVICE REG GRDAI,PUT DATA IN $BDDAT.
1609 011130 022737 000000 001166 CMP #0,$BDDAT ;READ THE INPUT
1610 011136 001401 BEQ .+4 ;BRANCH IF EQUAL
1611 011140 104400 ERROR ;ERROR, INPUT DID NOT EQUAL THE OUTPUT REG.
1612
1613 011142 104002 DRT16: SCOPE
1614 011144 012737 177777 001174 MOV #-1,$TMDAT ;LOAD THE OUTPUT
1615
1616 ;* MOV $TMDAT,AGRDIO ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRDIO
1617
1618 ;* MOV AGRDIO,$BDDAT ;/READ DEVICE REG GRDIO,PUT DATA IN $BDDAT.
1619 011172 022737 177777 001166 CMP #-1,$BDDAT ;READ THE INPUT
1620 011200 001401 BEQ .+4 ;BRANCH IF EQUAL
1621 011202 104400 ERROR ;ERROR, INPUT DID NOT EQUAL THE OUTPUT
1622 ;IS THIS REALLY A LPS-11-DR ?? OR IS IT AN
1623 ;LPS-11-DRA ?? IF DRA USE MD-11-DZLPI TEST
1624
1625 011204 104002 DRT17: SCOPE
1626 011206 012737 052525 001174 MOV #52525,$TMDAT ;LOAD THE OUTPUT
1627
1628 ;* MOV $TMDAT,AGRDIO ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRDIO
1629
1630 ;* MOV AGRDAL,$BDDAT ;/READ DEVICE REG GRDAI,PUT DATA IN $BDDAT.
1631 011234 022737 052525 001166 CMP #52525,$BDDAT ;READ THE INPUT
1632 011242 001401 BEQ .+4 ;BRANCH IF EQUAL
1633 011244 104400 ERROR ;ERROR, INPUT DID NOT EQUAL OUTPUT
1634
1635 011246 104002 DRT18: SCOPE
1636 011250 012737 025252 001174 MOV #25252,$TMDAT ;LOAD THE OUTPUT
1637
1638 ;* MOV $TMDAT,AGRDIO ;/ PUT DATA FROM $TMDAT TO DEVICE REG GRDIO
1639
1640 ;* MOV AGRDAL,$BDDAT ;/READ DEVICE REG GRDAI,PUT DATA IN $BDDAT.
1641 011276 022737 025252 001166 CMP #25252,$BDDAT ;READ THE INPUT
1642 011304 001401 BEQ .+4 ;BRANCH IF EQUAL
1643 011306 104400 ERROR ;ERROR, INPUT DID NOT EQUAL OUTPUT
1644
1645 ;TEST THAT WHEN OUTPUTTING THE INPUT DATA READY FLAG SETS
1646
1647 011310 104003 DRT19: SCOPE1
1648
1649 ;* MOV $ZERO,AGRSTAT ;/ PUT DATA FROM $ZERO TO DEVICE REG GRSTAT
1650
1651 ;* MOV $ZERO,AGRDIO ;/ PUT DATA FROM $ZERO TO DEVICE REG GRDIO
1652 ;OUTPUT 0
1653 011332 022727 000000 000000 CMP #0,#0 ;DELAY

```



```

1654
1655
1656 011350 105737 001166
1657 011354 100401
1658 011356 104400
1659
1660
1661
1662 011360 104002
1663
1664
1665
1666
1667 011402 022727 000000 000000
1668
1669
1670 011420 013700 001174
1671
1672
1673 011434 005737 001166
1674 011440 100401
1675 011442 104400

;*      MOV      @GRSTAT,$BDDAT  ;/READ DEVICE REG GRSTAT,PUT DATA IN $BDDAT.
      TSTB      $BDDAT
      BMI       .+4
      ERROR
;INPUT DATA READY FLAG FAILED TO SET

;TEST THAT WHEN THE INPUT BUFFER IS READ THE OUTPUT FLAG IS SET
DRT20:  SCOPE
;*      MOV      $ZERO,@GRSTAT  ;/ PUT DATA FROM $ZERO TO DEVICE REG GRSTAT
;*      MOV      $ZERO,@GRDIO   ;/ PUT DATA FROM $ZERO TO DEVICE REG GRDIO
      CMP      #0,#0
;*      MOV      @GRDAI,$TMDAT  ;/READ DEVICE REG GRDAI,PUT DATA IN $TMDAT.
      MOV      $TMDAT,$R0
;*      MOV      @GRSTAT,$BDDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN $BDDAT.
      TSTB      $BDDAT
      BMI       .+4
      ERROR
;INPUT DATA READY FLAG FAILED TO SET

```

```

1676
1677 011444 104003
1678 011446 042737 177437 001010
1679 011454 001001
1680 011456 000000
1681 011460 022737 000340 001010
1682 011466 001001
1683 011470 000000
1684 011472 013737 001010 016702
1685 011500 162737 000040 016702
1686 011506 013737 001010 016704
1687 011514 013737 001010 016706
1688 011522 062737 000040 016706
1689
1690
1691
1692
1693 011530 004737 023054
1694 011534 005737 000042
1695 011540 001402
1696 011542 000137 002142
1697 011546 004737 015650
1698 011552 005237 001042
1699 011556 000137 007656

;PRE INTERRUPT SETUP
DRT28: SCOPE1
      BIC      #177437,DIOBRL
      BNE      .+4
      HALT
      CMP      #340,DIOBRL
      BNE      .+4
      HALT
      MOV      DIOBRL,BRLEV1
      SUB      #40,BRLEV1
      MOV      DIOBRL,BRLEV2
      MOV      DIOBRL,BRLEV3
      ADD      #40,BRLEV3

;BR LEVEL INDICATED FOR DIGITAL I/O WAS 0
;BR LEVEL INDICATED FOR DIGITAL I/O WAS 7

;LOGICAL END OF 'DIGITAL I/O LOGIC TEST'

      JSR      PC,$RESET
      TST      @#42
      BEQ      .+6
      JMP      WHATC
      JSR      7,BELL
      INC      PASSCT
      JMP      IOTSTA

;REPORT END OF PASS

```

```

1700 ;*****
1701 ; LPSVC POINT PLOT SCOPE CONTROL
1702 ;*****
1703
1704 011562 004737 023054 VCTEST: JSR PC,$RESET
1705 011566 104000 PRINT
1706 011570 016276 MESS ;IDENTIFY TEST
1707 011572 005037 001042 VTEST1: CLR PASSCT
1708
1709 011576 013706 016632 VCTST0: MOV STACK,SP
1710 011602 013777 001042 167230 MOV PASSCT,$DISPLA
1711 011610 012737 011662 015572 MOV $VCT0+2,RETURN
1712 011616 005077 167200 CLR $PSW
1713 011622 052777 000100 167176 BIS $BIT6,$TKS
1714
1715 ;TEST FOR NO BUSS ERRORS
1716
1717
1718 ;* MOV $ZERO,$VCSTAT ;/ PUT DATA FROM $ZERO TO DEVICE REG VCSTAT
1719 ;* MOV $ZERO,$VCXREG ;/ PUT DATA FROM $ZERO TO DEVICE REG VCXREG
1720 ;* MOV $ZERO,$VCYREG ;/ PUT DATA FROM $ZERO TO DEVICE REG VCYREG
1721
1722 ;TEST THAT RESET SETS READY BIT
1723
1724
1725
1726 011660 104002 VCT0: SCOPE
1727 011662 004737 023054 JSR PC,$RESET
1728
1729 ;* MOV $VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1730 011676 105737 001166 TSTB $BDDAT
1731 011702 100401 BMI .+4
1732 011704 104400 ERROR ;ERROR, VCSTAT NOT = 200
1733
1734 ;TEST THAT FAST INTENSIFY (BIT 1) CAN BE SET AND CLEARED
1735
1736
1737 011706 104003 VCT1: SCOPE1
1738 011710 052777 000100 167110 BIS $BIT6,$TKS
1739 011716 012737 000002 001174 MOV $BIT1,$TMDAT ;LOAD DISPLAY STATUS
1740
1741 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
1742 ;* MOV $VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1743 011744 022737 000202 001166 CMP $BIT7:$BIT1,$BDDAT
1744 011752 001401 BEQ .+4
1745 011754 104400 ERROR ;ERROR, VC STATUS NOT = 202
1746
1747 ;TEST THAT MODE (BIT 2) CAN BE SET AND CLEARED
1748
1749
1750 011756 104002 VCT2: SCOPE
1751 011760 012737 000004 001174 MOV $BIT2,$TMDAT ;LOAD DISPLAY STATUS
1752
1753 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT

```



```

1754
1755 012006 022737 000204 001166 ;* MOV @VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1756 012014 001401 CMP #BIT7:BIT2,$BDDAT
1757 012016 104400 BEQ .+4 ;ERROR, VC STATUS NOT = 204
1758
1759 ;TEST THAT MODE (BIT 3) CAN BE SET AND CLEARED
1760
1761 012020 104002 VCT3: SCOPE
1762 012022 012737 000010 001174 MOV #BIT3,$TMDAT ;LOAD DISPLAY STATUS
1763
1764 ;* MOV $TMDAT,@VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
1765
1766 ;* MOV @VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1767 012050 022737 000210 001166 CMP #BIT7:BIT3,$BDDAT
1768 012056 001401 BEQ .+4 ;ERROR, VC STATUS NOT = 210
1769 012060 104400 ERROR
1770

```

```

1771
1772 ;TEST THAT EXT DEL (BIT 4) CAN BE SET AND CLEARED
1773
1774 012062 104002
1775 012064 012737 000020 001174 VCT4: SCOPE
1776 ;* MOV #BIT4,$TMDAT ;LOAD DISPLAY STATUS
1777 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
1778 ;* MOV $VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1779 012112 022737 000220 001166 ;* CMP #BIT7:BIT4,$BDDAT
1780 012120 001401 BEQ .+4
1781 012122 104400 ERROR ;ERROR, VC STATUS NOT = 220
1782
1783 ;TEST THAT INTERRUPT ENABLE (BIT 6) CAN BE SET AND CLEARED
1784
1785 012124 104002
1786 012126 012737 000100 001174 VCT5: SCOPE
1787 ;* MOV #BIT6,$TMDAT ;LOAD DISPLAY STATUS
1788 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
1789 ;* MOV $VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1790 012154 022737 000300 001166 ;* CMP #BIT7:BIT6,$BDDAT
1791 012162 001401 BEQ .+4
1792 012164 104400 ERROR ;ERROR, VC STATUS NOT = 300
1793
1794 ;TEST THAT CHANNEL (BIT 9) CAN BE SET AND CLEARED
1795
1796 012166 104002
1797 012170 012737 001000 001174 VCT6: SCOPE
1798 ;* MOV #BIT9,$TMDAT ;LOAD DISPLAY STATUS
1799 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
1800 ;* MOV $VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1801 012216 022737 001200 001166 ;* CMP #BIT9:BIT7,$BDDAT
1802 012224 001401 BEQ .+4
1803 012226 104400 ERROR ;ERROR, VC STATUS NOT = 1200
1804
1805 ;TEST THAT STORE (BIT 10) CAN BE SET AND CLEARED
1806
1807 012230 104002
1808 012232 012737 002000 001174 VCT7: SCOPE
1809 ;* MOV #BIT10,$TMDAT ;LOAD DISPLAY STATUS
1810 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
1811 ;* MOV $VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1812 012260 022737 002200 001166 ;* CMP #BIT10:BIT7,$BDDAT
1813 012266 001401 BEQ .+4
1814 012270 104400 ERROR ;ERROR, VC STATUS NOT = 2200
1815
1816 ;TEST THAT WRITE THRU (BIT 11) CAN BE SET AND CLEARED
1817
1818 012272 104002
1819 012274 012737 004000 001174 VCT8: SCOPE
1820 ;* MOV #BIT11,$TMDAT ;LOAD DISPLAY STATUS
1821
1822
1823
1824

```

H05

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 48

SEQ 0059

1825 ;*
1826
1827 ;*
1828 012322 022737 004200 001166
1829 012330 001401
1830 012332 104400
1831

MOV STMDAT,@VCSTAT ;/ PUT DATA FROM STMDAT TO DEVICE REG VCSTAT
MOV @VCSTAT,\$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN \$BDDAT.
CMP #BIT11:BIT7,\$BDDAT
BEQ .+4
ERROR ;ERROR, VC STATUS NOT = 4200


```

1832
1833
1834
1835 012334 104002
1836 012336 012737 002525 001174 VCT11: SCOPE
1837
1838
1839
1840
1841 012364 022737 002525 001166
1842 012372 001401
1843 012374 104400
1844
1845 012376 104002
1846 012400 012737 005252 001174 VCT12: SCOPE
1847
1848
1849
1850
1851 012426 022737 005252 001166
1852 012434 001401
1853 012436 104400
1854
1855
1856
1857 012440 104002
1858 012442 012737 002525 001174 VCT15: SCOPE
1859
1860
1861
1862
1863 012470 022737 002525 001166
1864 012476 001401
1865 012500 104400
1866
1867 012502 104002
1868 012504 012737 005252 001174 VCT16: SCOPE
1869
1870
1871
1872
1873 012532 022737 005252 001166
1874 012540 001401
1875 012542 104400
1876
1877
1878
1879 012544 104002
1880 012546 012737 001234 001174 VCT17: SCOPE
1881
1882
1883 012564 012737 004321 001174
1884
1885

```

;TEST THAT THE X REGISTER (BITS 0-11) CAN BE SET AND CLEARED
 ;* MOV \$TMDAT, @VCXREG ;/ PUT DATA FROM \$TMDAT TO DEVICE REG VCXREG
 ;* MOV @VCXREG, \$BDDAT ;/ READ DEVICE REG VCXREG, PUT DATA IN \$BDDAT.
 ;* CMP #2525, \$BDDAT
 ;* BEQ .+4
 ;* ERROR ;ERROR, VC X REGISTER NOT = 2525
 ;TEST THAT THE Y REGISTER (BITS 0-11) CAN BE SET AND CLEARED
 ;* MOV \$TMDAT, @VCYREG ;/ PUT DATA FROM \$TMDAT TO DEVICE REG VCYREG
 ;* MOV @VCYREG, \$BDDAT ;/ READ DEVICE REG VCYREG, PUT DATA IN \$BDDAT.
 ;* CMP #2525, \$BDDAT
 ;* BEQ .+4
 ;* ERROR ;ERROR, VC Y REGISTER NOT = 2525
 ;TEST THAT THE X-Y REGISTER CAN HOLD DIFFERENT DATA
 ;* MOV \$TMDAT, @VCXREG ;/ PUT DATA FROM \$TMDAT TO DEVICE REG VCXREG
 ;* MOV #4321, \$TMDAT ;LOAD X REGISTER
 ;* MOV \$TMDAT, @VCYREG ;/ PUT DATA FROM \$TMDAT TO DEVICE REG VCYREG

1886									
1887									
1888	012612	022737	001234	001166	;	*	MOV	@VCXREG,\$BDDAT	;/READ DEVICE REG VCXREG,PUT DATA IN \$BDDAT.
1889	012620	001401					CMP	#1234,\$BDDAT	
1890	012622	104400					BEQ	+.4	
1891							ERROR		;ERROR, SELECT X REGISTER INCORRECTLY
1892									
1893									
1894	012634	022737	004321	001166	;	*	MOV	@VCYREG,\$BDDAT	;/READ DEVICE REG VCYREG,PUT DATA IN \$BDDAT.
1895	012642	001401					CMP	#4321,\$BDDAT	
1896	012644	104400					BEQ	+.4	
1897							ERROR		;ERROR, SELECTED Y REGISTER INCORRECTLY

```

1898
1899
1900 ;TEST THAT WHEN INTENSIFY BIT IS SET THAT THE READY BIT SETS
1901 012646 104002 SCOPE
1902 012650 012700 000100 MOV #100,R0
1903 012654 012737 000001 001174 MOV #BIT0,$TMDAT ;INTENSIFY
1904
1905 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
1906 012672 VCTST2:
1907
1908 ;* MOV $VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1909 012702 105737 001166 TSTB $BDDAT
1910 012706 100403 BMI VCTST3 ;SET EXIT
1911 012710 005300 DEC R0 ;DELAY
1912 012712 001367 BNE VCTST2
1913 012714 104400 ERROR ;READY FAILED TO SET AFTER A DELAY
1914
1915 ;TEST THAT WHEN COLOR IS CHANGED READY SETS AFTER A DELAY
1916
1917 012716 104003 VCTST3: SCOPE1
1918 012720 012700 000400 MOV #400,R0 ;SET UP A DELAY
1919 012724 012737 000400 001174 MOV #BIT8,$TMDAT ;CHANGE TO RED
1920
1921 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
1922
1923 ;* MOV $VCSTAT,$TMDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $TMDAT.
1924 012752 032737 000400 001174 BIT #BIT8,$TMDAT
1925 012760 001001 BNE .+4 ;IS IT SET ?
1926 012762 104400 ERROR ;ERROR, COLOR BIT FAILED TO SET
1927 012764
1928 VCTST4:
1929 ;* MOV $VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1930 012774 105737 001166 TSTB $BDDAT
1931 013000 100403 BMI VCTST5 ;IS IT SET ?
1932 013002 005300 DEC R0 ;NO, DELAY
1933 013004 001367 BNE VCTST4
1934 013006 104400 ERROR ;ERROR, READY FAILED TO SET AFTER A COLOR CHANGE
1935
1936 ;TEST THAT WHEN COLOR IS CHANGED READY SETS AFTER A DELAY
1937
1938 013010 104001 VCTST5: SCOPE0
1939 013012 012700 020000 MOV #20000,R0 ;SET UP A DELAY
1940
1941 ;* MOV $ZERO,$VCSTAT ;/ PUT DATA FROM $ZERO TO DEVICE REG VCSTAT
1942
1943 ;* MOV $VCSTAT,$TMDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $TMDAT.
1944 013036 032737 000400 001174 BIT #BIT8,$TMDAT
1945
1946 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
1947
1948 ;* MOV $VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1949 013064 105737 001166 TSTB $BDDAT
1950 013070 100001 BPL .+4 ;DID IT SET
1951 013072 104400 ERROR ;ERROR, READY FAILED TO CLEAR

```


1952 013074

VCTST6:

1953

;*

1954

MOV @VCSTAT,\$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN \$BDDAT.

1955 013104 105737 001166

TSTB \$BDDAT

1956 013110 100403

BMI VCTST7

1957 013112 005300

DEC R0

1958 013114 001367

BNE VCTST6

1959 013116 104400

ERROR

;ERROR, READY FAILED TO SET AFTER A COLOR CHANGE

