LPA-11

SYSTEM EXERCISER

MD-11-DRLPA-A

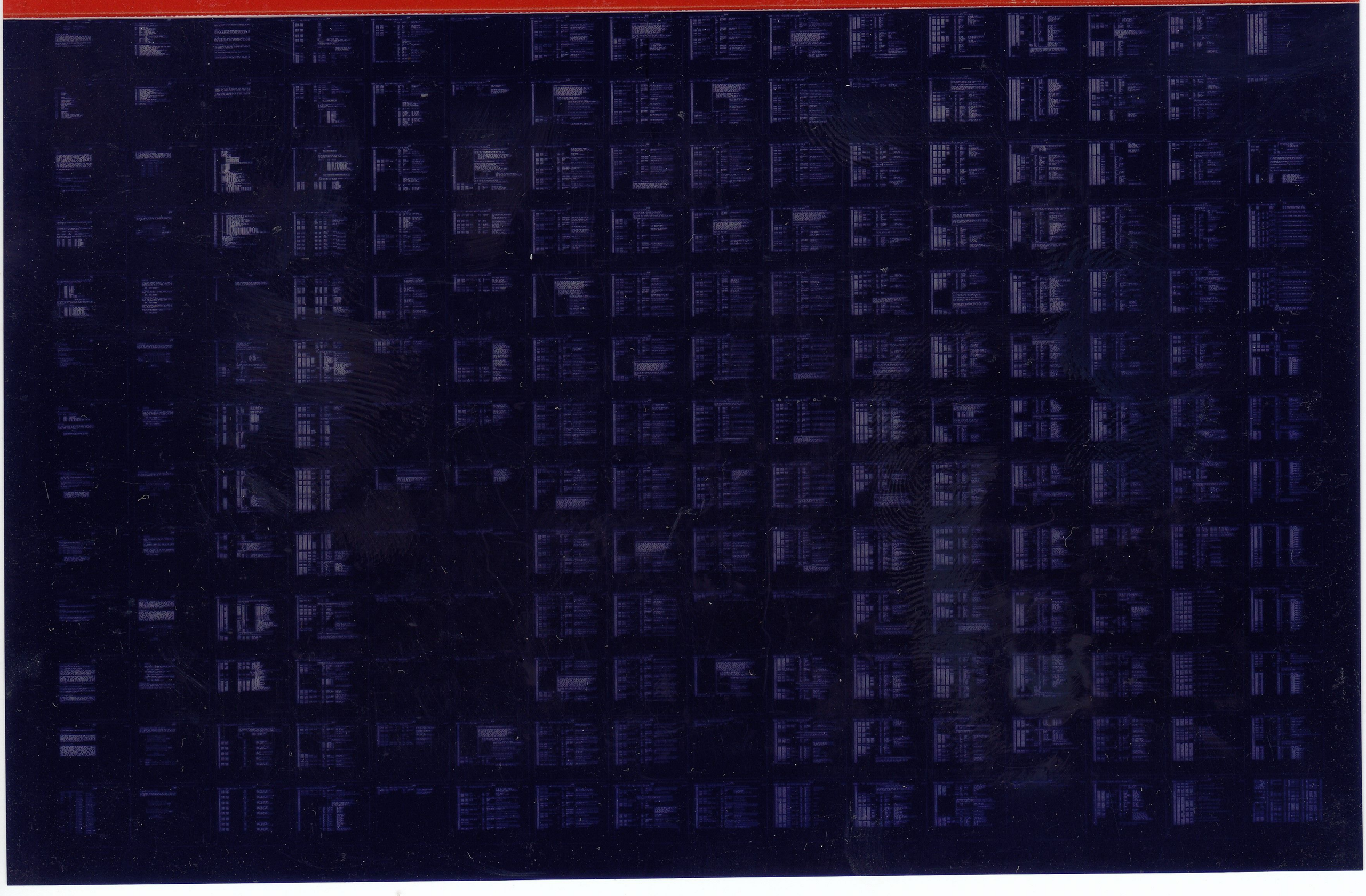
EP-DRLPA-A-DL

FICHE 1 OF 2

MAR 1978

digital

MADE IN USA



LPA-11

SYSTEM EXERCISER

MD-11-DRLPA-A

EP-DRLPA-A-DL

FICHE 2 OF 2

MAR 1978

digital

MADE IN USA

EOF1DECPOS01 00010000 780223 IDENTIFICATION1 HDR1DRLPAASEQ 00010000 780223 SEQ 0001

PRODUCT CODE: MAINDEC-11-DRLPA-A-D

PRODUCT NAME: LPA-11 SYSTEM EXERCISER

DATE CREATED: JAN. 1978

TABLE OF CONTENTS

1.0 ABSTRACT

THIS PROGRAM WAS DESIGNED TO EXERCISE THE LPA-11XX SUBSYSTEM. IT IS DIVIDED INTO TWO SECTIONS. THE FIRST SECTION EXERCISES EACH INDIVIDUAL HARDWARE COMPONENT ON THE SYSTEM. PLEASE NOTE THAT THE DEFAULT SYSTEM WHICH THE PROGRAM USES, HAS ONE AD11K AND ONE KW11K ON IT. THE USER MUST INFORM THE PROGRAM AS TO ANY DIFFERENCES IN CONFIGURATION. IF THE PROGRAM DETECTS A HARDWARE PROBLEM, IT INFORMS THE USER OF IT. THE USER SHOULD RUN THE DIAGNOSTIC DESIGNED TO DIAGNOSE THE SECTION OF HARDWARE THAT FAILED. (EXAMPLE IF DRLPA INFORMED THE USER THAT THE AD11K FAILED, THE USER SHOULD RUN THE LPA/AD11K DIAGNOSTIC.)

THE SECOND PART OF THE DIAGNOSTIC IS DESIGNED TO RUN WITH USER (KMC) MICRO-CODE. THIS IS THE FIRST (AND ONLY) TIME THAT M8254 (IPBM) INTERRUPT ARBITRATION LOGIC IS CHECKED. IF ANY PROBLEMS OCCUR HERE, ITS A GOOD BET THAT THE M8254 MODULE IS BAD. IF NOT, YOU'RE GOING TO HAVE TO CABLE THE I/O BUS TO THE UNIBUS IN ORDER TO RUN THE MORE DETAILED DIAGNOSTIC AVAILABLE FOR THE OPTIONS.

GOOD SCOPE LOOPS ARE HARD TO COME BY SINCE THIS PROGRAM MUST TRY TO KEEP THREE PROCESSORS IN SYNC.

2.0 REQUIREMENTS

2.1 EQUIPMENT

1. PDP-11 FAMILY COMPUTER WITH 16K OF MEMORY (OR MORE) AND CONSOLE I/O FACILITIES (I.E. TTY)
2. LPA-11X TO BE EXERCISED
3. (OPTIONAL) KW11L OR KW11P (IMPROVES PROGRAM EXECUTION)

2.2 STORAGE

THIS PROGRAM OCCUPIES AND USES 16K OF MEMORY.

NOTE

IF 20K OR MORE MEMORY IS AVAILABLE, THIS PROGRAM WILL RUN A/D SAMPLING AT MAXIMUM SPEED.

3.0 LOADING PROCEDURE

3.1 METHOD

STANDARD PROCEDURE FOR NORMAL BINARY PROGRAM SHOULD BE FOLLOWED. THIS PROGRAM IS SUPPLIED ON MULTI-MEDIA AND CAN BE LOADED BY XXDP,ACT, OR APT.

3.2 NON-STANDARD ADDRESS,VECTOR, CONFIGURATION

THIS PROGRAM IS SET UP TO CHECK AN LPA-11X WITH STANDARD SET-UP AS LISTED BELOW. IT IS IMPORTANT THAT IF THE EQUIPMENT ADDRESSES VARY, THAT YOU ONLY CHANGE THESE LOCATIONS.

TAG	ADDRESS	CONTENTS	COMMENTS
$BASE:	001250	170460	;;BASE ADDRESS OF EQUIPMENT
$VECT1:	001244	000300	;;VECTOR LOCATION

THE FOLLOWING ARE A LIST OF DEVICE ADDRESSES. YOU MAY CHANGE THE ADDRESS OF A DEVICE ONLY BY MODIFYING THE FOLLOWING LIST. DO NOT CHANGE ANY OTHER ADDRESSES!

AD11K:	1566	170400	;AD11K ADDRESS.
KW11K:	1570	170404	;KW11K ADDRESS.
DR11K1:	1572	167770	;DR11K #1 ADDR.
AA11K:	1574	170416	;AA11 ADDRESS.
AD11K2:	1576	170440	;AD11K #2 ADDRESS.
DR11K2:	1600	167760	;DR11K #2 ADDRESS.
DR11K3:	1602	167750	;DR11K #3 ADDRESS.
DR11K4:	1604	167740	;DR11K #4 ADDRESS.
DR11K5:	1606	166730	;DR11K #5 ADDRESS.
AR11:	1610	170400	;AR11 ADDRESS.
LPS11:	1612	170400	;LPS11 BASE ADDRESS.

SR1 INFORMS THIS DIAGNOSTIC AS TO WHAT DEVICES ARE ON THE I/O BUS. SR1 DEFAULTS TO 1 KW11K AND 1 AD11K IF YOUR CONFIGURATION IS DIFFERENT, YOU MUST CHANGE THIS LOCATION.

WORD BIT=1	OCTAL	DEVICE
0	000001	1ST AD11K
1	000002	1ST KW11K
2	000004	1ST DR11K
3	000010	1ST AA11K
4	000020	2ND AD11K#2
5	000040	2ND DR11K
6	000100	RESERVED
7	000200	3RD DR11K
8	000400	4TH DR11K
9	001000	5TH DR11K
10	002000	AR11
11	004000	RESERVED
12	010000	LPSAD (LPS A/D)
13	020000	LPSKW (LPS REAL TIME CLOCK)
14	040000	LPSVC (LPS D/A)
15	100000	LPSDR (LPS DIGITAL J/O)

```
SR1:    1562    003             ;DEVICE PRESENT FLAG DEFAULT
                                ;IS 1 KW11K, 1 AD11K
```

SR2 INFORMS THIS DIAGNOSTIC AS TO HOW THE DEVICES SELECTED BY SR1 ARE SET-UP FOR TEST. WHILE NO SPECIAL SETUP IS REQUIRED TO RUN THIS TEST, THE DEPTH OF COVERAGE WILL INCREASE IF SPECIAL SETUPS ARE PERFORMED.

WORD BIT=1	OCTAL	FUNCTION
0	000001	AD11K HAS G5036 WRAP-AROUND MODULE.
2	000004	DR11K #1 HAS LOOP BACK CABLE
3	000010	AA11K HAS SCOPE DISPLAY (VISUAL TEST).
4	000020	AD11K #2 HAS G5036 WRAP AROUND MODULE.
5	000040	DR11K #2 HAS LOOPBACK CABLE
7	000200	DR11K #3 HAS LOOP BACK CABLE.
8	000400	DR11K #4 HAS LOOP BACK CABLE.
9	001000	DR11K #5 HAS LOOP BACK CABLE.
10	002000	AR11 HAS G5034 WRAP AROUND MODULE.
11	004000	AR11 HAS SCOPE DISPLAY (VISUAL TEST)
14	040000	LPS-11 D/A HAS SCOPE DISPLAY (VISUAL TEST)
15	100000	LPS-11 DIGITAL I/O HAS LOOP BACK CABLE

```
SR2:    1564    0               ;DEVICE SETUP REG.
```

SEQ 0006

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

BEFORE STARTING THE DIAGNOSTIC, SET ALL SWITCH REGISTER BITS AS DESIRED, SEE SECTION 5.1.

4.2 STARTING ADDRESS

200 START OF TEST

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY
2. LOAD ADDRESS 200
3. SET SWITCH REGISTER TO DESIRED SETTING
4. SET UP LOCATIONS SR1: AND SR2: TO REFLECT THE SYSTEM CONFIGURATION.
5. START PROGRAM

5.0 OPERATING PROCEDURE

5.1 SWITCH REGISTER FUNCTION

SWR BIT	OCTAL	FUNCTION WHEN SET
15	100000	HALT ON ERROR
14	040000	LOOP ON TEST
13	020000	INHIBIT ERROR TYPEOUT
11	004000	INHIBIT TEST ITERATIONS
10	002000	RUN ONLY DEDICATED MODE USER MICRO-CODE
9	001000	RUN ONLY MULTIUSER MODE USER MICRO-CODE
8	000400	RUN TEST SELECTED BY <7:0>

5.2 SCOPE LOOPS

IF AN ERROR OCCURS AND THE USER WISHES TO sCOPE THE ERROR SWITCH REG. BIT 15 SHOULD BE SET TO HALT ON ERROR. WHEN THE CPU IS HALTED ON ERROR, SWITCH REG BIT 14 (LOOP ON TEST) AND SWR BIT 13 (INHIBIT ERROR TYPEOUT) SHOULD BE SET. SWR BIT 15 SHOULD BE CLEARED AND CPU SHOULD BE CONTINUED.

NOTE

SOME SCOPE LOOPS MAY BE IMPOSSIBLE TO OBTAIN OR NOT REPEATIBLE DUE TO THE FACT THAT THREE ASYNCONISE CPUS ARE RUNNING TO GENERATE THE ERROR.

5.3 PROGRAM AND/OR OPERATOR ACTION

1. WHEN THE PROGRAM IS INITIALLY STARTED IT WILL TYPE:

   MD-11-DRLPA-A

   THE FIRST "PASS" THROUGH THE PROGRAM IS QUICK VERIFY OR A SHORT ONE. ALL OTHER PASSES WILL ITERATE ON EACH SUBTEST UNLESS INHIBITED.

2. THE PROGRAM PERFORMS TESTS ON THE KMC11.

3. THE PROGRAM PERFORMS TESTS ON THE M8200-YC AND M8254.

4. THE PROGRAM TESTS EACH OPTION ON THE I/O BUSS SELECTED BY THE OPERATOR.

5. THE PROGRAM LOADS USER MICRO-CODE INTO THE KMCII AND EXERCISES TOTAL LPAII-KX SYSTEM.

6. PROGRAM END PASS.

NOTE

ON ALL EVEN PASSES THROUGH THE PROGRAM, THE PROGRAM SELECTS MULTIUSER MICRO-CODE FOR USER MICRO CODE; AND ON ALL ODD NUMBERED PASSES THE PROGRAM SELECTS DEDICATED MODE MICRO-CODE FOR USER MICRO-CODE. THE OPERATOR MAY INFORM THE PROGRAM TO ONLY RUN ON VERSION OF USER-MICRO-CODE BY USE OF THE SWITCH REGISTER (SEE SECTION 5.1)

6.0 ERRORS

6.1 ERROR PRINT-OUT

PRINTOUT VARIES WITH THE ERROR DETECTED. THE ERROR PC TYPED OUT IS THE ACTUAL LOCATION OF THE ERROR CALL.

THE FOLLOWING IS A LIST OF LABELS ASSOCIATED WITH LPA11 FUNCTIONAL ERROR.

TSTWD - TEST IN WHICH THE ERROR WAS DETECTED.

USJNO - USER JOB NUMBER ASSIGNED TO THE JOB BY THE MICRO-CODE.

ALPCO - LPA11 CONTROL OUT REGISTER CONTENTS.

ALPCI - LPA11 CONTROL IN REGISTER CONTENTS.

ALPSO - LPA11 STATUS OUT REGISTER CONTENTS.

NOTE

ERROR REPORTS MAY OR MAY NOT SPECIFY A DEVICE. IF SO, THAT DEVICE IS NOT NECESSARILY THE FAULTY UNIT. HOWEVER, IT IS THE DEVICE UNDER TEST WHEN THE ERROR WAS DETECTED. FOR FURTHER INFORMATION, THE OPERATOR MUST CONSULT THE LISTING.

7.0 RESTRICTIONS

NONE.

SEQ 0010

8.0 MISCELLANEOUS

8.1 POWER FAIL

THIS PROGRAM WILL NOT SUPPORT POWER FAILURES, IF A POWER FAILURE OCCURS, YOU MUST RELOAD AND RESTART THIS PROGRAM.

8.2 XXDP,ACT,APT

THIS PROGRAM IS CHAINABLE UNDER XXDP, ACT, OR APT. ALTOUGH "APT HOOKS" HAVE BEEN INSTALLED, THEY HAVE NOT BEEN TESTED.

8.3 EXECUTION TIME

THE EXECUTION TIME WILL VARY BETWEEN CPUS. EXECUTION TIME ALSO VARIES WITH THE NUMBER AND TYPE OF OPTIONS ON THE LPA-11XX SYSTEM. THE APPROXIMATE TIMES ARE LISTED BELOW:

1.0 MINUTE (60 SEC) -NO ERRORS-ITERATIONS INHIBITED
3.0 MINUTE (180 SEC) -NO ERRORS-WITH ITERATIONS.
(LISTED UNDER MIS.)

8.4 LPA11 (SYSTEM) DIAGNOSTIC SUMMARY

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.

8.5 LPA11 VERSION 3 MICRO-CODE ERROR SUMMARY

MICRO-CODE ERRORS ARE DFINED AS ANY ERROR RETURNED TO THE USER BY THE MICRO-CODE THROUGH THE LPA11 STATUS REGISTER.

WITHIN THE DIAGNOSTIC WE CHECK TO BE SURE THAT WE CAN GENERATE SOME OF THESE ERRORS. OTHERS MAY COME UP AT ANY TIME WHILE THE LPA11-XX MICRO-CODE IS BEING RUN.

8.5.1 FATAL HARDWARE ERRORS

THE FATAL HARDWARE ERRORS CONSIST OF DATA ERRORS, IMPROPER STATUS CONDITIONS, OR MALFUNCTIONS WHICH ARE DETECTED DURING THE INTIALIZATION OF THE LPA11 SUBSYSTEM OR DURING THE PROCESSING OF DATA. NO SUBSEQUENT DATA WILL BE TRANSFERRED AND THE LPA11 WILL NOT RESPOND TO ADDTIONAL COMMANDS FROM THE PDP-11. THE LPA11 MUST BE RE-INTIALIZED PRIOR TO ACCEPTING ANOTHER COMMAND FROM THE PDP-11.

THESE ERRORS MAY COME UP IF THERE IS AN ARBITRATION ERROR.

8.5.2 START REQUEST ERRORS

THE START REQUEST ERRORS CONSIST OF ERRORS DETECTED BY THE MASTER DURING THE PROCESSING OF A NEW PDP-11 COMMAND TO THE LPA11.

8.5.3 USER REQUEST ERRORS

THE USER REUQEST ERRORS ARE SPECIFIED BY ERROR CODE 0-4 AND ERROR STATUS BITS 0 AND 1 OF THE STATUS OUT, BYTE 3. THE REQUEST IS DEALLOCATED AND MUST BE REISSUED IN ORDER TO RESUME. THE USER REQUESTS ERROR CODES FROM 240 TO 247 ARE GENRATED BY THE MASTER MICROPROCESSOR AND ERROR CODES FROM 250 TO 254 AND 257 ARE GENERATED BY CONDITIONS DETECTED IN THE SLAVE MICROPROCESSOR AND TRANSFERRED TO THE CSR OF THE MASTER. IN THE MULTI REQUEST MODE THE USER FOR WHICH THE ERROR APPLIES IS IDENTIFIED BY THE USER INDEX CODE OF THE CONTROL OUT WORD BYTE 2.

8.5.4 NORMAL STATUS RETURNS

THE NORMAL STATUS RETURNS ARE INDICATED BY ERROR CODE 0-4. THE STATUS RETURNS INDICATE A FULL OR OVERRUN CONDITION OF THE PDP-11 MEMORY BUFFERS ASSIGNED TO STORE DATA FROM THE LPA11. THE BUFFER ADDRESSES ARE SPECIFIED BY THE RDA INFORMATION CONTAINED IN THE I/O DEVICE START COMMAND. THE BUFFER FULL AND OVERRUN STATUS CAN BE SPECIFIED AS A FATAL ON NON FATAL CONDITION BY THE RDA VALID BUFFER MASK CONFIGURATION OF THE USER STATUS WORD. ANY OF THE THREE NORMAL STATUS RETURNS WILL CAUSE AN LPA11 CONTROL OUT INTERRUPT REQUEST TO BE GENERATED. THE BUFFER FULL CONDITION EXISTS WHEN THE LPA11 HAS FILLED AN ASSIGNED BUFFER WITH DATA. THE BUFFER OVERRUN CONDITION IS A RESULT OF ALL THE ASSIGNED BUFERS BEING LOADED BEFORE THE PDP-11 HAS PROCESSED OR TRANSFERRED THE DATA FROM THE BUFFER. IF THE BUFFER OVERRUN CONDITIONS OCCUR CONTINUALLY; A USER REQUEST ERROR WILL RESULT.

THE LPA11-KX DIAGNOSTIC KIT WILL INCLUDE:

OPTION	GROUP	DIAG. #	DIAG. TITLE
LPA11-KX	LEVEL 2 DIAG.	MD-11-DRLPA	LPA11-KX SYSTEM
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-K	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-DRLPL	LPA/M8200-YC BASIC MICRO-CPU R/W TEST
	B	MD-11-DRLPM	LPA/M8200-YC JMP+ROM READ TEST

8.6 LPA-11 VERSION 3 MICRO-CODE ERROR LIST

```
      ********************  FATAL HARDWARE ERRORS  ********************

      322 --- ADDRESS OF NON-EXISTENT DEVICE
      340 --- SLAVE POWER LOW
      341 --- MASTER FIFO READ / WRITE ERROR
      342 --- I/O BUS SACK TIME OUT
      343 --- MASTER INITIAL CONDITION ERROR
              BYTE 6 --- EXPECTED STATUS
              BYTE 7 --- BAD STATUS
      344 --- MASTER / SLAVE VERSION ERROR
              BYTE 6 --- MASTER VERSION NUMBER
              BYTE 7 --- SLAVE VERSION NUMBER
      345 --- SLAVE COLD START TIME-OUT
      346 --- FATAL SLAVE ERROR
              BYTE 6 --- 200 --- SLAVE FIFO READ / WRITE ERROR
                         201 --- KW11-K CLOCK "A" OVERRUN
                         202 --- SLAVE FIRMWARE FIFO SEQUENCE ERROR
      347 --- FPATH DATA ERROR
              BYTE 6 --- BAD VALUE
              BYTE 7 --- EXPECTED VALUE
      350 --- FIFO DATA ERROR
              BYTE 6 --- BAD VALUE
              BYTE 7 --- EXPECTED VALUE

      ********************  START REQUEST ERRORS  ********************

      300 --- NO ROOM FOR REQUEST
      301 --- "GO" SET WITH "RDY IN" CLEAR
      302 --- MULTI-USER REQUEST WITH DEDICATED MODE MICRO CODE LOADED
      304 --- DEDICATED MODE REQUEST WITH MULTI-USER MICRO CODE LOADED
      306 --- "START COMMAND WITH NO "INITIALIZE"
      310 --- MULTIPLE "INITIALIZE" COMMANDS
      312 --- "STOP" COMMAND WITH USER NOT ACTIVE
      314 --- ODD ADDRESS SPEDCIFIED FOR REQUEST DESCRIPTOR ARRAY
      316 --- "INITIALIZE" WITH WRONG VERSION SPECIFIED
      320 --- "START" FOR DEVICE NOT IN CONFIGURATION
      322 --- "START" WITH ILLEGAL FUNCTION SPECIFIED
      324 --- NON EXISTANT MEMORY IN REQUEST DESCRIPTOR ARRAY
      325 --- ODD ADDRESS SPECIFIED FOR BUFFER OR USER STATUS WORD
      326 --- DEVICE NOT FOUND ON I/O BUS DURING "INITIALIZE" COMMAND
              BYTE 6,7 --- ADDRESS OF NON EXISTANT DEVICE
```

********** USER REQUEST ERRORS **********

240 --- NON FATAL ERROR COUNT EXCEEDED
241 --- ERROR STATUS OVERRUN
242 --- NON EXISTANT MEMORY IN RANDOM CHANNEL LIST
243 --- BUFFER OVERRUN / UNDERRUN
244 --- NON EXISTANT MEMORY IN BUFFER
245 --- NON EXISTANT MEMORY IN USER STATUS WORD
246 --- INVALID BUFFER INDEX SPECIFIED IN USER STATUS WORD
247 --- MASTER FIFO 7/8 FULL
250 --- REQUEST TERMINATED BY USER STATUS WORD REQUEST
251 --- NO FIRMWARE FIFO BUFFER AVAILABLE FOR REQUEST
252 --- SLAVE FIFO 7/8 FULL
253 --- RANDOM CHANNEL ADDRESS UNDERRUN
254 --- DATA UNDERRUN
255 --- NON EXISTANT CHANNEL OR DEVICE
260 --- MULTIPLE EXTERNAL TRIGGER DIGITAL OUTPUT REQUESTS

********** NORMAL STATUS RETURNS **********

000 --- START REQUEST PROCESSED
001 --- BUFFER FULL
002 --- BUFFER OVERRUN / UNDERRUN

8.7 LPA-11 (KMC-11) REGISTER DEFINITIONS

THE CONTROL AND STATUS INFORMATION, TRANSFERRED BETWEEN THE PDP-11 UNIBUS AND THE LPA-11 SUBSYSTEM, IS STORED IN THE CONTROL AND STATUS REGISTERS (CSR) OF THE MASTER MICROPROCESSOR (M8204). THE CSR'S CONSIST OF EIGHT, 8-BIT BYTES IMPLEMENTED AS FOUR 16-BIT WORDS OF MULTIPORT RANDOM ACCESS MEMORY (RAM'S). THE CSR LOCATIONS ARE ADDRESSABLE FROM EITHER THE PDP-11 OR THE M8204 MICROPROCESSOR PROGRAM. THE UNIBUS ADDRESSES FOR EACH CSR IS AS FOLLOWS:

MASTER MICROPROCESSOR CSR

BYTE	UNIBUS	NUM	WORD
0 (LOW)	76 XXX 0	LPCI	1
1 (HIGH)	76 XXX 1	MAIN	1
2 (LOW)	76 XXX 2	LPCO	2
3 (HIGH)	76 XXX 3	LPSO	2
4 (LOW)	76 XXX 4	LPADL	3
5 (HIGH)	76 XXX 5		4
6 (LOW)	76 XXX 6	LPMS1	4
7 (HIGH)	76 XXX 7		4

CONTROL IN (BYTE 0)

THE CONTROL IN BYTE CONSISTS OF 8 BITS AT ADDRESS 76XXX0 USED BY THE PDP-11 PROGRAM TO INITIATE THE TRANSFER OF COMMANDS TO THE LPA-11 SUBSYSTEM. THE FORMAT AND DESCRIPTION OF BYTE INFORMATION IS SHOWN ON FIGURE 8-1.

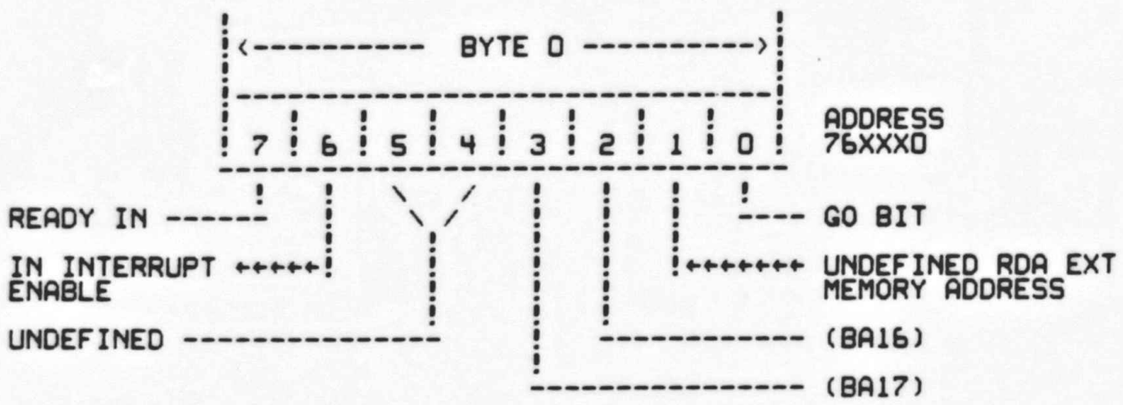


FIGURE 8-1

CONTROL IN (BYTE 0)

BIT	DESCRIPTION
0	GO BIT- INTIAL CONDITION IS LOW (0). SET TO HIGH (1) BY THE PDP-11 PROGRAM TO INDICATE THAT A VALID REQUEST DESCRIPTOR ARRAY (RDA) ADDRESS IS AVAILABLE FOR PROCESSING BY THE LPA-11. CLEARED LOW (0) BY THE MASTER MICROPROGRAM AFTER THE USER'S REQUEST IS INITIATED. READ - LPA-11 MICROPORGRAM WRITE - PDP-11 PROGRAM
1	NOT SPECIFIED
2,3	BUS ADDRESS (BA16,BA17) - EXTENDED MEMORY BITS FOR REQUEST DESCRIPTOR ARRAY (RDA) ADDRESS (BYTES 4 AND 5) OF CSR. READ - PDP-11 PROGRAM WRITE - PDP-11 PROGRAM
4,5	NOT SPECIFIED (MST BE LOWD)
6	IN INTERRUPT ENABLE - INITIAL CONDITION IS LOW (0). SET TO HIGH (1) BY PDP-11 PROGRAM TO ENABLE INTERUPT REQUEST TO BE GENERATED WHEN THE READY IN, BIT 07, IS HIGH (1) OR DURING A LOW TO hIGH TRANSITION. READ - PDP-11 PROGRAM WRITE - PDP-11 PROGRAM AN INTERRUPT OCCURS AT VECTOR ADDRESS +4. A READY IN INTERRUPT MAY OCCUR EITHER IMMEDIATELY OR AFTER A USEC DELAY. THE LPA-11 WILL CAUSE AN OUT INTERRUPT OR OUT STATUS BEFORE THE READY IN BIT IS SET IF BOTH ARE PENDING SIMULTANEOUSLY.
7	READY IN - INTIAL CONDITION IS LOW 0. SET TO HIGH (1) BY MICROPROCESSOR WHEN READY TO ACCEPT USER'S REQUEST. CLEARED LOW 0 AFTER REQUEST IS GRANTED. READ - PDP-11 PROGRAM WRITE - MICROPROCESSOR

MAINTENANCE (BYTE 1)

THE MAINTENANCE BYTE IS AN EIGHT BIT, HIGH BYTE AT ADDRESS 76XXX1 WHICH CAN BE MONITORED BY THE PDP-11 PROGRAM DURING DIAGNOSTIC FUNCTIONS. REFER TO THE PROGRAMMING SECTION 4 OF THE KMC-11 MAINTENANCE MANUAL FOR A DESCRIPTION OF THE BITS NOT DEFINED. FIGURE 8-2 SHOWS THE BIT CONFIGURATION ON THE MAINTENANCE BYTE.

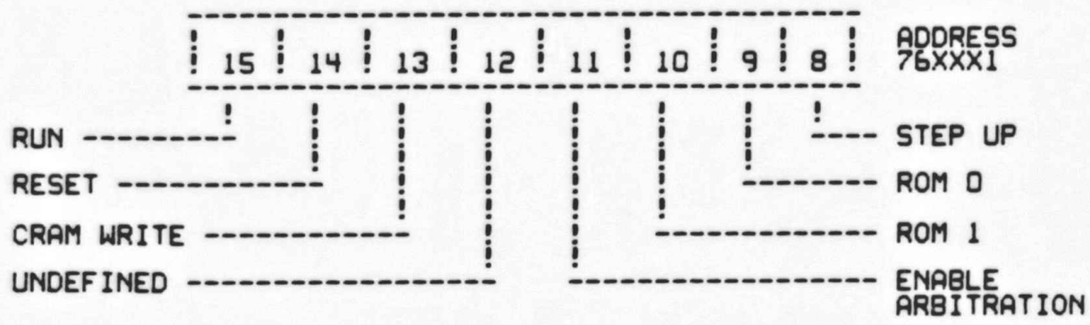


FIGURE 8-2

MAINTENANCE (BYTE 1)

BIT	DESCRIPTION
8,9,10	REFER TO KMC-11 MAINTENANCE MANUAL
11	ENABLE ARBITRATION - SET HIGH (1) BY PDP-11 PROGRAM TO ALLOW THE M8254 MODULE TO ARBITRATE NON PROCESSOR REQUESTS (NPR) AND BUS REQUESTS BR BETWEEN THE SLAVE PROCESSOR AND DEVICES ON THE I/O BUS. WHEN CLEARED LOW (0) BY PDP-11 PROGRAM THE ARBITRATION LOGIC IS INHIBITED. READ - PDP-11 PROGRAM AND MICROPROCESSOR WRITE - PDP-11 PROGRAM
12,13	REFER TO KMC-11 MAINTENANCE MANUAL
14	RESET - SET HIGH (1) BY PDP-11 PROGRAM TO CLEAR ALL PERTINENT REGISTERS IN THE LPA-11 SUBSYSTEM. READ - LPA-11 MICROPROGRAM WRITE - PDP-11 PROGRAM ONLY
15	REFER TO KMC-11 MAINTENANCE MANUAL

CONTROL OUT (BYTE 2)

THE CONTROL OUT BYTE CONSISTS OF 8 BITS AT ADDRESS 76XXX2 AND IS USED BY THE LPA-11 TO INDICATE TO THE PDP-11 THE AVAILABILITY OF THE LPA STATUS INFORMATION. THE FOMAT AND DESCRIPTION OF THE BYTE INFOMATION IS SHOWN ON FIGURE 8-3.

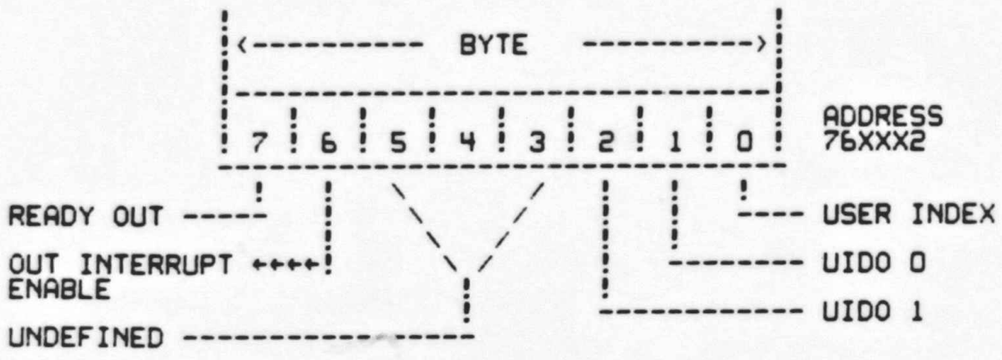


FIGURE 8-3

CONTROL OUT (BYTE 2)

BIT	DESCRIPTION
0,1,2	USER'S INDEX - AN OCTAL CODE (0-7) ASSIGNED BY THE MASTER MICROPROCESOR IN THE MULTI REQUEST MODE TO INDENTIFY THE STATUS AS RELATED TO UP TO EIGHT USER REQUESTS.
3,4,5	NOT SPECIFIED
6	OUT INTERRUPT ENABLE - SET HIGH (1) BY MASTER MICROPROCESSOR PROGRAM TO ALLOW AN INTERRUPT REQUEST TO BE GENERATED AT VECTOR +00 WHEN THE READY OUT (BIT 07) IS SET. WRITE - PDP-11 PROGRAM
7	READY OUT - SET HIGH (1) BY THE MASTER MICROPROCESSOR PROGRAM TO INDICATE THAT THE LPA-11 SUBSYSTEM HAS STATUS INFORMATION FOR THE PDP-11. CLEARED LOW (0) BY THE PDP-11 PROGRAM TO ACKNOWLEDGE THE RECEPTION OF THE LPA-11 STATUS INFORMATION. READ - PDP-11 PROGRAM WRITE - PDP-11 PROGRAM

USER INDEX CODE

A USER INDEX CODE IS SPECIFIED BY THE LPA-11 TO IDENTIFY THE ERROR OR STATUS INFOMATION AS RELATED TO ONE OF EIGHT USER'S REQUESTS. THE USER'S INDEX IS AN OCTAL CODE FORM 0 - 7 AND IS ASSIGNED TO THE USER IN ORDER THAT THE REQUESTS ARE GRANTED. THIS EFFECTIVELY PROVIDES A VARIABLE INDEX TO IDENTIFY FURTHER COMMUNICATIONS WITH AN ASSOCIATED USER. THE USER'S INDEX CODE IS NOT VALID FOR FATAL HARDWARE ERROR CONDITION IN LPA-11 WHICH EFFECT ALL ACTIVE USER'S AND FOR START REQUEST ERRORS WHICH OCCUR PRIOR TO ESTABLISHING THE NEW USER'S REQUEST.