```
1960
1961 ;TEST THAT MODE 1 (INTENSIFY ON X)
1962 ;CLEARS THE READY FLAG AND THEN SETS IT
1963
1964 013120 104001
1965 013122 012700 000100
1966 013126 012737 000004 001174 VCTST7: SCOPE0
1967 ;MOV #100,R0 ;SET UP DELAY
1968 ;MOV #BIT2,STMDAT ;LOAD MODE 1
1969
1970 ;* MOV STMDAT,AVCSTAT ;/ PUT DATA FROM STMDAT TO DEVICE REG VCSTAT
1971 ;* MOV AVCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1972 013154 105737 001166 TSTB $BDDAT
1973 013160 100401 BMI .+4
1974 013162 104400 ERROR ;ERROR, MODE 1 SHOULD NOT CLEAR READY UNTIL X IS
1975
1976 ;* MOV $ZERO,AVCXREG ;/ PUT DATA FROM $ZERO TO DEVICE REG VCXREG
1977
1978 013174 VCTST8:
1979
1980 ;* MOV AVCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1981 013204 105737 001166 TSTB $BDDAT
1982 013210 100403 BMI VCTST9 ;SET, NEXT TEST
1983 013212 005300 DEC R0 ;DELAY
1984 013214 001367 BNE VCTST8 ;TEST READY AGAIN
1985 013216 104400 ERROR ;ERROR, READY FAILED TO SET
1986 ; AFTER MODE 1 OPERATION
1987
1988 ;TEST THAT MODE 2 (INTENSIFY ON Y)
1989 ;CLEARS THE READY FLAG AND THEN SETS IT
1990
1991 013220 104003
1992 013222 012700 000100
1993 013226 012737 000010 001174 VCTST9: SCOPE1
1994 ;MOV #100,R0 ;SET UP DELAY
1995 ;MOV #BIT3,STMDAT ;LOAD MODE 2
1996
1997 ;* MOV STMDAT,AVCSTAT ;/ PUT DATA FROM STMDAT TO DEVICE REG VCSTAT
1998 ;* MOV AVCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
1999 013254 005737 001166 TST $BDDAT
2000 013260 100001 BPL .+4
2001 013262 104400 ERROR ;ERROR, MODE 2 SHOULD NOT CLEAR READY UNTIL Y IS
2002
2003 ;* MOV $ZERO,AVCYREG ;/ PUT DATA FROM $ZERO TO DEVICE REG VCYREG
2004
2005 013274 VCTS10:
2006
2007 ;* MOV AVCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
2008 013304 105737 001166 TSTB $BDDAT
2009 013310 100403 BMI VCTS11 ;SET, NEXT TEST
2010 013312 005300 DEC R0 ;DELAY
2011 013314 001367 BNE VCTS10 ;TEST READY AGAIN
2012 013316 104400 ERROR ;ERROR, READY FAILED TO SET
2013 ; AFTER MODE 2 OPERATION
```

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

N05
MACY11 27(654) 15-DEC-77 08:35 PAGE 54

SEQ 0065

2014


```

2015 ;TEST THAT WHEN ERASE BIT IS SET THAT THE READY BIT
2016 ;CLEAR AND THEN SETS AFTER A DELAY
2017
2018 013320 104003 VCTS11: SCOPE1
2019 013322 032777 000040 165506 BIT #BITS,ASWR
2020 013330 001444 BEQ AVCT12
2021 013332 012700 000002 MOV #2,RO
2022 013336 005037 016674 CLR TEMP ;CLEAR DELAY
2023
2024 013352 052737 012000 001174 ;* MOV @VCSTAT,$TMDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $TMDAT.
2025 BIS #BIT12!BIT10,$TMDAT
2026
2027 ;* MOV $TMDAT,@VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2028
2029 ;* MOV @VCSTAT,$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
2030 013400 105737 001166 TSTB $BDDAT
2031 ;TEST THAT READY CLEARS
2032 013404 100001 BPL .+4
2033 013406 104400 ERROR ;ERROR, READY FAILED TO RESET
2034 ; UPON SETTING ERASE BIT
2035
2036 013410 TST11A:
2037
2038 ;* MOV @VCSTAT,$TMDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN $TMDAT.
2039 013420 105737 001174 TSTB $TMDAT
2040 013424 100406 BMI AVCT12 ;BRANCH IF SET
2041 013426 005337 016674 DEC TEMP ;DELAY
2042 013432 001366 BNE TST11A ;BRANCH IF NOT READY
2043 013434 005300 DEC RO ;DECREMENT COUNTER
2044 013436 001364 BNE TST11A ;BRANCH IF NOT COMPLETED
2045 013440 104400 ERROR ;ERROR, ERASE CLEARED READY AND FAILED
2046 ; TO SET READY AFTER A DELAY
2047
2048 013442 AVCT12:
2049 013442 VCTS13:
2050 013442 VCTS14:
2051 013442 VCTS15:

```

```

2052
2053
2054
2055 013442 042737 177437 001012 BIC #177437,VCBRL ;MASK TO PSW
2056 013450 001001 BNE .+4
2057 013452 000000 HALT ;LOCATION VCBRL CONTAILED A BR LEVEL 0
2058 013454 022737 000340 001012 CMP #340,VCBRL ;TEST FOR BR 7
2059 013462 001001 BNE .+4 ;LOCATION VCBRL CONTAINS BR LEVEL 7
2060 013464 000000 HALT
2061
2062 013466 013737 001012 016702 MOV VCBRL,BRLEV1 ;SET UP BR LEVELS
2063 013474 162737 000040 016702 SUB #40,BRLEV1 ; -1
2064 013502 013737 001012 016704 MOV VCBRL,BRLEV2 ; 0
2065 013510 013737 001012 016706 MOV VCBRL,BRLEV3 ; +1
2066 013516 062737 000040 016706 ADD #40,BRLEV3

```

```

2067
2068
2069
2070 013524 104003
2071 013526 012737 000036 001174
2072
2073
2074 013544 004737 023054
2075
2076
2077 013560 013700 001166
2078 013564 042700 000200
2079 013570 005700
2080 013572 001401
2081 013574 104400
2082
2083
2084
2085 013576 104003
2086 013600 012737 007100 001174
2087
2088
2089 013616 004737 023054
2090
2091
2092 013632 013700 001166
2093 013636 042700 000200
2094 013642 005700
2095 013644 001401
2096 013646 104400
2097
2098
2099
2100 013650 104003
2101 013652 012737 177777 001174
2102
2103
2104 013670 004737 023054
2105
2106
2107 013704 005737 001166
2108 013710 001401
2109 013712 104400
2110
2111
2112
2113 013714 104003
2114 013716 012737 177777 001174
2115
2116
2117 013734 004737 023054
2118
2119
2120 013750 005737 001166

```

;TEST THAT RESET CLEARS MODE 0,2 ,FAST INTEN. AND EXT DELAY
 SCOPE1
 MOV #BIT4!BIT3!BIT2!BIT1,\$TMDAT
 ;* MOV \$TMDAT,\$VCSTAT ;/ PUT DATA FROM \$TMDAT TO DEVICE REG VCSTAT
 JSR PC,\$RESET
 ;* MOV \$VCSTAT,\$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN \$BDDAT.
 MOV \$BDDAT,R0
 BIC #BIT7,R0
 TST R0
 BEQ .+4
 ERROR ;ERROR, RESET FAILED TO CLEAR VC STATUS REG
 ;TEST THAT RESET CLEARS INTERRUPT ENABLE, CHANNEL, STORE, WRITE THRU
 SCOPE1
 MOV #BIT11!BIT10!BIT9!BIT6,\$TMDAT
 ;* MOV \$TMDAT,\$VCSTAT ;/ PUT DATA FROM \$TMDAT TO DEVICE REG VCSTAT
 JSR PC,\$RESET
 ;* MOV \$VCSTAT,\$BDDAT ;/READ DEVICE REG VCSTAT,PUT DATA IN \$BDDAT.
 MOV \$BDDAT,R0
 BIC #BIT7,R0
 TST R0
 BEQ .+4
 ERROR ;ERROR, RESET FAILED TO CLEAR VC STATUS
 ;TEST THAT RESET CLEARS X REGISTER
 SCOPE1
 MOV #-1,\$TMDAT
 ;* MOV \$TMDAT,\$VCXREG ;/ PUT DATA FROM \$TMDAT TO DEVICE REG VCXREG
 JSR PC,\$RESET
 ;* MOV \$VCXREG,\$BDDAT ;/READ DEVICE REG VCXREG,PUT DATA IN \$BDDAT.
 TST \$BDDAT
 BEQ .+4
 ERROR ;ERROR, RESET FAILED TO CLEAR VC X REGISTER
 ;TEST THAT RESET CLEARS Y REGISTER
 SCOPE1
 MOV #-1,\$TMDAT
 ;* MOV \$TMDAT,\$VCYREG ;/ PUT DATA FROM \$TMDAT TO DEVICE REG VCYREG
 JSR PC,\$RESET
 ;* MOV \$VCYREG,\$BDDAT ;/READ DEVICE REG VCYREG,PUT DATA IN \$BDDAT.
 TST \$BDDAT

E06

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 58

SEQ 0069

2121	013754	001401	
2122	013756	104400	
2123			
2124	013760	104003	
2125	013762	004737	023054
2126	013766	005737	000042
2127	013772	001402	
2128	013774	000137	002154
2129	014000	004737	015650
2130	014004	005237	001042
2131	014010	000137	011576

BEQ .+4
ERROR

;ERROR, RESET FAILED TO CLEAR VC Y REGISTER

SCOPE1
JSR PC,\$RESET
TST 2#42
BEQ .+6
JMP WHATD
JSR 7,BELL
INC PASSCT
JMP VCTSTO

;LOGICAL END OF THIS TEST

```

2132 ;SUBROUTINE TO ISSUE N SPACES
2133 ;N IS ONE PLUS VALUE CONTAINED IN SPACEX
2134 ;SPACEX IS CLEARED WITHIN THE SUBROUTINE, SO THAT A CALL ON
2135 ;SPACE WITHOUT LOADING SPACEX ISSUES ONLY ONE SPACE
2136
2137 014014 105777 165012 XSPACE: TSTB @TPS ;WAIT FOR TTY READY
2138 014020 100375 BPL -4
2139 014022 012777 000240 165004 MOV #240,@TPB ;OUTPUT A SPACE
2140 014030 005337 014044 DEC SPACEX ;DECREMENT COUNT
2141 014034 003367 BGT XSPACE ;LOOP IF NOT DONE
2142 014036 005037 014044 CLR SPACEX ;RESET COUNT TO ZERO
2143 014042 000002 RTI ;RETURN
2144 014044 000000 SPACEX: 0
2145
2146 ;KEYBOARD SERVICE ROUTINE
2147
2148 014046 010046 XTTYIN: MOV R0,-(SP)
2149 014050 010146 MOV R1,-(SP)
2150 014052 010246 MOV R2,-(SP)
2151 014054 010346 MOV R3,-(SP)
2152 014056 010446 MOV R4,-(SP)
2153 014060 010546 MOV R5,-(SP)
2154 014062 012704 014446 NEWIN: MOV #INBUF,R4 ;SETUP CHARACTER BUFFER
2155 014066 042777 000100 BIC #BIT6,@TKS
2156 014074 005037 016640 CLR CHRCNT ;CLEAR CHARACTER COUNTER
2157 014100 005037 014446 CLR INBUF
2158 014104 005037 014450 CLR INBUF+2
2159 014110 005037 014452 CLR INBUF+4
2160 014114 005037 014454 CLR INBUF+6
2161 014120 005037 014456 CLR INBUF+10
2162 014124 005037 014460 CLR INBUF+12
2163 014130 005037 014462 CLR INBUF+14
2164 014134 105777 164666 INPUTA: TSTB @TKS ;CHARACTER READY?
2165 014140 100375 BPL INPUTA ;NO, WAIT IT OUT
2166 014142 017701 164662 MOV @TKB,R1 ;SAVE CHARACTER
2167 014146 042701 177600 BIC #177600,R1 ;STRIP PARITY BIT
2168 014152 120127 000060 CMPB R1,#60 ;IS IT A SPECIAL CHARACTER
2169 014156 100420 BMI SPCHR ;YES, TEST IT
2170 014160 122701 000173 CMPB #173,R1
2171 014164 100415 BMI SPCHR
2172 014166 010124 INPUTB: MOV R1,(R4)+ ;SAVE CHARACTER
2173 014170 005237 016640 INC CHRCNT ;INCREMENT THE CHARACTER COUNT.
2174 014174 022737 000006 016640 CMP #6,CHRCNT
2175 014202 100516 BMI SPCHRS
2176 014204 105777 164622 OUTPTA: TSTB @TPS ;ECHO CHARACTER
2177 014210 100375 BPL OUTPTA
2178 014212 110177 164616 MOVB R1,@TPB
2179 014216 000746 BR INPUTA ;WAIT FOR NEXT CHARACTER
2180
2181 ;SUBROUTINE TO TEST FOR SPECIAL CHARACTERS : 'tA','tC','G','CR','.',', ' OR 'RUBOUT'
2182
2183 014220 122701 000001 SPCHR: CMPB #1,R1 ;CHAR.='tA'
2184 014224 001013 BNE SPCHR1 ;NO, NOT 'tA'
2185 014226 104000 PRINT ;ECHO 'tA'

```

2186	014230	016545		CNTRLA		;RESTORE 'SP'
2187	014232	012605		MOV	(SP)+,R5	
2188	014234	012604		MOV	(SP)+,R4	
2189	014236	012603		MOV	(SP)+,R3	
2190	014240	012602		MOV	(SP)+,R2	
2191	014242	012601		MOV	(SP)+,R1	
2192	014244	012600		MOV	(SP)+,R0	
2193	014246	022626		CMP	(SP)+,(SP)+	
2194	014250	000177	002360	JMP	DAVECTR	;YES, EXIT VIA 'IA' VECTOR ADDRESS.
2195	014254	122701	000003	SPCHR1: CMPB	#3,R1	;CHAR. = 'C'
2196	014260	001002		BNE	.+6	;NO, NOT 'C'
2197	014262	000137	001512	JMP	MONITR	;YES, EXIT TO MONITOR
2198	014266	122701	000177	CMPB	#177,R1	;CHAR. = 'RUBOUT'
2199	014272	001011		BNE	SPCHR3	;IGNORE CHAR. & EXIT
2200	014274	005737	016640	TST	CHRCNT	;IS RUBOUT LEGAL?
2201	014300	001715		BEQ	INPUTA	;NO, IGNORE IT
2202	014302	005337	016640	DEC	CHRCNT	
2203	014306	012701	000134	MOV	#134,R1	;TYPE '\ ' TO INDICATE RUBOUT
2204	014312	005744		TST	-(R4)	;POP OFF LAST CHARACTER
2205	014314	000733		BR	OUTPTA	;WAIT FOR NEXT CHARACTER
2206	014316	122701	000054	SPCHR3: CMPB	#54,R1	;TEST FOR ' '
2207	014322	001721		BEQ	INPUTB	;LEGAL CHAR., SAVE IT
2208	014324	122701	000015	SPCHR4: CMPB	#15,R1	;=TO 'CARRIAGE RETURN' TO TERMINATE?
2209	014330	001011		BNE	1\$	;NO, CONTINUE
2210	014332	104000		PRINT		;YES, TYPE 'CR-LF'
2211	014334	016577		CRLF		
2212	014336	012605		4\$: MOV	(SP)+,R5	
2213	014340	012604		MOV	(SP)+,R4	
2214	014342	012603		MOV	(SP)+,R3	
2215	014344	012602		MOV	(SP)+,R2	
2216	014346	012601		MOV	(SP)+,R1	
2217	014350	012600		MOV	(SP)+,R0	
2218	014352	000002		RTI		;EXIT
2219	014354	122701	000007	1\$: CMPB	#7,R1	;TEST IF CTRL G
2220	014360	001027		BNE	SPCHR5 ;BR IF NOT	
2221	014362	104000		PRINT		
2222	014364	016550		CNTRLG		
2223	014366	104005		PRTCT		;REPORT OLD
2224	014370	000170		SOFTSW		
2225	014372	104000		PRINT		
2226	014374	016566		NEWSWR		;ASK FOR NEW
2227	014376	104006		TTYIN		
2228	014400	005001		CLR	R1	;CLEAR NEW
2229	014402	012700	014446	MOV	#INBUF,R0	;LOAD NEW POINTER
2230	014406	005710		2\$: TST	(R0)	;TEST FOR TERM
2231	014410	001410		BEQ	3\$	;BR IF TERM
2232	014412	012002		MOV	(R0)+,R2	;GET A VALUE
2233	014414	042702	177770	BIC	#177770,R2	;MASK
2234	014420	006301		ASL	R1	
2235	014422	006301		ASL	R1	
2236	014424	006301		ASL	R1	
2237	014426	060201		ADD	R2,R1	
2238	014430	000766		BR	2\$	
2239	014432	010137	000170	3\$: MOV	R1,SOFTSW	;SAVE SWITCH VALUE


```

2240 014436 000737 BR 4$
2241
2242 014440 104000
2243 014442 016604
2244 014444 000606
2245 014446 000000
2246
2247
2248 014464
2249
2250
2251 014464 010046
2252 014466 010146
2253 014470 010246
2254 014472 010346
2255 014474 010446
2256 014476 010546
2257 014500 013746 000024
2258 014504 010637 016636
2259 014510 012737 014520 000024
2260 014516 000000
2261
2262
2263
2264 014520 012777 000340 164274
2265 014526 013706 016636
2266 014532 012637 000024
2267 014536 012605
2268 014540 012604
2269 014542 012603
2270 014544 012602
2271 014546 012601
2272 014550 012600
2273 014552 004737 023054
2274 014556 104000
2275 014560 016607
2276 014562 000137 001512

;POWER FAIL HANDLER
PWRFAL: MOV R0,-(SP)
MOV R1,-(SP)
MOV R2,-(SP)
MOV R3,-(SP)
MOV R4,-(SP)
MOV R5,-(SP)
MOV 24,-(SP)
MOV SP,PROC
MOV #PWRUP,24
HALT

;POWER UP HANDLER
PWRUP: MOV #340,SPSW
MOV PROC,SP
MOV (SP)+,24
MOV (SP)+,R5
MOV (SP)+,R4
MOV (SP)+,R3
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,R0
JSR PC,$RESET
PRINT
MES21
JMP MONITR

```

;OTHERWISE TYPE '?'

; WAIT FOR NEW ENTRY
; CHARACTER STORAGE BUFFER

```

2277
2278
2279      ; EMT DISPATCH SERVICE ROUTINE
2280      ; ARGUMENT OF EMT IS EXTRACTED AND USED AS OFFSET TO OBTAIN POINTER
2281      ; TO THE SELECTED SUBROUTINE.
2282
2283 014566 011646      EMTSRV: MOV      (SP), -(SP)      ; GET PC FOR TO RETURN
2284 014570 162716 000002 SUB      #2, (SP)      ; PC OF EMT
2285 014574 017616 000000 MOV      @ (SP), (SP)      ; GET EMT
2286 014600 005716      TST      (SP)      ; IS EMT VALID?
2287 014602 001001      BNE      EMTOK
2288 014604 000000      HALT
2289 014606 006316      EMTOK: ASL      (SP)      ; INVALID EMT
2290 014610 042716 177001 BIC      #177001, (SP)      ; MULTIPLY EMT ARG BY '2'
2291 014614 062716 014626 ADD      #EMTTAB, (SP)      ; CLEAR UNWANTED BITS
2292 014620 017616 000000 MOV      @ (SP), (SP)      ; POINTER TO SUBROUTINE ADDRESS
2293 014624 000136      JMP      @ (SP)+      ; SUBROUTINE ADDRESS
2294      ; GO TO SUBROUTINE
2295
2296      ; EMT DISPATCH TABLE
2297
2297 014626 014650      EMTTAB: TYPMES      ; MESSAGE PRINT ROUTINE
2298 014630 015634      SCOPEI      ; SCOPE ROUTINE
2299 014632 015500      SCOPEC      ; LOGIC TEST SCOPE ROUTINE
2300 014634 015574      SCOPEH      ; LOGIC TEST SCOPE LOOP (10)
2301 014636 014014      XSPACE      ; SUBROUTINE TO TYPE SPACES
2302 014640 015006      OCTPRT      ; OCTAL PRINT ROUTINE
2303 014642 014046      XTTYIN      ; TELEPRINTER SERVICE ROUTINE
2304 014644 015142      TKSFLG      ; SUBROUTINE TO TEST FOR KEYBOARD FLAG
2305 014646 015634      SCOPEI      ; SCOPE ROUTINE
2306

```