STATUS OUT (BYTE 3)

THE STATUS OUT IS AN 8-BIT, HIGH BYTE AT ADDRESS 76XXX3 WHICH INDICATES STATUS AND ERROR CONDITIONS TO THE PDP-11 PROGRAM DURING THE ESTABLISHMENT OF USER REQUESTS OR DURING THE TRANSFER OF DATA. THE ERRORS ARE CLASSIFIED AS FATAL HARDWARE ERRORS WHICH RESULT IN A HALT OF THE MICROPROCESSOR OR PROGRAM AND NON FATAL ERRORS WHICH INDICATE SPECIFIC STATUS CONDITIONS WHICH HAVE OCCURRED. THE HALT CONDITION PREVENTS THE LPA-11 FROM ACCEPTING ANY ADDITONAL USER REQUESTS OR FROM TRANSFERRING ANY DATA TO OR FROM THE PDP-11. THE STATUS OUT BYTE CONTAINS FIVE ERROR CODE BITS 8 - 12, TWO ERROR STATUS BITS 13,14 AND FATAL ERROR INDICATOR, BIT 15, AS SHOWN ON FIGURE 8-4. THE ERROR CODES LISTED ON TABLE 8-1, 8-2 AND 8-3 ARE GROUPED AS SHOWN ON THE STATUS OUT BIT CONFIGURATION.

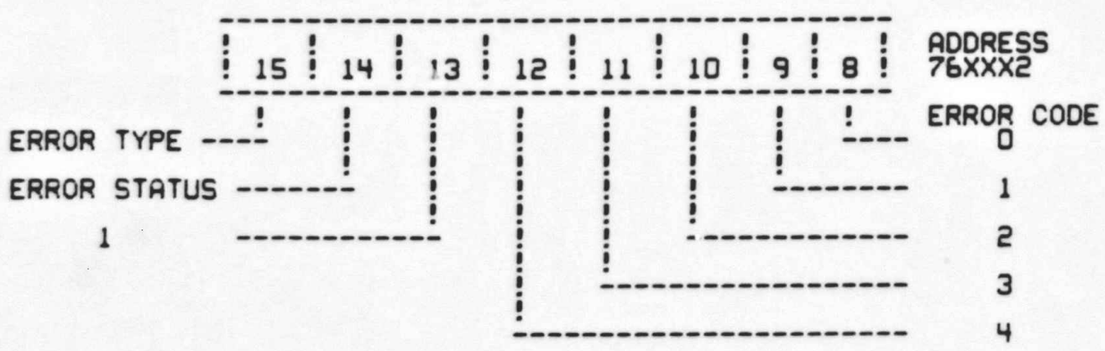


FIGURE 8-4
STATUS OUT (BYTE 3)

BIT	DESCRIPTION
8-12	ERROR CODE 0-4 - A SUB CODE AS LISTED ON TABLE WHICH FURTHER DEFINES THE ERROR CODE INDICATED BY THE ERROR STATUS 0 (BIT 14) AND ERROR STATUS 1 (BIT 1). READ - PDP-11 PROGRAM WRITE - MASTER MICROPROCESSOR
13,14	ERROR STATUS 0 AND 1 - INDICATES THE CONDITIONS FROM WHICH THE ERROR OR STATUS ORIGINATED READ - PDP-11 PROGRAM WRITE - MASTER MICROPROCESSOR
15	STATUS/ERROR INDICTOR - A SINGLE BIT WHICH INDICATES STATUS OR FATAL ERROR CONDITIONS REQUIRING ATTENTION. A HIGH (1) INDICATES AN ERROR AND A LOW (0) INDICATES STATUS. READ - PDP-11 PROGRAM WRITE - MASTER MICROPROCESSOR

ERROR/STATUS CODES

THE ERROR/STATUS CODES INDICATED BY THE STATUS OUT BYTE 3 ARE DEFINED AS FATAL HARDWARE ERRORS, START REQUEST ERRORS, USER REQUEST ERRORS AND NORMAL STATUS RETURNS.

THE ERROR/STATUS CODES AS LISTED ON THE FOLLOWING TABLES INCLUDE ALL EIGHT BITS STARTING AT BIT 0 TO BIT 7.

REQUEST DEScRIPTOR ARRAY ADDRESS (BYTE 4 AND 5)

THE BUS ADDRESSES OF THE REQUEST DESCRIPTOR ARRAYS (RDA) IN THE PDP-11 MEMORY ARE CONTAINED WITHIN BYTES 4 AND 5 OF THE LPA-11 CSR. BYTE 4 AND 5 EACH CONTAIN 8-ADDRESS BITS (BA00-BA07 AND BA08-BA15) AS SHOWN ON FIGURE 8-5 AND BYTE 0 CONTAINS TWO ADDRESS BITS (BA16 AND BA17).

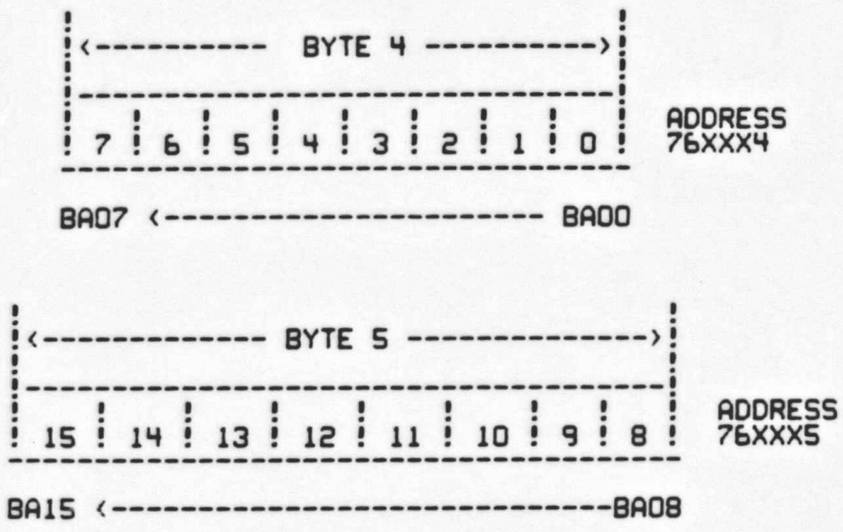


FIGURE 8-5

REQUEST DESCRIPTOR ARRAY ADDRESS

BIT	DEScRIPTION
0-7	BA00-BA07 - LOW BYTE OF THE RDA ADDRESS IN PDP-11 MEMORY
8-15	BA08-BA15 - HIGH BYTE OF THE RDA ADDRESS IN PDP-11 MEMORY.

MAINTENANCE STATUS (BYTE 6 AND 7)

THE MAINTENANCE STATUS, BYTES 6 AND 7, ARE LISTED ON TABLE 8-6 AND INDICATE THE SPECIFIC STATUS AND ERROR CHECK VALUES ASSOCIATED WITH THE FATAL HARDWARE ERRORS DEFINED ON TABLE 8-2. THE FORMAT OF THE LOW AND HIGH 8-BIT ERROR STATUS CODES ARE SHOWN ON FIGURE 8-6.

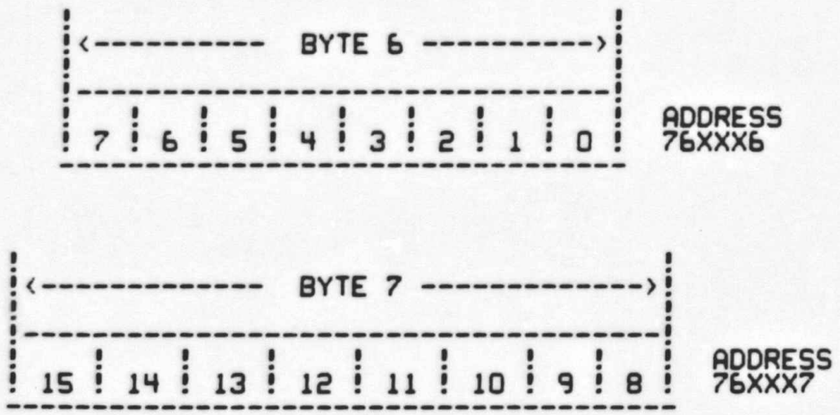


FIGURE 8-6

ERROR STATUS (BYTES 6 AND 7)

8.8 MICRO-CODE PROGRAMS

THERE ARE FOUR SEPARATE MICRO-CODE PROGRAMS LOADED INTO THE KMC11 DURING THE COURSE OF THIS DIAGNOSTIC. A BRIEF DESCRIPTION OF EACH FOLLOWS.

8.8.1 DRLPX1

THIS IS THE FIRST MICRO-CODE PROGRAM LOADED INTO THE KMC11. ITS FUNCTION IS TO VERIFY PROPER OPERATION OF THE KMC11 BOARD. IT WILL PERFORM CRAM READ/WRITE TESTS, BRANCH TESTS, ALU TESTS, NPR TESTS, AND INTERRUPT TESTS. IF THE KMC11 IS FOUND TO BE FAULTY, THE OPERATOR MUST ABORT DRLPA AND RUN THE KMC11 DIAGNOSTICS.

8.8.2 DRLPX0

THE MICRO-CODE PROGRAM "DRLPX0" IS LOADED INTO THE KMC11 AT THE ONSET OF TEST #3. DRLPX0 PROVIDES COMMUNICATIONS FACILITIES BETWEEN THE KMC11 AND THE SLAVE MICRO-PROCESSOR. IT IS USED DURiNG TESTING OF THE SLAVE MICRO-PROCESSOR, THE IPBM, AND THE OPTIONS ON THE I/O BUS.

8.8.3 USER MICRO-CODE

THERE ARE TWO SEPARATE USER MICRO-CODE PROGRAMS LOADED INTO THE KMC11; DEDICATED AND MULTI-USER. THEIR FUNCTION IN RELATION TO THIS DIAGNOSTIC IS TO EXERCISE THE OPTIONS ON THE I/O BUS. A BRIEF DESCRIPTION OF EACH FOLLOWS.

DEDICATED MICRO-CODE

THIS PROGRAM ALLOWS HIGH SPEED SAMPLING OF A/D DEVICES ON THE I/O BUS. ONLY ONE JOB CAN BE RUN AT ANY GIVEN TIME.

MULTI-USER MICRO-CODE

WHEN RUNNING MULTI-USER MICRO-CODE, ANY NUMBER OF JOBS FROM ONE TO EIGHT MAY BE RUN AT THE SAME TIME BY THE LPA11 SUBSYSTEM. THE JOBS MAY BE DIGITAL INPUT, A/D SAMPLING, AND/OR D/A CONVERSTIONS.

8.9 ILLEGAL INTERRUPTS ON START

SINCE THE KMC11 CAN NOT BE INITIALIZED UNTIL TEST #1 IS IN PROGRESS, THERE EXISTS THE POSSIBILITY THAT PREVIOUSLY LOADED MICRO-CODE WILL CAUSE A PROCESSOR INTERRUPT. THIS WILL BE INDICATED BY A PROCESSOR HALT AT THE KMC11 INTR. VECTOR LOCATION.

IN ORDER TO CORRECT THIS PROBLEM, THE OPERATOR MUST MANUALLY CLEAR ALL FOUR KMC11 REGISTERS SEEN BY THE UNIBUS. THIS IS DONE BY LOADING THE BASE ADDRESS OF THE KMC11 AND DEPOSITING ZERO'S IN FOUR CONSECUTIVE WORD LOCATIONS.

THE PROGRAM CAN THEN BE RESTARTED.

```
                        .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 AVAILIBLE FOR TESTING THE LPA SYSTEM.

                                GOOD LUCK !

[
.GLOBL  DRLPX2
```

```
                                ;<BADGER>LPD.P11.54,  1-AUG-77 13:16:41, EDIT BY BADGER
                                .TITLE  MAINDEC -11- DRLPA-A
                                ;*COPYRIGHT (C) 1977
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY EDWARD C. BADGER, GREGORY GLEZMAN
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
                                ;*
000001                          $TN=1

                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                        .ENABL  GBL
                                .SBTTL  TRAP CATCHER

000000                                  .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
000174                                  .=174
000174  000000                 DISPREG: .WORD   0              ;;SOFTWARE DISPLAY REGISTER
000176  000000                 SWREG:   .WORD   0              ;;SOFTWARE SWITCH REGISTER
000100                                  .=100
000100  000104  000200  000002          .WORD   104,200,2

000200                                  .=200
000200  000137  002016                  JMP     START
000220                                  .=220
000220  000137  034546                  JMP     MAKEI

000240                                  .=240
000240  000137  041324                  JMP     RBUFR

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
```

```
                        
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
                        
                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER
                        
                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7
                        
                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
```

```
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
170460                          ABASE=  170460
000300                          AVECT1= 300
000001                          $TN=1
                        .SBTTL  ACT11 HOOKS

                        ;;**************************************************************
                        ;HOOKS REQUIRED BY ACT11
```

```
                    000244                          $SVPC=.                 ;SAVE PC
                    000046                          .=46
000046              030772                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
                    000052                          .=52
000052              000000                          .WORD   0               ;;2)SET LOC.52 TO zERO
                    000244                          .=$SVPC                 ;; RESTORE PC

                    001000                          .=1000
                                        .SBTTL  APT PARAMETER BLOCK

                                        ;;**************************************************************
                                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                        ;;**************************************************************
                    001000                          .$X=.   ;;SAVE CURRENT LOCATION
                    000024                          .=24    ;;SET POWER FAIL TO pOINT TO START OF PROGRAM
000024              000200                          200     ;;FOR APT START UP
                    000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044              001000                          $APTHDR ;;POINT TO APT HEADER BLOCK
                    001000                          .=.$X   ;;RESET LOCATION COUNTER
                                        ;;**************************************************************
                                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                        ;INTERFACE SPEC.

001000                                  $APTHD:
001000  000000      $HIBTS: .WORD   0        ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001174      $MBADR: .WORD   $MAIL    ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000002      $TSTM:  .WORD   2        ;;RUN TIM OF LONGEST TESt
001006  000170      $PASTM: .WORD   120.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000170      $UNItM: .WORD   120.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000031              .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

                                ;;*****************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                    001100              .=1100
001100                          $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                      .WORD   0
001102  000                 $TSTNM: .BYTE   0                    ;;CONTAINS THE TEST NUMBER
001103  000                 $ERFLG: .BYTE   0                    ;;CONTAINS ERROR FLAG
001104  000000              $ICNT:  .WORD   0                    ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000              $LPADR: .WORD   0                    ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000              $LPERR: .WORD   0                    ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000              $ERTTL: .WORD   0                    ;;CONTAINS TOTaL ERRORS DETECTED
001114  000                 $ITEMB: .BYTE   0                    ;;CONTAINS ITEM CONTROL BYTE
001115  001                 $ERMAX: .BYTE   1                    ;;CONTAINS MAX. ERRORS PER TESt
001116  000000              $ERRPC: .WORD   0                    ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000              $GDADR: .WORD   0                    ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000              $BDADR: .WORD   0                    ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000              $GDDAT: .WORD   0                    ;;CONTAINS 'GOOD' DATA
001126  000000              $BDDAT: .WORD   0                    ;;CONTAINS 'BAD' DATA
001130  000000                      .WORD   0                    ;;RESERVED--NOT TO BE USED
001132  000000                      .WORD   0
001134  000                 $AUTOB: .BYTE   0                    ;;AUTOMATIC MODE INDICATOR
001135  000                 $INTAG: .BYTE   0                    ;;INTERRUPT MODE INDICATOR
001136  000000                      .WORD   0
001140  177570              SWR:    .WORD   DSWR                 ;;ADDRESS OF SWITCH REGISTER
001142  177570              DISPLAY: .WORD  DDISP                ;;ADDRESS OF DISPLAY REGISTER
001144  177560              $TKS:   177560                       ;;TTY KBD STATUS
001146  177562              $TKB:   177562                       ;;TTY KBD BUFFER
001150  177564              $TPS:   177564                       ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566              $TPB:   177566                       ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000                 $NULL:  .BYTE   0                    ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002                 $FILLS: .BYTE   2                    ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012                 $FILLC: .BYTE   12                   ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000                 $TPFLG: .BYTE   0                    ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000              $TIMES: 0                            ;;MAX. NUMBER OF ITERATIONS
001162  000000              $ESCAPE:0                            ;;ESCAPE ON ERROR ADDRESS
001164  177607  000377        $BELL:  .ASCIZ  <207><377><377>    ;;CODE FOR BELL
001170  077                 $QUES:  .ASCII  /?/                  ;;QUESTION MARK
001171  015                 $CRLF:  .ASCII  <15>                 ;;CARRIAGE RETURN
001172  000012              $LF:    .ASCIZ  <12>                 ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001174                          $MAIL:                          ;;APT MAILBOX
001174  000000              $MSGTY: .WORD   AMSGTY               ;;MESSAGE TYPE CODE
001176  000000              $FATAL: .WORD   AFATAL               ;;FATAL ERROR NUMBER
001200  000000              $TESTN: .WORD   ATESTN               ;;TEST NUMBER
001202  000000              $PASS:  .WORD   APASS                ;;PASS COUNT
001204  000000              $DEVCT: .WORD   ADEVCT               ;;DEVICE COUNT
```

```
                                $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
                                $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
                                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
                                $ETABLE:                ;;APT ENVIRONMENT TABLE
                                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
                                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
                                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
                                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
                                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                                ;*                      BITS 15-11=CPU TYPE
                                ;*                           11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                           11/70=06,PDQ=07,Q=10
                                ;*                      BIT 10=REAL TIME CLOCK
                                ;*                      BIT  9=FLOATING POINT PROCESSOR
                                ;*                      BIT  8=MEMORY MANAGEMENT
                                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
                                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                ;*                      MEM.TYPE BYTE    --  (HIGH BYTE)
                                ;*                           900 NSEC CORE=001
                                ;*                           300 NSEC BIPOLAR=002
                                ;*                           500 NSEC MOS=003
                                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                .*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
                                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
                                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
                                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
                                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
                                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
                                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
                                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
                                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
                                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
                                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
                                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
                                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
                                $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
                                $ETEND:
                                .MEXIT
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT

001256                          $ERRTB:

                                        ;ITEM   1

001256  041504                    EM1                   ;LPA-11 ADDRESS ERROR
001260  042240                    DH1                   ;ERRPC    ADDRESS
001262  042516                    DT1                   ;$ERRPC,KMADO
001264  042606                    DF0                   ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   2

001266  041527                    EM2                   ;LPA(KMC-11) DATA ERROR
001270  042261                    DH2                   ;ERRPC    EXP'D     REC'D
001272  042524                    DT2                   ;$ERRPC,$GDDAT,$BDDAT
001274  042606                    DF0                   ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   3

001276  041560                    EM3                   ;LPA (KMC-11) INSTRUCTION ERROR
001300  042312                    DH3                   ;ERRPC
001302  042534                    DT3                   ;$ERRPC
001304  042606                    DF0                   ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   4

001306  041620                    EM4                   ;LPA (M8254) INIT.ERROR
001310  042312                    DH3                   ;ERRPC
001312  042534                    DT3                   ;$ERRPC
001314  042606                    DF0                   ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   5

001316  041620                    EM4                   ;LPA (M8254) INIT.ERROR
001320  042261                    DH2                   ;ERRPC    EXP'ED    REC'ED
001322  042524                    DT2                   ;$ERRPC,$GDDAT,$BDDAT
001324  042606                    DF0                   ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   6
```

```
                                ;ITEM   7

001336  041705                    EM7             ;LPA (M8254) FAST PATH DATA ERROR
001340  042261                    DH2             ;ERRPC   EXP'ED   REC'ED
001342  042524                    DT2             ;$ERRPC,$GDDAT,$BDDAT
001344  042606                    DFO             ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   10

001346  041747                    EM10            ;LPA (AD11K) CSR ERROR
001350  042261                    DH2             ;ERRPC   EXP'ED  REC'ED
001352  042524                    DT2             ;$ERRPC,$GDDAT,$BDDAT
001354  042606                    DFO             ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   11

001356  041776                    EM11            ;LPA (KW11K) CSR ERROR
001360  042261                    DH2             ;ERRPC   EXP'ED   REC'ED
001362  042524                    DT2             ;$ERRPC,$GDDAT,$BDDAT
001364  042606                    DFO             ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   12

001366  042025                    EM12            ;LPA (DR11K) ERROR
001370  042261                    DH2             ;ERRPC   EXP'ED  REC'ED
001372  042524                    DT2             ;$ERRPC,$GDDAT,$BDDAT
001374  042606                    DFO             ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   13

001376  000000                    0               ;NO-ERROR 13 (BAD LUCK!)
001400  000000                    0
001402  000000                    0
001404  000000                    0

                                ;ITEM   14

001406  042050                    EM14            ;LPA (AA-11K) ERROR
001410  042261                    DH2             ;ERRPC   EXP'ED  REC'ED
001412  042524                    DT2             ;$ERRPC,$GDDAT,$BDDAT
001414  042606                    DFO             ;ALL NUMBER ARE IN OCTAL FORM.

                                ;ITEM   15

001416  042074                    EM15            ;LPA-11 (AR11K) ERROR
001420  042261                    DH2             ;ERRPC   EXP'ED  REC'ED
001422  042524                    DT2             ;$ERRPC,$GDDAT,$BDDAT
001424  042606                    DFO             ;ALL NUMBER ARE IN OCTAL FORM.
```

```
                                        ;ITEM   16
                                        
001426  042116                          EM16                    ;LPA-11 (LPS-11) ERROR
001430  042261                          DH2
001432  042524                          DT2
001434  042606                          DFO                     ;ALL NUMBER ARE IN OCTAL FORM.
                                        
                                        ;ITEM 17
                                        
001436  042142                          EM17                    ;LPA-11 FUNCTIONAL ERROR
001440  042321                          DH17
001442  042540                          DT17
001444  042606                          DFO                     ;ALL NUMBER ARE IN OCTAL FORM.
                                        
                                        ;ITEM 20
                                        
001446  042173                          EM20
001450  042361                          DH20
001452  042552                          DT20
001454  042606                          DFO                     ;ALL NUMBER ARE IN OCTAL FORM.
                                        
                                        ;ITEM 21
001456  042142                        EM17            ;LPA FUNCTIONAL ERROR
001460  042427                        DH21            ;ERRPC   TSTNO   USJNO   USRDA   ALPCO   ALPCI   ALPSO
001462  042566                        DT21            ;ERRPC,$TEsTN,USJNO,USRDA,ALPCO,ALPCI,APSO
001464  042606                          DFO                   ;ALL NUMBER ARE IN OCTAL FORM.
                                        
                                        ;DEVICE ADDRESSES AS USED BY PROGRAM.
                                        ;DO NOT "PATCH" THESE LOCATIONS
                                        ;LOCATIONS ARE ALTERED BY PROGRAM BASED ON YOUR INFO IN $BASE
                                        
001466                          KMCSR:
                                        ;
                                        ;ADDRESS OF KMC-11 OF LPA-11    THE ADDR FOR KMADO MAY BE
                                        ;                               CHANGED BY THE USER TO REFLECT
                                        ;                               A DIFFERENT KMC-11 ADDR. THE
                                        ;                               REST OF THE ADDRESSES WILL
                                        ;                               BE CHANGED BY THE PROGRAM.
                                        ;
001466                          LPCI:
001466  170460                 KMAD0:  .WORD   ABASE           ;BASE KMC ADDR. MAY BE PATCHED BY USER.
                                        
001470                          LPMR:
001470  170461                 KMAD1:  .WORD   ABASE+1         ;>DO NOT        <;KMC-CSR ADDR
001472                          LPCO:
001472  170462                 KMAD2:  .WORD   ABASE+2         ;>PATCH         <;
001474                          LPSO:
001474  170463                 KMAD3:  .WORD   ABASE+3         ;>THIS AREA     <
001476                          LPADL:
001476  170464                 KMAD4:  .WORD   ABASE+4         ;
001500                          LPADH:
001500  170465                 KMAD5:  .WORD   ABASE+5         ;>DO NOT        <
001502                          LPMS1:
```

```
001502  170466          KMAD6:  .WORD   ABASE+6         ;>PATCH         <
001504                  LPMS2:
001504  170467          KMAD7:  .WORD   ABASE+7         ;>THIS AREA     <

001506  000300          VECTOR: .WORD   AVECT1&777      ;BASE VECTOR OF KMC
001510  000304          VECTPS: .WORD   4+AVECT1&777    ;VECOTR ADDR.+2

001512  000004          VERSN:  .WORD   4               ;CURRENT VERSION NUMBER OF MICROCODE.

001514  000000          .DVLS:  .WORD   0               ;/DEVICE LIST OF I/O ADDR. DEFINED
001516  000020                  .BLKW   16.             ;/BY INIT.


001556  000300          VECT1:  .WORD   AVECT1

001560  000304          VECT2:  .WORD   AVECT1+4

                        .SBTTL          ############################
                        .SBTTL          SR1     DEVICE PRESENT REG
                        .SBTTL          ############################

                                .REM    %

                                SR1 INFORMS THIS DIAGNOSTIC AS TO WHAT DEVICES
                                ARE ON THE I/O BUS.
                                SR1 DEFAULTS TO 1 KW11K,AND  1 AD11K.
                                IF YOUR CONFIGURATION IS DIFFERENT, YOU MUST CHANGE
                                THIS LOCATION

                                WORD BIT=1      OCTAL   DEVICE
                                ----------      -----   ------

                                    0           000001  1ST AD11K
                                    1           000002  1ST KW11K
                                    2           000004  1ST DR11K
                                    3           000010  1ST AA11K
                                    4           000020  2ND AD11K#2
                                    5           000040  2ND DR11K
                                    6           000100  RESERVED
                                    7           000200  3RD DR11K
                                    8           000400  4TH DR11K
                                    9           001000  5TH DR11K
                                   10           002000  AR11
                                   11           004000  RESERVED
                                   12           010000  LPSAD (LPS A/D)
                                   13           020000  LPSKW (LPS REAL TIME CLOCK)
                                   14           040000  LPSVC (LPS D/A)
                                   15           100000  LPSDR (LPS DIGITAL I/O)

                                %

001562  000003          SR1:    .WORD   003             ;DEVICE PRESENT FLAG DEFAULT
                                                        ;IS 1 KW11K, 1 AD11K
```

```
                                .SBTTL          ##########################
                                .SBTTL          SR2     DEVICE SET-UP REG
                                .SBTTL          ##########################

                                        .REM    %

                                        SR2 INFORMS THIS DIAGNOSTIC AS TO HOW THE
                                        DEVICES SELECTED BY SR1 ARE SET-UP FOR TEST.
                                        WHILE NO SPECIAL SETUP IS REQUIRED TO RUN THIS TEST,
                                        THE DEPTH OF COVERAGE WILL INCREASE IF SPECIAL SETUPS
                                        ARE PERFORMED.

                                        WORD BIT=1      OCTAL   FUNCTION
                                        ----------      -----   --------

                                            0           000001  AD11K HAS G5034 WRAP-AROUND MODULE.
                                            2           000004  DR11K #1 HAS LOOP BACK CABLE
                                            3           000010  AA11K HAS SCOPE DISPLAY (VISUAL TEST)
                                            4           000020  AD11K #2 HAS G5034 WRAP AROUND MODULE.
                                            5           000040  DR11K #2 HAS LOOPBACK CABLE
                                            7           000200  DR11K #3 HAS LOOP BACK CABLE.
                                            8           000400  DR11K #4 HAS LOOP BACK CABLE.
                                            9           001000  DR11K #5 HAS LOOP BACK CABLE.
                                           10           002000  AR11 HAS G5034 WRAP AROUND MODULE.
                                           11           004000  AR11 HAS SCOPE DISPLAY (VISUAL TEST)
                                           14           040000  LPS-11 D/A HAS SCOPE DISPLAY (VISUAL TEST)
                                           15           100000  LPS-11 DIGITAL I/O HAS LOOP BACK CABLE
                                        %

001564  000000                  SR2:    .WORD   0

                                .SBTTL          ##################################################
                                .SBTTL          LPA-11 I/O BUS DEVICE ADDRESS DEFINITION AND LIST
                                .SBTTL          ##################################################

                                        .REM %
                                        THE FOLLOWING ARE A LIST OF DEVICE ADDRESSES.  YOU MAY
                                        CHANGE THE ADDRESS OF A DEVICE ONLY BY MODIFING THE FOLLOWING LIST.
                                        DO NOT CHANGE ANY OTHER ADDRESSES!
                                        %

001566  170400                  AD11K:  .WORD   170400          ;AD11K ADDRESS.
001570  170404                  KW11K:  .WORD   170404          ;KW11K ADDRESS.
001572  167770                  DR11K1: .WORD   167770          ;DR11K #1 ADDR.
001574  170416                  AA11K:  .WORD   170416          ;AA11 ADDRESS.
001576  170440                  AD11K2: .WORD   170440          ;AD11K #2 ADDRESS.
001600  167760                  DR11K2: .WORD   167760          ;DR11K #2 ADDRESS.
001602  167750                  DR11K3: .WORD   167750          ;DR11K #3 ADDRESS.
001604  167740                  DR11K4: .WORD   167740          ;DR11K #4 ADDRESS.
001606  167730                  DR11K5: .WORD   167730          ;DR11K #5 ADDRESS.
001610  170400                  AR11K:  .WORD   170400          ;AR11 ADDRESS.
001612  170400                  LPS11:  .WORD   170400          ;LPS11 BASE ADDRESS.
```

```
                                .REM    $
                                        END OF PATCHABLE DEVICE ADDRESSES.
                                $
                                                .SBTTL  AD11-K ADDRESSES


001614  170400                  STREG1: .WORD   170400          ;CONTROL AND STATUS REGISTER.

001616                          ADBUF1:                         ;ADDRESS OF BUFFER REG. AND
001616  170402                  ADAC:   .WORD   170402          ;DIGITAL OUTPUT REG.
001620  170440                  STREG2: .WORD   170440          ;AD11K #2 CSR
001622                          ADBUF2:
001622  170442                  ADAC2:  .WORD   170442          ;AD11K #2 OUTPUT/INPUT
                                                .SBTTL  AA11-K ADDRESSES

001624  170416                  AACSR:  .WORD   170416          ;AA11K CSR
001626  170420                   DAC0:  .WORD   170420
001630  170422                   DAC1:  .WORD   170422
001632  170424                   DAC2:  .WORD   170424
001634  170426                   DAC3:  .WORD   170426


                                                .SBTTL  DR11K #1 ADDRESS
001636  167770                  DRCR1:  .WORD   167770          ;DR11K #1 CSR.
001640  167774                  DROA1:  .WORD   167770+4                ;DR11K OUTPUT REG.
001642  167772                  DRIA1:  .WORD   167770+2                ;DR11K INPUT REG.

                                                .SBTTL  DR11K #2 ADDRESS
001644  167760                  DRCR2:  .WORD   167760          ;DR11K #2 CSR.
001646  167764                  DROA2:  .WORD   167760+4                ;DR11K OUTPUT REG.
001650  167762                  DRIA2:  .WORD   167760+2                ;DR11K INPUT REG.

                                                .SBTTL  DR11K #3 ADDRESS
001652  167750                  DRCR3:  .WORD   167750          ;DR11K #3 CSR.
001654  167754                  DROA3:  .WORD   167750+4                ;DR11K OUTPUT REG.
001656  167752                  DRIA3:  .WORD   167750+2                ;DR11K INPUT REG.

                                                .SBTTL  DR11K #4 ADDRESS
001660  167740                  DRCR4:  .WORD   167740          ;DR11K #4 CSR.
001662  167744                  DROA4:  .WORD   167740+4                ;DR11K OUTPUT REG.
001664  167742                  DRIA4:  .WORD   167740+2                ;DR11K INPUT REG.

                                                .SBTTL  DR11K #5 ADDRESS
001666  167730                  DRCR5:  .WORD   167730          ;DR11K #5 CSR.
001670  167734                  DROA5:  .WORD   167730+4                ;DR11K OUTPUT REG.
001672  167732                  DRIA5:  .WORD   167730+2                ;DR11K INPUT REG.


                                                .SBTTL  KW11K ADDRESS

001674  170404                  KWADR0: .WORD   170404          ;CLOCK A CSR

001676  170406                  KWADR1: .WORD   170406          ;CLOCK A PRESENT REG.
```

```
                                KWADR2: .WORD   170430          ;CLOCK A COUNTER REG.

                                KWADR4: .WORD   170432          ;CLOCK B CSR

                                KWADR5: .WORD   170434          ;CLOCK B PRESENT REG.

                                KWADR6: .WORD   170436          ;CLOCK B COUNT REG.

                                        .SBTTL  AR11 ADDRESSES

                                ARADS:  .WORD   170400          ;A/D STATUS REG
                                ARADB:  .WORD   170402          ;A/D BUFFER
                                ARCS:   .WORD   170404          ;CLOCK STATUS
                                ARCB:   .WORD   170406          ;CLOCK BUFFER/PRESET
                                ARDS:   .WORD   170410          ;DISPLAY STATUS
                                ARXB:   .WORD   170412          ;X BUFFER
                                ARYB:   .WORD   170414          ;Y BUFFER
                                ARCC:   .WORD   170416          ;CLOCK COUNT REG.


                                        .SBTTL  LPS-11 ADDRESSES

                                LPADSR: .WORD   170400          ;A/D STATUS REG.
                                LPADBR: .WORD   170402          ;A/D BUFFER REG.
                                LPCKSR: .WORD   170404          ;CLOCK STATUS REG.
                                LPCKBR: .WORD   170406          ;CLOCK BUFFER REG.
                                LPIOSR: .WORD   170410          ;I/O STATUS REG.
                                LPIOIR: .WORD   170412          ;I/O INPUT REG.
                                LPIOOR: .WORD   170414          ;I/O OUTPUT REG.
                                LPDASR: .WORD   170416          ;D/A STATUS REG.
                                LPDAXR: .WORD   170420          ;D/A X REG.
                                LPDAYR: .WORD   170422          ;D/A Y REG.

                                        ;MISC. STORAGE LOCATIONS

                                MYTEMP: .WORD   0               ;TEMP STORAGE
                                USJNO:  .WORD   0               ;USERS JOB NUMBER (IN R  FLASH)
                                USRDA:  .WORD   0               ;USERS RDA ADDR.
                                ALPCI:  .WORD   0               ;CONTENTS OF CONTROL IN REG.
                                ALPCO:  .WORD   0               ;CONTENTS OF CONTROL OUT REG.
                                ALPSO:  .WORD   0               ;CONTETS OF HIGH BYT OF ABOVE,(STATUS REG)
                                ALPADL: .WORD   0               ;CONTENTS OF ADDR. REG.
                        003     $VERSN: .BYTE   0,3             ;CONTAIN UCODE SYMBOLS
                                TAXING: .WORD   0
                                ERCNT:  .WORD   0               ;NO OF ERRORS.

                                                                ;TO BE KW11P ADDR. OR NO ADDR. AT ALL


                                STPCOD=73
                                LOPC=74
                                VBM=7
```

```
                    000007                          USWH=7
002000  177546                  KW11L:  .WORD   177546          ;ADDR. OF KW11L
002002  172540                  KW11P:  .WORD   172540          ;ADDR. OF KW11P.

002004  002006                  NOCLK:  .WORD   CLKTMP           ;ADDR. OF NO CLOCK.
002006  000000                  CLKTMP: .WORD   0
002010  000100                  KWVECL: .WORD   100              ;VECOTR OF KW11L
002012  000104                  KWVECP: .WORD   104              ;VECTOR OF KW11L
002014  000000                  FUDGE:  .WORD   0                ;FUDGE FACTOR


002016                          START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
002016  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
002022  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
002024  022706  001140                  CMP     #$WR,R6 ;;DONE?
002030  001374                          BNE     .-6             ;;LOOP BACK IF NO
002032  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
002036  012737  032014  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
002044  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
002052  012737  031464  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
002060  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
002066  012737  041242  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
002074  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
002102  012737  033704  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002110  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002116  005037  001160                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002122  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002126  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002134  012737  002134  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002142  012737  002142  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1"  SETUP FOR A SOFTWARE SWITCH REGISTER.
002150  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002154  012737  002210  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002162  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002170  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002176  022777  177777  176734          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002204  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
002206  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
002210  012716  002216          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
002214  000002                          RTI
002216  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
002224  012737  000174  001142          MOV     #DISPREG,DISPLAY
002232  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

002236  005037  001202                  CLR     $PASS           ;;CLEAR PASS COUNT
002242  132737  000200  001215          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
002250  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
002252  012737  001216  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
002260                          67$:
```

```
                                                
002260  013700  001250                  MOV     $BASE,R0        ;GET BASE KMC-ADDR.
002264  012701  001466                  MOV     #KMAD0,R1       ;STORAGE OF ADDR. LIST
002270  010021                  1$:     MOV     R0,(1)+         ;STORE ADDR.
002272  005200                          INC     R0              ;UPDATE ADDR.
002274  020127  001506                  CMP     R1,#KMAD7+2     ;DONE WHOLE LIST?
002300  001373                          BNE     1$
                                        
002302  005037  001514                  CLR     .DVLS
                                        
002306  013737  001244  001556          MOV     $VECT1,VECT1    ;GET VECTOR ADDR.
002314  042737  170000  001556          BIC     #170000,VECT1
002322  013737  001556  001560          MOV     VECT1,VECT2
002330  052737  000004  001560          BIS     #4,VECT2
                                        ;
                                        ;THE FOLLOWING SECTION OF CODE SETS UP DEVICE ADDRESSES
                                        ;BASED ON DEFAULT OR OPERATOR MODIFIED DATA IN LOCATIONS
                                        ;"AD11K" THROUGH "LPS11"
                                        ;
002336  013700  001566                  MOV     AD11K,R0        ;FIX AD11K'S ADDRESS
002342  010037  001614                  MOV     R0,STREG1
002346  010037  001616                  MOV     R0,ADAC
002352  062737  000002  001616          ADD     #2,ADAC
                                        
002360  013700  001570                  MOV     KW11K,R0        ;FIX KW11K ADDRESS
002364  010037  001674                  MOV     R0,KWADR0
002370  010037  001676                  MOV     R0,KWADR1
002374  010037  001700                  MOV     R0,KWADR2
002400  010037  001702                  MOV     R0,KWADR4
002404  010037  001704                  MOV     R0,KWADR5
002410  010037  001706                  MOV     R0,KWADR6
002414  062737  000002  001676          ADD     #2,KWADR1
002422  062737  000024  001700          ADD     #24,KWADR2
002430  062737  000026  001702          ADD     #26,KWADR4
002436  062737  000030  001704          ADD     #30,KWADR5
002444  062737  000032  001706          ADD     #32,KWADR6
                                        
                                        
                                        
002452  013700  001572                  MOV     DR11K1,R0       ;FIX DR11K #1 ADDRESS
002456  010037  001636                  MOV     R0,DRCR1
002462  010037  001640                  MOV     R0,DROA1
002466  010037  001642                  MOV     R0,DRIA1
002472  062737  000004  001640          ADD     #4,DROA1
002500  062737  000002  001642          ADD     #2,DRIA1
                                        
002506  013700  001600                  MOV     DR11K2,R0       ;FIX DR11K #2 ADDRESS
002512  010037  001644                  MOV     R0,DRCR2
002516  010037  001646                  MOV     R0,DROA2
002522  010037  001650                  MOV     R0,DRIA2
002526  062737  000004  001646          ADD     #4,DROA2
002534  062737  000002  001650          ADD     #2,DRIA2
                                        
002542  013700  001602                  MOV     DR11K3,R0       ;FIX DR11K #3 ADDRESS
```

```
  801  002546  010037  001652                MOV     R0,DRCR3
  802  002552  010037  001654                MOV     R0,DROA3
  803  002556  010037  001656                MOV     R0,DRIA3
  804  002562  062737  000004  001654        ADD     #4,DROA3
  805  002570  062737  000002  001656        ADD     #2,DRIA3
  806
  807  002576  013700  001604                MOV     DR11K4,R0       ;FIX DR11K #4 ADDRESS
  808  002602  010037  001660                MOV     R0,DRCR4
  809  002606  010037  001662                MOV     R0,DROA4
  810  002612  010037  001664                MOV     R0,DRIA4
  811  002616  062737  000004  001662        ADD     #4,DROA4
  812  002624  062737  000002  001664        ADD     #2,DRIA4
  813
  814  002632  013700  001606                MOV     DR11K5,R0       ;FIX DR11K #5 ADDRESS
  815  002636  010037  001666                MOV     R0,DRCR5
  816  002642  010037  001670                MOV     R0,DROA5
  817  002646  010037  001672                MOV     R0,DRIA5
  818  002652  062737  000004  001670        ADD     #4,DROA5
  819  002660  062737  000002  001672        ADD     #2,DRIA5
  820
  821  002666  013700  001574                MOV     AA11K,R0        ;FIX AA11K ADDRESS
  822  002672  010037  001624                MOV     R0,AACSR
  823  002676  010037  001626                MOV     R0,DAC0
  824  002702  010037  001630                MOV     R0,DAC1
  825  002706  010037  001632                MOV     R0,DAC2
  826  002712  010037  001634                MOV     R0,DAC3
  827  002716  062737  000002  001626        ADD     #2,DAC0
  828  002724  062737  000004  001630        ADD     #4,DAC1
  829  002732  062737  000006  001632        ADD     #6,DAC2
  830  002740  062737  000010  001634        ADD     #10,DAC3
  831
  832  002746  013700  001576                MOV     AD11K2,R0       ;FIX SECOND AD11K'S ADDR.
  833  002752  010037  001620                MOV     R0,STREG2
  834  002756  010037  001622                MOV     R0,ADAC2
  835  002762  062737  000002  001622        ADD     #2,ADAC2
  836
  837  002770  013700  001610                MOV     AR11K,R0        ;FIX AR11 ADDR.
  838  002774  010037  001710                MOV     R0,ARADS
  839  003000  010037  001712                MOV     R0,ARADB
  840  003004  010037  001714                MOV     R0,ARCS
  841  003010  010037  001716                MOV     R0,ARCB
  842  003014  010037  001720                MOV     R0,ARDS
  843  003020  010037  001722                MOV     R0,ARXB
  844  003024  010037  001724                MOV     R0,ARYB
  845  003030  010037  001726                MOV     R0,ARCC
  846  003034  062737  000002  001712        ADD     #2,ARADB
  847  003042  062737  000004  001714        ADD     #4,ARCS
  848  003050  062737  000006  001716        ADD     #6,ARCB
  849  003056  062737  000010  001720        ADD     #10,ARDS
  850  003064  062737  000012  001722        ADD     #12,ARXB
  851  003072  062737  000014  001724        ADD     #14,ARYB
  852  003100  062737  000016  001726        ADD     #16,ARCC
  853
  854  003106  013700  001612                MOV     LPS11,R0        ;FIX LPS-11 ADDRESSES
```

```
                                                MOV     R0,LPADSR
                                                MOV     R0,LPADBR
                                                MOV     R0,LPCKSR
                                                MOV     R0,LPCKBR
                                                MOV     R0,LPIOSR
                                                MOV     R0,LPIOIR
                                                MOV     R0,LPIOOR
                                                MOV     R0,LPDASR
                                                MOV     R0,LPDAXR
                                                MOV     R0,LPDAYR
                                                ADD     #2,LPADBR
                                                ADD     #4,LPCKSR
                                                ADD     #6,LPCKBR
                                                ADD     #10,LPIOSR
                                                ADD     #12,LPIOIR
                                                ADD     #14,LPIOOR
                                                ADD     #16,LPDASR
                                                ADD     #20,LPDAXR
                                                ADD     #22,LPDAYR
                                                CLR     FUDGE

                                                MOV     #10$,ERRVEC     ;SET UP FOR TIME-OUT IF NO KW11L
                                                TST     @KW11L          ;TRAP HERE IF NO L
                                                MOV     KW11L,RTCCSR    ;MUST BE HERE,RECORD ADDR.
                                                BR      50$
                                        10$:    MOV     #20$,ERRVEC     ;SET UP FOR TIME-OUT IF NO KW11P
                                                TST     @KW11P          ;TRAP HERE IF NO P
                                                MOV     KW11P,RTCCSR    ;MUST BE HERE,RECORD ADDR.
                                                BR      50$
                                        20$:    MOV     NOCLK,RTCCSR    ;NO CLOCK! OHOH.
                                        50$:    MOV     #STACK,SP       ;RESTORE STACK
                                                MOV     #6,ERRVEC       ;RESTORE ERROR VECTOR.

                                                MOV     #CLKINT,@KWVECL ;SET UP CLOCK VECTOR.
                                                MOV     #CLKINT,@KWVECP
                                .SBTTL  TYPE PROGRAM NAME
                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
                                                INC     #-1             ;;FIRST TIME?
                                                BNE     68$             ;;BRANCH IF NO
                                                TYPE    ,69$            ;;TYPE ASCIZ STRING
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
                                                TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
                                                BNE     70$             ;;BRANCH IF YES
                                                CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
                                                BEQ     70$             ;;BRANCH IF YES
                                                CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
                                                BNE     71$             ;;BRANCH IF NO
                                                GTSWR                   ;;GET SOFT-SWR SETTINGS
                                                BR      71$
                                70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
                                71$:
                                                BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <CRLF>#MD-11-DRLPA-A    LPA-11 DIAGNOSTIC #<CRLF>
```

```
                                68$:
003474                          RSTART: CLR     ERCNT
003474  005037  001776
003500                                  CLR     .DVLS
003500  005037  001514
                                ;;*****************************************************************
                                ;*TEST 1         *TEST THE ADDRESSABLITY OF THE KMC-11
                                ;;*****************************************************************
003504  000240                  TST1:   NOP
003506  012737  000050  001160          MOV     #50,$TIMES      ;;DO 50 ITERATIONS
003514  012737  003552  001106          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS

003522  012737  000001  001102          MOV     #1,$TSTNM
003530  012737  000001  001200          MOV     #1,$TESTN
003536  012737  003552  001106          MOV     #1$,$LPADR
003544  012737  003552  001110          MOV     #1$,$LPERR

003552  013746  000004          1$:     MOV     ERRVEC, -(SP)   ;SAVE CONTENTS OF LOC 4.
003556  012737  003626  000004          MOV     #2$, ERRVEC     ;SET TIME-OUT TO ERROR REPORTER
003564  013746  000006                  MOV     ERRVEC+2,-(SP)
003570  012737  000340  000006          MOV     #340,ERRVEC+2

003576  042777  000000  175662          BIC     #0,@KMADO       ;CAUSE A DATA IN, DATA OUT

003604  005077  175656                  CLR     @KMADO          ;CLEAR KMC
003610  005077  175656                  CLR     @KMAD2