```
2361 015116 005337 015132 DEC 10$
2362 015122 001350 BNE 1$
2363 015124 013705 015140 MOV SAV5,R5
2364 015130 000002 RTI
2365 015132 000000 10$: 0
2366 015134 000000 11$: 0
2367 015136 000376 MASK: 376
2368 015140 000000 SAV5: 0
2369 ;SUBROUTINE TO TEST FOR THE KEYBOARD FLAG BEING SET
2370 015142 105777 163660 TKSFLG: TSTB JTKS ;FLAG SET?
2371 015146 100001 BPL .+4 ;NO, EXIT
2372 015150 104006 TTYIN ;YES, INQUIRE
2373 015152 000002 RTI
2374
2375 ;ENTERED WITH SYSTEM TRAP CALL (ERROR)
2376 015154 104007 LOGERR: TSTTKS ;TEST FOR KEYBOARD INTERRUPT
2377 015156 037727 163654 020000 BIT JSWR, #20000 ;TEST FOR INHIBIT PRINT OUT
2378 015164 001135 BNE CK ;INHIBIT, CHECK FOR HALT
2379 015166 012737 015472 000004 MOV #LGERR2, #4 ;SET UP FOR BUSS ERROR
2380 015174 011637 016650 MOV (SP), KSTOR3 ;PC OF FAILING ROUTINE
2381 015200 162737 000002 016650 SUB #2, KSTOR3
2382 015206 004537 015702 JSR 5, LEDS
2383 015212 016650 KSTOR3
2384
2385 ;* MOV JCSR, KSTOR6 ;/READ DEVICE REG CSR, PUT DATA IN KSTOR6.
2386 ;* MOV JCSB, KSTOR7 ;/READ DEVICE REG CSB, PUT DATA IN KSTOR7.
2387 ;* MOV JGRSTAT, KSTOR8 ;/READ DEVICE REG GRSTAT, PUT DATA IN KSTOR8.
2388 ;* MOV JGRDIO, KSTOR9 ;/READ DEVICE REG GRDIO, PUT DATA IN KSTOR9.
2389 ;* MOV JGRDAI, KSTR10 ;/READ DEVICE REG GRDAI, PUT DATA IN KSTR10.
2390 ;* MOV JVCSTAT, KSTOR4 ;/READ DEVICE REG VCSTAT, PUT DATA IN KSTOR4.
2391 ;* MOV JVCXREG, KSTOR5 ;/READ DEVICE REG VCXREG, PUT DATA IN KSTOR5.
2392 ;* MOV JVCYREG, KSTR12 ;/READ DEVICE REG VCYREG, PUT DATA IN KSTR12.
2393 ;* MOV TEMP, KSTR11 ;SAVE TEMP
2394 ;* MOV #6, #4 ;RESET BUSS ERROR
2395 TST PRINT1
2396 BNE LGERR1
2397 PRINT
2398 CRLF
2399 PRINT
2400 MES1
2401 INC PRINT1
2402 LGERR1: PRINT ;OUTPUT CARRIAGE RETURN AND LINE FEED
2403 CRLF ;PRINT FAILING PC+2
2404 PRTCT
2405 KSTOR3
2406 SPACE
2407 PRTCT
2408
2409 015314 013737 016674 016670
2410 015322 012737 000006 000004
2411 015330 005737 016630
2412 015334 001006
2413 015336 104000
2414 015340 016577
2415 015342 104000
2416 015344 016111
2417 015346 005237 016630
2418 015352 104000
2419 015354 016577
2420 015356 104005
2421 015360 016650
2422 015362 104004
2423 015364 104005
```

2415	015366	016656			KSTOR6			
2416	015370	104004			SPACE			
2417	015372	104005			PRTOCT			
2418	015374	016660			KSTOR7			
2419	015376	104004			SPACE			
2420	015400	104005			PRTOCT			
2421	015402	016662			KSTOR8			
2422	015404	104004			SPACE			
2423	015406	104005			PRTOCT			
2424	015410	016664			KSTOR9			
2425	015412	104004			SPACE			
2426	015414	104005			PRTOCT			
2427	015416	016666			KSTR10			
2428	015420	104004			SPACE			
2429	015422	104005			PRTOCT			
2430	015424	016652			KSTOR4			
2431	015426	104004			SPACE			
2432	015430	104005			PRTOCT			
2433	015432	016654			KSTOR5			
2434	015434	104004			SPACE			
2435	015436	104005			PRTOCT			
2436	015440	016672			KSTR12			
2437	015442	104004			SPACE			
2438	015444	104005			PRTOCT			
2439	015446	016670			KSTR11			
2440	015450	104004			SPACE			
2441	015452	105777	163354		TSTB	@TPS		
2442	015456	100375			BPL	-4		
2443	015460	005777	163352	CK:	TST	@SWR		;CHECK SR FOR HALT SWITCH
2444	015464	100001			BPL	+4		;BRANCH IF NOT SET
2445	015466	000000			HALT			;HALT ON ERROR UP
2446	015470	000002			RTI			;RETURN TO MAIN LINE
2447	015472	062716	000002	LGERR2:	ADD	#2,(SP)		
2448	015476	000002			RTI			
2449								
2450								
2451								
2452								
2453	015500	104007			SCOPEC: TSTTKS			;TEST FOR KEYBOARD INIT
2454	015502	032777	040000	163326	BIT	#40000,@SWR		;TEST SR FOR SCOPE
2455	015510	001015			BNE	SCOPEB		;YES SCOPE
2456	015512	032777	004000	163316	BIT	#4000,@SWR		;NO-TEST FOR ITERATION
2457	015520	001016			BNE	SCOPEG		;INHIBIT ITERATION
2458	015522	005737	001042		TST	PASSCT		;TEST IF FIRST PASS
2459	015526	001413			BEQ	SCOPEG		;BR IF FIRST PASS -- QUICKK PASS
2460	015530	023737	015570	001044	CMP	SCOPEF,ICOUNT		;COMPARE CURRENT COUNT TO MAX NUMBER
2461	015536	001407			SCOPEJ: BEQ	SCOPEF		;EXIT-DONE
2462	015540	005237	015570		INC	SCOPEF		;INCREMENT COUNT
2463	015544	022606			SCOPEB: CMP	(6)+,SP		;REPOSITION STACK
2464	015546	012677	163250		MOV	(6)+,@PSW		;RESTORE PREVIOUS PROCESSOR STATUS
2465	015552	000177	000014		JMP	@RETURN		;REPEAT TEST
2466	015556	005037	015570		SCOPEG: CLR	SCOPEF		;CLEAR COUNT
2467	015562	011637	015572		MOV	@SP,RETURN		;SAVE SCOPE RETURN POINTER
2468	015566	000002			RTI			;RETURN INLINE-NEXT TEST


```
2469 015570 000000          SCOPEF: 0          ;COUNT LOCATION FOR ITERATION LOOP
2470 015572 002324          RETURN: KWTO          ;ADDRESS OF LAST TEST
2471
2472          ;SCOPE AND/OR INTERATION LOOP FOR SOME TESTS 2 TIMES
2473
2474 015574 104007          SCOPEH: TSTTKS
2475 015576 032777 040000 163232      BIT      #40000,ASWR
2476 015604 001357          BNE      SCOPEB
2477 015606 032777 004000 163222      BIT      #4000,ASWR
2478 015614 001360          BNE      SCOPEG
2479 015616 005737 001042          TST      PASSCT          ;TEST IF FIRST PASS
2480 015622 001755          BEQ      SCOPEG          ;BR IF YES
2481 015624 023727 015570 000002      CMP      SCOPEF,#2
2482 015632 000741          BR      SCOPEJ
2483
2484          ;SCOPE LOOP FOR SOME TEST 1 TIME
2485
2486 015634 104007          SCOPEI: TSTTKS          ;TEST KEYBOARD
2487 015636 032777 040000 163172      BIT      #40000,ASWR
2488 015644 001337          BNE      SCOPEB
2489 015646 000743          BR      SCOPEG
2490
2491 015650 104000          BELL:  PRINT
2492 015652 015662          BELLA: ENDPAS
2493 015654 104000          PRINT
2494 015656 015677          RING
2495 015660 000207          RTS      PC          ;EXIT
2496
2497 015662 042445 042116 050040 ENDPAS: .ASCII '%END PASS  '
2498 015670 051501 020123 020040
2499 015676 100
2500 015677 007 007 100 RING: .BYTE 7,7,100
2501 .EVEN
2502
2503          ;LOAD THE LPS DISPLAY LIGHTS
2504
2505 015702 012737 000006 016020 LEDS:  MOV      #6,CNTLED
2506 015710 005037 016026          CLR      LEDSV3
2507 015714 013537 016022          MOV      2(5)+,LEDSV1
2508 015720 005737 016030          TST      NOLEDS
2509 015724 001401          BEQ      LEDSA
2510 015726 000205          RTS      R5
2511 015730 013737 016022 016024 LEDSA: MOV      LEDSV1,LEDSV2
2512 015736 042737 177770 016024      BIC      #177770,LEDSV2
2513 015744 113737 016026 016025      MOV      LEDSV3,LEDSV2+1
2514 015752 013737 016024 001174      MOV      LEDSV2,$TMDAT
2515
2516          ;*
2517          MOV      $TMDAT,$ADDBR          ;/ PUT DATA FROM $TMDAT TO DEVICE REG ADDBR
2518 015770 006237 016022          ASR      LEDSV1
2519 015774 006237 016022          ASR      LEDSV1
2520 016000 006237 016022          ASR      LEDSV1
2521 016004 005237 016026          INC      LEDSV3
2522 016010 005337 016020          DEC      CNTLED
```


N06

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 67

SEQ 0078

2523	016014	001345
2524	016016	000205
2525		
2526	016020	000006
2527	016022	000000
2528	016024	000000
2529	016026	000000
2530	016030	000000
2531		

BNE	LEDSA
RTS	5

CNTLED: 6
LEDSV1: 0
LEDSV2: 0
LEDSV3: 0
NOLEDS: 0

2532					
2533					MESSAGES
2534	016032	022445	046045	051520	TITLE: .ASCII '%%LPS DIAGNOSTIC TEST II (MAINDEC-11-DRLPI-A)'
2535	016040	042040	040511	047107	
2536	016046	051517	044524	020103	
2537	016054	042524	052123	044440	
2538	016062	020111	046450	044501	
2539	016070	042116	041505	030455	
2540	016076	026461	051104	050114	
2541	016104	026511	024501	100	
2542					
2543	016111	040	050040	020103	MES1: .ASCII " PC "
2544	016116	020040			
2545	016120	045503	052123	052101	.ASCII "CKSTAT "
2546	016126	040			
2547	016127	103	041113	043125	.ASCII "CKBUFF "
2548	016134	020106			
2549	016136	047511	052123	052101	.ASCII "IOSTAT "
2550	016144	040			
2551	016145	111	026517	052517	.ASCII "IO-OUT "
2552	016152	020124			
2553	016154	044440	026517	047111	.ASCII " IO-IN "
2554	016162	040			
2555	016163	126	051503	040524	.ASCII "VCSTAT "
2556	016170	020124			
2557	016172	041526	740530	051530	.ASCII "VCXAXS "
2558	016200	040			
2559	016201	126	054503	054101	.ASCII "VCYAXS "
2560	016206	020123			
2561	016210	052040	046505	040120	.ASCII " TEMP "
2562					
2563	016216	021045	046103	041517	MES2: .ASCII "%CLOCK LOGIC TEST%"
2564	016224	020113	047514	044507	
2565	016232	020103	042524	052123	
2566	016240	022442	100		
2567	016243	045	042042	043511	MES3: .ASCII "%DIGITAL I/O LOGIC TEST%"
2568	016250	052111	046101	044440	
2569	016256	047457	046040	043517	
2570	016264	041511	052040	051505	
2571	016272	021124	040045		
2572	016276	021045	041523	050117	MES5: .ASCII "%SCOPE LOGIC TEST%"
2573	016304	020105	047514	044507	
2574	016312	020103	042524	052123	
2575	016320	022442	100		
2576	016323	045	053042	051511	MES6: .ASCII "%VISUAL DISPLAY TEST%"
2577	016330	040525	020114	044504	
2578	016336	050123	040514	020131	
2579	016344	042524	052123	022442	
2580	016352	100			
2581	016353	045	054524	042520	MES4: .ASCII "%TYPE LETTER ' ' TO RUN DESIRED TEST:%"
2582	016360	046040	052105	042524	
2583	016366	020122	020047	020047	
2584	016374	047524	051040	047125	
2585	016402	042040	051505	051111	

2586	016410	042105	052040	051505	
2587	016416	035124	045		
2588	016421	047	023501	041475	.ASCII "'A'=CLOCK LOGIC%"
2589	016426	047514	045503	046040	
2590	016434	043517	041511	045	
2591	016441	047	023502	044475	.ASCII "'B'=I/O LOGIC%"
2592	016446	047457	046040	043517	
2593	016454	041511	045		
2594	016457	047	023503	051475	.ASCII "'C'=SCOPE CONTROL LOGIC%"
2595	016464	047503	042520	041440	
2596	016472	047117	051124	046117	
2597	016500	046040	043517	041511	
2598	016506	045			
2599	016507	047	023504	053075	.ASCII "'D'=VISUAL SCOPE DISPLAY%a"
2600	016514	051511	040525	020114	
2601	016522	041523	050117	020105	
2602	016530	044504	050123	040514	
2603	016536	022531	100		
2604					
2605	016541	136	022503	100	CNTRLC: .ASCII 'tC%a'
2606					
2607	016545	136	040101		CNTRLA: .ASCII 'tAa'
2608	016550	043536	047445	042114	CNTRLG: .ASCII 'tG%OLD SWR = a'
2609	016556	051440	051127	036440	
2610	016564	040040			
2611	016566	020040	042516	020127	NEWSWR: .ASCII ' NEW = a'
2612	016574	020075	100		
2613					
2614	016577	045	100		CRLF: .ASCII '%a'
2615					
2616	016601	045	040056		DOT: .ASCII '%.a'
2617					
2618	016604	020077	100		QMARK: .ASCII '? a'
2619	016607	045	047520	042527	MES21: .ASCII '%POWER FAILURE a'
2620	016614	020122	040506	046111	
2621	016622	051125	020105	100	
2622					
2623	016630				.EVEN
2624					
2625					

;ADDRESS AND CONSTANTS TABLE

2626
2627
2628 016630 000000
2629 016632 001000
2630 016634 001504
2631 016636 000000
2632 016640 000000
2633 016642 000000
2634 016644 000000
2635 016646 000000
2636 016650 000000
2637 016652 000000
2638 016654 000000
2639 016656 000000
2640 016660 000000
2641 016662 000000
2642 016664 000000
2643 016666 000000
2644 016670 000000
2645 016672 000000
2646 016674 000000
2647 016676 000000
2648 016700 000000
2649 016702 000000
2650 016704 000000
2651 016706 000000
2652 016710 000000
2653 016712 000000
2654 016714 000020
2655 016716 000000
2656 016720 000000

PRINT1: 0
STACK: 1000
AVECTR: INITA
PROC: 0
CHRCNT: 0
COUNT: 0
DELAY: 0
KSTOR1: 0
KSTOR3: 0
KSTOR4: 0
KSTOR5: 0
KSTOR6: 0
KSTOR7: 0
KSTOR8: 0
KSTOR9: 0
KSTR10: 0
KSTR11: 0
KSTR12: 0
TEMP: 0
TEMP1: 0
TEMP2: 0
BRLEV1: 0
BRLEV2: 0
BRLEV3: 0
LOW: 0
HIGH: 0
INCR: 20
TIMSV: 0
TICKS: 0

; INITIAL SP. ADDRESS
; 'TA' VECTOR ADDRESS
; TEMP STORAGE FOR 'PSW'
; TEMP STORAGE
; TEMP STORAGE
; PERMANENT STORAGE
; PERMANENT STORAGE
; PERMANENT STORAGE
; TEMPORARY STORAGE
; TEMPORARY STORAGE

```

2657
2658
2659
2660
2661
2662 016722 004737 023054
2663 016726 104000
2664 016730 016323
2665 016732 005037 001042
2666 016736 013706 016632
2667 016742 013777 001042 162070
2668 016750 012737 017016 015572
2669 016756 005077 162040
2670 016762 042737 000002 020156
2671 016770 032777 000020 162040
2672 016776 001003
2673 017000 052737 000002 020156
2674 017006 052777 000100 162012
2675
2676
2677 017014 013737 001066 017056
2678 017022 013737 001070 017060
2679 017030 004737 017062
2680
2681
2682
2683 017034 013737 001070 017056
2684 017042 013737 001066 017060
2685 017050 004737 017062
2686 017054 000467
2687
2688 017056 000000
2689 017060 000000
2690
2691
2692 017062
2693
2694
2695 017072 013704 001064
2696 017076 012737 000050 016720
2697 017104 004737 020634
2698 017110 012703 007760
2699 017114 013702 016714
2700 017120 012737 004000 001174
2701
2702
2703 017136 012737 000000 001174
2704
2705
2706 017154
2707 017154 060237 001174
2708
2709
2710 017170 012737 000001 001166

```

 ; VISUAL DISPLAY TEST
 ;*****

VISUAL: JSR PC,\$RESET
 PRINT
 MES6

VTEST2: CLR PASSCT
 VSUALO: MOV STACK,SP ;LOAD THE STACK POINTER
 MOV PASSCT,\$DISPLA ;LOAD PASS COUNT
 MOV \$PICO+2,RETURN ;LOAD RETURN ADDRESS
 CLR \$PSW
 BIC \$BIT1,MODE
 BIT \$BIT4,\$SWR
 BNE VSLOA
 BIS \$BIT1,MODE
 VSLOA: BIS \$BIT6,\$TKS ;ENABLE KEYBOARD

;DISPLAY HORIZONTAL LINE USING INTERRUPT, NON STORE DISPLAY.
 PICO: MOV VCXREG,ROX
 MOV VCYREG,RIY
 JSR PC,PBB

;DISPLAY A VERTICAL LINE
 PIC1: MOV VCYREG,ROX
 MOV VCXREG,RIY
 JSR PC,PBB
 BR PIC3

ROX: .WORD 0
 RIY: .WORD 0

PBB:

;* MOV \$ZERO,\$VCSTAT ;/ PUT DATA FROM \$ZERO TO DEVICE REG VCSTAT
 MOV VCSTAT,R4
 MOV \$50,TICKS
 JSR PC,CHTIME ;CHECK TIMER
 PB: MOV \$7760,R3 ;SET HIGH LIMIT
 PD: MOV INCR,R2 ;INITIALIZE INCREMENTS BETWEEN POINTS
 PEEA: MOV \$4000,\$TMDAT

;* MOV \$TMDAT,\$RIY ;/ PUT DATA FROM \$TMDAT TO DEVICE REG RIY
 MOV \$0,\$TMDAT

;* MOV \$TMDAT,\$ROX ;/ PUT DATA FROM \$TMDAT TO DEVICE REG ROX
 PE: ADD R2,\$TMDAT

;* MOV \$TMDAT,\$ROX ;/ PUT DATA FROM \$TMDAT TO DEVICE REG ROX
 MOV \$1,\$BDDAT

F07

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 72

SEQ 0083

2711						
2712				;	*	MOV
2713						\$BDDAT,VCSTAT ;/ PUT DATA FROM \$BDDAT TO DEVICE REG VCSTAT
2714				;	*	MOV
2715	017216	023703	001174			ROX,\$BDDAT ;/READ DEVICE REG ROX,PUT DATA IN \$BDDAT.
2716	017222	001354				STMDAT,R3 ;DONE ALL POINTS?
2717	017224	004737	020532			PE ;NO
2718	017230	000733				PC,TIMER
2719	017232	000207				PEEA
2720						PC


```
2721
2722 ;PINCUSHION
2723 ;PLOT A SQUARE FROM LOWER LEFT TO LOWER RIGHT TO
2724 ;UPPER RIGHT TO UPPER LEFT TO LOWER LEFT.
2725 ;NON STORE DISPLAY
2726 017234 012737 007770 016712 PIC3: MOV #7770,HIGH
2727 017242 012737 000000 016710 MOV #0,LOW
2728
2729 ;* MOV $ZERO,AVCSTAT ;/ PUT DATA FROM $ZERO TO DEVICE REG VCSTAT
2730 017260 012737 000050 016720 MOV #50,TICKS
2731 017266 004737 020634 JSR PC,CHTIME
2732 017272 013701 001066 MOV VCXREG,R1
2733 017276 013702 001070 MOV VCYREG,R2
2734 017302 013703 001064 MOV VCSTAT,R3
2735 017306 013704 016714 MOV INCR,R4
2736 017312
2737 P3:
2738 ;* MOV LOW,AVCXREG ;/ PUT DATA FROM LOW TO DEVICE REG VCXREG
2739 ;* MOV LOW,VCYREG ;/ PUT DATA FROM LOW TO DEVICE REG VCYREG
2740 ;DRAW BOTTOM LINE
2741 017332 012700 000377 001174 MOV #377,R0
2742 017336 012737 000004 001174 MOV #4,$TMDAT ;ENABLE INTENSIFY ON LOADING X
2743
2744 ;* MOV $TMDAT,AVCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2745 017354 P3A:
2746 ;* MOV AVCXREG,$TMDAT ;/READ DEVICE REG VCXREG,PUT DATA IN $TMDAT.
2747 ADD R4,$TMDAT
2748 ;* MOV $TMDAT,AVCXREG ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCXREG
2749 017364 060437 001174 DEC R0
2750 BNE P3A ;NO
2751 ;DRAW RIGHT LINE
2752 017404 012737 000010 001174 MOV #10,$TMDAT ;ENABLE INTENSIFY ON LOADING Y
2753
2754 ;* MOV $TMDAT,AVCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2755 017422 012700 000377 MOV #377,R0
2756 P3B:
2757 ;* MOV VCYREG,$TMDAT ;/READ DEVICE REG VCYREG,PUT DATA IN $TMDAT.
2758 017436 060437 001174 ADD R4,$TMDAT
2759 ;* MOV $TMDAT,VCYREG ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCYREG
2760 017452 005300 DEC R0
2761 017454 001364 BNE P3B ;NO
2762 ;DRAW TOP LINE
2763 017456 012737 000004 001174 MOV #4,$TMDAT ;ENABLE INTENSIFY ON LOADING X
2764
2765 ;* MOV $TMDAT,AVCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2766 017474 012700 000377 MOV #377,R0
2767 P3C:
2768 ;* MOV AVCXREG,$TMDAT ;/READ DEVICE REG VCXREG,PUT DATA IN $TMDAT.
2769
2770
2771
2772
2773
2774
```

```
2775 017510 160437 001174 SUB R4,$TMDAT
2776
2777 ;* MOV $TMDAT,$VCXREG ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCXREG
2778 017524 005300 DEC R0
2779 017526 001364 BNE P3C ;NO
2780 ;DRAW LEFT LINE
2781 017530 012737 000010 001174 MOV #10,$TMDAT ;ENABLE INTENSIFY LOADING Y
2782
2783 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2784 017546 012700 000377 MOV #377,R0
2785 017552 P3D:
2786
2787 ;* MOV $VCYREG,$TMDAT ;/READ DEVICE REG VCYREG,PUT DATA IN $TMDAT.
2788 017562 160437 001174 SUB R4,$TMDAT
2789
2790 ;* MOV $TMDAT,$VCYREG ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCYREG
2791 017576 005300 DEC R0
2792 017600 001364 BNE P3D ;NO
2793 017602 004737 020532 JSR PC,TIMER
2794 017606 000401 BR 3$
2795 017610 000402 BR PIC4
2796 017612 000137 017312 3$: JMP P3
2797 ;PLOT AN X WITH NON STORE DISPLAY
2798 017616 012737 000000 016710 PIC4: MOV #0,LOW
2799 017624 012737 007770 016712 MOV #7770,HIGH
2800
2801 ;* MOV $ZERO,$VCSTAT ;/ PUT DATA FROM $ZERO TO DEVICE REG VCSTAT
2802 017642 012737 000050 016720 MOV #50,TICKS
2803 017650 004737 020634 JSR PC,CHTIME ;CHECK TIME
2804 017654 013701 001066 PIC4B: MOV VCXREG,R1
2805 017660 013702 001070 MOV VCYREG,R2
2806 017664 013703 001064 MOV VCSTAT,R3
2807 017670 013700 016712 MOV HIGH,R0
2808 017674 013704 016714 MOV INCR,R4
2809 017700 P4:
2810
2811 ;* MOV LOW,$VCYREG ;/ PUT DATA FROM LOW TO DEVICE REG VCYREG
2812
2813 ;* MOV LOW,$VCXREG ;/ PUT DATA FROM LOW TO DEVICE REG VCXREG
2814
2815 ;PLOT LINE BEGINNING IN LOWER LEFT CORNER
2816 017720 012737 000004 001174 MOV #4,$TMDAT ;ENABLE INTENSIFY ON LOADING X
2817
2818 ;* MOV $TMDAT,$VCSTAT ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2819 017736 012700 000377 MOV #377,R0
2820 017742 P4A:
2821
2822 ;* MOV $VCYREG,$GDDAT ;/READ DEVICE REG VCYREG,PUT DATA IN $GDDAT.
2823 017752 060437 001170 ADD R4,$GDDAT
2824
2825 ;* MOV $GDDAT,$VCYREG ;/ PUT DATA FROM $GDDAT TO DEVICE REG VCYREG
2826
2827 ;* MOV $VCXREG,$BDDAT ;/READ DEVICE REG VCXREG,PUT DATA IN $BDDAT.
2828 017776 060437 001166 ADD R4,$BDDAT
```

```

2829
2830      ;*      MOV      $BDDAT,AVCXREG      ;/ PUT DATA FROM $BDDAT TO DEVICE REG VCXREG
2831      020012      005300      DEC      RO
2832      020014      001352      BNE      P4A      ;NO
2833      ;PLOT LINE BEGINNING IN UPPER LEFT CORNER
2834
2835      ;*      MOV      HIGH,AVCYREG      ;/ PUT DATA FROM HIGH TO DEVICE REG VCYREG
2836
2837      ;*      MOV      LOW,AVCXREG      ;/ PUT DATA FROM LOW TO DEVICE REG VCXREG
2838      020036      012700      000377      MOV      #377,RO
2839      020042      P4B:
2840
2841      ;*      MOV      AVCYREG,$TMDAT      ;/READ DEVICE REG VCYREG,PUT DATA IN $TMDAT.
2842      020052      160437      001174      SUB      R4,$TMDAT
2843
2844      ;*      MOV      $TMDAT,AVCYREG      ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCYREG
2845
2846      ;*      MOV      AVCXREG,$TMDAT      ;/READ DEVICE REG VCXREG,PUT DATA IN $TMDAT.
2847      020076      060437      001174      ADD      R4,$TMDAT
2848
2849      ;*      MOV      $TMDAT,AVCXREG      ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCXREG
2850      020112      005300      DEC      RO
2851      020114      001352      BNE      P4B      ;NO
2852      020116      004737      020532      JSR      PC,TIMER
2853      020122      000666      BR      P4
2854
2855      .EVEN
2856
2857
2858
2859
2860      020124      004737      023054      PIC20: JSR      PC,$RESET
2861      020130      005737      000042      TST      #42
2862      020134      001402      BEQ      .+6
2863      020136      000137      002160      JMP      WHATE
2864      020142      004737      015650      JSR      PC,BELL
2865      020146      005237      001042      INC      PASSCT
2866      020152      000137      016736      JMP      VSUALO
2867
2868      020156      000010      MODE:      10
2869
2870      020160      012737      002000      001174      CLRVCA: MOV      #BIT10,$TMDAT      ;ENABLE STORE
2871
2872      ;*      MOV      $TMDAT,AVCSTAT      ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2873
2874      ;*      MOV      AVCSTAT,$TMDAT      ;/READ DEVICE REG VCSTAT,PUT DATA IN $TMDAT.
2875      020206      052737      010000      001174      BIS      #BIT12,$TMDAT      ;ERASE THE SCREEN
2876
2877      ;*      MOV      $TMDAT,AVCSTAT      ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2878      020224      012700      000020      MOV      #20,RO
2879      020230      005001      CLR      R1      ;SET UP DELAY
2880
2881      020232      CLRVCA:
2882

```



```

2883      ;*      MOV      @VCSTAT,$BDDAT      ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
2884      020242  105737  001165      TSTB      $BDDAT
2885      020246  100405      BMI      CLRVCB      ;BRANCH IF SET
2886      020250  005301      DEC      R1      ;DELAY
2887      020252  001367      BNE      CLRVCA
2888      020254  005300      DEC      R0      ;DELAY
2889      020256  001365      BNE      CLRVCA
2890      020260  104400      ERROR      ;ERROR, ERASE FAILED TO SET READY AFTER A DELAY
2891
2892      020262  000207      CLRVCB: RTS      PC
2893
2894      020264      LOADVC:
2895
2896      ;*      MOV      $ZERO,@VCSTAT      ;/ PUT DATA FROM $ZERO TO DEVICE REG VCSTAT
2897      020274  012737  007777  016676      MOV      #7777,TEMP1
2898      020302  013700  001064      MOV      VCSTAT,R0
2899      020306  013701  001066      MOV      VCXREG,R1
2900      020312  013702  001070      MOV      VCYREG,R2
2901      020316  012737  002000  001174      MOV      #BIT10,$TMDAT      ;SET STORE MODE
2902
2903      ;*      MOV      $TMDAT,@VCSTAT      ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2904
2905      ;*      MOV      TEMP1,@VCYREG      ;/ PUT DATA FROM TEMP1 TO DEVICE REG VCYREG
2906      020344  012737  007777  001174      LODVCA: MOV      #7777,$TMDAT
2907
2908      ;*      MOV      $TMDAT,@VCXREG      ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCXREG
2909      020362  000413      LODVCC: BR
2910      020364
2911
2912      ;*      MOV      @VCXREG,$TMDAT      ;/READ DEVICE REG VCXREG,PUT DATA IN $TMDAT.
2913      020374  162737  000010  001174      SUB      #10,$TMDAT
2914
2915      ;*      MOV      $TMDAT,@VCXREG      ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCXREG
2916      020412      LODVCC:
2917
2918      ;*      MOV      @VCSTAT,$TMDAT      ;/READ DEVICE REG VCSTAT,PUT DATA IN $TMDAT.
2919      020422  005237  001174      INC      $TMDAT
2920
2921      ;*      MOV      $TMDAT,@VCSTAT      ;/ PUT DATA FROM $TMDAT TO DEVICE REG VCSTAT
2922      020436  000240      NOP
2923      020440      IS:
2924
2925      ;*      MOV      @VCSTAT,$BDDAT      ;/READ DEVICE REG VCSTAT,PUT DATA IN $BDDAT.
2926      020450  105737  001166      TSTB      $BDDAT
2927      020454  100371      BPL      IS
2928
2929      ;*      MOV      @VCXREG,$TMDAT      ;/READ DEVICE REG VCXREG,PUT DATA IN $TMDAT.
2930      020466  022737  000007  001174      CMP      #7,$TMDAT
2931      020474  001333      BNE      LODVCB
2932      020476  104007      TSTTKS      ;TEST FOR INPUT FLAG
2933
2934      ;*      MOV      @VCYREG,$TMDAT      ;/READ DEVICE REG VCYREG,PUT DATA IN $TMDAT.
2935      020510  162737  000003  001174      SUB      #3,$TMDAT
2936

```

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

K07

MACY11 27(654) 15-DEC-77 08:35 PAGE 77

SEQ 0088

2937
2938 020526 001306
2939 020530 000207

;*

MOV
BNE
RTS

STMDAT,@VCYREG ;/ PUT DATA FROM STMDAT TO DEVICE REG VCYREG
LODVCA
PC

```

2940
2941
2942      ;TIMER ROUTINE
2943      ; ENTER VIA JSR PC,TIMER
2944
2945 020532 017737 160300 016716 TIMER: MOV @SWR,TIMSV
2946 020540 104007          TSTTKS
2947 020542 032737 000400 016716 TIMERA: BIT #BIT8,TIMSV
2948 020550 001006          BNE TIMER2      ;BIT 8 SET ?
2949 020552 005337 016720          DEC TICKS    ;NO, DECREMENT TICKS
2950 020556 001002          BNE TIMER1
2951 020560 062716 000002          ADD #2,(6)    ;ADD 2 TO STACK POINTER
2952 020564 000207          TIMER1: RTS PC        ;RETURN
2953
2954      ; SWR 8=1 SELECT TEST TO LOCK ON
2955      ; SWR 2-0= TEST NUMBER
2956
2957 020566 042737 177770 016716 TIMER2: BIC #177770,TIMSV
2958 020574 006337 016716          ASL TIMSV
2959 020600 062737 020624 016716          ADD #ROUTPT,TIMSV
2960 020606 017737 176104 016716          MOV @TIMSV,TIMSV
2961 020614 013706 016632          MOV STACK,SP
2962 020620 000177 176072          TIMER4: JMP @TIMSV
2963
2964 020624 017014          ROUTPT: PICO          ;DISPLAY A HORIZONTAL LINE
2965 020626 017034          PIC1                ;DISPLAY A VERTICAL LINE
2966 020630 017234          PIC3                ;DISPLAY A SQUARE
2967 020632 017616          PIC4                ;DISPALY A "X"
2968
2969 020634 013737 001014 016702 CHTIME: MOV PDPOLY,BRLEV1
2970 020642 005337 016702 CHTMA: DEC BRLEV1
2971 020646 001403          BEQ CHTMB
2972 020650 006337 016720          ASL TICKS
2973 020654 000772          BR CHTMA
2974 020656 000207          CHTMB: RTS PC

```



```
2975 ;SLOW RELAY SWITCH TEST
2976
2977 020660 013706 016632 RELAY: MOV STACK,SP ;LOAD THE STACK
2978 020664 005037 001124 CLR .DVLs
2979
2980 ;* MOV @GRSTAT,STMDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN STMDAT.
2981 020700 052737 000401 001174 BIS #BIT8:BIT0,STMDAT
2982
2983 ;* MOV STMDAT,@GRSTAT ;/ PUT DATA FROM STMDAT TO DEVICE REG GRSTAT
2984 020716 004737 020756 JSR PC,DLY
2985
2986 ;* MOV @GRSTAT,STMDAT ;/READ DEVICE REG GRSTAT,PUT DATA IN STMDAT.
2987 020732 042737 000401 001174 BIC #BIT8:BIT0,STMDAT
2988
2989 ;* MOV STMDAT,@GRSTAT ;/ PUT DATA FROM STMDAT TO DEVICE REG GRSTAT
2990 020750 004737 020756 JSR PC,DLY
2991 020754 000741 BR RELAY
2992
2993 020756 013737 001014 016644 DLY: MOV PDPDLY,DELAY
2994 020764 005037 021006 CLR DELAY1
2995 020770 005237 021006 DLYA: INC DELAY1
2996 020774 001375 BNE DLYA
2997 020776 005337 016644 DEC DELAY
2998 021002 001372 BNE DLYA
2999 021004 000207 RTS PC
3000 021006 000000 DELAY1: 0
3001
3002 ;SCOPE OUTPUT OF CLOCK OVERFLOW
3003
3004 021010 013706 016632 CKOVFL: MOV STACK,SP
3005 021014 012737 177766 001174 MOV #-10.,STMDAT ;LOAD COUNTER PRESET
3006 021022 005037 001124 CLR .DVLs
3007
3008 ;* MOV STMDAT,@CSB ;/ PUT DATA FROM STMDAT TO DEVICE REG CSB
3009 021036 012737 000403 001174 MOV #403,STMDAT ;LOAD RATE AND MODE
3010
3011 ;* MOV STMDAT,@CSR ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
3012 021054 SOCOA:
3013
3014 ;* MOV @CSR,$BDDAT ;/READ DEVICE REG CSR,PUT DATA IN $BDDAT.
3015 021064 105737 001166 TSTB $BDDAT
3016 021070 100371 BPL SOCOA
3017 021072 000746 BR CKOVFL
3018
3019 ;TEST FOR SCHMITT TRIGGER #1 AND #2
3020
3021 021074 005037 016676 021174 ST1: CLR TEMP1
3022 021100 042737 100000 BIC #BIT15,STSC
3023 021106 000406 BR STSA
3024
3025 021110 012737 001000 016676 ST2: MOV #1000,TEMP1 ;LOAD MODE
3026 021116 052737 100000 021174 BIS #BIT15,STSC
3027 021124 013706 016632 STSA: MOV STACK,SP
3028 021130 005037 016650 CLR KSTOR3
```