003614  005077  175656                  CLR     @KMAD4
003620  005077  175656                  CLR     @KMAD6

003624  000403                          BR      3$
                                                                ;OPERATION.

003626  104001                  2$:     ERROR   1               ;TIME-OUT TRAP TO LOC. 4 WHEN
                                                                ;KM-11 WAS ADDRESSED.
                                                                ;PLEASE DO NOT USE LOOP ERROR OPTION
                                                                ;BUT "LOOP ON TEST" (SWR14) IF YOU LOOP
                                                                ;ON THIS FAILURE.

003630  062706  000004                  ADD     #4,SP           ;TAKE CARE OF TRAP SP.

003634  012637  000006          3$:     MOV     (SP)+,ERRVEC+2  ;RESTORE ERROR VECTOR.
003640  012637  000004                  MOV     (SP)+,ERRVEC


                                ;;*****************************************************************
                                ;*TEST 2         *TEST KMC-11 FUNCTIONS

                                ;*
                                ;*IN THIS TEST WE ATTEMPT TO RUN AN INSTRUCTION TEST ON THE KMC-11.
                                ;*YOU WILL NOT BE ABLE TO LOOP ON AN INDIVIDUAL SUBTEST, BUT MUST LOOP ON
                                ;*THIS WHOLE TEST SINCE ONCE THE MICRO-CODE IS STARTED, IT EXECUTES IN-LINE
                                ;*WITH NO-LOOP BACK CAPABILITIES.
                                ;*IF THIS TEST FAILS YOU SHOULD RUN THE REGULAR KMC-11 DIAGNOSTICS
```

```
                                        ;*AVAILABLE FOR THE KMC-11
                                        ;*FOR EACH TEST, WE WILL LIST THE MICRO CODE THAT THE KMC EXECUTES
                                        ;;*****************************************************************
003644  000004                  TST2:   SCOPE
003646  012737  000050  001160          MOV     #50,$TIMES      ;;DO 50 ITERATIONS
                                        .SBTTL  KMC-11 MICRO-INITIALIZATION.
003654  004537  037534                  JSR     R5,$LOAD        ;LOAD KMC-DIAG. MICRO-CODE
                                        .GLOBL  DRLPX1
003660  000000G                         .WORD   DRLPX1
003662  012777  040000  175576          MOV     #BIT14,@KMADD   ;INITAILIZE LPA
003670  005000                          CLR     R0
003672  005200                  10$:    INC     R0              ;SHORT DELAY.
003674  001376                          BNE     10$

003676  112777  000210  175564          MOVB    #BIT7!BIT3,@KMAD1       ;SET RUN BIT ON KMC.

                                        .SBTTL  KMC-DIAG. SUB-TEST#1 CRAM WRITE ONES TEST
                                        .REM    %
                                        ;TEST #1
                                        ;               CRAM ONES WRITE TEST
                                        ;       ENTERED ON INIT.

                                TST1:   MOVE    # 377,BREG      ;GET ALL ONES
                                        MOVE    BREG,OUT1 <0>   ;PUT ALL ONES INTO CRAM
                                        MOVE    BREG,OUT1 <2>
                                        MOVE    BREG,OUT1 <3>
                                        MOVE    BREG,OUT1 <4>
                                        MOVE    BREG,OUT1 <5>
                                        MOVE    BREG,OUT1 <6>
                                        MOVE    BREG,OUT1 <7>

                                  A1:   MOVE    INP1 <0>,BREG   ;WHEN PDP-11 DONE VERIFICATION,
                                        BZ      A1              ;IT WILL CLEAR CRAM ADDR#0

                                        %

003704  005037  001126                  CLR     $BDDAT
003710  012737  000377  001124          MOV     #377,$GDDAT     ;EXPECT 377 IN ADDR. #0
003716  117737  175544  001126          MOVB    @KMADD, $BDDAT  ;READ ADDR 0
003724  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;GOOD READ?
003732  001401                          BEQ     11$             ;YES-NEXT SUB-TEST.

003734  104002                          ERROR   2               ;CRAM WRITE ONES TEST

003736  012737  177777  001124  11$:    MOV     #177777,$GDDAT  ;EXPECT 377,377,FROM ADDRS,2,3,4,5,6,AND 7
003744  017737  175522  001126          MOV     @KMAD2,$BDDAT   ;READ ADDRS 2 AND 3
003752  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;READ ALL ONES?
003760  001401                          BEQ     12$             ;YES-CHECK NEXT REG.

003762  104002                          ERROR   2               ;CRAM WRITE ONES TEST, ADDR. 2 OR 3

003764  017737  175506  001126  12$:    MOV     @KMAD4,$BDDAT   ;READ ADDRS. 4 AND 5.
003772  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;READ ALL ONES?
004000  001401                          BEQ     13$             ;YES-NEXT REGS.
```

```
1017
1018  004002  104002                          ERROR   2               ;CRAM WRITE ONES TEST, ADDR. 4 OR 5
1019
1020  004004  017737  175472  001126  13$:    MOV     @KMAD6,$BDDAT   ;READ ADDRS. 6 AND 7.
1021  004012  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;READ ALL ONES?
1022  004020  001401                          BEQ     14$             ;YES-NEXT TEST
1023
1024  004022  104002                          ERROR   2               ;CRAM WRITE ONES TEST ADDR 6 OR 7.
1025
1026  004024  105077  175436          14$:    CLRB    @KMADO          ;TELL THE MICRO-PROGRAM TO
1027                                                                  ;DO TEST #2.
1028
1029
1030                                          .SBTTL  KMC-DIAG SUB-TEST #2 CRAM WRITE ZEROES TEST
1031
1032                                          .REM    %
1033                                                  ;TEST #2
1034                                                  ;       CRAM ZEROES WRITE TEST
1035                                                  ;
1036                                                  ;       ENTERED WHEN PDP-11 ZEROES CRAM ADDR 0
1037                                                  ;
1038                                  TST2:   MOVE    # 0,BREG         ;GET ZERO.
1039                                          MOVE    BREG,OUT1 <0>   ;PUT INTO ALL CRAM.
1040                                          MOVE    BREG,OUT1 <2>
1041                                          MOVE    BREG,OUT1 <3>
1042                                          MOVE    BREG,OUT1 <4>
1043                                          MOVE    BREG,OUT1 <5>
1044                                          MOVE    BREG,OUT1 <6>
1045                                          MOVE    BREG,OUT1 <7>
1046
1047                                    A2:   MOVE    INP1 <0>,BREG   ;GET ZERO
1048                                          BZ      TST3            ;WHEN =377,PDP-11 READY FOR NEXT TEST.
1049                                          BR      A2
1050
1051                                          %
1052
1053  004030  000240                          NOP
1054  004032  000240                          NOP
1055  004034  005037  001124                  CLR     $GDDAT          ;EXPECT ZERO REGISTERS.
1056  004040  117737  175422  001126          MOVB    @KMADO,$BDDAT   ;READ ADR.0
1057  004046  001404                          BEQ     21$             ;IF ZERO, GOOD.
1058  004050  042737  177400  001126          BIC     #177400,$BDDAT
1059
1060  004056  104002                          ERROR   2               ;CRAM WRITE ZERO TEST, ADDR.0
1061
1062  004060  017737  175406  001126  21$:    MOV     @KMAD2,$BDDAT   ;READ ADDRS. 2,3
1063  004066  001401                          BEQ     22$             ;GOOD IF=0
1064
1065  004070  104002                          ERROR   2               ;CRAM WRITE ZERO TEST, ADDR 2 OR 3
1066
1067  004072  017737  175400  001126  22$:    MOV     @KMAD4,$BDDAT   ;READ ADDRS 4,5
1068  004100  001401                          BEQ     23$             ;GOOD IF=0
1069
1070  004102  104002                          ERROR   2               ;CRAM WRITE ZERO TEST, ADDR. 4 OR 5
```

```

004104  017737  175372  001126    23$:    MOV     @KMAD6,$BDDAT       ;READ ADDRS. 6,7
004112  001401                            BEQ     24$                 ;GOOD IF=0

004114  104002                            ERROR   2                   ;CRAM WRITE ZERO TEST ADDR. 6,7.

004116  112777  000377  175342    24$:    MOVB    #377, @KMAD0        ;TELL MICRO-CODE WE'RE READY FOR NEXT TEST.

                                          .SBTTL  KMC-DIAG SUB TEST #3 BRANCH TEST
                                          .REM    %
                                                  ;
                                                  ;TEST #3
                                                  ;                          BRANCH TEST
                                                  ;
                                                  ;                 ENTERED WHEN PDP-11 RETURNS 377 TO ADDR.0
                                                  ;
                                                  ;                 NOTE WE HAVE TO ASSUME BR AND BZ WORK
                                                  ;                 OR WE COULDN'T HAVE GOT THIS FAR.
                                                  ;                 UPON SUCCESSFULL COMPLETION OF THIS TEST WE
                                                  ;                 PUT CODE 377 INTO CRAM ADDR 2
                                                  ;                 AND 0 INTO CRAM ADDR.0. ON FAILURE
                                                  ;                 WE ONLY ZERO CRAM ADDR.0.
                                                  ;
                                  TST3:   MOVE    # 376,BREG          ;GET ZERO BIT0
                                          BB0     T3E                 ;ERROR IF BR
                                          MOVE    # 375,BREG          ;BIT 1=0
                                          BB1     T3E                 ;ERROR IF BR
                                          MOVE    # 357,BREG          ;BIT 4=0
                                          BB4     T3E                 ;ERROR IF BR
                                          MOVE    # 177,BREG          ;BIT 7=0
                                          BB7     T3E                 ;ERROR IF BR

                                          MOVE    # 1,BREG            ;POSITIVE BR TEST
                                          BB0     A3                  ;SHOULD BRANCH
                                          BR      T3E                 ;IF NOT, ERROR.
                                  A3:     MOVE    # 2,BREG            ;BR OR BIT 1
                                          BB1     B3                  ;SHOULD BR.
                                          BR      T3E                 ;IF NOT ERROR.
                                  B3:     MOVE    # 20,BREG           ;BR ON BIT 4
                                          BB4     C3                  ;SHOULD BR.
                                          BR      T3E                 ;IF NOT ERROR.
                                  C3:     MOVE    # 200,BREG          ;BR OF BIT 7
                                          BB7     D3                  ;SHOULD BR.
                                          BR      T3E                 ;IF NOT ERROR.
                                  D3:     MOVE    # 0,BREG            ;TEST BZ NEGATIVE
                                          BZ      T3E                 ;ERROR IF BR.
                                          MOVE    # 377,BREG          ;POS BR
                                          BZ      E3                  ;SHOULD BR.
                                          BR      T3E                 ;ERROR IF NOT.

                                  E3:     MOVE    BREG,OUT1 <2>       ;PUT 377 IN OUTPUT REG #2
                                          BR      F3                  ;IF BR INSTR FAILS TO WORK
                                          MOVE    # 0,BREG            ;THEN A ZERO GETS PUT IN #2
                                          MOVE    BREG,OUT1 <2>       ;A SIGN OF AN ERROR
```

```
                                        F3:
                                T3E:    MOVE    # 0,BREG          ;SIGNAL PDP-11 THAT WE ARE
                                        MOVE    BREG,OUT1 <0>    ;THROUGH BR-TEST.

                                        %
004124  005000                          CLR     R0               ;TIME OUT COUNTER.
004126  105777  175334        30$:     TSTB    @KMAD0           ;MICRO CODE DONE?
004132  001404                          BEQ     31$              ;YES-EXIT
004134  005200                          INC     R0               ;NO-CHECK FOR TIME OUT.
004136  001373                          BNE     30$              ;NO-TIMEOUT, LOOP.

004140  104003                          ERROR   3                ;KMC-11 FAILED TO FINISH BRANCH TEST.
004142  000550                          BR      TST3             ;;

004144  105777  175322        31$:     TSTB    @KMAD2           ;DID ALL BRANCHES WORK?,CODE 377 IN ADDR 2
004150  001002                          BNE     40$

004152  104003                          ERROR   3                ;KMC-11 FAILED BRANCH TEST.
004154  000543                          BR      TST3             ;;

                                        .SBTTL  KMC-DIAG SUB-TEST #4 ALU TEST
                                        .REM    %
                                        ;TEST #4
                                        ;
                                        ;       ALU TEST
                                        ;
                                        ;       ENTERED WHEN PDP-11 PUTS A CODE 377 INTO
                                        ;       CRAM ADDR 0
                                        ;       ADDR 2=0 IF BAD
                                        ;
                                TST4:   MOVE    INP1 <0>,BREG    ;GET CRAM ADDR0
                                        BZ      A4               ;WHEN =377 DO TEST
                                        BR      TST4

                                  A4:   MOV     # 0,BREG         ;GET=0
                                        MOV     BREG,OUT1 <2>
                                        MOVE    BREG,SPAD <4>
                                        DEC     SPAD <4>         ;MAKE = 377
                                        BZ      B4               ;BR IF GOOD SEC.
                                        BR      T4E
                                  B4:   DEC     SPAD <4>         ;SEC = 376
                                        INC     SPAD <4>         ;INC = 377
                                        BZ      C4               ;BR IF = 377
                                        BR      T4E
                                  C4:   MOVE    # 377,BREG
                                        MOVE    BREG,OUT1 <2>    ;RETURN GOOD CODE

                                T4E:    MOVE    # 0,.BREG       ;RETURN CODE 0
                                        MOVE    BREG,OUT1 <0>
                                        %

004156  112777  000377  175302  40$:    MOVB    #377,@KMAD0      ;START TEST.
004164  005000                          CLR     R0               ;TIME OUT COUNTER
```

```
1179  004166  105777  175274          41$:  TSTB@KMADO              ;DONE?
1180  004172  001404                        BEQ     42$             ;YES-CHECK RESULTS
1181  004174  005200                        INC     R0              ;NO-TIME OUT?
1182  004176  001373                        BNE     41$             ;NO-LOOP.
1183
1184  004200  104003                        ERROR   3               ;KMC-11 FAILED TO FINISH ALU TEST.
1185  004202  000530                        BR      TST3            ;;
1186
1187  004204  122777  000377  175260  42$:  CMPB    #377,@KMAD2     ;DID IT DO IT RIGHT?
1188  004212  001401                        BEQ     50$
1189
1190  004214  104003                        ERROR   3               ;KMC-11 ALU TEST FAILED.
1191
1192                                        .SBTTL  KMC-DIAG SUB-TEST #5 NPR TEST ZERO TRANSFER
1193
1194                                        .REM    %
1195
1196                                        ;
1197                                        ;TEST 5
1198                                        ;
1199                                        ;       NPR TEST
1200                                        ;
1201                                        ;       ENTERED IN LINE WHEN PDP-11 PUTS CODE 377 INTO
1202                                        ;       CRAM ADDR.0
1203                                        ;       DOES AN INPUT DATA XFERR FROM LOC 1124
1204                                        ;       IN PDP-11 MEM TO LOC 1126
1205                                        ;       DOES NO DATA CHECKING RETURNS
1206                                        ;       CODE 0 TO CRAM ADDR.0 WHEN DONE.
1207                                        ;
1208
1209                                TST5:   MOVE    INP1 <0>,BREG   ;WAIT FOR PDP-11
1210                                        BZ      A5
1211                                        BR      TST5
1212
1213                                A5:     MOVE    # 2,BREG        ;SET TO DO NPR IN
1214                                        MOVE    BREG,OUT0 <5>   ;SET HIGH ADDR.
1215                                        MOVE    # 124,BREG      ;GET LOW ADDR OF ADDR. 1124
1216                                        MOVE    BREG,OUT0 <4>   ;SET LOW ADDR.
1217                                        MOVE    # 1,BREG        ;SET TO NPR IN
1218                                        MOVE    BREG,OUT1 <10>  ;CAUSE NPR IN
1219                                B5:     MOVE    INP1 <10>,BREG  ;WAIT FOR NPR DONE.
1220                                        BB0     B5
1221
1222                                        MOVE    INP0 <0>,BREG   ;GET LOW BYTE DATA.
1223                                        MOVE    BREG,OUT0 <2>   ;OUTPUT NPR DATA REG.
1224                                        MOVE    INP0 <1>,BREG   ;GET HIGH BYTE.
1225                                        MOVE    BREG,OUT0 <3>   ;SAVE FOR OUTPUT.
1226                                        MOVE    # 2,BREG        ;GET OUTPUT ADDR.=1126
1227                                        MOVE    BREG,OUT0 <7>
1228                                        MOVE    # 126,BREG      ;LOW PART.
1229                                        MOVE    BREG,OUT0 <6>
1230                                        MOVE    INP1 <11>,SPAD <4>      ;/-STN-
1231                                        AND     SPAD <4>,BREG,SPAD
1232                                        MOVE    # 0,BREG
```

```
1233                                            OR      SPAD <4>,BREG,BREG
1234                                            MOVE    BREG,OUT1 <11>
1235                                            MOVE    # 21,BREG
1236                                            MOVE    BREG,OUT1 <10>  ;SET NPR OUT.
1237
1238                                    C5;     MOVE    INP1 <10>,BREG  ;WAIT TILL DONE
1239                                            BBO     C5
1240
1241                                            MOVE    # 0,BREG        ;TELL PDP-11 WE'RE DONe.
1242                                            MOVE    BREG,OUT1 <0>
1243
1244                                             %
1245
1246  004216  005037  001124                    50$:    CLR     $GDDAT          ;CLEAR $GDDAT (LOC 1124)
1247  004222  012737  177777  001126                    MOV     #-1,$BDDAT
1248  004230  112777  000377  175230                    MOVB    #377,@KMADO     ;TELL MICRO-CODE TO GO!
1249  004236  005000                                    CLR     R0              ;CLEAR TIME OUT COUNTER.
1250
1251  004240  105777  175222                    51$:    TSTB    @KMADO          ;KMC-MICROCODE DONe?
1252  004244  001403                                    BEQ     52$
1253  004246  005200                                    INC     R0              ;NO-TIME OUT OCCUR?
1254  004250  001373                                    BNE     51$
1255
1256  004252  104003                                    ERROR   3               ;KMC-11 FAILED TO FINISH NPR TEST.
1257
1258  004254  023737  001124  001126    52$:    CMP     $GDDAT,$BDDAT   ;NPR ALL ZEROS CORRECTLY?
1259  004262  001401                                    BEQ     60$             ;YES-NEXT TEST.
1260
1261  004264  104002                                    ERROR   2               ;KMC-11 NPR TEST ZERO TRANSFER
1262
1263                                            .SBTTL  KMC-DIAG SUB-TEST #6 NPR TEST ONES TRANSFER.
1264                                            .REM     %
1265                                            ;
1266                                            ;TEST 6
1267                                            ;
1268                                            ;       NPR TEST
1269                                            ;
1270                                            ;       ENTERED INLINE WHEN PDP-11 PUTS CODE 377 INTO
1271                                            ;       CRAM ADDR.0
1272                                            ;       DOES AN INPUT DATA XFERR FROM LOC 1124
1273                                            ;       IN PDP-11 MEM TO LOC 1126
1274                                            ;       DOES NO DATA CHECKING RETURNS
1275                                            ;       CODE 0 TO CRAM ADDR.0 WHEN DONe.
1276                                            ;
1277
1278                                    TST6:   MOVE    INP1 <0>,BREG   ;WAIT FOR PDP-11
1279                                            BZ      A6
1280                                            BR      TST6
1281
1282                                      A6:   MOVE    # 2,BREG         ;SET TO DO NPR IN
1283                                            MOVE    BREG,OUT0 <5>   ;SET HIGH ADDR.
1284                                            MOVE    # 124,BREG      ;GET LOW ADDR OF ADDR. 1124
1285                                            MOVE    BREG,OUT0 <4>   ;SET LOW ADDR.
1286                                            MOVE    # 1,BREG        ;SET TO NPR IN
```

```
1287                                        MOVE    BREG,OUT1 <10>  ;CAUSE NPR IN
1288                                B6:     MOVE    INP1 <10>,BREG  ;WAIT FOR NPR DONE.
1289                                        BBO     B6
1290
1291                                        MOVE    INPO <0>,BREG   ;GET LOW BYTE DATA.
1292                                        MOVE    BREG,OUTO <2>   ;OUTPUT NPR DATA REG.
1293                                        MOVE    INPO <1>,BREG   ;GET HIGH BYTE.
1294                                        MOVE    BREG,OUTO <3>   ;SAVE FOR OUTPUT.
1295                                        MOVE    # 2,BREG        ;GET OUTPUT ADDR.=1126
1296                                        MOVE    BREG,OUTO <7>
1297                                        MOVE    # 126,BREG      ;LOW PART.
1298                                        MOVE    BREG,OUTO <6>
1299                                        MOVE    INP1 <11>,SPAD <4>      ;/-STN-
1300                                        AND     SPAD <4>,BREG,SPAD
1301                                        MOVE    # 0,BREG
1302                                        OR      SPAD <4>,BREG,BREG
1303                                        MOVE    BREG,OUT1 <11>
1304                                        MOVE    # 21,BREG
1305                                        MOVE    BREG,OUT1 <10>  ;SET NPR OUT.
1306
1307                                C6:     MOVE    INP1 <10>,BREG  ;WAIT TILL DONE
1308                                        BBO     C6
1309
1310                                        MOVE    # 0,BREG        ;TELL PDP-11 WE'RE DONe
1311                                        MOVE    BREG,OUT1 <0>
1312
1313                                        %
1314
1315  004266  005037  001126        60$:    CLR     $BDDAT          ;NOW XFERR $GDDAT TO $BDDAT
1316  004272  012737  177777  001124       MOV     #177777,$GDDAT
1317
1318  004300  112777  000377  175160       MOVB    #377,@KMADO     ;START NPR XFERR
1319
1320  004306  005000                       CLR     R0              ;TIME OUT COUNTER
1321
1322  004310  105777  175152        61$:    TSTB    @KMADO          ;DONE NPR?
1323  004314  001404                       BEQ     62$
1324  004316  005200                       INC     R0              ;NO-TIME OUT?
1325  004320  001373                       BNE     61$
1326
1327  004322  104003                       ERROR   3               ;KMC-11 NPR TESt FAILED TO FINISH.
1328  004324  000457                       BR      TST3            ;;
1329
1330  004326  023737  001124  001126  62$:    CMP     $GDDAT,$BDDAT   ;ONES XFERR OK?
1331  004334  001401                       BEQ     70$             ;YES-NEXT TEST.
1332
1333  004336  104002                       ERROR   2               ;KMC-11 NPR ONES XFERR FAILED.
1334
1335                                        .SBTTL  KMC-DIAG SUB-TEST #7 INTERRRUPT TEST.
1336                                        .REM    %
1337
1338                                        ;TEST #7
1339                                        ;
1340                                        ;       INTERRUPT TEST
```

```
1341                                            ;
1342                                            ;       THIS TEST GENERATES TWO INTERRUPTS
1343                                            ;
1344                                            ;       THE 1ST INTERRUPT TO VECTOR XX0 WHEN
1345                                            ;       CRAM ADDR.0 =377 BY PDP-11 CONTROL.
1346                                            ;       THE 2ND INTERRUPT TO vECTOR XX4 WHEN
1347                                            ;       CRAM ADDR0 =377 (1ST MODE=0 AGAIN).
1348                                            ;
1349
1350                                    TST7:   MOVE    INP1 <0>,BREG   ;WAIT TILL PDP-11 READY!
1351                                            BZ      A7
1352                                            BR      TST7
1353
1354                                     A7:    MOVE    INP1 <11>,SPAD <4>
1355                                            AND     SPAD <4>,BREG,SPAD
1356                                            MOVE    # 200,BREG
1357                                            OR      SPAD <4>,BREG,BREG
1358                                            MOVE    BREG,OUT1 <11>  ;SET INTR TO ADDR. XX0
1359                                     B7:    MOVE    INP1 <11>,BREG  ;WAIT TILL DONe
1360                                            BB7     B7
1361
1362                                            MOVE    # 0,BREG        ;TELL PDP-11 WE INTERRUPTED
1363                                            MOVE    BREG,OUT1 <0>   ;IN CASE WE DIDN'T.
1364
1365                                     C7:    MOVE    INP1 <0>,BREG   ;NOW WAIT FOR PDP-11 TO TELL
1366                                            BZ      D7              ;US TO INTR. TO VECTOR XX4
1367                                            BR      C7
1368
1369                                     D7:    MOVE    INP1 <11>,SPAD <4>
1370                                            AND     SPAD <4>,BREG,SPAD
1371                                            MOVE    # 300,BREG
1372                                            OR      SPAD <4>,BREG,BREG
1373                                            MOVE    BREG,OUT1 <11>  ;SET INTR TO ADDR. XX4
1374
1375                                     E7:    MOVE    INP1 <11>,BREG  ;WAIT TILL DONE.
1376                                            BB7     E7
1377
1378                                            MOVE    # 0,BREG        ;TELL PDP-11 WE THOUGHT
1379                                            MOVE    BREG,OUT1 <0>   ;WE HAD INTERRUPTED!
1380
1381                                            BR      .
1382                                             %
1383
1384  004340  012777  004372  175210    70$:   MOV     #72$,@VECT1     ;SET-UP FOR 1ST INTERRUPT.
1385  004346  005037  177776                   CLR     PS              ;LOWER PROCESSOR STATUS.
1386  004352  112777  000377  175106           MOVB    #377,@KMADO     ;TELL MICRO-CODE WE'RE READY!
1387  004360  005000                           CLR     R0
1388  004362  105200                    71$:   INCB    R0              ;TIME-OUT COUNT IN CASE OF NO INTERRUPT.
1389  004364  100376                           BPL     71$
1390
1391  004366  104003                           ERROR   3               ;KMC-FAILED TO iNTERRRUPT AT XX0
1392  004370  000435                           BR      TST3            ;;
1393
1394  004372  062706  000004            72$:   ADD     #4,SP           ;RESTORE STACK.
```

```
1395  004376  012777  004444  175154          MOV     #74$,@VECT2     ;SET NEW INTERRUPT VECTOR
1396  004404  013777  001556  175144          MOV     VECT1,@VECT1    ;RESTORE OLD VECTOR.
1397  004412  062777  000002  175136          ADD     #2,@VECT1
1398  004420  005037  177776                  CLR     PS              ;ALLOW INTERRUPTS
1399  004424  112777  000377  175034          MOVB    #377,@KMADO     ;TELL MICRO CODE WE'RE READY AGAIN.
1400  004432  005000                          CLR     R0
1401
1402  004434  105200                  73$:    INCB    R0              ;TIME OUT COUNTER IN CASE OF NO INTR
1403  004436  100376                          BPL     73$
1404
1405  004440  104003                          ERROR   3               ;KMC-FAILED TO INTERRUPT AT 004
1406  004442  000410                          BR      TST3            ;;
1407
1408  004444  062706  000004          74$:    ADD     #4,SP           ;RESTORE STACK
1409  004450  013777  001560  175102          MOV     VECT2,@VECT2    ;RESTORE VECTOR.
```

```
 1410  004456  062777  000002  175074          ADD     #2,@VECT2
 1411
```

```
1412
1413                                    ;;*****************************************************************
1414                                    ;*TEST 3        *TEST SLAVE MICRO PROCESSOR START
1415
1416                                    ;*
1417                                    ;*IN THIS TEST WE'LL LOAD IN MICROCODE INTO THE KMC-11
1418                                    ;*THAT WILL ALLOW US TO tALK TO THE SLAVE MICRO-PROCESSOR.
1419
1420                                    ;*
1421
1422                                    ;*WHEN THE SLAVE MICRO-PROCESSOR IS INTITIALIZED (IT GETS INITED
1423                                    ;*WHEN BIT 14 OF THE KMC'S CSR GET SET) IT SENDS US A VERSION
1424                                    ;*NUMBER THROUGH THE IPBM'S FAST PATH REGISTER IF THE
1425                                    ;*INITIAL CSR OF IPBM (AS SEEN BY THE SLAVE) IS CORRECT.
1426                                    ;*IF BAD, IT SEND CODE 377 THROUGH BOTH FAST PATH AND
1427                                    ;*SILO TO INDICATE AN ERROR.
1428
1429                                    ;;*****************************************************************
1430  004464  000004                    tST3:   SCOPE
1431  004466  012737  000010  001160            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1432
1433  004474  004537  037534                    JSR     R5,$LOAD                ;LOAD MICRO CODE INTO KMC-11
1434                                            .GLOBL  DRLPX0
1435  004500  000000G                           .WORD   DRLPX0          ;FILE "DRLPX0"
1436
1437  004502  052777  040000  174756            BIS     #BIT14,@KMADO   ;SET INIT BIT, WHEN THIS BIT
```

```
 1438
 1439  004510  005000                    CLR     R0          ;CLEARS IN SLAVE, RUN STARTS.
```

```
1440  004512  005200                    1$:   INC     R0              ;LITTLE DELAY LOOP.
1441  004514  001376                          BNE     1$
1442
1443  004516  012777  104000  174742          MOV     #BIT15!BIT11,@KMADO     ;CLEAR INTIT, SET RUN
1444  004524  005000                          CLR     R0
```

```
  1445  004526  005200                    2$:   INC     R0              ;LITTLE DELAY.
```

```
1446  004530  001376                          BNE     2$
1447
1448  004532  032777  000040  174726          BIT     #BIT5,@KMADO    ;DID SLAVE SEND DATA THROUGH F.P.?
1449  004540  001410                          BEQ     4$              ;BR IF YES (LOOKING AT IPBM CSR BIT 5)
1450                                                                  ;SENDS EITHER VERSION OR CODE 377.
1451
1452  004542  032777  000002  174716          BIT     #BIT1,@KMADO    ;RIGHT NOW WE KNOW AN ERROR
1453  004550  001002                          BNE     3$              ;HAS OCCURRED. IF THERE IS DATA
1454                                                                  ;IN THE SILO, MASTER'S IPBM
1455                                                                  ;CSR BIT 1=0
1456
1457                                                                  ;IPBM INIT ERROR
1458  004552  104004                          ERROR   4               ;ERROR (IPBM) M8254 FAILED TO
1459                                                                  ;INITIALIZE PROPERLY. AS SEEN FROM
1460  004554  000427                          BR      TST4            ;;
1461
1462
1463  004556  104004                  3$:     ERROR   4               ;ERROR IPBM OR (MOST LIKELY)
1464                                                                  ;SLAVE MICRO PROCESSOR FAILED
1465                                                                  ;TO START PROPERLY.
1466  004560  000425                          BR      TST4            ;;
1467
1468  004562  112777  000004  174702  4$:     MOVB    #4,@KMAD2       ;ISSUE CMMD TO READ FAST PATH
1469
1470  004570  122777  000377  174674  5$:     CMPB    #377,@KMAD2     ;WAIT FOR READ.
1471  004576  001374                          BNE     5$
```

```
1472  004600  013737  001512  001124          MOV     VERSN,$GDDAT    ;GET MICRO-CODE VERSION NUMBER.
1473  004606  117737  174664  001126          MOVB    @KMAD4,$BDDAT   ;IF=MINUS, ERROR CODE RETURNED,
1474  004614  100002                          BPL     6$              ;IF PLUS, NEXT TEST
1475
1476  004616  104005                          ERROR   5               ;ERROR-SLAVE MICRO PROCESSOR
1477                                                                  ;INDICATED M8254 FAILED TO
1478                                                                  ;INITIALIZE PROPERLY.
```

```
 1479  004620  000405                          BR      TST4            ;;
```

```
1480
1481  004622  123737  001124  001126  6$:  CMPB  $GDDAT,$BDDAT  ;DOES THE MICRO-CODE VERSION
1482                                        ;NUMBER AGREE WITH THE
1483                                        ;VERSION WE EXPECT?
1484  004630  001401                  BEQ  TST4                 ;;
1485
1486  004632  104005                  ERROR  5                  ;WRONG MICRO-CODE VERSION #
1487                                        ;RETURNED BY SLAVE MICRO-PROCESSOR.
1488                                        ;NOTE:(1) YOU COULD BE RUNNING THE
1489                                        ;WRONG VERSION OF THE DIAGNOSTIC
1490                                        ;(2) SLAVE MICRO COULD BE HAD
1491                                        ;OR (3) M8254 MIGHT BE BAD
1492                                        ;(FAST PATH)
1493
1494                              .SBTTL  **PHASE 2 IPBM/DMC TESTING.
```

```
                        ;;*****************************************************************
                        ;*TEST 4         *TEST DATA LOOP BACK THROUGH IPBM,M8254 MODULE
                        ;*
                        ;*IN THIS TEST WE WILL SEND DATA THROUGH THE SILO
                        ;*AND FAST PATH. TO UNDERSTAND HOW WE'RE ABLE TO DO THIS,
                        ;*YOU MUST FIRST UNDERSTAND WHAT THE SLAVE MICRO-PROCESSOR
                        ;*IS DOING.
                        ;*      WHEN THE SLAVE RECEIVES CODE 200 IN ITS FAST PATH REGISTER
                        ;*      WHILE IT IS COMMAND MODE, IT MONITORS THE F.P. AND READS
                        ;*      THE F.P. INPUT (WHEN DATA IS PRESENT) AND LOOPS IT BACK TO ITS
                        ;*      F.P. OUTPUT. IT DOES THIS 256 (DECIMAL) TIMES. NEXT THE SLAVE
                        ;*      MONITORS THE SILO REGISTER AND LOOPS SILO DATA BACK
                        ;*      FOR 256 TIMES. AT THE CONCLUSION OF THIS, IT RETURNS TO
                        ;*      COMMAND MODE.
                        ;*
                        ;*THIS TEST USES KMC-11 MICRO-CODE FILE "DRLPX0" PRE LOADED
                        ;*INTO THE KMC-11 BY THE PREVIOUS TEST.

                        ;*THIS MICRO-CODE RUNS CONTINOUS READING THE IPBM STATUS
                        ;*PUTTING IT IN KMC-11 ADDR.0, AND LOOKING FOR A COMMAND TO BE
                        ;*PUT INTO KMC-11 ADDR.2. IT RETURNS CODE
                        ;*"377" TO ADDR.2 WHEN FINISHED COMMAND. DATA IS XFERRED
                        ;*THROUGH KMC-11 ADDR.4.
                        ;*HERE IS A LIST OF COMMANDS FOR FILE "DRLPX0"
                        ;*IF YOU HALT AT THE END OF THIS TEST YOU MAY USE
                        ;*THIS MICRO-CODE. CAUTION: BIT15 OF KMC-11 ADDR. MUST REMAIN
                        ;*SET WHEN A COMMAND IS GIVEN (THIS IS THE "RUN" BIT) (INITIALIZE
                        ;*ALSO CLEARS BIT). I USE BYTE ADDRESSING TO AVOID CLEARING IT.
                        ;*
                        ;*              "DRLPX0" COMMAND          FUNCTION
                        ;*              ----------------          --------
                        ;*                     1                   NOP
                        ;*                     2                 READ SILO, PUT IN LOW BYTE KMC ADDR.4
                        ;*                     3                 WRITE SILO, DATA IN LOW BYTE OF KMC ADDR.4
                        ;*                     4                 READ FAST PATH REG., PUT IN LOW BYTE OF KMC ADDR
                        ;*                     5                 WRITE FAST PATH REG., DATA IN LOW BYTE OF KMC AD
                        ;*
                        ;*COMMAND "200" SENT TO THE SLAVE THROUGH THE FAST PATH
                        ;*REG TO PUT IT INTO LOOP BACK MODE.
                        ;*
                        ;*YOU MAY NOT LOOP ON A SPECIFIC DATA PATTERN FAILURE
                        ;*YOU MAY USE LOOP ON TEST OPTION (SWR BIT14).
                        ;*IF THIS TEST FAILS YOU SHOULD RUN THE M8254(IPBM) DIAGNOSTIC.

                        ;;*****************************************************************
004634  000004         TST4:   SCOPE
004636  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS

004644  005037  005266          1$:     CLR     2$              ;DATA LOOP COUNTER
004650  012737  000001  001106          MOV     #1,$LPADR       ;SET FOR LOOP ON ADDR.
004656  012737  004672  001106          MOV     #1$,$LPADR      ;IN CASE LOOP ERROR IS USED.
004664  012737  000000  005270          MOV     #0,21$          ;SET INITIAL DATA PATTERN.
004672  005737  005266                  TST     2$              ;CHECK FOR LEGAL TEST ENTRY.
004676  001006                          BNE     2$              ;BAD ENTRY, PROBABLY FROM "LOOP ON ERROR"
```

```
1549                                                                    ;WE'LL TRY TO rECOVER
1550
1551   004700  112777  000200  174570          MOVB    #200,@KMAD4     ;PUT "LOOP BACK" CMMD INTO XFERR REG.
1552   004706  112777  000005  174556          MOVB    #5,@KMAD2       ;MAKE "DRLPX0" GIVE FAST PATH CMMD TO
1553                                                                    ;SLAVE.
1554                                                                    ;AT THIS POINT, SLAVE WILL
1555                                                                    ;LOOP 256 (DECIMAL) PIECES OF DATA
1556                                                                    ;THROUGH THE FAST PATH  REG.
1557
1558   004714  105737  005267          2$:     TSTB    20$+1           ;IF IN SILO SECTION, EXIT F.P. SECTION.
1559   004720  001064                          BNE     10$
1560   004722  004737  034062                  JSR     PC,WIR          ;WAIT FOR PROGRAM "DRLPX0" TO BECOME
1561                                                                    ;READY FOR ANOTHER COMMAND.
1562
1563   004726  012737  000052  001124          MOV     #52,$GDDAT      ;AT THIS POINT, EXPECT IPBM (MASTER
1564   004734  005037  001126                  CLR     $BDDAT          ;SIDE) CSR=52
1565   004740  117737  174522  001126          MOVB    @KMAD0,$BDDAT   ;READ CSR ("DRLPX0" REFLECTS CSR IN
1566                                                                    ;KMC'S ADDR.0)
1567   004746  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;-RIGHT CODE?
1568   004754  001401                          BEQ     3$              ;YES-SKIP ERROR.
1569
1570   004756  104005                          ERROR   5               ;CSR BAD. IPBM MASTER CSR S/B 52.
1571
1572   004760  113777  005270  174510  3$:     MOVB    21$,@KMAD4      ;PUT DATA PATTERN IN PRG. "DRLPX0"
1573   004766  005237  005266                  INC     20$             ;XFERR REG.
1574   004772  112777  000005  174472          MOVB    #5,@KMAD2       ;ISSUE WRITE F.P. CMMD.
1575   005000  004737  034062                  JSR     PC,WIR          ;WAITE FOR "DRLPX0" TO BECOME READY AGAIN.
1576                                                                    ;AT THIS POINT IN TIMES THE
```

```
1577                                                                    ;SLAVE SHOULD "SEE" DATA IN THE
1578                                                                    ;F.P. AND LOOP IT BACK TO US.
1579  005004  005000                          CLR     R0
1580  005006  032777  000040  174452  4$:    BIT     #BIT5,@KMADO      ;WAIT TILL F.P. HAS DATA TO BE READ.
1581  005014  001403                          BEQ     5$                ;WHEN READY, EXIT LOOP,
1582  005016  105200                          INCB    R0                ;DON'T WAIT TOO AWFUL LONG!
1583  005020  100372                          BPL     4$                ;IF INC R0 OVERFLOW, THE SLAVE
1584                                                                    ;NEVER WILL RETURN DATA.
1585
1586  005022  104004                          ERROR   4                 ;SLAVE MICRO-PROCESSOR FAILED TO
1587                                                                    ;LOOP BACK DATA THROUGH FAST PATH REG.
1588
1589
1590  005024  112777  000004  174440  5$:    MOVB    #4,@KMAD2         ;DATA IN F.P. READY TO BE READ, BETS
1591                                                                    ;TELL "DRLPXO" TO READ THE F.P..
1592  005032  004737  034062                  JSR     PC,WIR            ;WAITE FOR DRLPXO.
1593
1594  005036  117737  174434  001126          MOVB    @KMAD4,$BDDAT     ;READ DATA
1595  005044  113737  005270  001124          MOVB    21$,$GDDAT        ;21$ CONTAINED DATA SENT
```

```
                                                    ;DID DATA LOOP BACK OK?
005052  123737  001124  001126        CMPB    $GDDAT,$BDDAT
005060  001401                      BEQ     6$
                                                    ;FAST PATH  DATA ERROR
005062  104007                      ERROR   7
                                                    ;UPDATE LOOP BACK PATTERN.
005064  105237  005270          6$: INCB    21$     ;NOTE: IF YOU REALY WANT TO, YOU
                                                    ;CAN MODIFY THIS SECTION OF TEST
                                                    ;TO SEND ONLY A SPECIFIC PATTERN.
                                                    ;TO DO THIS, MODIFY
                                                    ;THE INSTRUCTOR "MOV #0,21$" TO
                                                    ;ANY STARTING PATTERN YOU WANT.
                                                    ;MODIFIED TO "NOP,NOP" IF YOU WISH
                                                    ;TO SEND THE SAME PATTERN, OR
                                                    ;TO ANY OTHER AUNRY INSTRUCTION THAT
                                                    ;WOULD MODIFY THE PATTERN.
005070  000711                      BR      2$      ;IF NOT DONE 256 TIMES LOOP.
                                                    ;WARNING: THE PROGRAM CAN ONLY
                                                    ;SEND 256 DATA BYTES THROUGH
                                                    ;THE FAST PATH REG. BEFORE IT MUST
                                                    ;LOOP 256 DATA BYTES THROUGH
                                                    ;THE SILO.
005072                          10$:                ;WE COME HERE ONLY AFTER
                                                    ;256 DATA BYTES HAVE BEEN
                                                    ;XFERRED THROUGH THE FAST PATH
                                                    ;NOW WE'LL LOOP 256 OF DATA
                                                    ;THROUGH THE SILO.
005072  012737  000000  005270      MOV     #0,21$  ;PUT NEW PATTERN IN PATTERN LOC
                                                    ;IF YOU DESIRE ANOTHER STARTING
                                                    ;PATTERN, YOU MAY REPLACE THE 0
                                                    ;WITH IT (REMEMBER ONLY 8 BITS LONG).
005100  032737  001000  005266  11$: BIT     #BIT9,20$
005106  001064                      BNE     18$     ;DONE?
005110  004737  034062              JSR     PC,WIR  ;YES-EXIT.
                                                    ;MAKE SURE "DRLPX0" IS READY.
005114  012737  000052  001124      MOV     #52,$GDDAT
005122  005037  001126              CLR     $BDDAT  ;AT THIS POINT, EXPECT IPBM (MASTER
005126  117737  174334  001126      MOVB    @KMAD0,$BDDAT ;SIDE) SCR=52
                                                    ;READ CSR ("DRLPX0" REFLECTS CSR IN
                                                    ;KMC'S ADDR.0)
005134  123737  001124  001126      CMPB    $GDDAT,$BDDAT
005142  001401                      BEQ     12$     ;-RIGHT CODE?