```

3029 021134 005037 001124 001174 STSB: CLR .DVLS
3030 021140 013737 016676 MOV TEMP1,STMDAT ;LOAD STATUS
3031
3032 ;* MOV STMDAT,ACSR ;/ PUT DATA FROM STMDAT TO DEVICE REG CSR
3033 021156 004537 015702 JSR 5,LEDS
3034 021162 016650 KSTOR3
3035
3036 021164 STST:
3037
3038 ;* MOV ACSR,SBDDAT ;/READ DEVICE REG CSR,PUT DATA IN SBDDAT.
3039 021174 005737 001166 STSC: TST SBDDAT
3040 021200 100371 BPL STST
3041
3042 ;* MOV $ZERO,ACSR ;/ PUT DATA FROM $ZERO TO DEVICE REG CSR
3043 021212 005237 016650 INC KSTOR3
3044 021216 004737 015654 JSR PC,BELLA
3045 021222 000746 BR STSB
3046
3047 ;*
3048 ;*THIS SUB CODE IS USED TO INITIALIZE THE LPA-11
3049 ;*FIRST WE WILL LOAD MICROCODE INTO KMC-11
3050 ;*NEXT WE WILL INIT BOTH UPROCESSORS
3051 ;*THEN WE WILL LOAD DEVICE TABLE IN SLAVE UP.
3052 ;*THE ORDER OF LOAD IS DETERMINED BY THE USER.
3053 ;*
3054 ;* CALL= JSR RS,SLPAI
3055 ;* .WORD 0 ;ADDR. OF DEVICE ADDRESS.
3056 ;* ROUTINES REQUIRED: .LOADLP
3057 ;* PROGRAMS REQUIRED: DRLPX2
3058 ;*
3059 ;*
3060 ;* ;RETURNS WITH SAERR=1 IF SLAVE
3061 ;* ;MICRO SAYS AN ADDR. DOES NOT EXSIST. IN THE LIST.
3062 ;*
3063 021224 SLPAI:
3064 021224 013746 000004 MOV 4,-(SP)
3065
3066 021230 000413 BR 31$
3067
3068 ;FIELD DOES NOT HAVE A BUS SWITCH TO
3069 ;WORRY ABOUT,SO WE WILL UNCONDITIONALLY
3070 ;BRANCH AROUND THE NEXT CODE THAT
3071 ;WORKS BASED ON A BUS SWITCH.
3072 ;CODE LEFT IN HERE FOR IN HOUSE
3073 ;PERSONAL WHO MAY PATCH THIS BRANCH
3074 ;INSTRUCTION TO A <NOP> OCTAL <240>
3075 ;IN ORDER TO RUN PROGRAM WITH A SWITCH.
3076
3077 ;NOTE THIS "SWITCH" IS A PIECE OF INHOUSE
3078 ;TEST EQUIPMENT ONLY IT CONNECTS
3079 ;THE UNIBUS TO THE I/O BUS FOR
3080 ;CERTAIN TESTING.
3081
3082 021232 012737 021256 000004 MOV #30$,4
3083 021240 005237 170000 INC 170000
3084 021244 104000 PRINT
3085 021246 021252 64$

```


3083	021250	000401			BR	65\$	
3084	021252			64\$:	;ASCIZ	<7>##	
3085	021254			65\$:			
3086	021254	000401			BR	31\$	
3087	021256	022626		30\$:	CMP	(SP)+,(SP)+	
3088	021260	012637	000004	31\$:	MOV	(SP)+,4	;ALL THIS JUNK MUST BE REMOVED!!
3089	021264	005037	022102		CLR	\$AERR	
3090	021270	004537	022104		JSR	R5,\$LOAD	;LOAD MICRO-CODE.
3091	021274	000000G			.WORD	DRLPX2	;FILE "DRLPX2.OBJ"
3092							
3093	021276	052777	040000	157572	BIS	#BIT14,\$KMADO	;ISSUE KMC+DMC INIT.
3094							
3095	021304			1\$:			; "HANGS" HERE THEN KMC-11 ERROR.
3096							
3097	021304	010146			MOV	R1,-(SP)	
3098	021306	005001			CLR	R1	
3099	021310	005201		2\$:	INC	R1	;STALL FOR DMC-UP
3100	021312	001376			BNE	2\$	
3101	021314	012777	104000	157554	MOV	#BIT15!BIT11,\$KMADO	;SET RUN, AND ENABLE ARBITRATION.
3102	021322	105201		25\$:	INCB	R1	
3103	021324	001376			BNE	25\$	
3104							
3105	021326	032777	000040	157542	BIT	#BITS,\$KMADO	;SLAVE READY? (READING IPBM SR)
3106	021334	001401			BEQ	3\$	
3107							;FATAL LPA-11 ERROR SLAVE NOT READY.
3108	021336	104400			ERROR		
3109							
3110	021340	012777	000004	157534	3\$:	MOV	#4,\$KMAD2
3111	021346			4\$:			;READ FAST PATH
3112	021346	004537	022720		JSR	R5,\$TOUT	; -TOUT-CHECK FOR TIMEOUT
3113							
3114	021352	104400			ERROR		; /TIME-OUT ERROR
3115							; /WE FAILED TO COMPLETE
3116							; /CURRENT OPERATION.
3117							; /CONTINUES IN THIS LOOP
3118							; /WOULD MAKE US "HANG" HERE
3119							
3120	021354	000774			BR	4\$	
3121							
3122							; /RETURNS HERE-FROM-TIMED OUT.
3123	021356	122777	000377	157516	CMPB	#377,\$KMAD2	; WAIT TILL KMC DONE COMMAND.
3124	021364	001370			BNE	4\$	
3125	021366	122777	000377	157512	CMPB	#377,\$KMAD4	; IF FAST PATH=377 THEN ERROR.
3126	021374	001001			BNE	35\$	
3127	021376	104400			ERROR		; IPBM ERROR (SLAVE SIDE)
3128							; YOU MUST RUN IPBM DIAGNOSTIC.
3129							
3130	021400	122777	000004	157500	35\$:	CMPB	#4,\$KMAD4
3131	021406	001543			BEQ	5\$	; IS THIS THE CORRECT VERSION OF MICRO-CODE?
3132	021410	005227	177777		INC	#-1	; YES-CONTINUE.
3133	021414	001140			BNE	5\$	
3134	021416	005227	177777		INC	#-1	
3135	021422	001135			BNE	5\$	
3136	021424	104000			PRINT		

C08

Address	Hex	Hex	Hex	Hex	Label	Assembly	Comments
3137	021426	021432					
3138	021430	000440					
3139	021432				66\$:	BR	67\$
3140	021532				67\$:	;ASCIZ	<200>"W A R N I N G THIS PROGRAM WAS DESIGNED TO RUN WITH VERSION 4"
3141	021532	104000				PRINT	
3142	021534	021540				68\$	
3143	021536	000430				BR	69\$
3144	021540				68\$:	;ASCIZ	<200>"MICRO-CODE. ANOTHER VERSION CODE WAS DETECTED."
3145	021620				69\$:		
3146	021620	104000				PRINT	
3147	021622	021626				70\$	
3148	021624	000434				BR	71\$
3149	021626				70\$:	;ASCIZ	<200>"THIS MAY OR MAYNOT CAUSE FALSE ERROR TO BE REPORTED."<200><200>
3150	021716				71\$:		
3151							
3152	021716	112737	177777	022050	5\$:	MOVB	#0-1,11\$;DAC CODE FOR SLAVE.
3153	021724	012501				MOV	(5)+,R1 ;GET NEXT DEVICE ADDR.
3154	021726	021127	000000		6\$:	CMP	(R1),#0 ;TERM REACHED?
3155	021732	001444				BEQ	10\$
3156	021734	105237	022050			INCB	11\$
3157	021740	113777	022050	157140		MOVB	11\$,2KMAD4 ;FIFO DATA
3158	021746	004737	022052			JSR	PC,20\$;ISSUE SEND
3159	021752	112177	157130			MOVB	(R1)+,2KMAD4 ;SEND LOW BYTE OF DEVICE ADDR TO SLAVE.
3160	021756	004737	022052			JSR	PC,20\$;ISSUE SEND
3161	021762	112177	157120			MOVB	(R1)+,2KMAD4 ;SEND HIGH BYTE OF DEVICE ADDR. TO SLAVE.
3162	021766	004737	022052			JSR	PC,20\$
3163							
3164	021772	032777	000002	157076	7\$:	BIT	#BIT1,2KMAD0 ;WAIT FOR FIFO DATA
3165	022000	001374				BNE	7\$;=1 NO DATA. =0 DATA.
3166	022002	112777	000002	157072		MOVB	#2,2KMAD2 ;READ FIFO.
3167							
3168	022010				8\$:		
3169	022010	004537	022720			JSR	RS, \$TOUT ;-TOUT-CHECK FOR TIMEOUT
3170							
3171	022014	104400				ERROR	;/TIME-OUT ERROR
3172							;/WE FAILED TO COMPLETE
3173							;/CURRENT OPERATION.
3174							;/CONTINUES IN THIS LOOP
3175							;/WOULD MAKE US "HANG" HERE
3176							
3177	022016	000774				BR	8\$
3178							
3179							;/RETURNS HERE-FROM-TIMED OUT.
3180	022020	122777	000377	157054		CMPB	#377,2KMAD2 ;WAIT FOR READ.
3181	022026	001370				BNE	8\$
3182	022030	105777	157052			TSTB	2KMAD4 ;WAS A ZERO RETURNED?
3183	022034	001734				BEQ	6\$;YES GET NEXT ADDR.
3184							SLAVE WILL RETURN CODE 0 IF
3185	022036	005237	022102			INC	\$AERR ;DEV PRESENT. ELSE
3186							EXIT \$AERR=1 IF SLAVE GIVES ERROR.
3187	022042	005041				CLR	-(1) ;GET RID OF REFERENCE TO BAD ADDR.
3188	022044	012601			10\$:	MOV	(SP)+,R1
3189	022046	000205				RTS	RS ;RETURN ALL ADDR. CHECKED.
3190							

```
3191 022050 000000 11$: .WORD 0 ;HOLDS DAC CODE PLUS OFFSET
3192 ;TO SLAVES ADDR. TABLE.
3193
3194 022052 112777 000003 157022 20$: MOVB #3,AKMAD2 ;ISSUE FIFO WRITE
3195 022060 004537 022720 21$: JSR RS, $TOuT ;-TOUT-CHECK FOR TIMEOUT
3196
3197 022064 104400 ERROR ;/TIME-OUT ERROR
3198 ;/WE FAILED TO COMPLETE
3199 ;/CURRENT OPERATION.
3200 ;/CONTINUES IN THIS LOOP
3201 ;/WOULD MAKE US "HANG" HERE
3202
3203
3204 022066 000774 BR 21$
3205
3206 ;/RETURNS HERE-FROM-TIMED OUT.
3207 022070 122777 000377 157004 CMPB #377,AKMAD2 ;KMC CODE WILL RETURN A "377"
3208 022076 001370 BNE 21$ ;WHEN DONE COMMAND.
3209 022100 000207 RTS PC
3210
3211 022102 000000 $AERR: .WORD 0 ;=0 IF ADDR. LIST OK,=1 IF BAD.
3212
3213 ;*
3214 ;*THIS SUB CODE USED TO LOAD MICRO-CODE INTO LPA-11.
3215 ;* CALL = JSR RS,$LOAD
3216 ;* .WORD XX ;ADDR. OF MICRO CODE.
3217 ;* ;RETURNS HERE
3218 ;* NOTE: MICRO CODE FILE MUST END IN -1 DATA.
3219 ;*
3220
3221 022104 010446 $LOAD: MOV R4,-(SP) ;SAVE R4.
3222 022106 010046 MOV RO,-(SP) ;SAVE RO.
3223 022110 012500 1$: MOV (5)+,RO ;GET PROG. ADDR.
3224 022112 005077 156760 CLR AKMAD0 ;CLEAR CSR
3225 022116 005077 156764 CLR AKMAD4 ;CLEAR CRAM ADDR.
3226 022122 052777 002000 156746 2$: BIS #2000,AKMAD0 ;SELECT CRAM.
3227 022130 012077 156756 MOV (0)+,AKMAD6 ;WRITE DATA.
3228 022134 052777 020000 156734 BIS #20000,AKMAD0 ;SET CRAM WRITE
3229 022142 005077 156730 CLR AKMAD0 ;DISABLE CRAM.
3230 022146 005277 156734 INC AKMAD4 ;UPDATE CRAM ADDR.
3231 022152 021027 177777 CMP (0), #-1 ;ALL DONE?
3232 022156 001361 BNE 2$ ;NO LOOP.
3233 022160 005077 156722 CLR AKMAD4 ;CLEAR CRAM ADDR.
3234 022164 016500 177776 MOV -2(5),RO ;GET MICRO CODE ADDR.
3235
3236 022170 052777 002000 156700 3$: BIS #2000,AKMAD0 ;SELECT CRAM
3237 022176 022077 156710 CMP (RO)+,AKMAD6 ;DATA OK?
3238 022202 001013 BNE 5$ ;NO - REPORT AN ERROR.
3239 022204 021027 177777 CMP (0), #-1 ;ALL DONE?
3240 022210 001405 BEQ 4$ ;YES - EXIT
3241 022212 005077 156660 CLR AKMAD0 ;NO - DESELECT CRAM.
3242 022216 005277 156664 INC AKMAD4 ;UPDATE CRAM ADDR.
3243 022222 000762 BR 3$
3244
```



```

3245 022224 012600      4$:  MOV      (SP)+,R0      ;RESTORE R0
3246 022226 012604      MOV      (SP)+,R4      ;RESTORE R4
3247 022230 000205      RTS        R5          ;EXIT
3248
3249 022232      5$:      TST        -(5)        ;COME HERE ON LOAD ERROR
3250 022232 005745      INCB      R4          ;UPDATE ERROR COUNTER.
3251 022234 105204      BPL        1$          ;IF NOT TOO MANY, TRY AGAIN.
3252 022236 100324      HALT      1$          ;MICRO CODE LOAD ERROR.
3253 022240 000000      BR        1$          ;KMC-11 FAULT. YOU COULD TRY
3254 022242 000722      ;TO PRESS CONTINUE TO GIVE IT
3255 ;ANOTHER CHANCE, BUT I DOUBT
3256 ;THAT THAT WOULD WORK. SINCE I'VE
3257 ;ALREADY GIVEN IT '77 (OCTAL) CHANCES.
3258 ;TRY RUNNING THE KMC-11 DIAGNOSTIC.
3259
3260
3261
3262
3263 ;*THIS ROUTINE ISSUES A WRITE COMMAND TO THE LPA-11
3264 ;*
3265 ;*      CALL = JSR      R5,$TLKW
3266 ;*      .WORD      0      ;OFFSET OF DEVICE ADDR.
3267 ;*      .WORD      0      ;DATA TO BE WRITTEN
3268 ;*
3269 022244 010046      $TLKW:  MOV      R0,-(SP)      ;SAVE R0
3270 022246 012500      MOV      (5)+,R0      ;GET DEVICE OFFSET
3271 022250 052700 000340  BIS      #340,R0      ;ADD WRITE CODE.
3272 022254 004737 022526  JSR      PC,$LPW      ;WAIT FOR FAST PATH READY
3273 022260 010037 022352  MOV      R0,W1
3274 022264 010077 156616  MOV      R0,$KMAD4
3275 022270 112777 000005 156604  MOVB     #5,$KMAD2      ;ISSUE FAST PATH WRITE
3276 022276 004737 022526  JSR      PC,$LPW      ;WAIT FOR RDY
3277 022302 011537 022354  MOV      (5),W2
3278 022306 112577 156574  MOVB     (5)+,$KMAD4      ;WRITE LOW BYTE DATA.
3279
3280 022312 112777 000005 156562  MOVB     #5,$KMAD2      ;FP WRITE
3281 022320 004737 022526  JSR      PC,$LPW
3282 022324 111537 022356  MOVB     (5),W3
3283 022330 112577 156552  MOVB     (5)+,$KMAD4      ;WRITE HIGH BYTE
3284 022334 112777 000005 156540  MOVB     #5,$KMAD2
3285 022342 004737 022526  JSR      PC,$LPW
3286 022346 012600      MOV      (SP)+,R0
3287 022350 000205      RTS        R5          ;EXIT DONE.
3288 022352 000000      W1:      0
3289 022354 000000      W2:      0
3290 022356 000000      W3:      0
3291
3292
3293 ;*THIS ROUTINE ISSUES A READ COMMAND TO THE LPA-11
3294 ;*
3295 ;*      CALL = JSR      R5,$TLKr
3296 ;*      .WORD      0      ;OFFSET OF DEVICE
3297 ;*      .WORD      0      ;RETURNS HERE
3298 ;*      .WORD      0      ;DATA IN WORD $DATA

```



```
3299 ;*
3300
3301 022360 010046 $TLKR: MOV R0, -(SP) ;SAVE R0
3302 022362 012500 MOV (5)+, R0 ;GET OFFSET
3303 022364 052700 000300 BIS #300, R0 ;ADD READ CODE
3304 022370 004737 022526 JSR PC, SLPW ;WAIT TILL READY
3305 022374 110077 156506 MOVB R0, @KMAD4
3306 022400 112777 000005 156474 MOVB #5, @KMAD2 ;ISSUE WRITE FP
3307 022406 004737 022526 JSR PC, SLPW
3308 022412 010037 022522 MOV R0, RD1
3309 022416
3310 022416 004537 022720 1$: JSR R5, $TOUT ; -TOUT-CHECK FOR TIMEOUT
3311
3312 022422 104400 ERROR ;/TIME-OUT ERROR
3313 ;/WE FAILED TO COMPLETE
3314 ;/CURRENT OPERATION.
3315 ;/CONTINUES IN THIS LOOP
3316 ;/WOULD MAKE US "HANG" HERE
3317
3318 022424 000774 BR 1$
3319
3320 ;/RETURNS HERE-FROM-TIMED OUT.
3321 022426 032777 000040 156442 BIT #BITS, @KMADO ;FAST PATH GOT DATA?
3322 022434 001370 1$ BNE 1$
3323 022436 112777 000004 156436 MOVB #4, @KMAD2 ;ISSUE FAST PATH READ
3324 022444 004737 022526 JSR PC, SLPW
3325 022450 117737 156432 022524 MOVB @KMAD4, $DATR ;GET LOW BYTE
3326 022456 2$: JSR R5, $TOUT ; -TOUT-CHECK FOR TIMEOUT
3327 022456 004537 022720 ERROR ;/TIME-OUT ERROR
3328 022462 104400 ;/WE FAILED TO COMPLETE
3329 ;/CURRENT OPERATION.
3330 ;/CONTINUES IN THIS LOOP
3331 ;/WOULD MAKE US "HANG" HERE
3332
3333 BR 2$
3334
3335 022464 000774 BR 2$
3336
3337 ;/RETURNS HERE-FROM-TIMED OUT.
3338 022466 032777 000040 156402 BIT #BITS, @KMADO ;FAST PATH READY?
3339 022474 001370 2$ BNE 2$
3340 022476 112777 000004 156376 MOVB #4, @KMAD2 ;ISSUE FAST PATH READ
3341 022504 004737 022526 JSR PC, SLPW
3342 022510 117737 156372 022525 MOVB @KMAD4, $DATR+1 ;SAVE HIGH BYTE
3343 022516 012600 MOV (SP)+, R0
3344 022520 000205 RTS R5
3345 022522 000000 RD1: 0
3346 022524 000000 $DATR: .WORD 0
3347
3348 ;THIS ROUTINE WAITS FOR KMC-CODE TO BECOME READY AS WELL
3349 ;AS FAST PATH TO BE READ.
3350
3351 CALL = JSR PC, SLPW
3352
```

;IT WILL TIME OUT IF TOO MUCH TIME IS TAKEN BY
;THE MICRO-PROCESSORS AND REPORT AN ERROR, THEN HALT.

```
3353
3354
3355
3356
3357 022526 010146 $LPW: MOV R1,-(SP) ;SAVE R1
3358 022530 005001 CLR R1
3359 022532 122777 000377 156342 1$: CMPB #377,AKMAD2 ;FINISHED INSTRUCTION?
3360 022540 001403 BEQ 2$
3361 022542 005201 INC R1 ;TIME OUT?
3362 022544 001372 BNE 1$
3363 022546 000411 BR 10$
3364
3365 022550 032777 000020 156320 2$: BIT #BIT4,AKMADO ;FAST PATH READ?
3366 022556 001403 BEQ 3$
3367 022560 005201 INC R1 ;NO - TIME OUT?
3368 022562 001372 BNE 2$
3369 022564 000402 BR 10$ ;YES - REPORT AN ERROR
3370
3371 022566 012601 3$: MOV (SP)+,R1 ;RESTORE R1
3372 022570 000207 RTS PC ;EXIT
3373
3374 022572 10$: PRINT
3375 022572 104000 64$
3376 022574 022600 BR 65$
3377 022576 000407 64$: ;ASCIZ <200>#LPA-11 FAULT#
3378 022600 65$:
3379 022616
3380
3381 022616 000000 11$: HALT ;LPA-11 FAULT RUN LPA-11
3382 022620 000776 BR 11$ ;DIAGNOSTICS.
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397 022622 010046 $OUTLP: MOV R0,-(SP) ;SAVE R0
3398 022624 010146 MOV R1,-(SP) ;SAVE R1
3399
3400 022626 012700 001124 MOV #.DVLS,R0 ;PROGRAM DEFINED LIST.
3401 022632 005001 CLR R1
3402 022634 005710 1$: TST (0) ;TERMINATOR REACHED?
3403 022636 001421 BEQ 10$ ;YES NEXT STEP.
3404 022640 027520 000000 CMP @ (5), (0)+ ;MATCH WITH ADDR IN LIST?
3405 022644 001402 BEQ 2$
3406 022646 005201 INC R1
```

```
3407 022650 000771 BR 1$
3408
3409 022652 010137 022670 2$: MOV R1,3$ ;SAVE OFFSET, DEVICE KNOWN.
3410 022656 005725 TST (5)+
3411 022660 013537 022672 MOV R5,(5)+,4$ ;GET DATA TO BE WRITTEN
3412 022664 004537 022244 JSR R5,$TLKW ;DO WRITE
3413 022670 000000 3$: .WORD 0 ;DEVICE OFFSET
3414 022672 000000 4$: .WORD 0 ;DATA TO BE WRITTEN.
3415 022674 012601 MOV (SP)+,R1
3416 022676 012600 MOV (SP)+,R0
3417 022700 000205 RTS R5
3418 022702 017520 000000 10$: MOV R5,(5),(0)+ ;SAVE ADDR.
3419 022706 005010 CLR (0)
3420 022710 004537 021224 JSR R5,$LPAI
3421 022714 001124 .WORD $VLS
3422 022716 000755 BR 2$
3423
3424 ;*
3425 ;*STOUT ROUTINE USED TO WATCH IF
3426 ;*WE'RE IN A LOOP TOO-LONG
3427 ;*CALL= JSR R5,$STOUT
3428 ;*ERROR X ;RETURNS HERE ON TIMEOUT
3429 ;*BR
3430 ;*RETURNS HERE NO ERROR
3431 ;*
3432
3433 022720 020537 022754 $STOUT: CMP R5,$$AD ;SAME ADDR?
3434 022724 001405 BEQ 1$
3435 022726 010537 022754 MOV R5,$$AD ;NO-SAVE THIS ADDR.
3436 022732 005037 022756 CLR $CNT ;CLR CNT AT ADDR.
3437 022736 000403 BR 2$
3438 022740 005237 022756 1$: INC $CNT ;OVERFLOW?
3439 022744 100402 BMI 3$ ;YES-ERROR RETURN
3440 022746 062705 000004 2$: ADD #4,R5 ;NO-NON ERROR RETURN
3441 022752 000205 3$: RTS R5 ;RETURN.
3442
3443 022754 000000 $$AD: .WORD 0 ;CONTAINS LOOP ADDR.
3444 022756 000000 $CNT: .WORD 0 ;# OF TIMES AT ADDR.
3445
3446 ;*
3447 ;*THIS ROUTINE PROVIDES THE LINKAGE FROM USER CODE
3448 ;*TO A DEVICE ADDR. ON THE I/O BUSS FOR READ ONLY.
3449 ;*
3450 ;*FIRST WE WILL DETERMINE IF THE ADDRESS HAS BEEN
3451 ;*USED BEFORE. IF NOT, WE HAVE TO INITIALIZE THE LPA
3452 ;*WITH THE NEW ADDR.
3453 ;*WHEN THE ADDR IS KNOWN WE CAN DO OUTPUT THROUGH
3454 ;*$TLKR
3455 ;*CALL THROUGH MOVEI DATA,ADDR.
3456 ;*WHICH EQUALS:
3457 ;* JSR R5,$INLP
3458 ;*.WORD XX ADDR OF DEVICE
3459 ;*.WORD YY ADDR TO STORE READ DATA.
3460 022760 010046 $INLP: MOV R0,-(SP) ;SAVE R0
```



```

3461 022762 010146      MOV      R1,-(SP)      ;SAVE R1
3462
3463 022764 012700 001124      MOV      #.DVLS,PO      ;PROG DEFINED ADDR. LIST.
3464 022770 005001      CLR      R1
3465 022772 005710      1$:      TST      (0)      ;EOL REACHED?
3466 022774 001420      BEQ      10$      ;YES - DEFINE NEW ADDR.
3467
3468 022776 027520 000000      CMP      2(5),(0)+      ;ADDR. MATCH?
3469 023002 001402      BEQ      2$
3470 023004 005201      INC      R1
3471 023006 000771      BR       1$
3472
3473 023010 010137 023022      2$:      MOV      R1,3$      ;SAVE LIST OFFSET
3474 023014 005725      TST      (5)+
3475 023016 004537 022360      JSR      R5,$TLKf      ;GO READ DEVICE
3476
3477 023022 000000      $OFS=.      .WORD      0      ;OFFSET OF DEVICE
3478 3$:
3479 023024 013735 022524      MOV      $DATA,2(5)+      ;STORE DATA.
3480 023030 012601      MOV      (SP)+,R1      ;RESTORE R1
3481 023032 012600      MOV      (SP)+,R0      ;RESTORE R2
3482 023034 000205      RTS      R5      ;EXIT
3483
3484 023036 017520 000000      10$:     MOV      2(5),(0)+
3485 023042 005010      CLR      (0)
3486 023044 004537 021224      JSR      R5,$LPAI
3487 023050 001124      .WORD      .DVLS
3488 023052 000756      BR       2$
3489
3490
3491      ;*
3492      ;* THIS ROUTINE REPLACES WHAT THE USER WOULD ORDINARILY
3493      ;* USE FOR A RESET. FIRST WE DO A RESET INSTRUCTION.
3494      ;* THEN WE CLR ".DVLST" WHICH FORCES US TO RESET BOTH THE
3495      ;* KMC AND DMC AS SOON AS A DEVICE IS REFERENCED.
3496
3497      ;*
3498      ;*      CALL=JSR      PC,$RESET      ;REPLACES "RESET INSTRUCTION
3499      ;*      ;RETURNS HERE.
3500
3501 023054 000005      $RESET:  RESET      ;RESET THE WORLD.
3502
3503      ;*
3504      MOV      2$,$1$      ;/READ DEVICE REG 2$,PUT DATA IN 1$.
3505      TST      $AERR      ;IF NO ERROR,LOOP
3506      BNE      10$      ;THERE WAS AN ERROR.
3507      ADD      #2,2$      ;UPDATE DEVICE ADDR.
3508      ;YOU SEE, WE HAVE TO PROTECT OUR SELF!
3509      ;IF 2$ CONTAINED A VALID ADDR,WE
3510      ;MUST KEEP TRYING UNTIL WE GENERATE
3511      ;AN INVALID ADDR.
3512
3513 023102 000764      10$:     BR       $RESET
3514 023104
3515 023104 000207      RTS      PC
3516 023106 000000      1$:      .WORD      0      ;JUNK LOC.
3517 023110 160000      2$:      .WORD      160000      ;DUMB ADDR. FORCES INIT OF DMC/KMC.

```

3515
3516
3517
3518
3519
3520
3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568

023112	005737	023174
023112	100016	
023116	012737	000002 023164
023120	052777	000115 000040
023126	005037	001024
023134	005737	023164
023140	001375	
023144	005077	000022
023152	105237	023164
023154	001375	
023160	000207	
023162		
023164	000000	
023166	005337	023164
023172	000002	
023174	000000	

SDELAY:

```
TST RTCCSR ;CLOCK PRESENT?
BPL 10$
MOV #2, TIME
BIS #115, RTCCSR ;START CLOCK
CLR PS
1$: TST TIME
BNE 1$
CLR RTCCSR ;STOP CLOCK
```

10\$:

```
RTS PC
INCB TIME
BNE 10$
RTS PC
```

TIME:

.WORD 0

CLKINT:

DEC TIME

RTCCSR:

.WORD 0 ;CLOCK CSR IF USED.

THIS ROUTINE LOOKS THROUGH CURENT DVLS FOR A/D ADDR.
IF UNFOUND, GENERATES IT. THIS ROUTINE'S WHOLE PURPOSE IS
TO SET UP THE USER PROGRAM TO LINK TO FILE "DRLPX2" FOR
SAMPLE TAKEING PURPOSES.

TO TAKE SAMPLES, THE USER PROGRAM MUST SET UP
A/D CSR IN BSEL 4 AND 5.

(2) HE MUST CALL THIS ROUTINE:

```
JSR RS, SPUTS ;CALL SET UP ROUTINE.
;WORD ADCSR ;ADDR. OF A/D CSR.
;RETURNS HERE ;KMC BSEL 3, 6, 7 PERMINENTLY SET UP
; (UNTILL ONE DOES A RESET)
```

(3) THE USER MUST PUT CODE 006 INTO KMC REG 2 TO
START CONVERSION. CAUTION*DO WITH MOV B INSTR. !
(4) MONITOR KMC REG 2 FOR CODE 377 (DRLPX2 IS DONE)
(5) READ KMC REG 4, 5 FOR A/D RESULT.
(6) TO TAKE MORE SAMPLES, SIMPLY PUT A/D CSR INTO
BSEL 4, 5 AND CODE 6 INTO BSEL 2.

K08

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

MACY11 27(654) 15-DEC-77 08:35 PAGE 90

SEQ 0101

3569	023176	012537	023206		\$PUTS:	MOV	(5)+, 1\$		
3570	023202	004537	022760			JSR	R5, \$INLP		;GET ADDR OF ADDR. OF A/D
3571	023206	000000			1\$:	.WORD	0		
3572	023210	023304				.WORD	10\$		
3573	023212	113777	023022	155672		MOVB	\$OFS, 2KMA06		
3574	023220	113777	023022	155666		MOVB	\$OFS, 2KMA07		
3575	023226	013737	023206	023246		MOV	1\$, 2\$		
3576	023234	062737	000002	023246		ADD	#2, 2\$		
3577	023242	004537	022760			JSR	R5, \$INLP		
3578	023246	000000			2\$:	.WORD	0		
3579	023250	023304				.WORD	10\$		
3580	023252	113777	023022	155624		MOVB	\$OFS, 2KMA03		
3581	023260	152777	000340	155624		BISB	#340, 2KMA06		
3582	023266	152777	000300	155620		BISB	#300, 2KMA07		
3583	023274	152777	000300	155602		BISB	#300, 2KMA03		
3584	023302	000205				RTS	R5		
3585	023304	000000			10\$:	.WORD	0		
3586									
3587		000001				.END			


```

ADDRBL      001004
ADCS        001046
ADDBR       001050
ADINT       001176
ADINT1      001200
AVCT12      013442
AVECTR      016634
AVECT1=     000300
BEGIN       002252
BELL        015650
BELLA       015654
BIT0        000001

BIT1        000002
BIT10       002000

BIT11       004000
BIT12       010000
BIT13       020000
BIT14       040000
BIT15       100000
BIT2        000004
BIT3        000010
BIT4        000020
BIT5        000040
BIT6        000100