                                                    ;-YES-SKIP ERROR.
005144  104005                      ERROR   5       ;CSR BAD, IPBM MASTER SIDE, S/B 52
                                                    ;WE WERE OK WHILE WORKING
                                                    ;WITH FAST PATH, CSR WENT BAD DURING
                                                    ;SILO TESTING.
005146  113777  005270  174322  12$: MOVB    21$,@KMAD4
005154  005237  005266              INC     20$     ;PUT DATA PATTERN IN PROG "DRLPX0"
005160  112777  000003  174304      MOVB    #3,@KMAD2 ;XFERR REG.
                                                    ;MAKE "DRLPX0" WRITE SILO.
```

```
1650  005166  004737  034062                JSR     PC,WIR          ;WAIT TILL "DRLPXO" READY.
1651                                                                  ;NOW-WE PUT DATA IN SILO.
1652                                                                  ;SLAVE MICRO-CODE READS SILO
1653                                                                  ;DATA IN AND PUTS IT IN SILO
1654                                                                  ;OUT. THATS WHERE WE PICK IT UP.
1655
1656  005172  005000                        CLR     R0
1657  005174  032777  000002  174264  13$:  BIT     #BIT1,@KMADO    ;WAIT TILL SILO HAS DATA TO BE READ
1658  005202  001403                        BEQ     14$             ;WHEN READY, EXIT LOOP.
1659  005204  105200                        INCB    R0              ;DON'T WAIT TOO LONG!
1660  005206  100372                        BPL     13$             ;IF INC R0 OVERFLOW, THE SLAVE
1661                                                                  ;NEVER WILL RETURN DATA.
1662
1663  005210  104004                        ERROR   4               ;SLAVE MICRO FAILED TO LOOP BACK
1664                                                                  ;DATA THROUGH SILO..
1665
1666  005212  112777  000002  174252  14$:  MOVB    #2,@KMAD2       ;DATA IN SILO READY TO BE READ, TELL
1667                                                                  ;DRLPXO TO READ IT.
1668  005220  004737  034062                JSR     PC,WIR          ;WAIT FOR DRLPXO READY
1669
1670  005224  117737  174246  001126        MOVB    @KMAD4,$BDDAT   ;GET DATA.
1671  005232  113737  005270  001124        MOVB    21$,$GDDAT      ;GET S/B
1672
1673  005240  123737  001124  001126        CMPB    $GDDAT,$BDDAT   ;DID DATA LOOP BACK OK?
1674  005246  001401                        BEQ     15$
1675
1676  005250  104006                        ERROR   6               ;SILO DATA ERROR.
1677
```

```
1678
1679  005252  105237  005270          15$:   INCB    21$          ;UPDATE LOOP BACK PATTERN
1680                                         ;NOTE: IF YOU REALLY WANT TO,
1681                                         ;YOU CAN MODIFY THIS
1682                                         ;SECTION OF TEST TO SEND ONLY A
1683                                         ;SPECIFIC PATTERN.
1684                                         ;TO DO THIS, CHANGE THE
1685                                         ;INSTRUCTION "INCB 21$" (2 WORDS)
1686                                         ;ALONG WITH THE "MOV#0,21$"
1687
1688  005256  000710                         BR      11$          ;IF NOT DONE 256 TIMES, LOOP.
1689                                         ;WARNING: THIS PROGRAM MUST
1690                                         ;SEND 256 DATA LOOP BACK PATTERNS
1691                                         ;THROUGH THE SILO REG SO
1692                                         ;THAT THE SLAVE MICRO CODE
1693                                         ;WILL EXIT FROM LOOP BACK
1694                                         ;MODE TO COMMAND MODE.
1695
1696
1697  005260  005037  005266          18$:   CLR     20$          ;EXIT, ALL DONe!
1698  005264  000402                         BR      TST5         ;;
1699
```

```
 1700  005266  000000                20$:   .WORD   0          ;CONTAINS CURRENT LOOP COUNT.
 1701  005270  000000                21$:   .WORD   0          ;CONTAINS CURRENT LOOP BACK PATTERN.
```

```
1702
1703                                    .SBTTL  PHASE 3 I/B BUS TESTING.
```

```
1704
```

```
                        ;*****************************************************************
                        ;*TEST 5        I/O DEVICE TEST-AD11-K OPTION
                        ;*
                        ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
                        ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
                        ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
                        ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
                        ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
                        ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
                        ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
                        ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
                        ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
                        ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
                        ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
                        ;*
                        ;*      IN THIS TEST, WE WILL EXERCISE THE AD11-K
                        ;*      IF DVID1 IS UNALTERED, WE WILL DO THIS TEST SENCE THE
                        ;*      AD11-K IS A DEFAULT I/O DEVICE. IF AN AD11-K IS NOT ON THE
                        ;*      I/O BUS, YOU MUST DELETE IT FROM SR.
                        ;*
                        ;*      TESTS:          1.  MAKE SURE AD11-K RESPONDS TO ADDRESS.
                        ;*                          (NOTE IF THE AD11-K ON YOU I/O BUS HAS NON-STANDARD
                        ;*                          ADDRESS, PLEASE REFLECT THE NEW ADDRESS IN ADDRESS "STREG"
                        ;*
                        ;*                      2.  MAKE SURE WE CAN WRITE A ZERO INTO CSR (EXCEPT BIT 7:15).
                        ;*
                        ;*
                        ;*                      3.  WRITE BITS 13,11,9,6 MAKE SURE THEY WRITE.
                        ;*
                        ;*                      4.  WRITE BITS 12,10,8,4 MAKE SURE THEY WRITE.
                        ;*
                        ;*                      5.  SET A/D START, MAKE SURE IT CLEARS, DONE FLAG SETS.
                        ;*
                        ;*
                        ;*                      6.  SET A/D START AGAIN, MAKE SURE ERROR FLAG SETS,
                        ;*
                        ;*                      7.  MAKE SURE ERROR FLAG CAN BE CLEARED.
                        ;*
                        ;*****************************************************************
005272  000004          TST5:   SCOPE
005274  032737  000001  001562          BIT     #BIT0,SR1       ;IS THIS DEVICE SELECTED FOR TEST?
                                                                ;NOTE: DEFAULT=YES.
005302  001002          10$:    BNE     10$
005304  000137  005750          JMP     7$
005310                          CLR     $BDDAT
005310  005037  001126
                        ;*      MOV     $BDDAT,@STREG1  ;/ PUT DATA FROM $BDDAT TO DEVICE REG STREG1
                                                        ;OK-WHAT WE JUST DID IS ASKED THE
                                                        ;SLAVE MICRO-CODE TO WRITE THE CONTENTS
```

```
                                                        ;OF THE AD11K'S CSR TO ZEROES.
                                                        ;IF THE AD11K FAILS TO rESPOND TO ITS
                                                        ;ADDRESS-THE SLAVE WILL SEND US AN
                                                        ;ERROR CODE THAT CAUSES US TO SET $AERR=1
                                                        ;IN THE SUPPORT ROUTINES.

005324  005737  037532              TST     $AERR       ;DEVICE PRESENT?
005330  001403                      BEQ     11$         ;YES-CONTINUE
005332  104010                      ERROR   10          ;A/D #1 FAILED TO RESPOND
005334  000137  005750              JMP     7$          ;TO ADDR.
005340                        11$:

                              ;*    MOV     @STREG1,$BDDAT  ;/READ DEVICE REG STREG1,PUT DATA IN $BDDAT.
005350  042737  100200  001126      BIC     #BIT15!BIT7,$BDDAT ;CLEAR OUT FLAGS
005356  005037  001124              CLR     $GDDAT
005362  005737  001126              TST     $BDDAT      ;IS CSR ZERO?
005366  001402                      BEQ     1$

005370  104010                      ERROR   10          ;FAILED TO ZERO AD11K CSR #1
005372  000572                      BR      TST6        ;;
005374  012737  025100  001124  1$: MOV     #BIT13!BIT11!BIT9!BIT6,$GDDAT ;NOW TRY A NEW BIT PATTERN

                              ;*    MOV     $GDDAT,@STREG1  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG1

                              ;*    MOV     @STREG1,$BDDAT  ;/READ DEVICE REG STREG1,PUT DATA IN $BDDAT.
005422  042737  100200  001126      BIC     #BIT15!BIT7,$BDDAT ;IGNORE FLAGS
005430  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;BIT SET OK?
005436  001402                      BEQ     2$

005440  104010                      ERROR   10          ;FAILED TO WRITE #1 AD11K CSR CORRECTLY
005442  000546                      BR      TST6        ;;

005444  012737  012420  001124  2$: MOV     #BIT12!BIT10!BIT8!BIT4,$GDDAT ;NOW TRY A NEW BIT PATTERN.

                              ;*    MOV     $GDDAT,@STREG1  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG1

                              ;*    MOV     @STREG1,$BDDAT  ;/READ DEVICE REG STREG1,PUT DATA IN $BDDAT.
005472  042737  100200  001126      BIC     #BIT15!BIT7,$BDDAT ;IGNORE FLAGS
005500  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;BITS SET OK?
005506  001402                      BEQ     3$

005510  104010                      ERROR   10          ;FAILED TO wRITE  #1 AD11K CSR CORRECTLY.
005512  000522                      BR      TST6        ;;

005514  005037  001124  3$: CLR     $GDDAT              ;NOW CLEAR ALL BITS SET

                              ;*    MOV     $GDDAT,@STREG1  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG1

                              ;*    MOV     @STREG1,$BDDAT  ;/READ DEVICE REG STREG1,PUT DATA IN $BDDAT.
005540  042737  000200  001126      BIC     #BIT7,$BDDAT    ;IGNORE DONE FLAG.
005546  005737  001126              TST     $BDDAT      ;CSR CLEAR?
005552  001402                      BEQ     4$
```

```
1813
1814  005554  104010                          ERROR   10              ;#1 AD11K CSR FAILED TO CLEAR.
1815  005556  000500                          BR      TST6            ;;
1816
1817  005560                          4$:
1818
1819                                  ;*      MOV     @ADBUF1,MYTEMP  ;/READ DEVICE REG ADBUF1,PUT DATA IN MYTEMP.
1820  005570  012737  000001  001124         MOV     #BIT0,$GDDAT    ;NOW WE'LL SET THE A/D START BIT.
1821
1822                                  ;*      MOV     $GDDAT,@STREG1  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG1
1823  005606  012737  000200  001124         MOV     #BIT7,$GDDAT    ;WHEN WE READ CSR BACK EXPECT
1824                                                                  ;START BIT TO CLEAR, DONE TO SET.
1825
1826                                  ;*      MOV     @STREG1,$BDDAT  ;/READ DEVICE REG STREG1,PUT DATA IN $BDDAT.
1827
1828  005624  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;START OK?
1829  005632  001402                         BEQ     5$
1830
1831  005634  104010                          ERROR   10              ;CSR BAD AFTER #1 A/D START
1832                                                                  ;BIT 0 SHOULD CLEAR ONLY A/D DONe SET.
1833  005636  000450                          BR      TST6            ;;
1834
1835  005640  012737  000001  001124    5$:   MOV     #BIT0,$GDDAT    ;OK-NOW WE'RE GONNA START ANOTHER
1836                                                                  ;A/D CONVERSION WITHOUT READING
1837                                                                  ;THE RESULTS OF THE LAST ONE. THIS
1838                                                                  ;SHOULD CAUSE THE ERROR FLAG
1839                                                                  ;AS WELL AS THE DONe FLAG TO sET.
1840
1841                                  ;*      MOV     $GDDAT,@STREG1  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG1
1842
1843
1844                                  ;*      MOV     @STREG1,$BDDAT  ;/READ DEVICE REG STREG1,PUT DATA IN $BDDAT.
1845  005666  012737  100200  001124         MOV     #BIT15!BIT7,$GDDAT ;S/B
1846
1847  005674  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;IS 15 AND 7 SET?
1848
1849  005702  001401                         BEQ     6$
1850
1851  005704  104010                          ERROR   10              ;ERROR #1 AD11K BIT 15 AND 7 SHOULD BE SET.
1852
1853
1854  005706  005037  001124      6$:   CLR     $GDDAT          ;MAKE SURE ERROR FLAG CLEARS
1855
1856                                ;*      MOV     $GDDAT,@STREG1  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG1
1857
1858                                ;*      MOV     @STREG1,$BDDAT  ;/READ DEVICE REG STREG1,PUT DATA IN $BDDAT.
1859  005732  042737  000200  001126         BIC     #BIT7,$BDDAT    ;IGNORE DONE FLAG.
1860  005740  005737  001126                 TST     $BDDAT          ;IS CSR CLEAR?
1861  005744  001401                         BEQ     7$
1862  005746  104010                          ERROR   10              ;AD11K CSR FAILED TO CLEAR.
1863
1864  005750                          7$:
1865
1866                                  ;*      MOV     @ADBUF1,MYTEMP  ;/READ DEVICE REG ADBUF1,PUT DATA IN MYTEMP.
```

```
                        ;*****************************************************************
                        ;*TEST 6        I/O DEVICE TEST-AD11-K OPTION
                        ;*
                        ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
                        ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
                        ;*KMC-11 AND THE DMC-11 WORK, IN THIS TEST, THE ARBITRATION CIRCUITRY IS
                        ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
                        ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
                        ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
                        ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
                        ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
                        ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
                        ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
                        ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
                        ;*
                        ;*      IN THIS TEST, WE WILL EXERCISE THE AD11-K
                        ;*      IF DVID1 IS UNALTERED, WE WILL DO THIS TEST SENCE THE
                        ;*      AD11-K IS A DEFAULT I/O DEVICE. IF AN AD11-K IS NOT ON THE
                        ;*      I/O BUS, YOU MUST DELETE IT FROM SR.
                        ;*
                        ;*      TESTS:          1.  MAKE SURE AD11-K RESPONDS TO ADDRESS.
                        ;*                          (NOTE IF THE AD11-K ON YOU I/O BUS HAS NON-STANDARD
                        ;*                          ADDRESS, PLEASE REFLECT THE NEW ADDRESS IN ADDRESS "STREG"
                        ;*
                        ;*                      2.  MAKE SURE WE CAN WRITE A ZERO INTO CSR (EXCEPT BIT 7:15).
                        ;*
                        ;*                      3.  WRITE BITS 13,11,9,6 MAKE SURE THEY WRITE.
                        ;*
                        ;*                      4.  WRITE BITS 12,10,8,4 MAKE SURE THEY WRITE.
                        ;*
                        ;*                      5.  SET A/D START, MAKE SURE IT CLEARS, DONE FLAG SETS.
                        ;*
                        ;*
                        ;*                      6.  SET A/D START AGAIN, MAKE SURE ERROR FLAG SETS,
                        ;*
                        ;*                      7.  MAKE SURE ERROR FLAG CAN BE CLEARED.
                        ;*
                        ;*****************************************************************
005760  000004         TST6:   SCOPE
005762  032737  000020  001562          BIT     #BIT4,SR1       ;IS THIS DEVICE SELECTED FOR TEST?
                                                                ;NOTE: DEFAULT=YES.
005770  001002                        BNE     10$
005772  000137  006436              JMP     7$
005776                         10$:
005776  005037  001126              CLR     $BDDAT
                        ;*      MOV     $BDDAT,@STREG2  ;/ PUT DATA FROM $BDDAT TO DEVICE REG STREG2
                                                        ;OK-WHAT WE JUST DID IS ASKED THE
                                                        ;SLAVE MICRO-CODE TO WRITE THE CONTENTS
```

```
                                                                ;OF THE AD11K'S CSR TO ZEROES.
                                                                ;IF THE AD11K FAILS TO RESPOND TO ITS
                                                                ;ADDRESS-THE SLAVE WILL SEND US AN
                                                                ;ERROR CODE THAT CAUSES US TO SET $AERR=1
                                                                ;IN THE SUPPORT ROUTINES.

006012  005737  037532                TST     $AERR           ;DEVICE PRESENT?
006016  001403                         BEQ     11$             ;YES-CONTINUE
006020  104010                         ERROR   10              ;A/D #2 FAILED TO RESPOND
006022  000137  006436                 JMP     7$              ;TO ADDR.
006026                          11$:
                                ;*      MOV     @STREG2,$BDDAT  ;/READ DEVICE REG STREG2,PUT DATA IN $BDDAT.
006036  042737  100200  001126         BIC     #BIT15!BIT7,$BDDAT ;CLEAR OUT FLAGS
006044  005037  001124                 CLR     $GDDAT
006050  005737  001126                 TST     $BDDAT          ;IS CSR ZERO?
006054  001402                         BEQ     1$

006056  104010                         ERROR   10              ;FAILED TO ZERO AD11K CSR #2
006060  000572                         BR      TST7            ;;
006062  012737  025100  001124     1$: MOV     #BIT13!BIT11!BIT9!BIT6,$GDDAT ;NOW TRY A NEW BIT PATTERN

                                ;*      MOV     $GDDAT,@STREG2  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG2

                                ;*      MOV     @STREG2,$BDDAT  ;/READ DEVICE REG STREG2,PUT DATA IN $BDDAT.
006110  042737  100200  001126         BIC     #BIT15!BIT7,$BDDAT ;IGNORE FLAGS
006116  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;BIT SET OK?
006124  001402                         BEQ     2$

006126  104010                         ERROR   10              ;FAILED TO WRITE #2 AD11K CSR CORRECTLY
006130  000546                         BR      TST7            ;;
006132  012737  012420  001124     2$: MOV     #BIT12!BIT10!BIT8!BIT4,$GDDAT ;NOW TRY A NEW BIT PATTERN.


                                ;*      MOV     $GDDAT,@STREG2  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG2

                                ;*      MOV     @STREG2,$BDDAT  ;/READ DEVICE REG STREG2,PUT DATA IN $BDDAT.
006160  042737  100200  001126         BIC     #BIT15!BIT7,$BDDAT ;IGNORE FLAGS
006166  023737  001124  001126         CMP     $GDDAT,$BDDAT   ;BITS SET OK?
006174  001402                         BEQ     3$

006176  104010                         ERROR   10              ;FAILED TO WRITE  #2 AD11K CSR CORRECTLY.
006200  000522                         BR      TST7            ;;

006202  005037  001124             3$: CLR     $GDDAT          ;NOW CLEAR ALL BITS SET

                                ;*      MOV     $GDDAT,@STREG2  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG2

                                ;*      MOV     @STREG2,$BDDAT  ;/READ DEVICE REG STREG2,PUT DATA IN $BDDAT.
006226  042737  000200  001126         BIC     #BIT7,$BDDAT    ;IGNORE DONE FLAG.
006234  005737  001126                 TST     $BDDAT          ;CSR CLEAR?
006240  001402                         BEQ     4$
```

```
                                                ERROR   10              ;#2 AD11K CSR FAILED TO CLEAR.
006244  000500                                  BR      TST7            ;;

006246                                  4$:

                                        ;*      MOV     @ADBUF2,MYTEMP  ;/READ DEVICE REG ADBUF2,PUT DATA IN MYTEMP.
006256  012737  000001  001124                  MOV     #BIT0,$GDDAT    ;NOW WE'LL SET THE A/D START BIT.

                                        ;*      MOV     $GDDAT,@STREG2  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG2
006274  012737  000200  001124                  MOV     #BIT7,$GDDAT    ;WHEN WE READ CSR BACK EXPECT
                                                                        ;START BIT TO CLEAR, DONE TO SET.

                                        ;*      MOV     @STREG2,$BDDAT  ;/READ DEVICE REG STREG2,PUT DATA IN $BDDAT.

006312  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;START OK?
006320  001402                                  BEQ     5$

006322  104010                                  ERROR   10              ;CSR BAD AFTER #2 A/D START
                                                                        ;BIT 0 SHOULD CLEAR ONLY A/D DONE SET.
006324  000450                                  BR      TST7            ;;

006326  012737  000001  001124          5$:     MOV     #BIT0,$GDDAT    ;OK-NOW WE'RE GONNA START ANOTHER
                                                                        ;A/D CONVERSION WITHOUT READING
                                                                        ;THE RESULTS OF THE LAST ONE. THIS
                                                                        ;SHOULD CAUSE THE ERROR FLAG
                                                                        ;AS WELL AS THE DONE FLAG TO SET.

                                        ;*      MOV     $GDDAT,@STREG2  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG2

                                        ;*      MOV     @STREG2,$BDDAT  ;/READ DEVICE REG STREG2,PUT DATA IN $BDDAT.
006354  012737  100200  001124                  MOV     #BIT15!BIT7,$GDDAT ;S/B

006362  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;IS 15 AND 7 SET?

006370  001401                                  BEQ     6$

006372  104010                                  ERROR   10              ;ERROR #2 AD11K BIT 15 AND 7 SHOULD BE SET.

006374  005037  001124                  6$:     CLR     $GDDAT          ;MAKE SURE ERROR FLAG CLEARS

                                        ;*      MOV     $GDDAT,@STREG2  ;/ PUT DATA FROM $GDDAT TO DEVICE REG STREG2

                                        ;*      MOV     @STREG2,$BDDAT  ;/READ DEVICE REG STREG2,PUT DATA IN $BDDAT.
006420  042737  000200  001126                  BIC     #BIT7,$BDDAT    ;IGNORE DONE FLAG.
006426  005737  001126                          TST     $BDDAT          ;IS CSR CLEAR?
006432  001401                                  BEQ     7$
006434  104010                                  ERROR   10              ;AD11K CSR FAILED TO CLEAR.

006436                                  7$:

                                        ;*      MOV     @ADBUF2,MYTEMP  ;/READ DEVICE REG ADBUF2,PUT DATA IN MYTEMP.
```

```
2029
2030
2031                    ;;**********************************************************************
2032                    ;*TEST 7        I/O DEVICE TEST-KW11K OPTION
2033                    ;
2034                    ;*
2035                    ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
2036                    ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
2037                    ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
2038                    ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
2039                    ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
2040                    ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
2041                    ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
2042                    ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
2043                    ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
2044                    ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
2045                    ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
2046                    ;*
2047                    ;*      IN THIS TEST WE WILL EXERCISE THE KW11-K. IF
2048                    ;*      SR1 IS UNALTERED, WE DO THIS TEST SINCE THE
2049                    ;*      KW11K IS A DEFAULT I/O DEVICE.
2050                    ;*      PLEASE NOTE: THIS IS ONLY A BRIEF TEST OF THE KW11K,
2051                    ;*      TO SEE THAT IT ROUGHLY WORKS. FOR TESTING, YOU
2052                    ;*      MUST RUN THE KW11-K DIAGNOSTIC. COMPLETE
2053                    ;*
2054                    ;*      TESTS:          1.  MAKE SURE KW11K RESPONDS TO ADDRESS
2055                    ;*                          (NOTE: IF THE KW11K ON YOUR I/O BUS USES A NON
2056                    ;*                          STANDARD ADDRESS, MODIFY "KWADD" WITH THE CORRECT ADDR.)
2057                    ;*
2058                    ;*                      2.  CLOCK A CSR BITS 14,9,6,3, AND 1 SET
2059                    ;*
2060                    ;*                      3.  CLOCK A CSR BITS 13,8 AND 2 SET
2061                    ;*
2062                    ;*                      4.  CLOCK A CSR ZEROS.
2063                    ;*
2064                    ;*                      5.  CLOCK B CSR BITS 11,6,4 AND 2 SET
2065                    ;*
2066                    ;*                      6.  CLOCK B CSR BITS 5,3, AND 0 SET
2067                    ;*
2068                    ;*                      7.  CLOCK B CSR ZEROS
2069                    ;*
2070                    ;*                      8.  START CLOCK A MODE 0, RATE 1MHZ,
2071                    ;*                          CHECK "ENABLE CNTR A" CLEARS,
2072                    ;*                          "MODE FLAG" AND "A OVERFLOW FLAG" SETS
2073                    ;*
2074                    ;*                      9.  CLOCK A CSR ZEROS
2075                    ;*
2076                    ;*                      10. START CLOCK B RATE 1MHZ
2077                    ;*                          CHECK "ENB CNTR B" CLEARS, "B OVERFL
2078                    ;*                          FLAG" SETS
2079                    ;*
2080                    ;**********************************************************************
2081  006446  000004    TST7:   SCOPE
2082
```

```
2083  006450  032737  000002  001562          BIT     #BIT1,SR1       ;IS THIS DEVICE SELECTED FOR TEST?
2084                                                                    ;NOTE: DEFAULT=YES.
2085  006456  001002                          BNE     10$
2086  006460  000137  007254                  JMP     11$
2087  006464                          10$:
2088
2089  006464  012737  041110  001124          MOV     #BIT14!BIT9!BIT6!BIT3,$GDDAT ;TEST THESE BITS
2090
2091
2092                                  ;*      MOV     $GDDAT,@KWADR0  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR0
2093
2094  006502  005737  037532                  TST     $AERR           ;DID KW11K RESPOND
2095  006506  001403                          BEQ     12$             ;TO ADDR? $AERR=0?
2096                                                                  ;IF YES-NEXT TEST.
2097  006510  104010                          ERROR   10              ;KW11K FAILED TO rESPOND TO ADDR.
2098  006512  000137  007254                  JMP     11$
2099  006516                          12$:
2100
2101
2102                                  ;*      MOV     @KWADR0,$BDDAT  ;/READ DEVICE REG KWADR0,PUT DATA IN $BDDAT.
2103
2104  006526  023737  001124  001126          CMP     $GDDAT,$BDDAT ;DID THESE BITS SET?
2105  006534  001403                          BEQ     1$
2106
2107  006536  104011                          ERROR   11              ;KW11K R/W CSR ERROR.
2108  006540  000137  007254                  JMP     11$
2109
2110  006544  012737  020404  001124  1$:     MOV     #BIT13!BIT8!BIT2,$GDDAT ;GET NEW PATTERN
2111
2112                                  ;*      MOV     $GDDAT,@KWADR0  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR0
2113
2114                                  ;*      MOV     @KWADR0,$BDDAT  ;/READ DEVICE REG KWADR0,PUT DATA IN $BDDAT.
2115
2116  006572  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID THE BITS SET?
2117  006600  001403                          BEQ     2$
2118
2119  006602  104011                          ERROR   11              ;KW11K R/W CSR ERROR.
2120  006604  000137  007254                  JMP     11$
2121
2122  006610  005037  001124          2$:     CLR     $GDDAT          ;WRITE ALL ZEROS.
2123
2124                                  ;*      MOV     $GDDAT,@KWADR0  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR0
2125
2126                                  ;*      MOV     @KWADR0,$BDDAT  ;/READ DEVICE REG KWADR0,PUT DATA IN $BDDAT.
2127  006634  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
2128  006640  001403                          BEQ     3$              ;YES-GOOD.
2129
2130  006642  104011                          ERROR   11              ;KW11K FAILED TO ZERO CSR.
2131  006644  000137  007254                  JMP     11$
2132
2133  006650  012737  004124  001124  3$:     MOV     #BIT11!BIT6!BIT4!BIT2,$GDDAT ;TEST CLOCK B'S CSR.
2134
2135                                  ;*      MOV     $GDDAT,@KWADR4  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR4
2136
```

```
                                        ;*      MOV     @KWADR4,$BDDAT  ;/READ DEVICE REG KWADR4,PUT DATA IN $BDDAT.
002137
002138
002139  006676  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID THE BITS SET OK?
002140  006704  001402                          BEQ     4$
002141
002142  006706  104011                          ERROR   11              ;KW11K CLK B CSR FAILURE.
002143  006710  000561                          BR      TST10           ;;
002144
002145  006712  012737  000050  001124  4$:     MOV     #BIT5!BIT3,$GDDAT ;TRY THESE BITS
002146
002147
002148                                  ;*      MOV     $GDDAT,@KWADR4  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR4
002149
002150                                  ;*      MOV     @KWADR4,$BDDAT  ;/READ DEVICE REG KWADR4,PUT DATA IN $BDDAT.
002151  006740  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID BITS SET OK?
002152  006746  001402                          BEQ     5$
002153
002154  006750  104011                          ERROR   11              ;KW11K-CLK B CSR ERROR
002155  006752  000540                          BR      TST10           ;;
002156
002157  006754  005037  001124            5$:   CLR     $GDDAT          ;ZERO CSR
002158
002159                                  ;*      MOV     $GDDAT,@KWADR4  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR4
002160
002161                                  ;*      MOV     @KWADR4,$BDDAT  ;/READ DEVICE REG KWADR4,PUT DATA IN $BDDAT.
002162  007000  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
002163  007004  001402                          BEQ     6$
002164
002165  007006  104011                          ERROR   11              ;KW11K-CLKB CSR ERROR.
002166  007010  000521                          BR      TST10           ;;
002167
002168  007012  012737  000003  001124    6$:   MOV     #BIT0!BIT1,$GDDAT ;PUT "ENABL CNTR A" AND RATE IMHZ IN CSR
002169  007020  012737  177777  001126          MOV     #177777,$BDDAT  ;PRELOAD CNTR.
002170
002171                                  ;*      MOV     $BDDAT,@KWADR1  ;/ PUT DATA FROM $BDDAT TO DEVICE REG KWADR1
002172
002173                                  ;*      MOV     $GDDAT,@KWADR0  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR0
002174
002175                                  ;*      MOV     @KWADR0,$BDDAT  ;/READ DEVICE REG KWADR0,PUT DATA IN $BDDAT.
002176                                                                  ;CLOCK SHOULD OVERFLOW, CLEAR
002177  007056  012737  000242  001124          MOV     #BIT5!BIT7!BIT1,$GDDAT ;BIT 0 ("ENABLE CNTR A) SET
002178                                                                  ;BIT 5 AND 7
002179  007064  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;HAPPEN OK?
002180  007072  001401                          BEQ     7$
002181                                                                  ;KW11K
002182  007074  104011                          ERROR   11              ;CLOCK CSR A PROBLEM
002183
002184  007076  005037  001124            7$:   CLR     $GDDAT          ;TRY CLEARING CSR
002185
002186                                  ;*      MOV     $GDDAT,@KWADR0  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR0
002187
002188                                  ;*      MOV     @KWADR0,$BDDAT  ;/READ DEVICE REG KWADR0,PUT DATA IN $BDDAT.
002189  007122  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
002190  007126  001402                          BEQ     8$
```

```
2191
2192  007130  104011                          ERROR   11              ;CSR FAILED TO CLEAR
2193  007132  000450                          BR      TST10           ;;
2194
2195  007134  012737  000003  001124  8$:     MOV     #BIT1!BIT0,$GDDAT ;SET "ENABL CNTR B" AND IMHZ IN CSR B
2196  007142  012737  000377  001126          MOV     #377,$BDDAT     ;PRESENT ENTR.
2197
2198                                  ;*      MOV     $BDDAT,@KWADR5  ;/ PUT DATA FROM $BDDAT TO DEVICE REG KWADR5
2199
2200                                  ;*      MOV     $GDDAT,@KWADR4  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR4
2201
2202                                                                  ;CLOCK SHOULD OVERFLOW,NOT CLEAR
2203                                                                  ;"ENB CNTR B" AND SET
2204                                                                  ;"B OVERFL FLAG"
2205
2206                                  ;*      MOV     @KWADR4,$BDDAT  ;/READ DEVICE REG KWADR4,PUT DATA IN $BDDAT.
2207  007200  012737  000203  001124          MOV     #BIT7!BIT1!BIT0,$GDDAT  ;EXPECT ONLY OVERFL FLAG SET.
2208  007206  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CSR OK?
2209  007214  001401                          BEQ     9$
2210
2211  007216  104011                          ERROR   11              ;KW11K CLKB COUNT UP ERROR.
2212
2213
2214  007220  005037  001124          9$:     CLR     $GDDAT          ;TRY CLEARING CSR
2215
2216                                  ;*      MOV     $GDDAT,@KWADR4  ;/ PUT DATA FROM $GDDAT TO DEVICE REG KWADR4
2217
2218                                  ;*      MOV     @KWADR4,$BDDAT  ;/READ DEVICE REG KWADR4,PUT DATA IN $BDDAT.
2219  007244  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
2220  007250  001401                          BEQ     TST10           ;;
2221
2222  007252  104011                          ERROR   11              ;KW11K CLOCK B SCR FAILED TO CLEAR.
2223
2224  007254                          11$:
2225
2226
2227
2228                                  ;*****************************************************************
2229                                  ;*TEST 10        *TEST THE DR11K OPTION,#1, IF "SR1" BIT 2=1(SET)
2230                                  ;*
2231                                  ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
2232                                  ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IP8M), THE
2233                                  ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
2234                                  ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
2235                                  ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
2236                                  ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
2237                                  ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
2238                                  ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
2239                                  ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
2240                                  ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
2241                                  ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
2242                                  ;*
2243                                  ;*IN THIS TEST WE WILL CHECK OUT DR11 #1 IF BIT 2 OF SR1=1
2244                                  ;*(SET). ALSO, DATA LOOP BACK (OUTPUT DATA BACK TO INPUT) WILL BE CHECKED
```

```
                                        ;*IF "SR2" BIT 2 =1 (SET). FOR THIS, YOU MUST HAVE INSTALLED A LOOP-BACK
                                        ;*CABLE.
                                        ;.*************************************************************
007254  000004                  TST10:  SCOPE
007256  032737  000004  001562          BIT     #BIT2,SR1
007264  001002                          BNE     10$             ;/-TDRT-
007266  000137  007762                  JMP     11$
007272                          10$:
007272  005037  001514                  CLR     .DVLS
007276  005037  001124                  CLR     $GDDAT          ;/SET TO xFERR ZEROS.

                                ;*      MOV     $GDDAT,@DRCR1   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR1
007312  005737  037532                  TST     $AERR           ;=1 IF DEVICE NOT PRESENT,
007316  001403                          BEQ     1$              ;IF 0, PRESENT, (CON'T).

007320  104012                          ERROR   12              ;DR11K #1 DID NOT RESPOND WHEN
                                                                ;ADDRESSED. "SR1" BIT 2 SELECTED
                                                                ;THIS OPTION. ADDRESS "DRCR"1
                                                                ;CONTAINS ITS ADDRESS

007322  000137  007762                  JMP     11$

007326                          1$:

                                ;*      MOV     @DRCR1,$BDDAT   ;/READ DEVICE REG DRCR1,PUT DATA IN $BDDAT.
007336  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
007342  001403                          BEQ     2$

007344  104012                          ERROR   12              ;DR11K #1 CSR FAILED TO CLEAR.
007346  000137  007762                  JMP     11$

007352                          2$:

                                ;*      MOV     @DROA1,$GDDAT   ;/READ DEVICE REG DROA1,PUT DATA IN $GDDAT.

                                ;*      MOV     @DRIA1,$GDDAT   ;/READ DEVICE REG DRIA1,PUT DATA IN $GDDAT.
007372  012737  040100  001124          MOV     #BIT6!BIT14,$GDDAT ;LOAD WRITEABLE BITS.

                                ;*      MOV     $GDDAT,@DRCR1   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR1

                                ;*      MOV     @DRCR1,$BDDAT   ;/READ DEVICE REG DRCR1,PUT DATA IN $BDDAT.
007420  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID BITS SET?
007426  001403                          BEQ     3$

007430  104012                          ERROR   12              ;DR11K #1 CSR BIT(S) FAILED.
007432  000137  007762                  JMP     11$

007436  005037  001124          3$:     CLR     $GDDAT          ;TRY CLEARING CSR.

                                ;*      MOV     $GDDAT,@DRCR1   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR1

                                ;*      MOV     @DRCR1,$BDDAT   ;/READ DEVICE REG DRCR1,PUT DATA IN $BDDAT.
007462  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
007466  001403                          BEQ     4$              ;YES-NEXT TEST.
```

```
2299
2300  007470  104012                        ERROR   12              ;DR11K #1 CSR FAILED TO CLEAR.
2301  007472  000137  007762                JMP     11$
2302
2303  007476  032737  000004  001564  4$:   BIT     #BIT2,SR2       ;DOES THIS DR11 HAVE A LOOP BACK
2304                                                                ;CABLE CONNECTED TO IT?
2305  007504  001526                        BEQ     TST11           ;;
2306                                                                ;IF SR2 BIT2=1 THEN LOOP BACK.
2307
2308  007506  012737  052525  001124        MOV     #052525,$GDDAT  ;SET FIRST PATTERN
2309
2310                            ;*          MOV     $GDDAT,@DROA1   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA1
2311
2312                            ;*          MOV     @DRCR1,$BDDAT   ;/READ DEVICE REG DRCR1,PUT DATA IN $BDDAT.
2313  007534  012737  000200  001124        MOV     #BIT7,$GDDAT    ;EXPECT FLAGS TO SET.
2314  007542  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;DID THEY?
2315  007550  001402                        BEQ     5$              ;YES-CON'T.
2316
2317  007552  104012                        ERROR   12              ;DR11K #1 FLAG(S) FAILED TO SET
2318                                                                ;WHEN DATA XFERRED. IS THE OUTPUT
2319                                                                ;REALLY CALLED BACK TO THE INPUT?
2320  007554  000502                        BR      TST11           ;;
2321
2322  007556  012737  052525  001124  5$:   MOV     #052525,$GDDAT  ;PATTERN THAT SHOULD HAVE XFERRED.
2323
2324                            ;*          MOV     @DRIA1,$BDDAT   ;/READ DEVICE REG DRIA1,PUT DATA IN $BDDAT.
2325  007574  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;DATA SENT=DATA RECEIVED?
2326  007602  001402                        BEQ     6$
2327
2328  007604  104012                        ERROR   12              ;DR11K #1 DATA XFERR ERROR.
2329  007606  000465                        BR      TST11           ;;
2330
2331  007610                          6$:
2332
2333                            ;*          MOV     @DRCR1,$BDDAT   ;/READ DEVICE REG DRCR1,PUT DATA IN $BDDAT.
2334  007620  005737  001126                TST     $BDDAT          ;DID "OUTPUT FLAG" SET.
2335  007624  100402                        BMI     7$              ;SHOULD HAVE.
2336
2337  007626  104012                        ERROR   12              ;DR11K #1, "OUTPUT FLAG" FAILED TO SET.
2338  007630  000454                        BR      TST11           ;;
2339
2340  007632                          7$:
2341
2342                            ;*          MOV     $GDDAT,@DRIA1   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA1
2343  007642  012737  125252  001124        MOV     #125252,$GDDAT  ;NEW PATTERN
2344
2345                            ;*          MOV     $GDDAT,@DROA1   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA1
2346
2347                            ;*          MOV     @DRIA1,$BDDAT   ;/READ DEVICE REG DRIA1,PUT DATA IN $BDDAT.
2348  007670  023737  001126  001124        CMP     $BDDAT,$GDDAT   ;PATTERN XFERR OK?
2349  007676  001402                        BEQ     8$
2350
2351  007700  104012                        ERROR   12              ;DR11K #1 DATA XFERR ERROR
2352  007702  000427                        BR      TST11           ;;
```

```
2353  007704                          8$:
2354
2355                                  ;*      MOV     $GDDAT,@DRIA1   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA1
2356
2357  007714  005037  001124                CLR     $GDDAT          ;NOW XFERR ZERO PATTERN.
2358
2359                                  ;*      MOV     $GDDAT,@DROA1   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA1
2360
2361                                  ;*      MOV     @DRIA1,$BDDAT   ;/READ DEVICE REG DRIA1,PUT DATA IN $BDDAT.
2362  007740  005737  001126                TST     $BDDAT          ;DID ZERO PATTERN GET XFERRED?
2363  007744  001402                        BEQ     9$
2364
2365  007746  104012                        ERROR   12              ;DR11K #1 DATA XFERR ERROR
2366  007750  000404                        BR      TST11           ;;
2367
2368  007752                          9$:
2369
2370                                  ;*      MOV     $GDDAT,@DRCR1   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR1
2371
2372  007762                          11$:
2373
2374                                  ;;*****************************************************************
2375                                  ;*TEST 11       *TEST THE DR11K OPTION,#2, IF "SR1" BIT 5=1(SET)
2376                                  ;*
2377                                  ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
2378                                  ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
2379                                  ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
2380                                  ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
2381                                  ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
2382                                  ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
2383                                  ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
2384                                  ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
2385                                  ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
2386                                  ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
2387                                  ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
2388                                  ;*
2389                                  ;*IN THIS TEST WE WILL CHECK OUT DR11 #2 IF BIT 5 OF SR1=1
2390                                  ;*(SET). ALSO, DATA LOOP BACK (OUTPUT DATA BACK TO INPUT) WILL BE CHECKED
2391                                  ;*IF "SR2" BIT 5 =1 (SET). FOR THIS, YOU MUST HAVE INSTALLED A LOOP-BACK
2392                                  ;*CABLE.
2393                                  ;;*****************************************************************
2394  007762  000004                 TST11:  SCOPE
2395  007764  032737  000040  001562         BIT     #BIT5,SR1
2396  007772  001002                        BNE     10$             ;/-TDRT-
2397  007774  000137  010470                JMP     11$
2398  010000                          10$:
2399  010000  005037  001514                CLR     .DVLS
2400  010004  005037  001124                CLR     $GDDAT          ;/SET TO XFERR ZEROS.
2401
2402                                  ;*      MOV     $GDDAT,@DRCR2   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR2
2403  010020  005737  037532                TST     $AERR           ;=1 IF DEVICE NOT PRESENT,
2404  010024  001403                        BEQ     1$              ;IF 0, PRESENT, (CON'T).
2405
2406  010026  104012                        ERROR   12              ;DR11K #2 DID NOT RESPOND WHEN
```

```
                                                                        ;ADDRESSED. "SR1" BIT 5 SELECTED
                                                                        ;THIS OPTION. ADDRESS "DRCR"2
                                                                        ;CONTAINS ITS ADDRESS

010030  000137  010470                 JMP     11$

010034                          1$:

                                ;*      MOV     @DRCR2,$BDDAT   ;/READ DEVICE REG DRCR2,PUT DATA IN $BDDAT.
010044  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
010050  001403                          BEQ     2$

010052  104012                          ERROR   12              ;DR11K #2 CSR FAILED TO CLEAR.
010054  000137  010470                  JMP     11$

010060                          2$:

                                ;*      MOV     @DROA2,$GDDAT   ;/READ DEVICE REG DROA2,PUT DATA IN $GDDAT.

                                ;*      MOV     @DRIA2,$GDDAT   ;/READ DEVICE REG DRIA2,PUT DATA IN $GDDAT.
010100  012737  040100  001124          MOV     #BIT6!BIT14,$GDDAT ;LOAD WRITEABLE BITS.

                                ;*      MOV     $GDDAT,@DRCR2   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR2

                                ;*      MOV     @DRCR2,$BDDAT   ;/READ DEVICE REG DRCR2,PUT DATA IN $BDDAT.
010126  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID BITS SET?
010134  001403                          BEQ     3$

010136  104012                          ERROR   12              ;DR11K #2 CSR BIT(S) FAILED.
010140  000137  010470                  JMP     11$

010144  005037  001124          3$:     CLR     $GDDAT          ;TRY CLEARING CSR.

                                ;*      MOV     $GDDAT,@DRCR2   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR2

                                ;*      MOV     @DRCR2,$BDDAT   ;/READ DEVICE REG DRCR2,PUT DATA IN $BDDAT.
010170  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
010174  001403                          BEQ     4$              ;YES-NEXT TEST.

010176  104012                          ERROR   12              ;DR11K #2 CSR FAILED TO CLEAR.
010200  000137  010470                  JMP     11$

010204  032737  000040  001564  4$:     BIT     #BIT5,SR2       ;DOES THIS DR11 HAVE A LOOP BACK
                                                                ;CABLE CONNECTED TO IT?
010212  001526                          BEQ     TST12           ;;
                                                                ;IF SR2 BIT5=1 THEN LOOP BACK.

010214  012737  052525  001124          MOV     #052525,$GDDAT  ;SET FIRST PATTERN

                                ;*      MOV     $GDDAT,@DROA2   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA2

                                ;*      MOV     @DRCR2,$BDDAT   ;/READ DEVICE REG DRCR2,PUT DATA IN $BDDAT.
010242  012737  000200  001124          MOV     #BIT7,$GDDAT    ;EXPECT FLAGS TO SET.
010250  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID THEY?
```

```
2461  010256  001402                          BEQ     5$              ;YES-CON'T.
2462
2463  010260  104012                          ERROR   12              ;DR11K #2 FLAG(S) FAILED TO SET
2464                                                                  ;WHEN DATA XFERRED. IS THE OUTPUT
2465                                                                  ;REALLY CALLED BACK TO THE INPUT?
2466  010262  000502                          BR      TST12           ;;
2467
2468  010264  012737  052525  001124  5$:     MOV     #052525,$GDDAT  ;PATTERN THAT SHOULD HAVE XFERRED.
2469
2470                           ;*             MOV     @DRIA2,$BDDAT   ;/READ DEVICE REG DRIA2,PUT DATA IN $BDDAT.
2471  010302  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DATA SENT=DATA RECEIVED?
2472  010310  001402                          BEQ     6$
2473
2474  010312  104012                          ERROR   12              ;DR11K #2 DATA XFERR ERROR.
2475  010314  000465                          BR      TST12           ;;
2476
2477  010316                          6$:
2478
2479                           ;*             MOV     @DRCR2,$BDDAT   ;/READ DEVICE REG DRCR2,PUT DATA IN $BDDAT.
2480  010326  005737  001126                  TST     $BDDAT          ;DID "OUTPUT FLAG" SET.
2481  010332  100402                          BMI     7$              ;SHOULD HAVE.
2482
2483  010334  104012                          ERROR   12              ;DR11K #2, "OUTPUT FLAG" FAILED TO SET.
2484  010336  000454                          BR      TST12           ;;
2485
2486  010340                          7$:
2487
2488                           ;*             MOV     $GDDAT,@DRIA2   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA2
2489  010350  012737  125252  001124          MOV     #125252,$GDDAT  ;NEW PATTERN
2490
2491                           ;*             MOV     $GDDAT,@DROA2   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA2
2492
2493                           ;*             MOV     @DRIA2,$BDDAT   ;/READ DEVICE REG DRIA2,PUT DATA IN $BDDAT.
2494  010376  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;PATTERN XFERR OK?
2495  010404  001402                          BEQ     8$
2496
2497  010406  104012                          ERROR   12              ;DR11K #2 DATA XFERR ERROR
2498  010410  000427                          BR      TST12           ;;
2499  010412                          8$:
2500
2501                           ;*             MOV     $GDDAT,@DRIA2   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA2
2502
2503  010422  005037  001124                  CLR     $GDDAT          ;NOW XFERR ZERO PATTERN.
2504
2505                           ;*             MOV     $GDDAT,@DROA2   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA2
2506
2507                           ;*             MOV     @DRIA2,$BDDAT   ;/READ DEVICE REG DRIA2,PUT DATA IN $BDDAT.
2508  010446  005737  001126                  TST     $BDDAT          ;DID ZERO PATTERN GET XFERRED?
2509  010452  001402                          BEQ     9$
2510
2511  010454  104012                          ERROR   12              ;DR11K #2 DATA XFERR ERROR
2512  010456  000404                          BR      TST12           ;;
2513
2514  010460                          9$:
```

```
                                        ;*      MOV     $GDDAT,@DRCR2   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR2
                                        11$:
                                        ;;*****************************************************************
                                        ;*TEST 12       *TEST THE DR11K OPTION,#3, IF "SR1" BIT 7=1(SET)
                                        ;*
                                        ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
                                        ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
                                        ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
                                        ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
                                        ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
                                        ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
                                        ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
                                        ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
                                        ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
                                        ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
                                        ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK OUT DR11 #3 IF BIT 7 OF SR1=1
                                        ;*(SET). ALSO, DATA LOOP BACK (OUTPUT DATA BACK TO INPUT) WILL BE CHECKED
                                        ;*IF "SR2" BIT 7 =1 (SET). FOR THIS, YOU MUST HAVE INSTALLED A LOOP-BACK
                                        ;*CABLE.
                                        ;;*****************************************************************
010470  000004                        TST12:  SCOPE
010472  032737  000200  001562               BIT     #BIT7,SR1
010500  001002                               BNE     10$             ;/-TDR+-
010502  000137  011176                       JMP     11$
010506                                  10$:
010506  005037  001514                       CLR     .DVLS
010512  005037  001124                       CLR     $GDDAT          ;/SET TO XFERR ZEROS.
                                        ;*      MOV     $GDDAT,@DRCR3   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR3
010526  005737  037532                       TST     $AERR           ;=1 IF DEVICE NOT PRESENT,
010532  001403                               BEQ     1$              ;IF 0, PRESENT, (CON'T).

010534  104012                               ERROR   12              ;DR11K #3 DID NOT RESPOND WHEN
                                                                    ;ADDRESSED. "SR1" BIT 7 SELECTED
                                                                    ;THIS OPTION. ADDRESS "DRCR"3
                                                                    ;CONTAINS ITS ADDRESS

010536  000137  011176                       JMP     11$

010542                                  1$:
                                        ;*      MOV     @DRCR3,$BDDAT   ;/READ DEVICE REG DRCR3,PUT DATA IN $BDDAT.
010552  005737  001126                       TST     $BDDAT          ;DID CSR CLEAR?
010556  001403                               BEQ     2$

010560  104012                               ERROR   12              ;DR11K #3 CSR FAILED TO CLEAR.
010562  000137  011176                       JMP     11$

010566                                  2$:
```

```
2569
2570                                    ;*      MOV     @DROA3,$GDDAT   ;/READ DEVICE REG DROA3,PUT DATA IN $GDDAT.
2571
2572                                    ;*      MOV     @DRIA3,$GDDAT   ;/READ DEVICE REG DRIA3,PUT DATA IN $GDDAT.
2573  010606  012737  040100  001124          MOV     #BIT6!BIT14,$GDDAT ;LOAD WRITEABLE BITS.
2574
2575                                    ;*      MOV     $GDDAT,@DRCR3   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR3
2576
2577                                    ;*      MOV     @DRCR3,$BDDAT   ;/READ DEVICE REG DRCR3,PUT DATA IN $BDDAT.
2578  010634  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID BITS SET?
2579  010642  001403                          BEQ     3$
2580
2581  010644  104012                          ERROR   12              ;DR11K #3 CSR BIT(S) FAILED.
2582  010646  000137  011176                  JMP     11$
2583
2584  010652  005037  001124          3$:     CLR     $GDDAT          ;TRY CLEARING CSR.
2585
2586                                    ;*      MOV     $GDDAT,@DRCR3   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR3
2587
2588                                    ;*      MOV     @DRCR3,$BDDAT   ;/READ DEVICE REG DRCR3,PUT DATA IN $BDDAT.
2589  010676  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
2590  010702  001403                          BEQ     4$              ;YES-NEXT TEST.
2591
2592  010704  104012                          ERROR   12              ;DR11K #3 CSR FAILED TO CLEAR.
2593  010706  000137  011176                  JMP     11$
2594
2595  010712  032737  000200  001564  4$:     BIT     #BIT7,SR2       ;DOES THIS DR11 HAVE A LOOP BACK
2596                                                                  ;CABLE CONNECTED TO IT?
2597  010720  001526                          BEQ     TST13           ;;
2598                                                                  ;IF SR2 BIT7=1 THEN LOOP BACK.
2599
2600  010722  012737  052525  001124          MOV     #052525,$GDDAT  ;SET FIRST PATTERN
2601
2602                                    ;*      MOV     $GDDAT,@DROA3   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA3
2603
2604                                    ;*      MOV     @DRCR3,$BDDAT   ;/READ DEVICE REG DRCR3,PUT DATA IN $BDDAT.
2605  010750  012737  000200  001124          MOV     #BIT7,$GDDAT    ;EXPECT FLAGS TO SET.
2606  010756  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID THEY?
2607  010764  001402                          BEQ     5$              ;YES-CON'T.
2608
2609  010766  104012                          ERROR   12              ;DR11K #3 FLAG(S) FAILED TO SET
2610                                                                  ;WHEN DATA XFERRED. IS THE OUTPUT
2611                                                                  ;REALLY CALLED BACK TO THE INPUT?
2612  010770  000502                          BR      TST13           ;;
2613
2614  010772  012737  052525  001124  5$:     MOV     #052525,$GDDAT  ;PATTERN THAT SHOULD HAVE XFERRED.
2615
2616                                    ;*      MOV     @DRIA3,$BDDAT   ;/READ DEVICE REG DRIA3,PUT DATA IN $BDDAT.
2617  011010  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DATA SENT=DATA RECEIVED?
2618  011016  001402                          BEQ     6$
2619
2620  011020  104012                          ERROR   12              ;DR11K #3 DATA XFERR ERROR.
2621  011022  000465                          BR      TST13           ;;
2622
```

```
011024                          6$:
                                ;*      MOV     @DRCR3,$BDDAT   ;/READ DEVICE REG DRCR3,PUT DATA IN $BDDAT.
011034  005737  001126          TST     $BDDAT                  ;DID "OUTPUT FLAG" SET.
011040  100402                  BMI     7$                      ;SHOULD HAVE.

011042  104012                  ERROR   12                      ;DR11K #3, "OUTPUT FLAG" FAILED TO SET.
011044  000454                  BR      TST13                   ;;

011046                          7$:
                                ;*      MOV     $GDDAT,@DRIA3   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA3
011056  012737  125252  001124  MOV     #125252,$GDDAT          ;NEW PATTERN

                                ;*      MOV     $GDDAT,@DROA3   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA3

                                ;*      MOV     @DRIA3,$BDDAT   ;/READ DEVICE REG DRIA3,PUT DATA IN $BDDAT.
011104  023737  001126  001124  CMP     $BDDAT,$GDDAT           ;PATTERN XFERR OK?
011112  001402                  BEQ     8$

011114  104012                  ERROR   12                      ;DR11K #3 DATA XFERR ERROR
011116  000427                  BR      TST13                   ;;
011120                          8$:
                                ;*      MOV     $GDDAT,@DRIA3   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA3

011130  005037  001124          CLR     $GDDAT                  ;NOW XFERR ZERO PATTERN.

                                ;*      MOV     $GDDAT,@DROA3   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA3

                                ;*      MOV     @DRIA3,$BDDAT   ;/READ DEVICE REG DRIA3,PUT DATA IN $BDDAT.
011154  005737  001126          TST     $BDDAT                  ;DID ZERO PATTERN GET XFERRED?
011160  001402                  BEQ     9$

011162  104012                  ERROR   12                      ;DR11K #3 DATA XFERR ERROR
011164  000404                  BR      TST13                   ;;

011166                          9$:
                                ;*      MOV     $GDDAT,@DRCR3   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR3

011176                          11$:

                                ;;*****************************************************************
                                ;*TEST 13        *TEST THE DR11K OPTION,#4, IF "SR1" BIT 8=1(SET)
                                ;*
                                ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
                                ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
                                ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
                                ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
                                ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
                                ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
                                ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
                                ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
```

```
                                        ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
                                        ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
                                        ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK OUT DR11 #4 IF BIT 8 OF SR1=1
                                        ;*(SET). ALSO, DATA LOOP BACK (OUTPUT DATA BACK TO INPUT) WILL BE CHECKED
                                        ;*IF "SR2" BIT 8 =1 (SET). FOR THIS, YOU MUST HAVE INSTALLED A LOOP-BACK
                                        ;*CABLE.
                                        ;;*****************************************************************
011176  000004                        TST13:  SCOPE
011200  032737  000400  001562                BIT     #BIT8,SR1
011206  001002                              BNE     10$             ;/-TDRT-
011210  000137  011704                      JMP     11$
011214                                  10$:
011214  005037  001514                      CLR     .DVLS
011220  005037  001124                      CLR     $GDDAT          ;/SET TO XFERR ZEROS.

                                        ;*      MOV     $GDDAT,@DRCR4   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR4
011234  005737  037532                      TST     $AERR           ;=1 IF DEVICE NOT PRESENT,
011240  001403                              BEQ     1$              ;IF 0, PRESENT, (CON'T).

011242  104012                              ERROR   12              ;DR11K #4 DID NOT RESPOND WHEN
                                                                    ;ADDRESSED. "SR1" BIT 8 SELECTED
                                                                    ;THIS OPTION. ADDRESS "DRCR"4
                                                                    ;CONTAINS ITS ADDRESS

011244  000137  011704                      JMP     11$

011250                                  1$:

                                        ;*      MOV     @DRCR4,$BDDAT   ;/READ DEVICE REG DRCR4,PUT DATA IN $BDDAT.
011260  005737  001126                      TST     $BDDAT          ;DID CSR CLEAR?
011264  001403                              BEQ     2$

011266  104012                              ERROR   12              ;DR11K #4 CSR FAILED TO CLEAR.
011270  000137  011704                      JMP     11$

011274                                  2$:

                                        ;*      MOV     @DROA4,$GDDAT   ;/READ DEVICE REG DROA4,PUT DATA IN $GDDAT.

                                        ;*      MOV     @DRIA4,$GDDAT   ;/READ DEVICE REG DRIA4,PUT DATA IN $GDDAT.
011314  012737  040100  001124              MOV     #BIT6!BIT14,$GDDAT ;LOAD WRITEABLE BITS.

                                        ;*      MOV     $GDDAT,@DRCR4   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR4

                                        ;*      MOV     @DRCR4,$BDDAT   ;/READ DEVICE REG DRCR4,PUT DATA IN $BDDAT.
011342  023737  001124  001126              CMP     $GDDAT,$BDDAT   ;DID BITS SET?
011350  001403                              BEQ     3$

011352  104012                              ERROR   12              ;DR11K #4 CSR BIT(S) FAILED.
011354  000137  011704                      JMP     11$

011360  005037  001124          3$:         CLR     $GDDAT          ;TRY CLEARING CSR.
```

```
2731
2732                                    ;*      MOV     $GDDAT,@DRCR4   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR4
2733
2734                                    ;*      MOV     @DRCR4,$BDDAT   ;/READ DEVICE REG DRCR4,PUT DATA IN $BDDAT.
2735  011404  005737  001126                      TST     $BDDAT          ;DID CSR CLEAR?
2736  011410  001403                               BEQ     4$              ;YES-NEXT TEST.
2737
2738  011412  104012                               ERROR   12              ;DR11K #4 CSR FAILED TO CLEAR.
2739  011414  000137  011704                       JMP     11$
2740
2741  011420  032737  000400  001564  4$:     BIT     #BIT8,SR2       ;DOES THIS DR11 HAVE A LOOP BACK
2742                                                                    ;CABLE CONNECTED TO IT?
2743  011426  001526                               BEQ     TST14           ;;
2744                                                                    ;IF SR2 BIT8=1 THEN LOOP BACK.
2745
2746  011430  012737  052525  001124               MOV     #052525,$GDDAT  ;SET FIRST PATTERN
2747
2748                                    ;*      MOV     $GDDAT,@DROA4   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA4
2749
2750                                    ;*      MOV     @DRCR4,$BDDAT   ;/READ DEVICE REG DRCR4,PUT DATA IN $BDDAT.
2751  011456  012737  000200  001124               MOV     #BIT7,$GDDAT    ;EXPECT FLAGS TO SET.
2752  011464  023737  001126  001124               CMP     $BDDAT,$GDDAT   ;DID THEY?
2753  011472  001402                               BEQ     5$              ;YES-CON'T.
2754
2755  011474  104012                               ERROR   12              ;DR11K #4 FLAG(S) FAILED TO SET
2756                                                                    ;WHEN DATA XFERRED. IS THE OUTPUT
2757                                                                    ;REALLY CALLED BACK TO THE INPUT?
2758  011476  000502                               BR      TST14           ;;
2759
2760  011500  012737  052525  001124  5$:     MOV     #052525,$GDDAT  ;PATTERN THAT SHOULD HAVE XFERRED.
2761
2762                                    ;*      MOV     @DRIA4,$BDDAT   ;/READ DEVICE REG DRIA4,PUT DATA IN $BDDAT.
2763  011516  023737  001126  001124               CMP     $BDDAT,$GDDAT   ;DATA SENT=DATA RECEIVED?
2764  011524  001402                               BEQ     6$
2765
2766  011526  104012                               ERROR   12              ;DR11K #4 DATA XFERR ERROR.
2767  011530  000465                               BR      TST14           ;;
2768
2769  011532                          6$:
2770
2771                                    ;*      MOV     @DRCR4,$BDDAT   ;/READ DEVICE REG DRCR4,PUT DATA IN $BDDAT.
2772  011542  005737  001126                      TST     $BDDAT          ;DID "OUTPUT FLAG" SET.
2773  011546  100402                               BMI     7$              ;SHOULD HAVE.
2774
2775  011550  104012                               ERROR   12              ;DR11K #4, "OUTPUT FLAG" FAILED TO SET.
2776  011552  000454                               BR      TST14           ;;
2777
2778  011554                          7$:
2779
2780                                    ;*      MOV     $GDDAT,@DRIA4   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA4
2781  011564  012737  125252  001124               MOV     #125252,$GDDAT  ;NEW PATTERN
2782
2783                                    ;*      MOV     $GDDAT,@DROA4   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA4
2784
```

```
                                        ;*      MOV     @DRIA4,$BDDAT   ;/READ DEVICE REG DRIA4,PUT DATA IN $BDDAT.
011612  023737  001126  001124                  CMP     $BDDAT,$GDDAT   ;PATTERN XFERR OK?
011620  001402                                  BEQ     8$
                                                ERROR   12              ;DR11K #4 DATA XFERR ERROR
011622  104012
011624  000427                                  BR      TST14           ;;
011626                                  8$:
                                        ;*      MOV     $GDDAT,@DRIA4   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA4
011636  005037  001124                          CLR     $GDDAT          ;NOW XFERR ZERO PATTERN.
                                        ;*      MOV     $GDDAT,@DROA4   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA4
                                        ;*      MOV     @DRIA4,$BDDAT   ;/READ DEVICE REG DRIA4,PUT DATA IN $BDDAT.
011662  005737  001126                          TST     $BDDAT          ;DID ZERO PATTERN GET XFERRED?
011666  001402                                  BEQ     9$
                                                ERROR   12              ;DR11K #4 DATA XFERR ERROR
011670  104012
011672  000404                                  BR      TST14           ;;
011674                                  9$:
                                        ;*      MOV     $GDDAT,@DRCR4   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR4
011704                                  11$:
                                        ;;*****************************************************************
                                        ;*TEST 14       *TEST THE DR11K OPTION,#5, IF "SR1" BIT 9=1(SET)
                                        ;*
                                        ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
                                        ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
                                        ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
                                        ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
                                        ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
                                        ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
                                        ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
                                        ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
                                        ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
                                        ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
                                        ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
                                        ;*
                                        ;*IN THIS TEST WE WILL CHECK OUT DR11 #5 IF BIT 9 OF SR1=1
                                        ;*(SET). ALSO, DATA LOOP BACK (OUTPUT DATA BACK TO INPUT) WILL BE CHECKED
                                        ;*IF "SR2" BIT 9 =1 (SET). FOR THIS, YOU MUST HAVE INSTALLED A LOOP-BACK
                                        ;*CABLE.
                                        ;*****************************************************************
011704  000004                  TST14:  SCOPE
011706  032737  001000  001562          BIT     #BIT9,SR1
011714  001002                          BNE     10$             ;/-TDRT-
011716  000137  012412                  JMP     11$
011722                          10$:
011722  005037  001514                  CLR     .DVLS
011726  005037  001124                  CLR     $GDDAT          ;/SET TO XFERR ZEROS.
```

```
2839
2840
2841  011742  005737  037532          ;*      MOV     $GDDAT,@DRCR5   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR5
2842  011746  001403                          TST     $AERR           ;=1 IF DEVICE NOT PRESENT,
2843                                           BEQ     1$              ;IF 0, PRESENT, (CON'T).
2844  011750  104012                          ERROR   12              ;DR11K #5 DID NOT RESPOND WHEN
2845                                                                   ;ADDRESSED. "SR1" BIT 9 SELECTED
2846                                                                   ;THIS OPTION. ADDRESS "DRCR"S
2847                                                                   ;CONTAINS ITS ADDRESS
2848
2849  011752  000137  012412                  JMP     11$
2850
2851  011756                          1$:
2852
2853                                  ;*      MOV     @DRCR5,$BDDAT   ;/READ DEVICE REG DRCR5,PUT DATA IN $BDDAT.
2854  011766  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
2855  011772  001403                          BEQ     2$
2856
2857  011774  104012                          ERROR   12              ;DR11K #5 CSR FAILED TO CLEAR.
2858  011776  000137  012412                  JMP     11$
2859
2860  012002                          2$:
2861
2862                                  ;*      MOV     @DROA5,$GDDAT   ;/READ DEVICE REG DROA5,PUT DATA IN $GDDAT.
2863
2864                                  ;*      MOV     @DRIA5,$GDDAT   ;/READ DEVICE REG DRIA5,PUT DATA IN $GDDAT.
2865  012022  012737  040100  001124          MOV     #BIT6!BIT14,$GDDAT ;LOAD WRITEABLE BITS.
2866
2867                                  ;*      MOV     $GDDAT,@DRCR5   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR5
2868
2869                                  ;*      MOV     @DRCR5,$BDDAT   ;/READ DEVICE REG DRCR5,PUT DATA IN $BDDAT.
2870  012050  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID BITS SET?
2871  012056  001403                          BEQ     3$
2872
2873  012060  104012                          ERROR   12              ;DR11K #5 CSR BIT(S) FAILED.
2874  012062  000137  012412                  JMP     11$
2875
2876  012066  005037  001124          3$:     CLR     $GDDAT          ;TRY CLEARING CSR.
2877
2878                                  ;*      MOV     $GDDAT,@DRCR5   ;/ PUT DATA FROM $GDDAT TO dEVICE REG DRCR5
2879
2880                                  ;*      MOV     @DRCR5,$BDDAT   ;/READ DEVICE REG DRCR5,PUT DATA IN $BDDAT.
2881  012112  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
2882  012116  001403                          BEQ     4$              ;YES-NEXT TEST.
2883
2884  012120  104012                          ERROR   12              ;DR11K #5 CSR FAILED TO CLEAR.
2885  012122  000137  012412                  JMP     11$
2886
2887  012126  032737  001000  001564  4$:     BIT     #BIT9,SR2       ;DOES THIS DR11 HAVE A LOOP BACK
2888                                                                   ;CABLE CONNECTED TO IT?
2889  012134  001526                          BEQ     TST15           ;;
2890                                                                   ;IF SR2 BIT9=1 THEN LOOP BACK.
2891
2892  012136  012737  052525  001124          MOV     #052525,$GDDAT  ;SET FIRST PATTERN
```

```
2893
2894                                    ;*      MOV     $GDDAT,@DROA5   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA5
2895
2896                                    ;*      MOV     @DRCR5,$BDDAT   ;/READ DEVICE REG DRCR5,PUT DATA IN $BDDAT.
2897  012164  012737  000200  001124          MOV     #BIT7,$GDDAT    ;EXPECT FLAGS TO SET.
2898  012172  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DID THEY?
2899  012200  001402                          BEQ     5$              ;YES-CON'T.
2900
2901  012202  104012                          ERROR   12              ;DR11K #5 FLAG(S) FAILED TO sET
2902                                                                  ;WHEN DATA XFERRED. IS THE OUTPUT
2903                                                                  ;REALLY CALLED BACK TO THE INPUT?
2904  012204  000502                          BR      TST15           ;;
2905
2906  012206  012737  052525  001124  5$:     MOV     #052525,$GDDAT  ;PATTERN THAT SHOULD HAVE XFERRED.
2907
2908                                    ;*      MOV     @DRIA5,$BDDAT   ;/READ DEVICE REG DRIA5,PUT DATA IN $BDDAT.
2909  012224  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;DATA SENT=DATA RECEIVED?
2910  012232  001402                          BEQ     6$
2911
2912  012234  104012                          ERROR   12              ;DR11K #5 DATA XFERR ERROR.
2913  012236  000465                          BR      TST15           ;;
2914
2915  012240                          6$:
2916
2917                                    ;*      MOV     @DRCR5,$BDDAT   ;/READ DEVICE REG DRCR5,PUT DATA IN $BDDAT.
2918  012250  005737  001126                  TST     $BDDAT          ;DID "OUTPUT FLAG" SET.
2919  012254  100402                          BMI     7$              ;SHOULD HAVE.
2920
2921  012256  104012                          ERROR   12              ;DR11K #5, "OUTPUT FLAG" FAILED TO SET.
2922  012260  000454                          BR      TST15           ;;
2923
2924  012262                          7$:
2925
2926                                    ;*      MOV     $GDDAT,@DRIA5   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA5
2927  012272  012737  125252  001124          MOV     #125252,$GDDAT  ;NEW PATTERN
2928
2929                                    ;*      MOV     $GDDAT,@DROA5   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA5
2930
2931                                    ;*      MOV     @DRIA5,$BDDAT   ;/READ DEVICE REG DRIA5,PUT DATA IN $BDDAT.
2932  012320  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;PATTERN XFERR OK?
2933  012326  001402                          BEQ     8$
2934
2935  012330  104012                          ERROR   12              ;DR11K #5 DATA XFERR ERROR
2936  012332  000427                          BR      TST15           ;;
2937  012334                          8$:
2938
2939                                    ;*      MOV     $GDDAT,@DRIA5   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRIA5
2940
2941  012344  005037  001124                  CLR     $GDDAT          ;NOW XFERR ZERO PATTERN.
2942
2943                                    ;*      MOV     $GDDAT,@DROA5   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DROA5
2944
2945                                    ;*      MOV     @DRIA5,$BDDAT   ;/READ DEVICE REG DRIA5,PUT DATA IN $BDDAT.
2946  012370  005737  001126                  TST     $BDDAT          ;DID ZERO PATTERN GET XFERRED?
```

```
2947  012374  001402                          BEQ     9$
2948
2949  012376  104012                          ERROR   12              ;DR11K #5 DATA XFERR ERROR
2950  012400  000404                          BR      TST15           ;;
2951
2952  012402                          9$:
2953
2954                                  ;*      MOV     $GDDAT,@DRCR5   ;/ PUT DATA FROM $GDDAT TO DEVICE REG DRCR5
2955
2956  012412                          11$:
2957
2958
2959                                  ;;*****************************************************************
2960                                  ;*TEST 15       *TEST THE AA-11 OPTION, IF "SR1" BIT 3=1 (SET)
2961                                  ;*
2962                                  ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
2963                                  ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
2964                                  ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
2965                                  ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
2966                                  ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
2967                                  ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
2968                                  ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
2969                                  ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
2970                                  ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
2971                                  ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
2972                                  ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
2973                                  ;*
2974                                  ;*IN THIS TEST WE WILL CHECK OUT THE AA-11 OPTION,
2975                                  ;*IF BIT 3 OF "SR1" IS SET.
2976                                  ;*
2977                                  ;*****************************************************************
2978  012412  000004                  TST15:  SCOPE
2979
2980  012414  032737  000010  001562          BIT     #BIT3,SR1       ;IS AA-11 SELECTED
2981  012422  001002                          BNE     20$             ;IF =0, NO, SKIP THIS TEST.
2982  012424  000137  013442                  JMP     19$
2983  012430                          20$:
2984  012430  005037  001124                  CLR     $GDDAT
2985
2986                                  ;*      MOV     $GDDAT,@AACSR   ;/ PUT DATA FROM $GDDAT TO DEVICE REG AACSR
2987  012444  005737  037532                  TST     $AERR           ;DID WE SUCCESSFULLY WRITE THE AA-11?
2988  012450  001403                          BEQ     1$              ;YES IF $AERR=0
2989
2990  012452  104014                          ERROR   14              ;FAILED TO ADDRESS AA-11. "SR1" BIT 3=1
2991                                                                  ;SELECTED THIS TEST. AA-11 ADDRESS IS
2992                                                                  ;IN LOC "AACSR"
2993  012454  000137  013442                  JMP     19$
2994
2995  012460                          1$:
2996
2997                                  ;*      MOV     @AACSR,$BDDAT   ;/READ DEVICE REG AACSR,PUT DATA IN $BDDAT.
2998  012470  042737  000200  001126          BIC     #BIT7,$BDDAT    ;CLEAR FLAG
2999  012476  001403                          BEQ     2$              ;DID CSR CLEAR?
3000
```

```
3001  012500  104014                          ERROR   14              ;CLEAR CSR FAILED (AA-11K)
3002  012502  000137  013442                  JMP     19$
3003
3004  012506  012737  005124  001124  2$:     MOV     #5124,$GDDAT    ;TRY WRITING THESE BITS
3005
3006                                  ;*      MOV     $GDDAT,@AACSR   ;/ PUT DATA FROM $GDDAT TO DEVICE REG AACSR
3007
3008                                  ;*      MOV     @AACSR,$BDDAT   ;/READ DEVICE REG AACSR,PUT DATA IN $BDDAT.
3009                                                                  ;READ THEM BACK
3010  012534  042737  000200  001126          BIC     #BIT7,$BDDAT    ;GET RID OF FLAG
3011  012542  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID PATTERN W/R OK?
3012  012550  001403                          BEQ     3$              ;YES-NEXT TEST.
3013
3014  012552  104014                          ERROR   14              ;AA11-K PATTERN 5124 FAILED TO W/R PROPERLY.
3015  012554  000137  013442                  JMP     19$
3016
3017  012560  012737  002012  001124  3$:     MOV     #2012,$GDDAT    ;TRY NEW PATTERN.
3018
3019                                  ;*      MOV     $GDDAT,@AACSR   ;/ PUT DATA FROM $GDDAT TO DEVICE REG AACSR
3020
3021                                  ;*      MOV     @AACSR,$BDDAT   ;/READ DEVICE REG AACSR,PUT DATA IN $BDDAT.
3022  012606  042737  000200  001126          BIC     #BIT7,$BDDAT    ;GET RID OF FLAG.
3023  012614  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID PATTERN XFERR OK?
3024  012622  001403                          BEQ     4$              ;YES-NEXT TEST.
3025
3026  012624  104014                          ERROR   14              ;AA-11K PATTERN 2012 FAILED TO W/R PROPERLY.
3027  012626  000137  013442                  JMP     19$
3028
3029  012632  005037  001124          4$:     CLR     $GDDAT          ;TRY TO CLEAR CSR.
3030
3031                                  ;*      MOV     $GDDAT,@AACSR   ;/ PUT DATA FROM $GDDAT TO DEVICE REG AACSR
3032
3033                                  ;*      MOV     @AACSR,$BDDAT   ;/READ DEVICE REG AACSR,PUT DATA IN $BDDAT.
3034  012656  042737  000200  001126          BIC     #BIT7,$BDDAT    ;CLEAR FLAG BIT.
3035  012664  001403                          BEQ     5$
3036
3037  012666  104014                          ERROR   14              ;AA-11K CSR FAILED TO CLEAR.
3038  012670  000137  013442                  JMP     19$
3039
3040  012674  012737  005252  001124  5$:     MOV     #5252,$GDDAT
3041
3042                                  ;*      MOV     $GDDAT,@DAC0    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC0
3043
3044                                  ;*      MOV     @DAC0,$BDDAT    ;/READ DEVICE REG DAC0,PUT DATA IN $BDDAT.
3045  012722  023737  001124  001126          CMP     $GDDAT,$BDDAT
3046  012730  001403                          BEQ     6$              ;OK?
3047
3048  012732  104014                          ERROR   14              ;AA-11K DAC0 FAILED TO SET PATTERN 5252.
3049  012734  000137  013442                  JMP     19$
3050
3051  012740                          6$:
3052
3053                                  ;*      MOV     $GDDAT,@DAC1    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC1
3054
```

```
3055                                        ;*      MOV     @DAC1,$BDDAT    ;/READ DEVICE REG DAC1,PUT DATA IN $BDDAT.
3056  012760  023737  001126  001124                CMP     $BDDAT,$GDDAT   ;OK
3057  012766  001403                              BEQ     7$              ;YES-
3058
3059  012770  104014                              ERROR   14              ;AA-11K DAC1 FAILED PATTERN 5252
3060  012772  000137  013442                      JMP     19$
3061
3062  012776                              7$:
3063
3064                                        ;*      MOV     $GDDAT,@DAC2    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC2
3065
3066                                        ;*      MOV     @DAC2,$BDDAT    ;/READ DEVICE REG DAC2,PUT DATA IN $BDDAT.
3067  013016  023737  001126  001124                CMP     $BDDAT,$GDDAT   ;OK?
3068  013024  001403                              BEQ     8$
3069
3070  013026  104014                              ERROR   14              ;AA-11K DAC2 FAILED PATTERN 5252
3071  013030  000137  013442                      JMP     19$
3072
3073  013034                              8$:
3074
3075                                        ;*      MOV     $GDDAT,@DAC3    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC3
3076
3077                                        ;*      MOV     @DAC3,$BDDAT    ;/READ DEVICE REG DAC3,PUT DATA IN $BDDAT.
3078  013054  023737  001126  001124                CMP     $BDDAT,$GDDAT   ;PATTERN OK?
3079  013062  001402                              BEQ     9$
3080
3081  013064  104014                              ERROR   14              ;AA-11K DAC3 FAILED PATTERN 5252
3082  013066  000565                              BR      TST16           ;;
3083
3084  013070  012737  002525  001124      9$:     MOV     #2525,$GDDAT
3085
3086                                        ;*      MOV     $GDDAT,@DAC0    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC0
3087
3088                                        ;*      MOV     @DAC0,$BDDAT    ;/READ DEVICE REG DAC0,PUT DATA IN $BDDAT.
3089  013116  023737  001124  001126                CMP     $GDDAT,$BDDAT
3090  013124  001402                              BEQ     10$             ;OK?
3091
3092  013126  104014                              ERROR   14              ;AA-11K DAC0 FAILED TO SET PATTER 2525
3093  013130  000544                              BR      TST16           ;;
3094
3095  013132                              10$:
3096
3097                                        ;*      MOV     $GDDAT,@DAC1    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC1
3098
3099                                        ;*      MOV     @DAC1,$BDDAT    ;/READ DEVICE REG DAC1,PUT DATA IN $BDDAT.
3100  013152  023737  001126  001124                CMP     $BDDAT,$GDDAT   ;OK?
3101  013160  001402                              BEQ     11$             ;YES-
3102
3103  013162  104014                              ERROR   14              ;AA-11K DAC1 FAILED PATTERN 2525
3104  013164  000526                              BR      TST16           ;;
3105
3106  013166                              11$:
3107
3108                                        ;*      MOV     $GDDAT,@DAC2    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC2
```

```
3109
3110                                        ;*      MOV     @DAC2,$BDDAT    ;/READ DEVICE REG DAC2,PUT DATA IN $BDDAT.
3111  013206  023737  001126  001124               CMP     $BDDAT,$GDDAT   ;OK?
3112  013214  001402                              BEQ     12$
3113
3114  013216  104014                              ERROR   14              ;AA-11K DAC2 FAILED PATTERN 2525.
3115  013220  000510                              BR      TST16           ;;
3116
3117  013222                              12$:
3118
3119                                        ;*      MOV     $GDDAT,@DAC3    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC3
3120
3121                                        ;*      MOV     @DAC3,$BDDAT    ;/READ DEVICE REG DAC3,PUT DATA IN $BDDAT.
3122  013242  023737  001126  001124               CMP     $BDDAT,$GDDAT   ;PATTERN OK?
3123  013250  001402                              BEQ     13$
3124
3125  013252  104014                              ERROR   14              ;AA-11K DAC3 FAILED PATTERN.
3126                                                                      ;2525
3127  013254  000472                              BR      TST16           ;;
3128
3129  013256  012737  000000  001124  13$:  MOV     #0,$GDDAT
3130
3131                                        ;*      MOV     $GDDAT,@DAC0    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC0
3132