BIT7        000200

BIT8        000400
BIT9        001000
BRLEV1      016702
BRLEV2      016704
BRLEV3      016706
CHRCNT      016640
CHTIME      020634
CHTMA       020642
CHTMB       020656
CK          015460
CKBR        001006
CKDLY       007640
CKOVFL      021010
CKTEST      002236
CKV         001204
CKVS        001206
CLKINT      023166
CLRCV       020160
CLRVCA      020232
CLRVCB      020262
CNTDEV      007634
CNTLED      016020
CNTRLA      016545
CNTRLC      016541
CNTRLG      016550

```

G

ENDPAS 015662
ERROR = 104400

2492	2497*	460	470	481	495	507	519	531	544	556	568	580
148*	450	612	624	631	644	663	683	705	729	753	779	807
593	605	867	895	914	943	962	990	1009	1038	1068	1079	1092
813	848	1131	1144	1159	1164	1176	1181	1192	1204	1216	1228	1241
1105	1118	1263	1304	1306	1420	1431	1442	1452	1465	1478	1489	1503
1247	1258	1542	1555	1568	1581	1596	1611	1621	1633	1643	1658	1675
1518	1530	1757	1769	1782	1794	1806	1818	1830	1843	1853	1865	1875
1732	1745	1913	1926	1934	1951	1959	1973	1985	2000	2012	2033	2045
1890	1896	2109	2122	2890	3108	3114	3127	3171	3198	3312	3329	
2081	2096											
167*	2333											
166*	2330											
197*	287*	288*										
189*	1409	1609	1631	1641	1670	2394						
190*	287	1411	1416	1418	1426	1429	1438	1440	1448	1450	1458	1461
1463	1471	1474	1476	1485	1487	1499	1501	1607	1617	1619	1629	1639
1652	1667	2392										
247*	335*											
250*	337*											
248*	335	336*										
251*	337	338*										
188*	377	1407	1514	1516	1526	1528	1538	1540	1551	1553	1564	1566
1577	1579	1590	1593	1650	1656	1665	1673	2390	2981	2984	2987	2990
1313	1316*											
2653*	2726*	2799*	2807	2836								
177*	2460											
350*	351	354	357	360	2154	2157*	2158*	2159*	2160*	2161*	2162*	2163*
2229	2245*											
2654*	2699	2735	2808									
138	260*											
316*	340	2630										
283*	286											
292*	295											
311	318	328*	365									
363*												
2164*	2165	2179	2201									
2172*	2207											
140	356	1393*										
1397*	1699											
395	1396*											
208*	268	3093*	3101*	3105	3164	3224*	3226*	3228*	3229*	3236*	3241*	3321
3338	3365											
211*	269											
213*	3110*	3123	3166*	3180	3194*	3207	3275*	3280*	3284*	3306*	3323*	3340*
3359												
215*	3580*	3583*										
217*	3125	3130	3157*	3159*	3161*	3182	3225*	3230*	3233*	3242*	3274*	3278*
3283*	3305*	3325	3342									
219*												
221*	3227*	3237	3573*	3581*								
223*	273	3574*	3582*									
2635*												
2380*	2381*	2383	2412	2636*	3028*	3034	3043*					
2396	2430	2637*										

FILCHR 001020
FILLS 001016
GRBHIO 001074
GRDAI 001060
GRDIO 001062GRIVA 001210
GRIVB 001214
GRIVSA 001212
GRIVSB 001216
GRSTAT 001056HERE 007322
HIGH 016712
ICOUNT 001044
INBUF 014446INCR 016714
INIT 001224
INITA 001504
INIT1A 001314
INIT1B 001356
INIT2 001542
INIT3 001766
INPUTA 014134
INPUTB 014166
IOTEST 007642
IOTSTA 007656
ITEST1 007652
KMADO 001076KMAD1 001100
KMAD2 001102KMAD3 001104
KMAD4 001106KMAD5 001110
KMAD6 001112
KMAD7 001114
KSTOR1 016646
KSTOR3 016650
KSTOR4 016652

KSTOR5	016654	2398	2433	2638#
KSTOR6	016656	2386	2415	2639#
KSTOR7	016660	2388	2418	2640#
KSTOR8	016662	2390	2421	2641#
KSTOR9	016664	2392	2424	2642#
KSTR10	016666	2394	2427	2643#
KSTR11	016670	2400*	2439	2644#
KSTR12	016672	2400	2436	2645#
KWT0	002324	430	442#	2470
KWT10	002744	536#		
KWT11	003006	548#		
KWT12	003050	560#		
KWT14	003112	572#		
KWT15	003154	585#		
KWT16	003216	597#		
KWT17	003306	616#		
KWT18	003376	635#		
KWT18A	003414	639#	646	
KWT19	003442	651#		
KWT2	002366	452#		
KWT20	003522	668#		
KWT22	003622	688#		
KWT23	003730	709#		
KWT24	004056	733#		
KWT25	004204	757#		
KWT26	004336	789#		
KWT26A	004376	797#	821	
KWT3	002430	462#		
KWT30	004524	808	814	831#
KWT30A	004640	850#	857	
KWT31	004744	878#		
KWT31A	005060	897#	904	
KWT32	005164	926#		
KWT32A	005300	945#	952	
KWT33	005404	973#		
KWT33A	005520	992#	999	
KWT34	005630	1020#		
KWT34A	005670	1028#	1046	
KWT35	005776	1039	1050#	
KWT36	006164	1083#		
KWT37	006230	1096#		
KWT38	006274	1109#		
KWT39	006340	1122#		
KWT4	002472	474#		
KWT40	006404	1135#		
KWT41	006456	1152#		
KWT42	006534	1169#		
KWT43	006612	1185#		
KWT44	006646	1197#		
KWT45	006702	1209#		
KWT46	006736	1221#		
KWT47	006772	1234#		
KWT48	007050	1251#		
KWT49	007256	1264	1296#	

[illegible]

		1154*	1171*	1187*	1199*	1211*	1223*	1236*	1253*	1311*	1316*	1341*	1342*	1344*
		1382*	1393*	1426*	1590*	1693*	1704*	1727*	2074*	2089*	2104*	2117*	2125*	2273*
		2495*	2662*	2679*	2685*	2697*	2717*	2719*	2731*	2793*	2803*	2852*	2860*	2864*
		2892*	2939*	2952*	2974*	2984*	2990*	2999*	3044*	3158*	3160*	3162*	3209*	3272*
		3276*	3281*	3285*	3304*	3307*	3324*	3341*	3372*	3511*	3538*	3541*		
PD	017114	2699*												
PDPDLY	001014	165*	2969	2993										
PE	017154	2706*	2716											
PEEA	017120	2700*	2718											
PICO	017014	2668	2677*	2964										
PIC1	017034	2683*	2965											
PIC20	020124	2860*												
PIC3	017234	2686	2726*	2966										
PIC4	017616	2795	2798*	2967										
PIC4B	017654	2804*												
PRINT =	104000	149*	314	316	324	347	363	424	1394	1705	2185	2210	2221	2225
		2242	2274	2404	2406	2409	2491	2493	2663	3081	3136	3141	3146	3375
PRINT1	016630	313*	2402	2408*	2628*									
PROC	016636	2258*	2265	2631*										
PRTOCT=	104005	154*	2223	2411	2414	2417	2420	2423	2426	2429	2432	2435	2438	
PS	001024	169*	3533*											
PSW	001022	168*	261*	322*	429*	1400*	1547*	1573*	1586*	1598*	1712*	2264*	2464*	2669*
PWRFAL	014464	120	328	2251*										
PWRUP	014520	2259	2264*											
P3	017312	2736*	2796											
P3A	017354	2746*	2753											
P3B	017426	2759*	2766											
P3C	017500	2772*	2779											
P3D	017552	2785*	2792											
P4	017700	2809*	2853											
P4A	017742	2820*	2832											
P4B	020042	2839*	2851											
QMARK	016604	364	2243	2618*										
RATE	007632	1153*	1170*	1186*	1198*	1210*	1222*	1235*	1252*	1277	1337*	1359*	1360	1384*
RD1	022522	3308*	3345*											
RELAY	020660	146	2977*	2991										
REPEAT	007336	1299	1332*											
RETURN	015572	430*	1397*	1711*	2465	2467*	2470*	2668*						
RING	015677	2494	2500*											
ROUTPT	020624	2959	2964*											
RTCCSR	023174	3529	3532*	3536*	3547*									
RO	=%000000	110*	266	268*	271*	272	277*	280*	289*	402*	1343	1345	1346*	1347*
		1349*	1355	1358*	1368*	1377*	1670*	1902*	1911*	1918*	1932*	1939*	1957*	1965*
		1983*	1992*	2010*	2021*	2043*	2077*	2078*	2079*	2092*	2093*	2094*	2148	2192*
		2217*	2229*	2230	2232	2251	2272*	2742*	2752*	2758*	2765*	2771*	2778*	2784*
		2791*	2807*	2819*	2831*	2838*	2850*	2878*	2888*	2898*	3222	3223*	3234*	3237
		3245*	3269	3270*	3271*	3273	3274	3286*	3301	3302*	3303*	3305	3308	3343*
		3397	3400*	3416*	3460	3463*	3481*							
		2677*	2683*	2688*	2706	2710	2715							
ROX	017056	111*	267	269*	273	276*	281*	283	284*	290*	292	293*	2149	2166*
R1	=%000001	2167*	2168	2170	2172	2178	2183	2191*	2195	2198	2203*	2206	2208	2216*
		2219	2228*	2234*	2235*	2236*	2237*	2239	2252	2271*	2732*	2804*	2879*	2886*
		2899*	3097	3098*	3099*	3102*	3153*	3154	3159	3161	3188*	3357	3358*	3361*
		3367*	3371*	3398	3401*	3406*	3409	3415*	3461	3464*	3470*	3473	3480*	

R1Y	017060	2678*	2684*	2689*	2703									
R2	=%000002	112*	282*	285*	291*	294*	2150	2190*	2215*	2232*	2233*	2237	2253	2270*
R3	=%000003	2699*	2707	2733*	2805*	2900*	2269*	2698*	2715	2734*	2806*			
R4	=%000004	113*	2151	2189*	2214*	2254	2204	2213*	2255	2268*	2695*	2735*	2749	2762
R5	=%000005	114*	2152	2154*	2172*	2188*	2204	2213*	2255	2268*	2695*	2735*	2749	2762
		2775	2788	2808*	2823	2828	2842	2847	3221	3246*	3251*			
		115*	305*	341*	374*	377*	380*	437*	439*	446*	448*	456*	458*	466*
		468*	479*	491*	493*	503*	505*	515*	517*	527*	529*	540*	542*	552*
		554*	564*	566*	576*	578*	589*	591*	601*	603*	608*	610*	620*	622*
		627*	629*	638*	642*	654*	656*	659*	661*	671*	674*	676*	679*	681*
		691*	693*	696*	698*	701*	703*	712*	715*	718*	720*	722*	725*	727*
		736*	739*	742*	744*	746*	749*	751*	760*	765*	768*	770*	772*	775*
		777*	792*	794*	797*	800*	803*	805*	811*	820*	834*	836*	839*	841*
		844*	846*	853*	856*	860*	863*	865*	881*	883*	886*	888*	891*	893*
		900*	903*	907*	910*	912*	929*	931*	934*	936*	939*	941*	948*	951*
		955*	958*	960*	976*	978*	981*	983*	986*	988*	995*	998*	1002*	1005*
		1007*	1023*	1025*	1028*	1031*	1034*	1036*	1042*	1045*	1053*	1056*	1059*	1061*
		1064*	1066*	1072*	1075*	1077*	1087*	1090*	1100*	1103*	1113*	1116*	1126*	1129*
		1139*	1142*	1157*	1162*	1174*	1179*	1190*	1202*	1214*	1226*	1239*	1245*	1256*
		1261*	1274*	1277*	1280*	1284*	1288*	1290*	1299*	1335*	1337*	1338*	1339*	1340*
		1353*	1354	1356*	1363*	1365*	1368*	1372*	1375*	1377*	1380*	1382*	1407*	1409*
		1411*	1416*	1418*	1426*	1429*	1438*	1440*	1448*	1450*	1458*	1461*	1463*	1471*
		1474*	1476*	1485*	1487*	1499*	1501*	1514*	1516*	1526*	1528*	1538*	1540*	1551*
		1553*	1564*	1566*	1577*	1579*	1590*	1593*	1607*	1609*	1617*	1619*	1629*	1631*
		1639*	1641*	1650*	1652*	1656*	1665*	1667*	1670*	1673*	1719*	1721*	1723*	1730*
		1741*	1743*	1753*	1755*	1765*	1767*	1778*	1780*	1790*	1792*	1802*	1804*	1814*
		1816*	1826*	1828*	1839*	1841*	1849*	1851*	1861*	1863*	1871*	1873*	1883*	1886*
		1888*	1894*	1906*	1909*	1922*	1924*	1930*	1942*	1944*	1947*	1949*	1955*	1969*
		1971*	1977*	1981*	1996*	1998*	2004*	2008*	2025*	2028*	2030*	2039*	2074*	2077*
		2089*	2092*	2104*	2107*	2117*	2120*	2153	2187*	2212*	2256	2267*	2313	2314*
		2318	2320*	2322	2324	2336	2345	2346*	2351*	2352*	2353*	2354	2363*	2386*
		2388*	2390*	2392*	2394*	2396*	2398*	2400*	2511*	2518*	2695*	2703*	2706*	2710*
		2713*	2715*	2730*	2739*	2741*	2746*	2749*	2752*	2758*	2762*	2765*	2771*	2775*
		2778*	2784*	2788*	2791*	2802*	2812*	2814*	2819*	2823*	2826*	2828*	2831*	2836*
		2838*	2842*	2845*	2847*	2850*	2873*	2875*	2878*	2884*	2897*	2904*	2906*	2909*
		2913*	2916*	2919*	2922*	2926*	2930*	2935*	2938*	2981*	2984*	2987*	2990*	3009*
		3012*	3015*	3033*	3039*	3043*	3090*	3112*	3169*	3189*	3196*	3247*	3287*	3310*
		3327*	3344*	3412*	3417*	3420*	3433	3435	3440*	3441*	3475*	3482*	3486*	3502*
		3570*	3577*	3584*										
		2313*	2320	2345*	2363	2368*								
		151*	442	452	462	499	511	523	536	548	560	572	585	597
		616	635	668	688	709	733	757	789	1083	1422	1444	1467	1481
		1495	1522	1534	1546	1560	1572	1585	1613	1625	1635	1662	1726	1749
		1761	1774	1786	1798	1810	1822	1835	1845	1857	1867	1879	1901	
		2455	2463*	2476	2488									
		2299	2453*											
		339*	2460	2462*	2466*	2469*	2481							
		2457	2459	2461	2466*	2478	2480	2489						
		2300	2474*											
		2298	2305	2486*										
		2461*	2482											
		150*	831	878	926	973	1020	1050	1938	1964				
		152*	474	487	651	1096	1109	1122	1135	1152	1169	1185	1197	1209
		1221	1234	1251	1298	1310	1412	1434	1454	1510	1601	1647	1677	1736
SAVS	015140													
SCOPE =	104002													
SCOPEB	015544													
SCOPEC	015500													
SCOPEF	015570													
SCOPEG	015556													
SCOPEH	015574													
SCOPEI	015634													
SCOPEJ	015536													
SCOPEO=	104001													
SCOPE1=	104003													

SAVS 015140
SCOPE = 104002SCOPEB 015544
SCOPEC 015500
SCOPEF 015570
SCOPEG 015556
SCOPEH 015574
SCOPEI 015634
SCOPEJ 015536
SCOPEO= 104001
SCOPE1= 104003

[illegible]

[illegible]

H09

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11 CROSS REFERENCE TABLE

MACY11 27(654) 15-DEC-77 08:35 PAGE 100

SEQ 0111

VCT3	012020	1761#															
VCT4	012062	1774#															
VCT5	012124	1786#															
VCT6	012166	1798#															
VCT7	012230	1810#															
VCT8	012272	1822#															
VCXREG	001066	193#	1721	1839	1841	1849	1851	1883	1888	1977	2104	2107	2398	2677			
		2684	2732	2739	2749	2752	2775	2778	2804	2814	2828	2831	2838	2847			
		2850	2899	2909	2913	2916	2930										
VCYREG	001070	194#	1723	1861	1863	1871	1873	1886	1894	2004	2117	2120	2400	2678			
		2683	2733	2741	2762	2765	2788	2791	2805	2812	2823	2826	2836	2842			
		2845	2900	2906	2935	2938											
VECTOR	001116	225#															
VECTPS	001120	226#															
VERSN	001122	228#															
VISUAL	016722	142	362	2662#													
VSLOA	017006	2672	2674#														
VSUALO	016736	2666#	2866														
VTEST1	011572	398	1707#														
VTEST2	016732	399	2665#														
WHAT	001774	345	368#	407													
WHATA	002116	382	384	386	390#												
WHATB	002130	391	393#	1314													
WHATC	002142	394	396#	1696													
WHATD	002154	399#	2128														
WHATE	002160	397	400#	2863													
WHAT1	002214	371	413#														
WHAT2	002222	374	415#														
WHAT3	002230	377	417#														
W1	022352	3273#	3288#														
W2	022354	3277#	3289#														
W3	022356	3282#	3290#														
XSPACE	014014	2137#	2141	2301													
XTTYIN	014046	123	2148#	2303													
\$AERR	022102	306	3089#	3185#	3211#	3502											
\$BDDAT	001166	233#	448	458	468	479	493	505	517	529	542	554	566	578			
		591	603	610	622	629	642	661	681	703	727	751	777	805			
		811	816#	817#	820	846	865	893	912	941	960	988	1007	1036			
		1066	1077	1090	1103	1116	1129	1142	1157	1162	1174	1179	1190	1202			
		1214	1226	1239	1245	1256	1261	1284	1290	1345*	1347	1350	1418	1429			
		1440	1450	1487	1501	1516	1528	1540	1553	1566	1579	1593*	1594	1609			
		1619	1631	1641	1656	1673	1730	1743	1755	1767	1780	1792	1804	1816			
		1828	1841	1851	1863	1873	1888	1894	1909	1930	1949	1955	1971	1981			
		1998	2008	2030	2077	2092	2107	2120	2710*	2713	2715	2828*	2831	2884			
		2926	3015	3039													
\$CNT	022756	3436#	3438#	3444#													
\$DATR	022524	3325#	3342#	3346#	3479												
\$GDDAT	001170	234#	1288	1290	1343*	1346	2823*	2826									
\$INLP	022760	448	458	468	479	493	505	517	529	542	554	566	578	591			
		603	610	622	629	642	656	661	676	681	698	703	722	727			
		746	751	772	777	800	805	811	841	846	853	860	865	888			
		893	900	907	912	936	941	948	955	960	983	988	995	1002			
		1007	1031	1036	1042	1061	1066	1072	1077	1090	1103	1116	1129	1142			
		1157	1162	1174	1179	1190	1202	1214	1226	1239	1245	1256	1261	1284			

\$LOAD 022104
\$LPAI 021224
\$LPW 022526
\$OFS = 023022
\$OUTLP 022622

\$PUTS 023176
\$RESET 023054

\$SAD 022754
\$TLKR 022360
\$TLKW 022244
\$TMDAT 001174

1288	1290	1365	1372	1377	1418	1429	1440	1450	1463	1476	1487	1501
1516	1528	1540	1553	1566	1579	1593	1609	1619	1631	1641	1656	1670
1673	1730	1743	1755	1767	1780	1792	1804	1816	1828	1841	1851	1863
1873	1888	1894	1909	1924	1930	1944	1949	1955	1971	1981	1998	2008
2025	2030	2039	2077	2092	2107	2120	2386	2388	2390	2392	2394	2396
2398	2400	2715	2749	2762	2775	2788	2823	2828	2842	2847	2875	2884
2913	2919	2926	2930	2935	2981	2987	3015	3039	3460*	3502	3570	3577
3090	3221*											
3063*	3420	3486										
3272	3276	3281	3285	3304	3307	3324	3341	3357*				
3476*	3573	3574	3580									
305	374	377	380	437	439	446	456	466	491	503	515	527
540	552	564	576	589	601	608	620	627	638	654	659	671
674	679	691	693	696	701	712	715	718	720	725	736	739
742	744	749	760	765	768	770	775	792	794	797	803	820
834	836	839	844	856	863	881	883	886	891	903	910	929
931	934	939	951	958	976	978	981	986	998	1005	1023	1025
1028	1034	1045	1053	1056	1059	1064	1075	1087	1100	1113	1126	1139
1274	1277	1280	1335	1337	1363	1368	1375	1380	1382	1407	1409	1411
1416	1426	1438	1448	1458	1461	1471	1474	1485	1499	1514	1526	1538
1551	1564	1577	1590	1607	1617	1629	1639	1650	1652	1665	1667	1719
1721	1723	1741	1753	1765	1778	1790	1802	1814	1826	1839	1849	1861
1871	1883	1886	1906	1922	1942	1947	1969	1977	1996	2004	2028	2074
2089	2104	2117	2518	2695	2703	2706	2710	2713	2730	2739	2741	2746
2752	2758	2765	2771	2778	2784	2791	2802	2812	2814	2819	2826	2831
2836	2838	2845	2850	2873	2878	2897	2904	2906	2909	2916	2922	2938
2984	2990	3009	3012	3033	3043	3397*						
3569*												
279	320	400	401	423	476	1087	1100	1113	1126	1139	1311	1393
1426	1590	1693	1704	1727	2074	2089	2104	2117	2125	2273	2662	2860
3499*	3509											
3433	3435*	3443*										
3301*	3475											
3269*	3412											
236*	443*	446	448	453*	456	458	463*	466	468	475*	488*	491
493	500*	503	505	512*	515	517	524*	527	529	537*	540	542
549*	552	554	561*	564	566	573*	576	578	586*	589	591	598*
601	617*	620	656*	659	671*	674	676*	679	693*	696	698*	701
712*	715*	718	722*	725	736*	739*	742	746*	749	760*	765*	768
772*	775	794*	797	800*	803	836*	839	841*	844	853*	856	860*
863	883*	886	888*	891	900*	903	907*	910	931*	934	936*	939
948*	951	955*	958	978*	981	983*	986	995*	998	1002*	1005	1025*