3133                                        ;*      MOV     @DAC0,$BDDAT    ;/READ DEVICE REG DAC0,PUT DATA IN $BDDAT.
3134  013304  023737  001124  001126               CMP     $GDDAT,$BDDAT
3135  013312  001402                              BEQ     14$             ;OK?
3136
3137  013314  104014                              ERROR   14              ;AA-11K DAC0 FAILED TO SET PATTERN 0
3138  013316  000451                              BR      TST16           ;;
3139
3140  013320                              14$:
3141
3142                                        ;*      MOV     $GDDAT,@DAC1    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC1
3143
3144                                        ;*      MOV     @DAC1,$BDDAT    ;/READ DEVICE REG DAC1,PUT DATA IN $BDDAT.
3145  013340  023737  001126  001124               CMP     $BDDAT,$GDDAT   ;OK?
3146  013346  001402                              BEQ     15$             ;YES-
3147
3148  013350  104014                              ERROR   14              ;AA-11K DAC1 FAILED PATTERN 0
3149  013352  000433                              BR      TST16           ;;
3150
3151  013354                              15$:
3152
3153                                        ;*      MOV     $GDDAT,@DAC2    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC2
3154
3155                                        ;*      MOV     @DAC2,$BDDAT    ;/READ DEVICE REG DAC2,PUT DATA IN $BDDAT.
3156  013374  023737  001126  001124               CMP     $BDDAT,$GDDAT   ;OK?
3157  013402  001402                              BEQ     16$
3158
3159  013404  104014                              ERROR   14              ;AA-11K DAC2 FAILED PATTERN 0
3160  013406  000415                              BR      TST16           ;;
3161
3162  013410                              16$:
```

```
3163
3164                                    ;*      MOV     $GDDAT,@DAC3    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC3
3165
3166                                    ;*      MOV     @DAC3,$BDDAT    ;/READ DEVICE REG DAC3,PUT DATA IN $BDDAT.
3167  013430  023737  001126  001124             CMP     $BDDAT,$GDDAT   ;PATTERN OK?
3168
3169  013436  001401                             BEQ     TST16           ;;
3170  013440  104014                             ERROR   14              ;AA-11K DAC3 FAILED PATTERN 0.
3171
3172
3173
3174  013442                          19$:
3175                                  ;;**************************************************************
3176                                  ;*TEST 16        I/O DEVICE TEST-AR11 OPTION
3177                                  ;;**************************************************************
3178  013442  000004                 TST16:  SCOPE
3179
3180  013444  032737  002000  001562         BIT     #BIT10,SR1      ;IS THIS DEVICE SELECTED FOR TEST?
3181                                                                  ;NOTE: DEFAULT=YES.
3182
3183  013452  001002                         BNE     10$
3184  013454  000137  014012                 JMP     7$
3185  013460                          10$:
3186  013460  005037  001126                 CLR     $BDDAT
3187  013464  005037  001124                 CLR     $GDDAT
3188
3189                                  ;*      MOV     $BDDAT,@ARADS   ;/ PUT DATA FROM $BDDAT TO DEVICE REG ARADS
3190                                                                  ;OK-WHAT WE JUST DID IS ASKED THE
3191                                                                  ;SLAVE MICRO-CODE TO WRITE THE CONTENTS
3192                                                                  ;OF THE AR11 CSR TO ZEROES.
3193                                                                  ;IF THE AR11 FAILS TO rESPOND TO ITS
3194                                                                  ;ADDRESS-THE SLAVE WILL SEND US AN
3195                                                                  ;ERROR CODE THAT CAUSES US TO SET $AERR=1
3196                                                                  ;IN THE SUPPORT ROUTINES.
3197
3198  013500  005737  037532                 TST     $AERR           ;DEVICE PRESENT?
3199  013504  001403                         BEQ     11$             ;YES-CONTINUE
3200  013506  104015                         ERROR   15              ;AR11 FAILED TO RESPOND
3201  013510  000137  014012                 JMP     7$              ;TO ADDR.
3202  013514                          11$:
3203
3204                                  ;*      MOV     @ARADS,$BDDAT   ;/READ DEVICE REG ARADS,PUT DATA IN $BDDAT.
3205  013524  042737  000200  001126         BIC     #BIT7,$BDDAT    ;CLEAR OUT FLAG
3206  013532  005037  001124                 CLR     $GDDAT
3207  013536  005737  001126                 TST     $BDDAT          ;IS CSR ZERO?
3208  013542  001402                         BEQ     1$
3209
3210  013544  104015                         ERROR   15              ;FAILED TO ZERO AR11 A/D CSR
3211  013546  000521                         BR      TST17           ;;
3212  013550  012737  025100  001124  1$:    MOV     #BIT13!BIT11!BIT9!BIT6,$GDDAT ;NOW TRY A NEW BIT PATTERN
3213
3214
3215                                  ;*      MOV     $GDDAT,@ARADS   ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARADS
3216
```

```
3217                                        ;*      MOV     @ARADS,$BDDAT   ;/READ DEVICE REG ARADS,PUT DATA IN $BDDAT.
3218  013576  042737  000200  001126               BIC     #BIT7,$BDDAT    ;IGNORE FLAG
3219  013604  023737  001124  001126               CMP     $GDDAT,$BDDAT   ;BIT SET OK?
3220  013612  001402                               BEQ     2$
3221
3222  013614  104015                               ERROR   15              ;FAILED TO WRITE AR11 A/D CSR CORRECTLY
3223  013616  000475                               BR      TST17           ;;
3224
3225  013620  012737  002420  001124  2$:          MOV     #BIT10!BIT8!BIT4,$GDDAT ;NOW TRY A NEW BIT PATTERN.
3226
3227
3228                                        ;*      MOV     $GDDAT,@ARADS   ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARADS
3229
3230                                        ;*      MOV     @ARADS,$BDDAT   ;/READ DEVICE REG ARADS,PUT DATA IN $BDDAT.
3231  013646  042737  000200  001126               BIC     #BIT7,$BDDAT    ;IGNORE FLAG
3232  013654  023737  001124  001126               CMP     $GDDAT,$BDDAT   ;BITS SET OK?
3233  013662  001402                               BEQ     3$
3234
3235  013664  104015                               ERROR   15              ;FAILED TO WRITE  #'NY' AD11K CSR CORRECTLY
3236  013666  000451                               BR      TST17           ;;
3237
3238  013670  005037  001124          3$:          CLR     $GDDAT          ;NOW CLEAR ALL BITS SET
3239
3240                                        ;*      MOV     $GDDAT,@ARADS   ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARADS
3241
3242                                        ;*      MOV     @ARADS,$BDDAT   ;/READ DEVICE REG ARADS,PUT DATA IN $BDDAT.
3243  013714  042737  000200  001126               BIC     #BIT7,$BDDAT    ;IGNORE DONE FLAG.
3244  013722  005737  001126                       TST     $BDDAT          ;CSR CLEAR
3245  013726  001402                               BEQ     4$
3246
3247  013730  104015                               ERROR   15              ;AR11 A/D CSR FAILED TO CLEAR.
3248  013732  000427                               BR      TST17           ;;
3249
3250  013734                          4$:
3251
3252                                        ;*      MOV     @ARADB,MYTEMP   ;/READ DEVICE REG ARADB,PUT DATA IN MYTEMP.
3253  013744  012737  000001  001124               MOV     #BIT0,$GDDAT    ;NOW WE'LL SET THE A/D START BIT.
3254
3255                                        ;*      MOV     $GDDAT,@ARADS   ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARADS
3256  013762  012737  000200  001124               MOV     #BIT7,$GDDAT    ;WHEN WE READ CSR BACK EXPECT
3257                                                                       ;START BIT TO CLEAR, DONE TO SET.
3258
3259                                        ;*      MOV     @ARADS,$BDDAT   ;/READ DEVICE REG ARADS,PUT DATA IN $BDDAT.
3260
3261  014000  023737  001124  001126               CMP     $GDDAT,$BDDAT   ;START OK?
3262  014006  001401                               BEQ     TST17           ;;
3263
3264  014010  104015                               ERROR   15              ;CSR BAD AFTER A/D START
3265                                                                       ;BIT 0 SHOULD CLEAR ONLY A/D DONe SET.
3266  014012                          7$:
3267
3268
3269                                        ;;**************************************************************
3270                                        ;*TEST 17        I/O DEVICE TEST-AR11 CLOCK OPTION
```

```
3271
3272                                    ;*
3273                                    ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
3274                                    ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
3275                                    ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
3276                                    ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
3277                                    ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
3278                                    ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
3279                                    ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
3280                                    ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
3281                                    ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
3282                                    ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
3283                                    ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
3284                                    ;*
3285                                    ;*      IN THIS TEST WE WILL EXERCISE THE AR11 CLOCK.  IF
3286                                    ;*      SR1 IS UNALTERED, WE DO THIS TEST SINCE THE
3287                                    ;*      KW11K IS A DEFAULT I/O DEVICE.
3288                                    ;*      PLEASE NOTE: THIS IS ONLY A BRIEF TEST OF THE AR11,
3289                                    ;*      TO SEE THAT IT ROUGHLY WORKS, FOR TESTING, YOU
3290                                    ;*      MUST RUN THE AR11 DIAGNOSTIC, COMPLETE
3291                                    ;*
3292                                    ;*      TESTS:          1.      MAKE SURE AR11 RESPONDS TO ADDRESS
3293                                    ;*                              (NOTE: IF THE AR11 ON YOU I/O BUS USES A NON
3294                                    ;*                              STANDARD ADDRESS, MODIFY "ARADS" WITH THE CORR
3295                                    ;*
3296                                    ;*                      2.      CLOCK CSR BITS 14,6,3, AND 1 SET
3297                                    ;*
3298                                    ;*                      3.      CLOCK CSR BITS 8 AND 2 SET
3299                                    ;*
3300                                    ;*                      4.      CLOCK CSR ZEROS.
3301                                    ;*
3302                                    ;*                      5.      START CLOCK A MODE 0, RATE 1MHZ,
3303                                    ;*                              CHECK "ENABLE CNTR A" CLEARS,
3304                                    ;*                              "A OVERFLOW FLAG" SETS
3305                                    ;*
3306                                    ;*                      6.      CLOCK A CSR ZEROS
3307                                    ;*
3308                                    ;**************************************************************
3309  014012  000004                    TST17:  SCOPE
3310
3311  014014  032737  002000  001562            BIT     #BIT10,SR1      ;IS THIS DEVICE SELECTED FOR TEST?
3312                                                                      ;NOTE: DEFAUT=YES.
3313  014022  001002                            BNE     10$
3314  014024  000137  014354                    JMP     11$
3315  014030                            10$:
3316
3317  014030  012737  040130  001124            MOV     #BIT14!BIT6!BIT4!BIT3,$GDDAT ;TEST THESE BITS
3318
3319
3320                                    ;*      MOV     $GDDAT,@ARCS    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARCS
3321
3322  014046  005737  037532                    TST     $AERR           ;DID AR11 RESPOND
3323  014052  001403                            BEQ     12$             ;TO ADDR? $AERR=0?
3324                                                                      ;IF YES-NEXT TEST.
```

```
3325  014054  104015                          ERROR   15              ;AR11 CLOCK FAILED TO RESPOND TO ADDR.
3326  014056  000137  014354                  JMP     11$
3327  014062                          12$:
3328
3329                                  ;*      MOV     @ARCS,$BDDAT    ;/READ DEVICE REG ARCS,PUT DATA IN $BDDAT.
3330
3331  014072  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID THESE BITS SET?
3332  014100  001403                          BEQ     1$
3333
3334  014102  104015                          ERROR   15              ;AR11 CLOCK R/W CSR ERROR.
3335  014104  000137  014354                  JMP     11$
3336
3337  014110  012737  000424  001124  1$:     MOV     #BIT8!BIT4!BIT2,$GDDAT ;GET NEW PATTERN
3338
3339                                  ;*      MOV     $GDDAT,@ARCS    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARCS
3340
3341                                  ;*      MOV     @ARCS,$BDDAT    ;/READ DEVICE REG ARCS,PUT DATA IN $BDDAT.
3342
3343  014136  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID THE BITS SET?
3344  014144  001403                          BEQ     2$
3345
3346  014146  104015                          ERROR   15              ;AR11 CLOCK R/W CSR ERROR.
3347  014150  000137  014354                  JMP     11$
3348
3349  014154  005037  001124          2$:     CLR     $GDDAT          ;WRITE ALL ZEROS.
3350
3351                                  ;*      MOV     $GDDAT,@ARCS    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARCS
3352
3353                                  ;*      MOV     @ARCS,$BDDAT    ;/READ DEVICE REG ARCS,PUT DATA IN $BDDAT.
3354  014200  012737  000020  001124          MOV     #BIT4,$GDDAT    ;BIT 4 ALWAYS SET
3355  014206  023737  001124  001126          CMP     $GDDAT,$BDDAT
3356  014214  001403                          BEQ     6$              ;YES-GOOD.
3357
3358  014216  104015                          ERROR   15              ;AR11 CLOCK FAILED TO ZERO CSR.
3359  014220  000137  014354                  JMP     11$
3360
3361  014224  012737  000003  001124  6$:     MOV     #BIT0!BIT1,$GDDAT ;PUT "ENABL CNTR A" AND RATE IMHZ IN CSR
3362  014232  012737  177777  001126          MOV     #177777,$BDDAT  ;PRELOAD CNTR.
3363
3364                                  ;*      MOV     $BDDAT,@ARCB    ;/ PUT DATA FROM $BDDAT TO DEVICE REG ARCB
3365
3366                                  ;*      MOV     $GDDAT,@ARCS    ;/ PUT DATA FROM $GDDAT TO DEViCE REG ARCS
3367
3368                                  ;*      MOV     @ARCS,$BDDAT    ;/READ DEVICE REG ARCS,PUT DATA IN $BDDAT.
3369                                                                  ;CLOCK SHOULD OVERFLOW, CLEAR
3370  014270  012737  000222  001124          MOV     #BIT7!BIT4!BIT1,$GDDAT ;BIT 0 ("ENABLE CNTR A) SET
3371                                                                  ;BIT 7
3372  014276  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;HAPPEN OK?
3373  014304  001401                          BEQ     7$
3374
3375  014306  104015                          ERROR   15              ;CLOCK CSR PROBLEM
3376
3377  014310  005037  001124          7$:     CLR     $GDDAT          ;TRY CLEARING CSR
3378
```

```
3379                                    ;*      MOV     $GDDAT,@ARCS    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARCS
3380
3381                                    ;*      MOV     @ARCS,$BDDAT    ;/READ DEVICE REG ARCS,PUT DATA IN $BDDAT.
3382  014334  012737  000020  001124            MOV     #BIT4,$GDDAT    ;BIT 4 ALWAYS SET
3383  014342  023737  001124  001126            CMP     $GDDAT,$BDDAT
3384  014350  001401                            BEQ     TST20           ;;
3385
3386  014352  104015                            ERROR   15              ;CSR FAILED TO CLEAR
3387  014354                          11$:
3388
3389
3390                                    ;;*****************************************************************
3391                                    ;*TEST 20       *TEST THE AR11 DISPLAY OPTION, IF "SR1" BIT 10=1 (SET)
3392                                    ;*
3393                                    ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
3394                                    ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
3395                                    ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
3396                                    ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
3397                                    ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
3398                                    ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
3399                                    ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
3400                                    ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
3401                                    ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
3402                                    ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
3403                                    ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
3404                                    ;*
3405                                    ;*IN THIS TEST WE WILL CHECK OUT THE AR11 DISPLAY OPTION,
3406                                    ;*IF BIT 10 OF "SR1" IS SET.
3407                                    ;*
3408                                    ;;*****************************************************************
3409  014354  000004                    TST20:  SCOPE
3410
3411  014356  032737  002000  001562            BIT     #BIT10,SR1      ;IS AR-11 SELECTED
3412  014364  001002                            BNE     20$             ;IF =0, NO, SKIP THIS TEST.
3413  014366  000137  015132                    JMP     19$
3414  014372                          20$:
3415  014372  005037  001124                    CLR     $GDDAT
3416
3417                                    ;*      MOV     $GDDAT,@ARDS    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARDS
3418  014406  005737  037532                    TST     $AERR           ;DID WE SUCCESSFULLY WRITE THE AR-11?
3419  014412  001403                            BEQ     1$              ;YES IF $AERR=0
3420
3421  014414  104015                            ERROR   15              ;FAILED TO ADDRESS AR-11.  "SR1" BIT 3=1
3422                                                                    ;SELECTED THIS TEST.  AR-11 ADDRESS IS
3423                                                                    ;IN LOC "ARADS"
3424  014416  000137  015132                    JMP     19$
3425
3426  014422                          1$:
3427
3428                                    ;*      MOV     @ARDS,$BDDAT    ;/READ DEVICE REG ARDS,PUT DATA IN $BDDAT.
3429  014432  042737  000200  001126            BIC     #BIT7,$BDDAT    ;CLEAR FLAG
3430  014440  001403                            BEQ     2$              ;DID CSR CLEAR?
3431
3432  014442  104015                            ERROR   15              ;CLEAR CSR FAILED (AR-11)
```

```
3433  014444  000137  015132                    JMP     19$
3434
3435  014450  012737  005104  001124  2$:       MOV     #5104,$GDDAT    ;TRY WRITING THESE BITS
3436
3437                                    ;*      MOV     $GDDAT,@ARDS    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARDS
3438
3439                                    ;*      MOV     @ARDS,$BDDAT    ;/READ DEVICE REG ARDS,PUT DATA IN $BDDAT.
3440                                                                    ;READ THEM BACK
3441  014476  042737  000200  001126            BIC     #BIT7,$BDDAT    ;GET RID OF FLAG
3442  014504  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;DID PATTERN W/R OK?
3443  014512  001403                            BEQ     3$              ;YES-NEXT TEST.
3444
3445  014514  104015                            ERROR   15              ;AR11 D/A PATTERN 5024 FAILED TO W/R PROPERLY.
3446  014516  000137  015132                    JMP     19$
3447
3448  014522  012737  002010  001124  3$:       MOV     #2010,$GDDAT    ;TRY NEW PATTERN.
3449
3450                                    ;*      MOV     $GDDAT,@ARDS    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARDS
3451
3452                                    ;*      MOV     @ARDS,$BDDAT    ;/READ DEVICE REG ARDS,PUT DATA IN $BDDAT.
3453  014550  042737  000200  001126            BIC     #BIT7,$BDDAT    ;GET RID OF FLAG.
3454  014556  023737  001124  001126            CMP     $GDDAT,$BDDAT   ;DID PATTERN XFERR OK?
3455  014564  001403                            BEQ     4$              ;YES-NEXT TEST.
3456
3457  014566  104015                            ERROR   15              ;AR-11 DISPLAY REG PATTERN 2010 FAILED TO W/R PROPERLY.
3458  014570  000137  015132                    JMP     19$
3459
3460  014574  005037  001124  4$:       CLR     $GDDAT          ;TRY TO CLEAR CSR.
3461
3462                                    ;*      MOV     $GDDAT,@ARDS    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARDS
3463
3464                                    ;*      MOV     @ARDS,$BDDAT    ;/READ DEVICE REG ARDS,PUT DATA IN $BDDAT.
3465  014620  042737  000200  001126            BIC     #BIT7,$BDDAT    ;CLEAR FLAG BIT.
3466  014626  001403                            BEQ     5$
3467
3468  014630  104015                            ERROR   15              ;AR-11 DISPLAY REG. CSR FAILED TO CLEAR.
3469  014632  000137  015132                    JMP     19$
3470
3471  014636  012737  001252  001124  5$:       MOV     #1252,$GDDAT
3472
3473                                    ;*      MOV     $GDDAT,@ARXB    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARXB
3474
3475                                    ;*      MOV     @ARXB,$BDDAT    ;/READ DEVICE REG ARXB,PUT DATA IN $BDDAT.
3476  014664  023737  001124  001126            CMP     $GDDAT,$BDDAT
3477  014672  001403                            BEQ     6$              ;OK?
3478
3479  014674  104015                            ERROR   15              ;AR-11 DACX FAILED TO sET PATTERN 1252.
3480  014676  000137  015132                    JMP     19$
3481
3482  014702                            6$:
3483
3484                                    ;*      MOV     $GDDAT,@ARYB    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARYB
3485
3486                                    ;*      MOV     @ARYB,$BDDAT    ;/READ DEVICE REG ARYB,PUT DATA IN $BDDAT.
```

```
3487  014722  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;OK
3488  014730  001403                          BEQ     7$              ;YES-
3489
3490  014732  104015                          ERROR   15              ;AR-11 DACY FAILED PATTERN 1252
3491  014734  000137  015132                  JMP     19$
3492
3493  014740  012737  000525  001124  7$:     MOV     #0525,$GDDAT
3494
3495                                  ;*      MOV     $GDDAT,@ARXB    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARXB
3496
3497                                  ;*      MOV     @ARXB,$BDDAT    ;/READ DEVICE REG ARXB,PUT DATA IN $BDDAT.
3498  014766  023737  001124  001126          CMP     $GDDAT,$BDDAT
3499  014774  001402                          BEQ     10$             ;OK?
3500
3501  014776  104015                          ERROR   15              ;AR-11 DACX FAILED TO sET PATTERN 0525
3502  015000  000454                          BR      TST21           ;;
3503  015002                          10$:
3504
3505                                  ;*      MOV     $GDDAT,@ARYB    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARYB
3506
3507                                  ;*      MOV     @ARYB,$BDDAT    ;/READ DEVICE REG ARYB,PUT DATA IN $BDDAT.
3508  015022  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;OK?
3509  015030  001402                          BEQ     13$             ;YES-
3510
3511  015032  104015                          ERROR   15              ;AR-11 DACY FAILED PATTERN 0525
3512  015034  000436                          BR      TST21           ;;
3513
3514  015036  012737  000000  001124  13$:     MOV     #0,$GDDAT
3515
3516                                  ;*      MOV     $GDDAT,@ARXB    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARXB
3517
3518                                  ;*      MOV     @ARXB,$BDDAT    ;/READ DEVICE REG ARXB,PUT DATA IN $BDDAT.
3519  015064  023737  001124  001126          CMP     $GDDAT,$BDDAT
3520  015072  001402                          BEQ     14$             ;OK?
3521
3522  015074  104015                          ERROR   15              ;AR-11 DACX FAILED TO sET PATTERN 0
3523  015076  000415                          BR      TST21           ;;
3524
3525  015100                          14$:
3526
3527                                  ;*      MOV     $GDDAT,@ARYB    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARYB
3528
3529                                  ;*      MOV     @ARYB,$BDDAT    ;/READ DEVICE REG ARYB,PUT DATA IN $BDDAT.
3530  015120  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;OK?
3531  015126  001401                          BEQ     19$             ;YES-
3532
3533  015130  104015                          ERROR   15              ;AR-11 DACY FAILED PATTERN 0
3534  015132                          19$:
3535
3536
3537                                  ;;**********************************************************************
3538                                  ;*TES1 21        *TEST THAT THE AR11 CAN DISPLAY A SQUARE, SELECTED BY BIT 10 OF SR1,SR2
3539                                  ;*IN THIS TEST WE'LL DISPLAY A SQUARE ON THE
3540                                  ;*DISPLAY SCOPE VIA THE AR-11.
```

```
                                        ;*IF YOU HAVE AN AR11 AND SCOPE YOU MUST SELECT THIS
                                        ;*TEST BY SETTING THE APPROPRIATE BITS IN "SR1" (INDICATING YOU HAVE
                                        ;*AN AR-11) AND "SR2" (INDICATING YOU HAVE A SCOPE).
                                        ;;*****************************************************************
015132  000004                        TST21:  SCOPE

015134  032737  002000  001562                BIT     #BIT10,SR1      ;/AR-11 SELECTED?
015142  001475                              BEQ     TST22           ;;
015144  032737  010000  001564              BIT     #BIT12,SR2      ;/SCOPE DISPLAY?
015152  001471                              BEQ     TST22           ;;

015154  005037  001124                      CLR     $GDDAT
015160  005037  001126                      CLR     $BDDAT
015164  012737  000001  001754              MOV     #1,MYTEMP
015172                                  1$:

                                        ;*      MOV     $GDDAT,@ARXB    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARXB

                                        ;*      MOV     MYTEMP,@ARDS    ;/ PUT DATA FROM MYTEMP TO DEVICE REG ARDS
015212  005237  001124                      INC     $GDDAT
015216  032737  002000  001124              BIT     #BIT10,$GDDAT
015224  001762                              BEQ     1$
015226                                  2$:

                                        ;*      MOV     $BDDAT,@ARYB    ;/ PUT DATA FROM $BDDAT TO DEVICE REG ARYB

                                        ;*      MOV     MYTEMP,@ARDS    ;/ PUT DATA FROM MYTEMP TO DEVICE REG ARDS
015246  005237  001126                      INC     $BDDAT
015252  032737  002000  001126              BIT     #BIT10,$BDDAT
015260  001762                              BEQ     2$
015262                                  3$:

                                        ;*      MOV     $GDDAT,@ARXB    ;/ PUT DATA FROM $GDDAT TO DEVICE REG ARXB

                                        ;*      MOV     MYTEMP,@ARDS    ;/ PUT DATA FROM MYTEMP TO DEVICE REG ARDS
015302  005337  001124                      DEC     $GDDAT
015306  001365                              BNE     3$
015310                                  4$:

                                        ;*      MOV     $BDDAT,@ARYB    ;/ PUT DATA FROM $BDDAT TO DEVICE REG ARYB

                                        ;*      MOV     MYTEMP,@ARDS    ;/ PUT DATA FROM MYTEMP TO DEVICE REG ARDS
015330  005337  001126                      DEC     $BDDAT
015334  001365                              BNE     4$


                                        ;;*****************************************************************
                                        ;*TEST 22        I/O DEVICE TEST LPS-11 A/D OPTION
                                        ;;*****************************************************************
015336  000004                        TST22:  SCOPE

015340  032737  010000  001562                BIT     #BIT12,SR1      ;IS THIS DEVICE SELECTED FOR TEST?
                                                                        ;NOTE: DEFAULT=YES.
```

```
3595  015346  001002                        BNE     10$
3596  015350  000137  016014                JMP     7$
3597  015354                        10$:
3598  015354  005037  001126                CLR     $BDDAT
3599
3600                                ;*      MOV     $BDDAT,@LPADSR  ;/ PUT DATA FROM $BDDAT TO DEVICE REG LPADSR
3601                                                                ;OK-WHAT WE JUST DID IS ASKED THE
3602                                                                ;SLAVE MICRO-CODE TO WRITE THE CONTENTS
3603                                                                ;OF THE A/D 'S CSR TO zEROES.
3604                                                                ;IF THE A/D FAILS TO rESPOND TO ITS
3605                                                                ;ADDRESS-THE SLAVE WILL SEND US AN
3606                                                                ;ERROR CODE THAT CAUSES US TO SET $AERR=1
3607                                                                ;IN THE SUPPORT ROUTINES.
3608
3609  015370  005737  037532                TST     $AERR           ;DEVICE PRESENT?
3610  015374  001403                        BEQ     11$             ;YES-CONTINUE
3611  015376  104016                        ERROR   16              ;A/D FAILED TO RESPOND
3612  015400  000137  016014                JMP     7$              ;TO ADDR.
3613
3614  015404                        11$:
3615
3616                                ;*      MOV     @LPADSR,$BDDAT  ;/READ DEVICE REG LPADSR,PUT DATA IN $BDDAT.
3617  015414  042737  100200  001126        BIC     #BIT15!BIT7,$BDDAT ;CLEAR OUT FLAGS
3618  015422  005037  001124                CLR     $GDDAT
3619  015426  005737  001126                TST     $BDDAT          ;IS CSR ZERO?
3620  015432  001402                        BEQ     1$
3621
3622  015434  104016                        ERROR   16              ;FAILED TO ZERO A/D CSR
3623  015436  000572                        BR      TST23           ;;
3624  015440  012737  025100  001124  1$:   MOV     #BIT13!BIT11!BIT9!BIT6,$GDDAT ;NOW TRY A NEW BIT PATTERN
3625
3626
3627                                ;*      MOV     $GDDAT,@LPADSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPADSR
3628
3629                                ;*      MOV     @LPADSR,$BDDAT  ;/READ DEVICE REG LPADSR,PUT DATA IN $BDDAT.
3630  015466  042737  100200  001126        BIC     #BIT15!BIT7,$BDDAT ;IGNORE FLAGS
3631  015474  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;BIT SET OK?
3632  015502  001402                        BEQ     2$
3633
3634  015504  104016                        ERROR   16              ;FAILED TO wRITE A/D CSR CORRECTLY
3635  015506  000546                        BR      TST23           ;;
3636
3637  015510  012737  012404  001124  2$:   MOV     #BIT12!BIT10!BIT8!BIT2,$GDDAT ;NOW TRY A NEW BIT PATTERN.
3638
3639
3640                                ;*      MOV     $GDDAT,@LPADSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPADSR
3641
3642                                ;*      MOV     @LPADSR,$BDDAT  ;/READ DEVICE REG LPADSR,PUT DATA IN $BDDAT.
3643  015536  042737  100200  001126        BIC     #BIT15!BIT7,$BDDAT ;IGNORE FLAGS
3644  015544  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;BITS SET OK?
3645  015552  001402                        BEQ     3$
3646
3647  015554  104016                        ERROR   16              ;FAILED TO WRITE A/D CSR CORRECTLY
3648  015556  000522                        BR      TST23           ;;
```

```
3649
3650  015560  005037  001124          3$:     CLR     $GDDAT          ;NOW CLEAR ALL BITS SET
3651
3652                                  ;*      MOV     $GDDAT,@LPADSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPADSR
3653
3654                                  ;*      MOV     @LPADSR,$BDDAT  ;/READ DEVICE REG LPADSR,PUT DATA IN $BDDAT.
3655  015604  042737  000200  001126          BIC     #BIT7,$BDDAT    ;IGNORE DONE FLAG.
3656  015612  005737  001126                  TST     $BDDAT          ;CSR CLEAR?
3657  015616  001402                          BEQ     4$
3658
3659  015620  104016                          ERROR   16              ;A/D CSR FAILED TO CLEAR.
3660  015622  000500                          BR      TST23           ;;
3661
3662  015624                          4$:
3663
3664                                  ;*      MOV     @LPADBR,MYTEMP  ;/READ DEVICE REG LPADBR,PUT DATA IN MYTEMP.
3665  015634  012737  000001  001124          MOV     #BIT0,$GDDAT    ;NOW WE'LL SET THE A/D START BIT.
3666
3667                                  ;*      MOV     $GDDAT,@LPADSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPADSR
3668  015652  012737  000200  001124          MOV     #BIT7,$GDDAT    ;WHEN WE READ CSR BACK EXPECT
3669                                                                  ;START BIT TO CLEAR, DONe TO sET.
3670
3671                                  ;*      MOV     @LPADSR,$BDDAT  ;/READ DEVICE REG LPADSR,PUT DATA IN $BDDAT.
3672
3673  015670  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;START OK?
3674  015676  001402                          BEQ     5$
3675
3676  015700  104016                          ERROR   16              ;CSR BAD AFTER A/D START
3677                                                                  ;BIT 0 SHOULD CLEAR ONLY A/D DONE SET.
3678  015702  000450                          BR      TST23           ;;
3679
3680  015704  012737  000001  001124  5$:     MOV     #BIT0,$GDDAT    ;OK-NOW WE'RE GONNA START ANOTHER
3681                                                                  ;A/D CONVERSION WITHOUT READING
3682                                                                  ;THE RESULTS OF THE LAST ONE.  THIS
3683                                                                  ;SHOULD CAUSE THE ERROR FLAG
3684                                                                  ;AS WELL AS THE DONE FLAG TO SET.
3685
3686                                  ;*      MOV     $GDDAT,@LPADSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPADSR
3687
3688
3689                                  ;*      MOV     @LPADSR,$BDDAT  ;/READ DEVICE REG LPADSR,PUT DATA IN $BDDAT.
3690  015732  012737  100200  001124          MOV     #BIT15!BIT7,$GDDAT ;S/B
3691
3692  015740  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS 15 AND 7 SET?
3693
3694  015746  001401                          BEQ     6$
3695
3696  015750  104016                          ERROR   16              ;ERROR A/D BIT 15 AND 7 SHOULD BE
3697
3698  015752  005037  001124          6$:     CLR     $GDDAT          ;MAKE SURE ERROR FLAG CLEARS
3699
3700                                  ;*      MOV     $GDDAT,@LPADSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPADSR
3701
3702                                  ;*      MOV     @LPADSR,$BDDAT  ;/READ DEVICE REG LPADSR,PUT DATA IN $BDDAT.
```

```
3703  015776  042737  000200  001126          BIC     #BIT7,$BDDAT    ;IGNORE DONe FLAG.
3704  016004  005737  001126                  TST     $BDDAT          ;IS CSR CLEAR?
3705  016010  001401                          BEQ     7$
3706  016012  104016                          ERROR   16              ;A/D CSR FAILED TO CLEAR.
3707
3708  016014                          7$:
3709
3710                                  ;*      MOV     @LPADBR,MYTEMP  ;/READ DEVICE REG LPADBR,PUT DATA IN MYTEMP.
3711
```

```
3712
3713                            ;;*****************************************************************
3714                            ;*TEST 23       I/O DEVICE TEST-LPS-CLOCK OPTION
3715
3716                            ;*
3717                            ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
3718                            ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
3719                            ;*KMC-11 AND THE DMC-11 WORK, IN THIS TEST, THE ARBITRATION CIRCUITRY IS
3720                            ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
3721                            ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
3722                            ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
3723                            ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
3724                            ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
3725                            ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
3726                            ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
3727                            ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
3728                            ;*
3729                            ;*      IN THIS TEST WE WILL EXERCISE THE LPS-11 CLOCK.  IF
3730                            ;*      SR1 IS UNALTERED, WE DO THIS TEST SINCE THE
3731                            ;*      LPS-11 CLOCK IS A DEFAULT I/O DEVICE.
3732                            ;*      PLEASE NOTE: THIS IS ONLY A BRIEF TEST OF THE LPS-11 CLOCK,
3733                            ;*      TO SEE THAT IT ROUGHLY WORKS.  FOR TESTING, YOU
3734                            ;*      MUST RUN THE LPS-11 CLOCK DIAGNOSTIC.  COMPLETE
3735                            ;*
3736                            ;*      TESTS:          1.  MAKE SURE LPS-11 CLOCK RESPONDS TO ADDRESS
3737                            ;*                          (NOTE: IF THE LPS-11 CLOCK ON YOU I/O BUS USES A NON
3738                            ;*                          STANDARD ADDRESS, MODIFY "LPADSR" WITH THE CORR
3739                            ;*
3740                            ;*                      2.  CLOCK CSR BITS 14,9,6,3, AND 1 SET
3741                            ;*
3742
3743                            ;*                      3.  CLOCK CSR BITS 13,8 AND 2 SET
3744                            ;*
3745
3746                            ;*                      4.  CLOCK CSR ZEROS.
3747                            ;*
3748                            ;*                      5.  START CLOCK A MODE 0, RATE 1MHZ,
3749
3750                            ;*                          CHECK "ENABLE CNTR A" CLEARS,
3751                            ;*                          "OVERFLOW FLAG" SETS
3752
3753                            ;*
3754                            ;*                      6.  CLOCK A CSR ZEROS
3755                            ;*
3756
3757                            ;;*****************************************************************
3758  016024  000004            TST23:  SCOPE
3759
3760  016026  032737  020000  001562        BIT     #BIT13,SR1      ;IS THIS DEVICE SELECTED FOR TEST?
```

```
                                                        ;NOTE: DEFAULT=YES.
016034  001002                          BNE     10$
016036  000137  016346                  JMP     11$
016042                          10$:
```

```
3766  016042  012737  041110  001124          MOV     #BIT14!BIT9!BIT6!BIT3,$GDDAT ;TEST THESE BITS
3767
3768
3769                                  ;*      MOV     $GDDAT,@LPCKSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPCKSR
3770
3771  016060  005737  037532                  TST     $AERR           ;DID CLOCK RESPOND
3772  016064  001403                          BEQ     12$             ;TO ADDR? $AERR=0?
3773                                                                  ;IF YES-NEXT TEST.
3774  016066  104016                          ERROR   16              ;CLOCK FAILED TO RESPOND TO ADDR.
3775  016070  000137  016346                  JMP     11$
3776  016074                          12$:
3777
3778
3779                                  ;*      MOV     @LPCKSR,$BDDAT  ;/READ DEViCE REG LPCKSR,PUT DATA IN $BDDAT.
3780
3781  016104  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID THESE BITS SET?
3782  016112  001403                          BEQ     1$
3783
3784  016114  104016                          ERROR   16              ;CLOCK R/W CSR ERROR.
3785  016116  000137  016346                  JMP     11$
3786
3787  016122  012737  020404  001124  1$:     MOV     #BIT13!BIT8!BIT2,$GDDAT ;GET NEW PATTERN
3788
3789                                  ;*      MOV     $GDDAT,@LPCKSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPCKSR
3790
3791                                  ;*      MOV     @LPCKSR,$BDDAT  ;/READ DEVICE REG LPCKSR,PUT DATA IN $BDDAT.
3792
3793  016150  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID THE BITS SET?
3794  016156  001403                          BEQ     2$
3795
3796  016160  104016                          ERROR   16              ;CLOCK R/W CSR ERROR.
3797  016162  000137  016346                  JMP     11$
3798
3799  016166  005037  001124          2$:     CLR     $GDDAT          ;WRITE ALL ZEROS.
3800
3801                                  ;*      MOV     $GDDAT,@LPCKSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPCKSR
3802
3803                                  ;*      MOV     @LPCKSR,$BDDAT  ;/READ DEVICE REG LPCKSR,PUT DATA IN $BDDAT.
3804  016212  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
3805  016216  001403                          BEQ     3$              ;YES-GOOD.
3806
3807  016220  104016                          ERROR   16              ;CLOCK FAILED TO ZERO CSR.
3808  016222  000137  016346                  JMP     11$
3809
3810  016226  012737  000003  001124  3$:     MOV     #BIT0!BIT1,$GDDAT ;PUT "ENABL CNTR A" AND RATE IMHZ IN CSR
3811  016234  012737  177777  001126          MOV     #177777,$BDDAT  ;PRELOAD CNTR.
3812
3813                                  ;*      MOV     $BDDAT,@LPCKBR  ;/ PUT DATA FROM $BDDAT TO DEVICE REG LPCKBR
3814
3815                                  ;*      MOV     $GDDAT,@LPCKSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPCKSR
3816
3817                                  ;*      MOV     @LPCKSR,$BDDAT  ;/READ DEVICE REG LPCKSR,PUT DATA IN $BDDAT.
3818                                                                  ;CLOCK SHOULD OVERFLOW, CLEAR
3819  016272  012737  000202  001124          MOV     #BIT7!BIT1,$GDDAT ;BIT 0 ("ENABLE CNTR A) SET
```

```
                                                                ;BIT 7
016300  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;HAPPEN OK?
016306  001401                          BEQ     7$
                                                                ;
016310  104016                          ERROR   16              ;CLOCK CSR PROBLEM
                                                                ;
016312  005037  001124          7$:     CLR     $GDDAT          ;TRY CLEARING CSR
                                                                ;
                              ;*        MOV     $GDDAT,@LPCKSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPCKSR
                                                                ;
                              ;*        MOV     @LPCKSR,$BDDAT  ;/READ DEVICE REG LPCKSR,PUT DATA IN $BDDAT.
016336  005737  001126                  TST     $BDDAT          ;DID CSR CLEAR?
016342  001401                          BEQ     TST24           ;;
                                                                ;
016344  104016                          ERROR   16              ;CSR FAILED TO cLEAR
016346                        11$:
                                                                ;
                              ;;****************************************************************
                              ;*TEST 24        *TEST THE LPS I/O, IF "SR1" BIT 15=1(SET)
                              ;*
                              ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
                              ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
                              ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
                              ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
                              ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
                              ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
                              ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
                              ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
                              ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
                              ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
                              ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
                              ;*
                              ;*IN THIS TEST WE WILL CHECK OUT LPS-I/O IF BIT 15 OF SR1=1