1028	1031*	1034	1042*	1045	1053*	1056*	1059	1061*	1064	1072*	1075	1084*
1087	1097*	1100	1110*	1113	1123*	1126	1136*	1139	1274*	1277*	1280	1360*
1363	1365*	1368	1372*	1375	1377	1413*	1416	1423*	1426	1435*	1438	1445*
1448	1455*	1458*	1461	1468*	1471*	1474	1482*	1485	1496*	1499	1511*	1514
1523*	1526	1535*	1538	1548*	1551	1561*	1564	1574*	1577	1587*	1590	1614*
1617	1626*	1629	1636*	1639	1670	1738*	1741	1750*	1753	1762*	1765	1775*
1778	1787*	1790	1799*	1802	1811*	1814	1823*	1826	1836*	1839	1846*	1849
1858*	1861	1868*	1871	1880*	1883*	1886	1903*	1906	1919*	1922	1924	1944
1947	1966*	1969	1993*	1996	2025*	2028	2039	2071*	2074	2086*	2089	2101*
2104	2114*	2117	2515*	2518	2700*	2703*	2706	2707*	2710	2715	2743*	2746
2749*	2752	2755*	2758	2762*	2765	2768*	2771	2775*	2778	2781*	2784	2788*
2791	2816*	2819	2842*	2845	2847*	2850	2870*	2873	2875*	2878	2901*	2904

.DVLS 001124

2906*	2909	2913*	2916	2919*	2922	2930	2935*	2938	2981*	2984	2987*	2990
3005*	3009*	3012	3030*	3033								
3112	3169	3196	3310	3327	3433#							
242#												
235#	305	374	377	380	437	439	608	627	638	654	671	691
693	712	720	736	744	760	770	792	794	834	836	881	883
929	931	976	978	1023	1025	1053	1274	1335	1337	1380	1382	1407
1409	1411	1607	1650	1652	1665	1667	1719	1721	1723	1942	1977	2004
2695	2730	2802	2897	3043								
119#	122#	125#	130#	132#	134#	158#	231#	344	352	355	358	361
388	449	459	469	480	494	506	518	530	543	555	567	579
592	604	611	623	630	643	662	682	704	728	752	778	806
847	866	894	913	942	961	989	1008	1037	1067	1078	1091	1104
1117	1130	1143	1158	1163	1175	1180	1191	1203	1215	1227	1240	1246
1257	1262	1305	1419	1430	1441	1451	1464	1477	1488	1502	1517	1529
1541	1554	1567	1580	1595	1603	1610	1620	1632	1642	1657	1674	1679
1682	1695	1731	1744	1756	1768	1781	1793	1805	1817	1829	1842	1852
1864	1874	1889	1895	1925	1950	1972	1999	2032	2056	2059	2080	2095
2108	2121	2127	2138	2196	2248#	2328	2350	2358	2371	2442	2444	2623#
2862	3476											
230#	275*	2978*	3006*	3029*	3400	3421	3463	3487				

[illegible]

ADD	284 2847 2234 2518 307 555 961 1430 1610 1864 2461 3466 1351 2141 350 3022 2355 288 900 1401 3236 3581 681 2671 2357 1305 604 1674 274 999 2011 2455 2889 3208 643 2444 298 2240 3017 3335 275 788 2159 3028 3485 1458 273 751 1450 1681 2058 3468 351 2322	293 2951 2235 2519 310 567 989 1441 1620 1874 2480 3469 2141 817 431 907 1433 3271 3582 1162 2947 623 1731 286 1046 2042 2457 2931 3232 1348 2927 318 2244 3023 3363 302 1019 2160 3029 3533 1471 299 777 1463 1743 2174 354 3123	1291 2959 2236 2520 344 579 1008 1451 1632 1889 2510 483 936 1599 3303 3583 1179 3105 662 1910 295 1246 2044 2476 2938 3238 1950 3016 365 2325 3045 3369 313 1280 2161 3089 3536 448 805 1476 1755 2193 357 3125	1688 3440 2289 382 592 1037 1464 1642 1895 2862 656 948 1713 3532 1245 3164 704 1931 352 1262 2056 2478 2948 3322 1999 3040 388 2337 3066 3377 330 1396 2162 3098 458 865 1516 1767 2460 360 3130	2066 3504 2958 384 611 1067 1477 1695 2020 2971 676 955 1737 1261 3321 812 1956 355 1292 2059 2488 2950 3339 2032 3252 394 2350 3083 3382 332 1400 2163 3187 468 912 1528 1780 2463 1487 3180	2237 3576 2972 386 728 1078 1488 1756 2080 3106 698 983 2025 1924 3365 1158 1972 358 1369 2184 2523 2996 3362 2138 3530 407 2482 3086 3407 334 1494 2228 3224 493 960 1540 1792 2481 1501 3207	2291 449 728 1051 1502 1744 2095 3131 722 995 2356 1944 3365 1175 1982 361 1492 2196 2672 2998 3368 2165 808 2489 3120 3422 336 1494 2466 3225 505 1007 1553 1804 2715 2168 3359	2315 459 752 1104 1517 1768 2108 3155 746 1002 2673 1944 3365 1191 2009 391 1506 2199 2716 3100 3503 2177 814 2686 3138 3437 338 1598 2507 3229 517 1036 1566 1816 2930 2170	2347 469 778 1117 1529 1781 2121 3183 772 1031 2674 2019 3365 1203 2040 397 1679 2209 2753 3103 3535 2317 1039 2718 3143 3471 339 1707 2665 3233 529 1066 1579 1828 3087 2183	2447 480 806 1130 1541 1793 2127 3240 800 1042 2875 2377 3124 3540 2328 1264 2794 3148 3488 369 370 2022 2669 3241 542 1077 1609 1841 3154 2195	2707 494 847 1143 1554 1805 2201 3360 816 1061 2981 2454 3126 3133 2332 1490 2795 3177 3488 369 370 2022 2669 3358 554 1290 1619 1851 3231 2198	2749 506 866 1163 1567 1817 2207 3366 518 894 913 1180 1580 1829 2231 3403 566 841 1072 3026 2456 3133 2335 1504 2853 3204 369 370 2022 2669 3401 578 1350 1631 1863 3237 2206	2762 518 894 1180 1580 1829 2231 3403 530 913 1313 1595 1842 2323 3405 591 1418 1653 1888 3404 2208	2823 530 913 1313 1595 1842 2323 3405 888 1372 3228 2477 3165 2371 2205 2973 3255 426 429 2157 3006 3436 591 1418 1653 1888 3404 2219	2828 543 942 1419 1603 1852 2459 3434 888 1372 3228 2487 3181 2442 2238 2991 3318 638 2158 3021 3464 727 1440 1667 1894 3433 2318
-----	--	--	---	---	--	---	--	---	--	--	--	--	---	--	---

COM	413	415	417											
DEC	285	294	856	903	951	998	1368	1911	1932	1957	1983	2010	2041	2043
	2202	2334	2361	2522	2752	2765	2778	2791	2831	2850	2886	2888	2949	2970
	3545													2140
EMT	149	150	151	152	153	154	155	156						
HALT	119	387	1680	1683	2057	2060	2260	2288	2445	3253	3381			
INC	271	820	1045	1317	1365	1698	2130	2173	2408	2462	2521	2865	2919	2995
	3080	3099	3132	3134	3185	3230	3242	3361	3367	3406	3438	3470		3043
INCB	645	1491	1505	3102	3156	3251	3539							
JMP	137	138	139	140	141	142	143	144	145	146	311	345	353	356
	362	392	395	398	399	1314	1318	1604	1696	1699	2128	2131	2194	2197
	2293	2465	2796	2863	2866	2962								359
JSR	279	305	320	341	374	377	380	400	401	403	423	437	439	446
	456	458	466	468	476	479	491	493	503	505	515	517	527	529
	542	552	554	564	566	576	578	589	591	601	603	608	610	620
	627	629	638	642	654	656	659	661	671	674	676	679	681	691
	696	698	701	703	712	715	718	720	722	725	727	736	739	742
	746	749	751	760	765	768	770	772	775	777	792	794	797	800
	805	811	820	834	836	839	841	844	853	856	856	860	863	865
	883	886	888	891	893	900	903	907	910	912	929	931	934	936
	941	948	951	955	958	960	976	978	981	983	986	988	995	998
	1005	1007	1023	1025	1028	1031	1034	1036	1042	1045	1053	1056	1059	1061
	1066	1072	1075	1077	1087	1090	1100	1103	1113	1116	1126	1129	1139	1142
	1157	1162	1171	1174	1179	1187	1190	1199	1202	1211	1214	1223	1226	1236
	1245	1253	1256	1261	1274	1277	1280	1284	1288	1290	1299	1311	1316	1335
	1341	1342	1344	1363	1365	1368	1372	1375	1377	1380	1382	1393	1407	1409
	1416	1418	1426	1429	1438	1440	1448	1450	1458	1461	1463	1471	1474	1476
	1487	1499	1501	1514	1516	1526	1528	1538	1540	1551	1553	1564	1566	1577
	1590	1593	1607	1609	1617	1619	1629	1631	1639	1641	1650	1652	1656	1665
	1670	1673	1693	1697	1704	1719	1721	1723	1727	1730	1741	1743	1753	1755
	1767	1778	1780	1790	1792	1802	1804	1814	1816	1826	1828	1839	1841	1849
	1861	1863	1871	1873	1883	1886	1888	1894	1906	1909	1922	1924	1930	1942
	1947	1949	1955	1969	1971	1977	1981	1996	1998	2004	2008	2025	2028	2030
	2074	2077	2089	2092	2104	2107	2117	2120	2125	2129	2273	2382	2386	2388
	2392	2394	2396	2398	2400	2518	2662	2679	2685	2695	2697	2703	2706	2710
	2715	2717	2730	2731	2739	2741	2746	2749	2752	2758	2762	2765	2771	2775
	2784	2788	2791	2793	2802	2803	2812	2814	2819	2823	2826	2828	2831	2836
	2842	2845	2847	2850	2852	2860	2864	2873	2875	2878	2884	2897	2904	2906
	2913	2916	2919	2922	2926	2930	2935	2938	2981	2984	2987	2990	3009	3012
	3033	3039	3043	3044	3090	3112	3158	3160	3162	3169	3196	3272	3276	3281
	3304	3307	3310	3324	3327	3341	3412	3420	3475	3486	3502	3570	3577	3285
MOV	260	261	266	267	268	269	272	276	277	280	281	282	283	287
	290	291	292	296	300	301	308	312	321	322	323	328	329	331
	335	337	340	346	371	374	377	380	402	427	428	430	443	453
	475	488	500	512	524	537	549	561	573	586	598	617	671	693
	715	736	739	760	765	794	836	849	883	896	931	944	978	991
	1053	1056	1084	1097	1110	1123	1136	1153	1170	1186	1198	1210	1222	1235
	1274	1277	1337	1338	1339	1340	1343	1345	1346	1352	1358	1360	1377	1397
	1399	1413	1423	1435	1445	1455	1468	1511	1523	1535	1547	1548	1561	1573
	1586	1587	1614	1626	1636	1670	1684	1686	1687	1709	1710	1711	1738	1750
	1775	1787	1799	1811	1823	1836	1846	1858	1868	1880	1883	1902	1903	1918
	1939	1965	1966	1992	1993	2021	2062	2064	2065	2071	2077	2086	2092	2101
	2139	2148	2149	2150	2151	2152	2153	2154	2166	2172	2187	2188	2189	2190
	2192	2203	2212	2213	2214	2215	2216	2217	2229	2232	2239	2251	2252	2253

	2255	2256	2257	2258	2259	2264	2265	2266	2267	2268	2269	2270	2271	2272	2283
	2285	2292	2313	2314	2320	2326	2329	2330	2345	2346	2348	2349	2360	2363	2379
	2380	2400	2401	2464	2467	2506	2508	2512	2515	2666	2667	2668	2677	2678	2683
	2684	2695	2696	2698	2699	2700	2703	2710	2726	2727	2730	2732	2733	2734	2735
	2742	2743	2755	2758	2768	2771	2781	2784	2798	2799	2802	2804	2805	2806	2807
	2808	2816	2819	2838	2870	2878	2897	2898	2899	2900	2901	2906	2945	2960	2961
	2969	2977	2993	3004	3005	3009	3025	3027	3030	3064	3079	3088	3097	3101	3110
	3153	3188	3221	3222	3223	3227	3234	3245	3246	3269	3270	3273	3274	3277	3286
	3301	3302	3308	3343	3357	3371	3397	3398	3400	3409	3411	3415	3416	3418	3435
	3460	3461	3463	3473	3479	3480	3481	3484	3531	3569	3575				
MOV B	1482	1496	2178	2324	2333	2354	2359	2514	3152	3157	3159	3161	3166	3194	3275
	3278	3280	3282	3283	3284	3305	3306	3323	3325	3340	3342	3573	3574	3580	
NEG	1349														
NOP	305	404	405	406	2311	2312	2343	2344	2922						
RESET	3499														
ROL	2351	2352	2353												
RTI	414	416	418	2143	2218	2321	2364	2373	2446	2448	2468	3546			
RTS	1293	1353	1356	1382	2495	2511	2524	2719	2892	2939	2952	2974	2999	3189	3209
	3247	3287	3344	3372	3417	3441	3482	3511	3538	3541	3584				
SUB	1347	1685	2063	2284	2381	2775	2788	2842	2913	2935					
TRAP	148														
TST	297	306	309	343	381	383	385	390	393	396	479	610	622	629	642
	661	846	893	941	988	1090	1103	1116	1129	1142	1312	1354	1429	1594	1673
	1694	1998	2079	2094	2107	2120	2126	2200	2204	2230	2286	2402	2443	2458	2479
ISTB	2509	2861	3039	3250	3402	3410	3465	3474	3502	3529	3534				
	603	703	811	1157	1174	1190	1202	1214	1226	1239	1256	1284	1656	1730	1909
	1930	1949	1955	1971	1981	2008	2030	2039	2137	2164	2176	2316	2327	2331	2336
	2370	2441	2884	2926	3015	3182									
.ASCII	2497	2534	2543	2545	2547	2549	2551	2553	2555	2557	2559	2561	2563	2567	2572
	2576	2581	2588	2591	2594	2599	2605	2607	2608	2611	2614	2616	2618	2619	3085
	3140	3145	3150	3379											
.ASECT	14														
.BLKW	231														
.BYTE	2500														
.ENABL	33														
.END	3587														
.ENDC	228	278	3115	3122	3172	3179	3199	3206	3313	3320	3330	3337	3549		
.EVEN	2501	2623	2855	3085	3140	3145	3150	3379							
.GLOBL	14														
.IF	225	278	3114	3120	3171	3177	3198	3204	3312	3318	3329	3335	3549		
.IFF	3114	3120	3171	3177	3198	3204	3312	3318	3329	3335					
.LIST	14	30	34	119	305	374	377	380	437	439	446	448	456	458	466
	468	479	491	493	503	505	515	517	527	529	540	542	552	554	564
	566	576	578	589	591	601	603	608	610	620	622	627	629	638	642
	654	656	659	661	671	674	676	679	681	691	693	696	698	701	703
	712	715	718	720	722	725	727	736	739	742	744	746	749	751	760
	765	768	770	772	775	777	792	794	797	800	803	805	811	820	834
	836	839	841	844	846	853	856	860	863	865	881	883	886	888	891
	893	900	903	907	910	912	929	931	934	936	939	941	948	951	955
	958	960	976	978	981	983	986	988	995	998	1002	1005	1007	1023	1025
	1028	1031	1034	1036	1042	1045	1053	1056	1059	1061	1064	1066	1072	1075	1077
	1087	1090	1100	1103	1113	1116	1126	1129	1139	1142	1157	1162	1174	1179	1190
	1202	1214	1226	1239	1245	1256	1261	1274	1277	1280	1284	1288	1290	1335	1337
	1363	1365	1368	1372	1375	1377	1380	1382	1407	1409	1411	1416	1418	1426	1429

	1438	1440	1448	1450	1458	1461	1463	1471	1474	1476	1485	1487	1499	1501	1514
	1516	1526	1528	1538	1540	1551	1553	1564	1566	1577	1579	1590	1593	1607	1609
	1617	1619	1629	1631	1639	1641	1650	1652	1656	1665	1667	1670	1673	1719	1721
	1723	1730	1741	1743	1753	1755	1765	1767	1778	1780	1790	1792	1802	1804	1814
	1816	1826	1828	1839	1841	1849	1851	1861	1863	1871	1873	1883	1886	1888	1894
	1906	1909	1922	1924	1930	1942	1944	1947	1949	1955	1969	1971	1977	1981	1996
	1998	2004	2008	2025	2028	2030	2039	2074	2077	2089	2092	2104	2107	2117	2120
	2386	2388	2390	2392	2394	2396	2398	2400	2518	2695	2703	2706	2710	2713	2715
	2730	2739	2741	2746	2749	2752	2758	2762	2765	2771	2775	2778	2784	2788	2791
	2802	2812	2814	2819	2823	2826	2828	2831	2836	2838	2842	2845	2847	2850	2873
	2875	2878	2884	2897	2904	2906	2909	2913	2916	2919	2922	2926	2930	2935	2938
	2981	2984	2987	2990	3009	3012	3015	3033	3039	3043	3085	3140	3145	3150	3379
	3502														
.MACRO	17	18	19	20	22	24	25	26	27	28	29	30	233	258	
.NLIST	14	30	35	119	305	374	377	380	437	439	446	448	456	458	466
	468	479	491	493	503	505	515	517	527	529	540	542	552	554	564
	566	576	578	589	591	601	603	608	610	620	622	627	629	638	642
	654	656	659	661	671	674	676	679	681	691	693	696	698	701	703
	712	715	718	720	722	725	727	736	739	742	744	746	749	751	760
	765	768	770	772	775	777	792	794	797	800	803	805	811	820	834
	836	839	841	844	846	853	856	860	863	865	881	883	886	888	891
	893	900	903	907	910	912	929	931	934	936	939	941	948	951	955
	958	960	976	978	981	983	986	988	995	998	1002	1005	1007	1023	1025
	1028	1031	1034	1036	1042	1045	1053	1056	1059	1061	1064	1066	1072	1075	1077
	1087	1090	1100	1103	1113	1116	1126	1129	1139	1142	1157	1162	1174	1179	1190
	1202	1214	1226	1239	1245	1256	1261	1274	1277	1280	1284	1288	1290	1335	1337
	1363	1365	1368	1372	1375	1377	1380	1382	1407	1409	1411	1416	1418	1426	1429
	1438	1440	1448	1450	1458	1461	1463	1471	1474	1476	1485	1487	1499	1501	1514
	1516	1526	1528	1538	1540	1551	1553	1564	1566	1577	1579	1590	1593	1607	1609
	1617	1619	1629	1631	1639	1641	1650	1652	1656	1665	1667	1670	1673	1719	1721
	1723	1730	1741	1743	1753	1755	1765	1767	1778	1780	1790	1792	1802	1804	1814
	1816	1826	1828	1839	1841	1849	1851	1861	1863	1871	1873	1883	1886	1888	1894
	1906	1909	1922	1924	1930	1942	1944	1947	1949	1955	1969	1971	1977	1981	1996
	1998	2004	2008	2025	2028	2030	2039	2074	2077	2089	2092	2104	2107	2117	2120
	2386	2388	2390	2392	2394	2396	2398	2400	2518	2695	2703	2706	2710	2713	2715
	2730	2739	2741	2746	2749	2752	2758	2762	2765	2771	2775	2778	2784	2788	2791
	2802	2812	2814	2819	2823	2826	2828	2831	2836	2838	2842	2845	2847	2850	2873
	2875	2878	2884	2897	2904	2906	2909	2913	2916	2919	2922	2926	2930	2935	2938
	2981	2984	2987	2990	3009	3012	3015	3033	3039	3043	3085	3140	3145	3150	3379
	3502														
.PSECT	14														
.REM	1	14	37												
.REPT	119														
.TITLE	32														
.WORD	208	211	213	215	217	219	221	223	225	226	228	230	242	305	374
	377	380	437	439	446	448	456	458	466	468	479	491	493	503	505
	515	517	527	529	540	542	552	554	564	566	576	578	589	591	601
	603	608	610	620	622	627	629	638	642	654	656	659	661	671	674
	676	679	681	691	693	696	698	701	703	712	715	718	720	722	725
	727	736	739	742	744	746	749	751	760	765	768	770	772	775	777
	792	794	797	800	803	805	811	820	834	836	839	841	844	846	853
	856	860	863	865	881	883	886	888	891	893	900	903	907	910	912
	929	931	934	936	939	941	948	951	955	958	960	976	978	981	983
	986	988	995	998	1002	1005	1007	1023	1025	1028	1031	1034	1036	1042	1045

1053	1056	1059	1061	1064	1066	1072	1075	1077	1087	1090	1100	1103	1113	1116
1126	1129	1139	1142	1157	1162	1174	1179	1190	1202	1214	1226	1239	1245	1256
1261	1274	1277	1280	1284	1288	1290	1335	1337	1363	1365	1368	1372	1375	1377
1380	1382	1407	1409	1411	1416	1418	1426	1429	1438	1440	1448	1450	1458	1461
1463	1471	1474	1476	1485	1487	1499	1501	1514	1516	1526	1528	1538	1540	1551
1553	1564	1566	1577	1579	1590	1593	1607	1609	1617	1619	1629	1631	1639	1641
1650	1652	1656	1665	1667	1670	1673	1719	1721	1723	1730	1741	1743	1753	1755
1765	1767	1778	1780	1790	1792	1802	1804	1814	1816	1826	1828	1839	1841	1849
1851	1861	1863	1871	1873	1883	1886	1888	1894	1906	1909	1922	1924	1930	1942
1944	1947	1949	1955	1969	1971	1977	1981	1996	1998	2004	2008	2025	2028	2030
2039	2074	2077	2089	2092	2104	2107	2117	2120	2386	2388	2390	2392	2394	2396
2398	2400	2518	2688	2689	2695	2703	2706	2710	2713	2715	2730	2739	2741	2746
2749	2752	2758	2762	2765	2771	2775	2778	2784	2788	2791	2802	2812	2814	2819
2823	2826	2828	2831	2836	2838	2842	2845	2847	2850	2873	2875	2878	2884	2897
2904	2906	2909	2913	2916	2919	2922	2926	2930	2935	2938	2981	2984	2987	2990
3009	3012	3015	3033	3039	3043	3091	3191	3211	3346	3413	3414	3421	3443	3444
3477	3487	3502	3512	3513	3543	3547	3571	3572	3578	3579	3585			

000000

ERRORS DETECTED: 0

LPS DIAGNOSTIC TEST II MAINDEC-11-DRLPI-A
DRLPI.P11

D10
MACY11 27(654) 15-DEC-77 08:35 PAGE 109

SEQ 0120

*DRLPI, DRLPI/SOL/CRF=DRLPA.MAC, DRLPI
RUN-TIME: 10 14 2 SECONDS
CORE USED: 21K