                              ;*(SET).  ALSO, DATA LOOP BACK (OUTPUT DATA BACK TO INPUT) WILL BE CHECKED
                              ;*IF "SR2" BIT 15 =1 (SET).  FOR THIS, YOU MUST HAVE INSTALLED A LOOP-BACK
                              ;*CABLE.
                              ;;****************************************************************
016346  000004              TST24:  SCOPE
016350  032737  100000  001562          BIT     #BIT15,SR1
016356  001002                        BNE     10$             ;1-TDRT-
016360  000137  017044                JMP     11$
016364                        10$:
016364  005037  001124                CLR     $GDDAT          ;/SET TO XFERR ZEROS.
                                                                ;
                              ;*        MOV     $GDDAT,@LPIOSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPIOSR
016400  005737  037532                  TST     $AERR           ;=1 IF DEVICE NOT PRESENT,
016404  001403                          BEQ     1$              ;IF 0, PRESENT, (CON'T).
                                                                ;
016406  104016                          ERROR   16              ;DR11K #'DRN' DID NOT RESPOND WHEN
                                                                ;ADDRESSED.  "SR1" BIT 'DRB' SELECTED
                                                                ;THIS OPTION. ADDRESS "DRCR" 'DRN'
                                                                ;CONTAINS ITS ADDRESS
                                                                ;
016410  000137  017044                JMP     11$
```

```
3874
3875  016414                                1$:
3876
3877                                        ;*      MOV     @LPIOSR,$BDDAT  ;/READ DEVICE REG LPIOSR,PUT DATA IN $BDDAT.
3878  016424  005737  001126                        TST     $BDDAT          ;DID CSR CLEAR?
3879  016430  001403                                BEQ     2$
3880
3881  016432  104016                                ERROR   16              ;LPS-I/O CSR FAILED TO CLEAR.
3882  016434  000137  017044                        JMP     11$
3883  016440                                2$:
3884
3885                                        ;*      MOV     @LPIOOR,$GDDAT  ;/READ DEVICE REG LPIOOR,PUT DATA IN $GDDAT.
3886
3887                                        ;*      MOV     @LPIOIR,$GDDAT  ;/READ DEVICE REG LPIOIR,PUT DATA IN $GDDAT.
3888
3889  016460  012737  040100  001124                MOV     #BIT6!BIT14,$GDDAT ;LOAD WRITEABLE BITS.
3890
3891                                        ;*      MOV     $GDDAT,@LPIOSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPIOSR
3892
3893                                        ;*      MOV     @LPIOSR,$BDDAT  ;/READ DEVICE REG LPIOSR,PUT DATA IN $BDDAT.
3894  016506  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;DID BITS SET?
3895  016514  001402                                BEQ     3$
3896
3897  016516  104016                                ERROR   16              ;LPS-I/O CSR BIT(S) FAILED.
3898  016520  000551                                BR      TST25           ;;
3899
3900  016522  005037  001124                3$:     CLR     $GDDAT          ;TRY CLEARING CSR.
3901
3902                                        ;*      MOV     $GDDAT,@LPIOSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPIOSR
3903
3904                                        ;*      MOV     @LPIOSR,$BDDAT  ;/READ DEVICE REG LPIOSR,PUT DATA IN $BDDAT.
3905  016546  005737  001126                        TST     $BDDAT          ;DID CSR CLEAR?
3906  016552  001402                                BEQ     4$              ;YES-NEXT TEST.
3907
3908  016554  104016                                ERROR   16              ;LPS-I/O CSR FAILED TO CLEAR.
3909  016556  000532                                BR      TST25           ;;
3910
3911  016560  032737  100000  001564  4$:     BIT     #BIT15,SR2      ;DOES THIS I/O HAVE A LOOP BACK
3912                                                                    ;CABLE CONNECTED TO IT?
3913  016566  001526                                BEQ     TST25           ;;
3914                                                                    ;IF SR2 BIT15=1 THEN LOOP BACK.
3915
3916  016570  012737  052525  001124                MOV     #052525,$GDDAT  ;SET FIRST PATTERN
3917
3918                                        ;*      MOV     $GDDAT,@LPIOOR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPIOOR
3919
3920                                        ;*      MOV     @LPIOSR,$BDDAT  ;/READ DEVICE REG LPIOSR,PUT DATA IN $BDDAT.
3921  016616  012737  000200  001124                MOV     #BIT7,$GDDAT    ;EXPECT FLAGS TO SET.
3922  016624  023737  001126  001124                CMP     $BDDAT,$GDDAT   ;DID THEY?
3923  016632  001402                                BEQ     5$              ;YES-CON'T.
3924
3925  016634  104016                                ERROR   16              ;LPS-I/O CSR FLAG(S) FAILED TO SET
3926                                                                    ;WHEN DATA XFERRED.  IS THE OUTPUT
3927                                                                    ;REALLY CALLED BACK TO THE INPUT?
```

```
3928  016636  000502                          BR      TST25           ;;
3929
3930  016640  012737  052525  001124  5$:     MOV     #052525,$GDDAT  ;PATTERN THAT SHOULD HAVE XFERRED.
3931
3932                                    ;*      MOV     @LPIOIR,$BDDAT  ;/READ DEVICE REG LPIOIR,PUT DATA IN $BDDAT.
3933  016656  023737  001126  001124            CMP     $BDDAT,$GDLAT   ;DATA SENT=DATA RECEIVED?
3934  016664  001402                          BEQ     6$
3935
3936  016666  104016                          ERROR   16              ;LPS-11 I/O DATA XFERR ERROR.
3937  016670  000465                          BR      TST25           ;;
3938
3939  016672                          6$:
3940
3941                                    ;*      MOV     @LPIOSR,$BDDAT  ;/READ DEVICE REG LPIOSR,PUT DATA IN $BDDAT.
3942  016702  005737  001126                  TST     $BDDAT          ;DID "OUTPUT FLAG" SET.
3943  016706  100402                          BMI     7$              ;SHOULD HAVE.
3944
3945  016710  104016                          ERROR   16              ;LPS-11 I/O "OUTPUT FLAG" FAILED TO SET.
3946  016712  000454                          BR      TST25           ;;
3947
3948  016714                          7$:
3949
3950                                    ;*      MOV     $GDDAT,@LPIOOR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPIOOR
3951  016724  012737  125252  001124            MOV     #125252,$GDDAT  ;NEW PATTERN
3952
3953                                    ;*      MOV     $GDDAT,@LPIOOR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPIOOR
3954
3955                                    ;*      MOV     @LPIOIR,$BDDAT  ;/READ DEVICE REG LPIOIR,PUT DATA IN $BDDAT.
3956  016752  023737  001126  001124            CMP     $BDDAT,$GDDAT   ;PATTERN XFERR OK?
3957  016760  001402                          BEQ     8$
3958
3959  016762  104016                          ERROR   16              ;LPS I/O DATA XFERR ERROR
3960  016764  000427                          BR      TST25           ;;
3961
3962  016766                          8$:
3963
3964                                    ;*      MOV     $GDDAT,@LPIOIR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPIOIR
3965  016776  005037  001124                  CLR     $GDDAT          ;NOW XFERR ZERO PATTERN.
3966
3967                                    ;*      MOV     $GDDAT,@LPIOOR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPIOOR
3968
3969                                    ;*      MOV     @LPIOIR,$BDDAT  ;/READ DEVICE REG LPIOIR,PUT DATA IN $BDDAT.
3970  017022  005737  001126                  TST     $BDDAT          ;DID ZERO PATTERN GET XFERRED?
3971  017026  001402                          BEQ     9$
3972
3973  017030  104016                          ERROR   16              ;LPS I/O DATA XFERR ERROR
3974  017032  000404                          BR      TST25           ;;
3975
3976  017034                          9$:
3977
3978                                    ;*      MOV     $GDDAT,@LPIOSR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPIOSR
3979  017044                          11$:
3980
3981
```

```
                                ;;*****************************************************************
                                ;*TEST 25       *TEST THE LPS D/A OPTION, IF "SR1" BIT 14=1 (SET)
                                ;*
                                ;*IN THIS TEST WE ARE CHECKING OUT AN I/O DEVICE ON THE LPA-11'S I/O BUS.
                                ;*WE HAVE A CERTAIN AMOUNT OF CONFIDENCE THAT THE M8254 (IPBM), THE
                                ;*KMC-11 AND THE DMC-11 WORK. IN THIS TEST, THE ARBITRATION CIRCUITRY IS
                                ;*NOT CHECK, NOR THE DEVICE'S INTERRRUPT CAPABILITIES.
                                ;*THE DIAGNOSTIC IS SET UP FOR CERTAIN DEFAULT DEVICES ON THE
                                ;*I/O BUS. IF THE DEFAULT DEVICE IS NOT ON THE I/O BUS, DELETE
                                ;*IT FROM "SR1" (NOTE: AT THAT LOCATION, WE LIST CODES, AND EXPLAIN
                                ;*HOW TO ADD AND DELETE OPTIONS). IF YOU HAVE A DEVICE ON THE
                                ;*I/O BUS THAT IS NOT A DEFAULT DEVICE, AD IT TO "SR1"
                                ;*IF A FAILURE OCCURS IN THIS TEST, YOU SHOULD RUN THE
                                ;*INDIVIDUAL DIAGNOSTIC THAT EXERCISES THIS OPTION.
                                ;*
                                ;*IN THIS TEST WE WILL CHECK OUT THE LPS D/A OPTION.
                                ;*IF BIT 14 OF "SR1" IS SET.
                                ;*
                                ;*****************************************************************
017044  000004                  TST25:  SCOPE

017046  032737  040000  001562          BIT     #BIT14,SR1      ;IS LPS D/A SELECTED?
017054  001002                          BNE     20$             ;IF =0, NO, SKIP THIS TEST.
017056  000137  017622                  JMP     19$
017062                          20$:
017062  005037  001124                  CLR     $GDDAT

                                ;*      MOV     $GDDAT,@LPDASR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDASR
017076  005737  037532                  TST     $AERR           ;DID WE SUCCESSFULLY WRITE THE LPS D/A?
017102  001403                          BEQ     1$              ;YES IF $AERR=0

017104  104016                          ERROR   16              ;FAILED TO ADDRESS LPS D/A.  "SR1" BIT 14=1
                                                                ;SELECTED THIS TEST.  LPS D/A ADDRESS IS
                                                                ;IN LOC "LPDASR"
017106  000137  017622                  JMP     19$

017112                          1$:

                                ;*      MOV     @LPDASR,$BDDAT  ;/READ DEVICE REG LPDASR,PUT DATA IN $BDDAT.
017122  042737  000200  001126          BIC     #BIT7,$BDDAT    ;CLEAR FLAG
017130  001403                          BEQ     2$              ;DID CSR CLEAR?

017132  104014                          ERROR   14              ;CLEAR CSR FAILED (LPS-D/A)
017134  000137  017622                  JMP     19$

017140  012737  005124  001124  2$:     MOV     #5124,$GDDAT    ;TRY WRITING THESE BITS

                                ;*      MOV     $GDDAT,@LPDASR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDASR

                                ;*      MOV     @LPDASR,$BDDAT  ;/READ DEVICE REG LPDASR,PUT DATA IN $BDDAT.
                                                                ;READ THEM BACK.
017166  042737  000200  001126          BIC     #BIT7,$BDDAT    ;GET RID OF FLAG
017174  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID PATTERN
017202  001403                          BEQ     3$              ;
```

```
4036
4037  017204  104016                          ERROR   16              ;LPS D/A PATTERN 5124 FAILED TO W/R PROPERLY.
4038  017206  000137  017622                  JMP     19$
4039
4040  017212  012737  002012  001124  3$:     MOV     #2012,$GDDAT    ;TRY NEW PATTERN.
4041
4042                                  ;*      MOV     $GDDAT,@LPDASR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDASR
4043
4044                                  ;*      MOV     @LPDASR,$BDDAT  ;/READ DEVICE REG LPDASR,PUT DATA IN $BDDAT.
4045  017240  042737  000200  001126          BIC     #BIT7,$BDDAT    ;GET RID OF FLAG.
4046  017246  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID PATTERN XFERR OK?
4047  017254  001403                          BEQ     4$              ;YES-NEXT TEST.
4048
4049  017256  104016                          ERROR   16              ;LPS D/A PATTERN 2012 FAILED TO W/R PROPERLY.
4050  017260  000137  017622                  JMP     19$
4051
4052  017264  005037  001124  4$:     CLR     $GDDAT          ;TRY TO CLEAR CSR.
4053
4054                                  ;*      MOV     $GDDAT,@LPDASR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDASR
4055
4056                                  ;*      MOV     @LPDASR,$BDDAT  ;/READ DEVICE REG LPDASR,PUT DATA IN $BDDAT.
4057  017310  042737  000200  001126          BIC     #BIT7,$BDDAT    ;CLEAR FLAG BIT.
4058  017316  001403                          BEQ     5$
4059
4060  017320  104016                          ERROR   16              ;LPS D/A CSR FAILED TO CLEAR.
4061  017322  000137  017622                  JMP     19$
4062
4063  017326  012737  005252  001124  5$:     MOV     #5252,$GDDAT
4064
4065                                  ;*      MOV     $GDDAT,@LPDAXR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDAXR
4066
4067                                  ;*      MOV     @LPDAXR,$BDDAT  ;/READ DEVICE REG LPDAXR,PUT DATA IN $BDDAT.
4068  017354  023737  001124  001126          CMP     $GDDAT,$BDDAT
4069  017362  001403                          BEQ     6$              ;OK?
4070
4071  017364  104016                          ERROR   16              ;D/A DACX FAILED TO SET PATTERN 5252.
4072  017366  000137  017622                  JMP     19$
4073
4074  017372                          6$:
4075
4076                                  ;*      MOV     $GDDAT,@LPDAYR  ;/ PUT DATA FROM $GDDAT TO DEViCE REG LPDAYR
4077
4078                                  ;*      MOV     @LPDAYR,$BDDAT  ;/READ DEVICE REG LPDAYR,PUT DATA IN $BDDAT.
4079  017412  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;OK
4080  017420  001403                          BEQ     7$              ;YES-
4081
4082  017422  104016                          ERROR   16              ;DACY FAILED PATTERN 5252
4083  017424  000137  017622                  JMP     19$
4084
4085  017430                          7$:
4086  017430  012737  002525  001124          MOV     #2525,$GDDAT
4087
4088                                  ;*      MOV     $GDDAT,@LPDAXR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDAXR
4089
```

```
                                        ;*      MOV     @LPDAXR,$BDDAT  ;/READ DEVICE REG LPDAXR,PUT DATA IN $BDDAT.
017456  023737  001124  001126                  CMP     $GDDAT,$BDDAT
017464  001402                                  BEQ     10$             ;OK?

017466  104016                                  ERROR   16              ;D/A DACY FAILED TO SET PATTER 2525
017470  000454                                  BR      TST26           ;;

017472                                  10$:

                                        ;*      MOV     $GDDAT,@LPDAYR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDAYR

                                        ;*      MOV     @LPDAYR,$BDDAT  ;/READ DEVICE REG LPDAYR,PUT DATA IN $BDDAT.
017512  023737  001126  001124                  CMP     $BDDAT,$GDDAT   ;OK?
017520  001402                                  BEQ     11$             ;YES-

017522  104016                                  ERROR   16              ;D/A DACY FAILED PATTERN 2525
017524  000436                                  BR      TST26           ;;

017526                                  11$:

017526  012737  000000  001124          13$:    MOV     #0,$GDDAT

                                        ;*      MOV     $GDDAT,@LPDAXR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDAXR

                                        ;*      MOV     @LPDAXR,$BDDAT  ;/READ DEVICE REG LPDAXR,PUT DATA IN $BDDAT.
017554  023737  001124  001126                  CMP     $GDDAT,$BDDAT
017562  001402                                  BEQ     14$             ;OK?

017564  104016                                  ERROR   16              ;D/A DACX FAILED TO SET PATTERN 0
017566  000415                                  BR      TST26           ;;

017570                                  14$:

                                        ;*      MOV     $GDDAT,@LPDAYR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDAYR

                                        ;*      MOV     @LPDAYR,$BDDAT  ;/READ DEVICE REG LPDAYR,PUT DATA IN $BDDAT.
017610  023737  001126  001124                  CMP     $BDDAT,$GDDAT   ;OK?
017616  001401                                  BEQ     TST26           ;;

017620  104016                                  ERROR   16              ;D/A DACY FAILED PATTERN 0
017622                                  19$:

017622                                  15$:


                                        ;;*****************************************************************
                                        ;*TEST 26       *TEST THAT THE LPS D/A CAN DISPLAY A SQUARE
                                        ;*IN THIS TEST WE'LL DISPLAY  A SQUARE ON A
                                        ;*DISPLAY SCOPE VIA THE LPS-D/A
                                        ;*IF YOU HAVE AN LPS-D/A AND SCOPE YOU MUST SELECT THIS
                                        ;*TEST BY SETTING THE APPROPRIATE BITS IN "SR1" (INDICATING YOU HAVE
                                        ;*AN LPS D/A) AND "SR2" (INDICATING YOU HAVE A SCOPE).
                                        ;;*****************************************************************
017622  000004                          TST26:  SCOPE
```

```
                                                        ;-DAAT-
017624  032737  040000  001562        BIT     #BIT14,SR1      ;/LPS D/A SELECTED?
017632  001475                      BEQ     TST27           ;;
017634  032737  040000  001564        BIT     #BIT14,SR2      ;/SCOPE DISPLAY?
017642  001471                      BEQ     TST27           ;;

017644  005037  001124              CLR     $GDDAT
017650  005037  001126              CLR     $BDDAT
017654  012737  000001  001754      MOV     #1,MYTEMP
017662                          1$:

                                ;*      MOV     $GDDAT,@LPDAXR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDAXR

                                ;*      MOV     MYTEMP,@LPDASR  ;/ PUT DATA FROM MYTEMP TO DEVICE REG LPDASR
017702  005237  001124              INC     $GDDAT
017706  032737  010000  001124      BIT     #BIT12,$GDDAT
017714  001762                      BEQ     1$
017716                          2$:

                                ;*      MOV     $BDDAT,@LPDAYR  ;/ PUT DATA FROM $BDDAT TO DEVICE REG LPDAYR

                                ;*      MOV     MYTEMP,@LPDASR  ;/ PUT DATA FROM MYTEMP TO DEVICE REG LPDASR
017736  005237  001126              INC     $BDDAT
017742  032737  010000  001126      BIT     #BIT12,$BDDAT
017750  001762                      BEQ     2$
017752                          3$:

                                ;*      MOV     $GDDAT,@LPDAXR  ;/ PUT DATA FROM $GDDAT TO DEVICE REG LPDAXR

                                ;*      MOV     MYTEMP,@LPDASR  ;/ PUT DATA FROM MYTEMP TO DEViCE REG LPDASR
017772  005337  001124              DEC     $GDDAT
017776  001365                      BNE     3$
020000                          4$:

                                ;*      MOV     $BDDAT,@LPDAYR  ;/ PUT DATA FROM $BDDAT TO DEVICE REG LPDAYR

                                ;*      MOV     MYTEMP,@LPDASR  ;/ PUT DATA FROM MYTEMP TO DEVICE REG LPDASR
020020  005337  001126              DEC     $BDDAT
020024  001365                      BNE     4$


                                ;;**********************************************************************
                                ;*TEST 27        *TEST THAT THE AA11K # CAN DISPLAY A SQUARE, SELECTED  BIT 3 IN SR1,SR2
                                ;*IN THIS TEST WE'LL DISPLAY A SQUARE ON THE
                                ;*DISPLAY SCOPE VIA THE AA-11K.
                                ;*IF YOU HAVE AN AA11-K AND SCOPE YOU MUST SELECT THIS
                                ;*TEST  SETTING THE APPROPRIATE BITS IN "SRI" (INDICATING YOU HAVE
                                ;*AN AA-11K) AND "SR2" (INDICATING YOU HAVE A SCOPE).
                                ;;**********************************************************************
020026  000004                TST27:  SCOPE
020030  012737  000001  001160      MOV     #1,$TIMES       ;;DO 1 ITERATION
```

```
                                                        ;-DAAT-
020036  032737  000010  001562        BIT     #BIT3,SR1       ;/AA-11K SELECTED?
020044  001475                      BEQ     TST30           ;;
020046  032737  000010  001564        BIT     #BIT3,SR2       ;/SCOPE DISPLAY?
020054  001471                      BEQ     TST30           ;;

020056  005037  001124                CLR     $GDDAT
020062  005037  001126                CLR     $BDDAT
020066  012737  000001  001754        MOV     #1,MYTEMP
020074                          1$:

                                ;*      MOV     $GDDAT,@DAC0    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC0

                                ;*      MOV     MYTEMP,@AACSR   ;/ PUT DATA FROM MYTEMP TO DEVICE REG AACSR
020114  005237  001124                INC     $GDDAT
020120  032737  010000  001124        BIT     #BIT12,$GDDAT
020126  001762                      BEQ     1$
020130                          2$:

                                ;*      MOV     $BDDAT,@DAC1    ;/ PUT DATA FROM $BDDAT TO DEVICE REG DAC1

                                ;*      MOV     MYTEMP,@AACSR   ;/ PUT DATA FROM MYTEMP TO DEVICE REG AACSR
020150  005237  001126                INC     $BDDAT
020154  032737  010000  001126        BIT     #BIT12,$BDDAT
020162  001762                      BEQ     2$
020164                          3$:

                                ;*      MOV     $GDDAT,@DAC0    ;/ PUT DATA FROM $GDDAT TO DEVICE REG DAC0

                                ;*      MOV     MYTEMP,@AACSR   ;/ PUT DATA FROM MYTEMP TO DEVICE REG AACSR
020204  005337  001124                DEC     $GDDAT
020210  001365                      BNE     3$
020212                          4$:

                                ;*      MOV     $BDDAT,@DAC1    ;/ PUT DATA FROM $BDDAT TO DEVICE REG DAC1

                                ;*      MOV     MYTEMP,@AACSR   ;/ PUT DATA FROM MYTEMP TO DEVICE REG AACSR
020232  005337  001126                DEC     $BDDAT
020236  001365                      BNE     4$
                                ;       DAAT    6,2

                                ;;*****************************************************************
                                ;*TEST 30        *TEST THAT USER MICRO-CODE CAN BE STARTED
                                        ;*IN THIS TEST, WE ARE GOING TO MAKE SURE
                                        ;*THAT RUN SETS, READY IN SETS, READY OUT CLEARS.
                                        ;*CALL INITIAL CONDITIONS
                                ;;*****************************************************************
020240  000004                TST30:  SCOPE

020242  004737  040504                JSR     PC,$RESET       ;ISSUE LPA-11 RESET.
020246  032737  000001  001202        BIT     #1,$PASS        ;ON ALTERNATE PASSES,WE ALTERNATE MICROCODE.
020254  001013                      BNE     10$             ;
020256  032777  002000  160654        BIT     #BIT10,@SWR     ;RUN ONLY DEDICATED
```

```
4252  020264  001007                          BNE     10$             ;CODE?
4253                                                                  ;YES DO THAT LOAD.
4254  020266  004537  037534          13$:    JSR     R5,$LOAD
4255  020272  044120                          MMAST
4256  020274  012737  000115  001772          MOV     #'M,$VERSN
4257  020302  000412                          BR      11$
4258  020304  004537  037534          10$:    JSR     R5,$LOAD
4259  020310  050124                          DMAST
4260  020312  012737  000104  001772          MOV     #'D,$VErSN
4261  020320  032777  001000  160612          BIT     #BIT9,@SWR      ;RUN ONLY MULITE-SER CODE?
4262  020326  001357                          BNE     13$             ;YES-GO BACK AND RELOAD
4263  020330                          11$:
4264  020330  113737  001512  001773          MOVB    VERSN,$VErSN+1
4265  020336  052777  040000  161122          BIS     #BIT14,@KMADO           ;SET INIT.
4266  020344  012701  000004                  MOV     #4,R1
4267  020350  004737  040542          12$:    JSR     PC,SDELAY
4268  020354  005301                          DEC     R1
4269  020356  001374                          BNE     12$
4270  020360  012777  104000  161100          MOV     #BIT15!BIT11,@KMADO     ;BIT 15 OF THE KMC CSR SHOULD BE
4271                                                                  ;SET AT THIS TIME.
4272  020366  012701  000062                  MOV     #50.,R1
4273
4274  020372  032777  000200  161066  1$:     BIT     #BIT7,@LPCI     ;BIT 7 OF CONTROL IN REG. (READY IN)
4275  020400  001014                          BNE     2$              ;SHOULD BE SET AT INIT.
4276  020402  004737  040542                  JSR     PC,SDELAY       ;DELAY FOR WHILE.
4277  020406  005301                          DEC     R1
4278  020410  001370                          BNE     1$
4279  020412  012737  000200  001124          MOV     #BIT7,$GDDAT
4280  020420  017737  161042  001126          MOV     @LPCI,$BDDAT
4281  020426  104017                          ERROR   17              ;LPA-11 ERROR - READY IN NOT SET AT INIT.
4282  020430  000424                          BR      TST31           ;;
4283
4284  020432  032777  000200  161032  2$:     BIT     #BIT7,@LPCO     ;CHECK READY OUT, IT SHOULD
4285  020440  001407                          BEQ     3$              ;BE CLEAR AT INIT.
4286  020442  017737  161024  001126          MOV     @LPCO,$BDDAT
4287  020450  005037  001124                  CLR     $GDDAT
4288  020454  104017                          ERROR   17              ;LPW-11 ERROR - READY OUT SET AT INIT.
4289  020456  000411                          BR      TST31           ;;
4290
4291  020460  105777  161010          3$:     TSTB    @LPSO           ;CHECK THE STATUS OUT REGISTER
4292                                                                  ;IT SHOULD BE CLEAR AT INIT
4293  020464  001406                          BEQ     TST31           ;;<IF CLEAR, NEXT TEST>
4294  020466  117737  161002  001126          MOVB    @LPSO,$BDDAT
4295  020474  005077  160424                  CLR     @$GDDAT
4296  020500  104017                          ERROR   17              ;LPA-11 ERROR STATUS OUT REGISTER NOT CLEAR AT INIT.
4297
4298                                  ;;**********************************************************************
4299                                  ;*TEST 31        *TEST FOR ERROR CODE 306
4300                                                  ;*IN THIS TEST, WE ARE GOING TO ISSUE A CLOCK START
4301                                                  ;*COMMAND. THE FIRST THING THAT SHOULD TAKE PLACE
4302                                                  ;*IS "READY OUT" SHOULD SET. NEXT WE SHOULD GET
4303                                                  ;*ERROR #306 (NO INIT COMMAND) SINCE WE HADN'T
4304                                                  ;*INITIALIZED FIRST. AFTER, WE'LL CLEAR "READY OUT"
4305                                                  ;*AND MAKE SURE THE STATUS OUT REG. CLEARS.
```

```
4306                                    ;;*****************************************************************
4307  020502  000004                    TST31:  SCOPE
4308  020504  012737  000001  001160            MOV     #1,$TIMES       ;;DO 1 ITERATION
4309
4310  020512  012737  000322  044112            MOV     #BIT1!BIT4!BIT6!BIT7,KWT ;SET START CLOCK IN CLOCK TABLE
4311                                                                     ;(REQUEST DESCRIPTOR ARRAY)
4312
```

```
4313  020520  012700  044112              MOV     #KWT,R0         ;GET ADDR. OF ARRAY
4314  020524  012760  000000  000002      MOV     #0,2(0)         ;SET CLOCK STATUS.
4315  020532  005060  000004              CLR     4(0)            ;SET CLOCK PRESET
4316  020536  010077  160734              MOV     R0,@LPADL
4317  020542  152777  000001  160716      BISB    #1,@LPCI        ;SET GO.
4318  020550  004737  040542              JSR     PC,$DELAY
4319  020554  105777  160712              TSTB    @LPCO           ;DID "READY OUT" SET?
4320  020560  100410                      BMI     1$
4321  020562  017737  160704  001126      MOV     @LPCO,$BDDAT
4322  020570  012737  000200  001124      MOV     #200,$GDDAT
4323  020576  104017                      ERROR   17              ;"READY OUT" FAILED TO SET (LPA-11 ERROR)
4324  020600  000432                      BR      TST32           ;;
4325
4326  020602  005037  001126      1$:     CLR     $BDDAT
4327  020606  012737  000306  001124      MOV     #306,$GDDAT             ;/EXPECT 306.
4328  020614  117737  160654  001126      MOVB    @LPSO,$BDDAT            ;/SEE WHAT WE GET.
4329  020622  023737  001124  001126      CMP     $GDDAT,$BDDAT   ;DID LPA-11 RETURN ERROR CODE
4330                                                              ;#306 - "NO INITIALIZE COMMAND"?
4331  020630  001402                      BEQ     2$
4332
4333  020632  104017                      ERROR   17              ;LPA-11 ERROR - ERROR CODE #306 NOT RETURNED
4334  020634  000414                      BR      TST32           ;;
4335
4336  020636  005077  160630      2$:     CLR     @LPCO           ;CLEAR "READY OUT"
4337                                                              ;ERROR OR STATUS OUT REG. SHOULD CLEAR.
4338  020642  105777  160626              TSTB    @LPSO           ;DID IT CLEAR?
4339  020646  001407                      BEQ     TST32           ;;<YES - NEXT TEST>
4340
4341  020650  012737  000000  001124      MOV     #0,$GDDAT               ;/EXPECT 0.
4342  020656  117737  160612  001126      MOVB    @LPSO,$BDDAT            ;/SEE WHAT WE GET.
4343  020664  104017                      ERROR   17              ;STATUS OUT REG. NOT CLEARED WHEN "READY OUT" CLEARED (L
4344
4345                          ;;**************************************************************
4346                          ;*TEST 32        *TEST THAT WE CAN GENERATE ERROR CODE 314
4347                                  ;*
4348                                  ;*IN THIS TEST WE WILL TRY TO GENERATE ERROR CODE 314
4349                                  ;*(ODD ADDRESS SPECIFIED) FOR REQUEST DESCRIPTOR ARRAY <RDA>)
4350                                  ;*
4351
4352                          ;;**************************************************************
4353  020666  000004          TST32:  SCOPE
4354  020670  012700  054130              MOV     #DEVLST,R0      ;GET AN ADDR.
4355  020674  005200                      INC     R0              ;MAKE IT ODD.
4356  020676  010077  160574              MOV     R0,@LPADL       ;STORE ADDR. AS ADDR OF A
4357                                                              ;REQUEST DESCRIPTOR ARRAY.
4358  020702  152777  000001  160556      BISB    #1,@LPCI        ;SET GO
4359  020710  005000                      CLR     R0              ;TIME OUT COUNTER
4360  020712                      1$:
4361  020712  105777  160554              TSTB    @LPCO           ;READY OUT SET?
4362  020716  100412                      BMI     2$              ;YES-EXIT LOOP
4363  020720  105200                      INCB    R0              ;NO-LOOP OVERFLOW OCCUR?
4364  020722  100373                      BPL     1$              ;NO-LOOP.
4365  020724  012737  000200  001124      MOV     #200,$GDDAT             ;/EXPECT 200.
4366  020732  017737  160534  001126      MOV     @LPCO,$BDDAT            ;/SEE WHAT WE GET.
```

```
4367  020740  104017                          ERROR   17              ;READY OUT NOT SET ON INIT CMND. (KMC-ERROR)
4368  020742  000414                          BR      TST33           ;;
4369
4370  020744                          2$:
4371  020744  012737  000314  001124          MOV     #314,$GDDAT             ;/EXPECT 314.
4372  020752  117737  160516  001126          MOVB    @LPS0,$BDDAT            ;/SEE WHAT WE GET.
4373  020760  122737  000314  001126          CMPB    #314,$BDDAT     ;DID CORRECT ERROR CODE GET RETURNED?
4374  020766  001402                          BEQ     TST33           ;;
4375
4376  020770  104017                          ERROR   17              ;ERROR CODE 314 NOT RETURNED FOR ODD ADRRESS FOR RDA
4377
4378  020772  000400                          BR      TST33           ;;
4379
4380                          ;;**************************************************************
4381                          ;*TEST 33        *TEST THAT THE DEVICE LIST CAN BE VERIFIED
4382                                  ;
4383                                  ;INIT TEST IN THIS TEST WE MAKE SURE THE KMC MICRO-
4384                                  ;CODE CAN FIND ALL ADDRESSES OF DEVICES TO BE TESTED.
4385                                  ;
4386                          ;;**************************************************************
4387  020774  000004          TST33:  SCOPE
4388
4389  020776  012700  000012          MOV     #10.,R0
4390  021002  012701  054130          MOV     #DEVLST,R1      ;CLEAR DEVICE LIST AREA
4391  021006  005021                  CLR     (1)+
4392  021010  005003                  CLR     R3
4393  021012  012721  000001  1$:     MOV     #1,(1)+         ;PUT A 1 INTO ALL POSTIONS OF
4394                                                           ;DEVICE LIST SO THAT LPA WON'T RETURN
4395                                                           ;ARR ERROR UNLESS BAD ADDR.
4396                                                           ;OTHERWIZE IT WOULD RETURN ADDR. ERROR
4397                                                           ;CODE ON ADDR. ZERO.
4398  021016  005300                  DEC     R0
4399  021020  001374                  BNE     1$
4400  021022  012701  054132          MOV     #DEVLST+2,R1
4401  021026  032737  000002  001562  BIT     #BIT1,SR1       ;KW11K CLOCK??
4402  021034  001403                  BEQ     10$             ;NO
4403  021036  013721  001570          MOV     KW11K,(1)+      ;YES FIX ADDR.
4404  021042  000425                  BR      20$
4405  021044  032737  002000  001562  10$:    BIT     #BIT10,SR1      ;AR11 CLOCK??
4406  021052  001405                  BEQ     11$             ;NO
4407  021054  013711  001610          MOV     AR11K,(1)       ;YES FIX AR11K CLOCK ADDR.
4408  021060  062721  000004          ADD     #4,(1)+
4409  021064  000414                  BR      20$
4410  021066  032737  020000  001562  11$:    BIT     #BIT13,SR1      ;LPS CLOCK?
4411  021074  001405                  BEQ     12$             ;NO-ERROR!!!
4412  021076  013711  001612          MOV     LPS11,(1)       ;YES FIX LPS11 CLOCK ADDR.
4413  021102  062721  000004          ADD     #4,(1)+
4414  021106  000403                  BR      20$
4415  021110                  12$:
4416  021110  104017                  ERROR   17              ;NO CLOCK SELECTED FOR TEST...
4417                                                           ;CAN NOT DO ANY MORE TESTS WITHOUT
4418                                                           ;CLOCK PRESENT!
4419                                                           ;YOU MUST SELECT A CLOCK.
4420  021112  000137  030576          JMP     ETEST
```

```
4421
4422  021116  012721  000001          20$:   MOV   #1,(1)+
4423  021122  032737  000001  001562          BIT   #BIT0,SR1      ;AD11K FOR TEST??
4424  021130  001403                          BEQ   21$
4425  021132  013721  001566                  MOV   AD11K,(1)+     ;YES FIX AD11K ADDR.
4426  021136  000420                          BR    30$
4427  021140  032737  002000  001562  21$:    BIT   #BIT10,SR1     ;AR11 A/D?
4428  021146  001403                          BEQ   22$
4429  021150  013721  001610                  MOV   AR11K,(1)+
4430  021154  000411                          BR    30$
4431  021156  032737  010000  001562  22$:    BIT   #BIT12,SR1     ;LPS A/D?
4432  021164  001403                          BEQ   23$
4433  021166  013721  001612                  MOV   LPS11,(1)+
4434  021172  000402                          BR    30$
4435  021174  012721  000001          23$:    MOV   #1,(1)+
4436
4437  021200  032737  000020  001562  30$:    BIT   #BIT4,SR1      ;2ND AD11K?
4438  021206  001403                          BEQ   31$
4439  021210  013721  001576                  MOV   AD11K2,(1)+
4440  021214  000402                          BR    32$
4441  021216  012721  000001          31$:    MOV   #1,(1)+
4442
4443  021222  032737  000010  001562  32$:    BIT   #BIT3,SR1      ;AA11K?
4444  021230  001403                          BEQ   33$
4445  021232  013721  001574                  MOV   AA11K,(1)+
4446  021236  000424                          BR    40$
4447  021240  032737  002000  001562  33$:    BIT   #BIT10,SR1     ;AR11?
4448  021246  001405                          BEQ   34$
4449  021250  013711  001610                  MOV   AR11K,(1)
4450  021254  062721  000016                  ADD   #16,(1)+
4451  021260  000413                          BR    40$
4452  021262  032737  040000  001562  34$:    BIT   #BIT14,SR1     ;LPS D/A?
4453  021270  001405                          BEQ   35$
4454  021272  013711  001612                  MOV   LPS11,(1)
4455  021276  062721  000016                  ADD   #16,(1)+
4456  021302  000402                          BR    40$
4457  021304  012721  000001          35$:    MOV   #1,(1)+
4458
4459  021310  032737  000004  001562  40$:    BIT   #BIT2,SR1      ;DR11K?
4460  021316  001403                          BEQ   41$
4461  021320  013721  001572                  MOV   DR11K1,(1)+
4462  021324  000413                          BR    50$
4463  021326  032737  100000  001562  41$:    BIT   #BIT15,SR1     ;LPS I/O?
4464  021334  001405                          BEQ   42$
4465  021336  013711  001612                  MOV   LPS11,(1)
4466  021342  062721  000010                  ADD   #10,(1)+       ;FIX LPS11 I/O ADDR.
4467  021346  000402                          BR    50$
4468  021350  012721  000001          42$:    MOV   #1,(1)+
4469
4470  021354  032737  000040  001562  50$:    BIT   #BIT5,SR1      ;DR11K#2?
4471  021362  001403                          BEQ   51$
4472  021364  013721  001600                  MOV   DR11K2,(1)+
4473  021370  000402                          BR    52$
4474  021372  012721  000001          51$:    MOV   #1,(1)+
```

```
4475  021376  032737  000200  001562  52$:    BIT     #BIT7,SR1       ;DR11K #3 ?
4476  021404  001403                          BEQ     53$
4477  021406  013721  001602                  MOV     DR11K3,(1)+
4478  021412  000402                          BR      54$
4479  021414  012721  000001          53$:    MOV     #1,(1)+
4480  021420  032737  000400  001562  54$:    BIT     #BIT8,SR1       ;DR11K #4 ?
4481  021426  001403                          BEQ     55$
4482  021430  013721  001604                  MOV     DR11K4,(1)+
4483  021434  000402                          BR      56$
4484  021436  012721  000001          55$:    MOV     #1,(1)+
4485  021442  032737  001000  001562  56$:    BIT     #BIT9,SR1       ;DR11K #5 ?
4486  021450  001403                          BEQ     57$
4487  021452  013721  001606                  MOV     DR11K5,(1)+
4488  021456  000402                          BR      60$
4489  021460  012711  000001          57$:    MOV     #1,(1)
4490  021464                          60$:
4491
4492
4493  021464  113737  001512  054131          MOVB    VERSN,DEVLST+1
4494
4495  021472  004737  040626          4$:     JSR     PC,SRESET       ;RESET SYSTEM
4496  021476  012777  054130  157772          MOV     #DEVLST,@LPADL  ;SET INIT POINTER
4497  021504  052777  000001  157754          BIS     #BIT0,@LPCI     ;SET GO!
4498  021512  005000                          CLR     R0
4499  021514  004737  040542          5$:     JSR     PC,SDELAY       ;DELAY
4500  021520  132777  000200  157740          BITB    #BIT7,@LPCI     ;DONE?
4501  021526  001010                          BNE     6$              ;YES!
4502  021530  105200                          INCB    R0              ;NO-WAIT
4503  021532  001370                          BNE     5$              ;BUT NOT TOO LONG...
4504  021534  005037  001124                  CLR     $GDDAT
4505  021540  005037  001126                  CLR     $BDDAT
4506
4507  021544  104017                          ERROR   17              ;LPA FAILED TO FINISH INIT.
4508  021546  000423                          BR      TST34           ;;
4509  021550  105777  157716          6$:     TSTB    @LPCO           ;DID CONTROL OUT SET WITH ERROR??
4510  021554  100020                          BPL     TST34           ;;
4511
4512  021556  117737  157712  001126          MOVB    @LPSO,$BDDAT    ;GET STATUS
4513  021564  012737  000000  001124          MOV     #0,$GDDAT
4514  021572  122737  000326  001126          CMPB    #326,$BDDAT     ;DEVICE ADDR ERROR?
4515  021600  001402                          BEQ     7$
4516  021602  104017                          ERROR   17              ;ERROR KNOWN CHECK RETURNED ERROR
4517                                                                  ;CODE AGAINST DOCUMENTATION.
4518  021604  000404                          BR      TST34           ;;
4519  021606  017737  157670  001124  7$:     MOV     @LPMS1,$GDDAT   ;BAD DEVICE ADDRESS
4520  021614  104017                          ERROR   17              ;DEVICE DOES NOT EXSISTS? SOMETHINGS
4521                                                                  ;FUNNY! PREVIOUS TESTS DIDN'T
4522                                                                  ;SHOW THIS ERROR-I'D SAY KMC-ERROR.
4523
4524                          ;;*****************************************************************
4525                          ;*TEST 34        *TRY TO GENERATE ERROR CODE 312
4526                                  ;*
4527                                  ;*TEST THAT WE CAN GENERATE ERROR CODE 312
4528                                  ;*"USER NOT ACTIVE FOR STOP COMMAND"
```

```
4529                                        ;*
4530                                ;;*****************************************************************
4531  021616  000004                TST34:  SCOPE
4532
4533  021620  012737  000403  055066          MOV     #403,JOBO       ;SET USER#1, STOP COMMAND IN MODE
4534                                                                    ;WORD OF RDA #0
4535  021626  012737  055164  055072          MOV     #JOBOU,JOBO+4   ;SET UP USW ADDR.
4536  021634  012777  055066  157634          MOV     #JOBO,@LPADL    ;LOAD ADDR OF RDA.
```

```
4537  021642  152777  000001  157616          BISB    #1,@LPCI        ;SET GO.
4538  021650  005000                          CLR     R0
4539  021652                          1$:
4540  021652  105777  157614                  TSTB    @LPCO           ;READ OUT SET?
4541  021656  100412                          BMI     2$              ;YES-EXIT LOOP
4542  021660  105200                          INCB    R0              ;NO-LOOP OVERFLOW OCCUR?
4543  021662  100373                          BPL     1$              ;NO-LOOP
4544
4545  021664  012737  000200  001124          MOV     #200,$GDDAT             ;/EXPECT 200.
4546  021672  117737  157574  001126          MOVB    @LPCO,$BDDAT            ;/SEE WHAT WE GET.
4547  021700  104017                          ERROR   17              ;READY OUT NOT SET ON CMND (KMC ERROR)
4548  021702  000413                          BR      TST35           ;;
4549
4550  021704                          2$:
4551  021704  012737  000312  001124          MOV     #312,$GDDAT             ;/EXPECT 312.
4552  021712  117737  157556  001126          MOVB    @LPSO,$BDDAT            ;/SEE WHAT WE GET.
4553  021720  122737  000312  001126          CMPB    #312,$BDDAT     ;DID CORRECT ERROR CODE GET RETURNED
4554  021726  001401                          BEQ     TST35           ;;
4555  021730  104017                          ERROR   17              ;ERROR CODE 312 NOT RETURNED STOP USER ISSUED AND NO USE
4556
4557
4558                                  ;;**************************************************************
4559                                  ;*TEST 35        *TRY TO GENERATE ERROR CODE 310
4560                                          ;*
4561                                          ;*MAKE USRE WE CAN GENERATE ERROR CODE 310
4562                                          ;*"MULTIPLE INITIALIZE COMMAND".
4563                                          ;*
4564                                  ;;**************************************************************
4565  021732  000004                  TST35:  SCOPE
4566
4567  021734  004737  040626                  JSR     PC,SRESET       ;LPA-11 RESET
4568
4569  021740  012777  054130  157530          MOV     #DEVLST,@LPADL  ;SET ADDR. OF DEVICE LIST IN RDA WORD
4570  021746  005037  054130                  CLR     DEVLST          ;INDICATE SIZING
4571  021752  113737  001512  054131          MOVB    VERSN,DEVLST+1  ;SET VERSION NUMBER
4572
4573  021760  152777  000001  157500          BISB    #1,@LPCI        ;SET GO.
4574  021766  004737  040542                  JSR PC, SDELAY
4575  021772                          1$:
4576  021772  105777  157470                  TSTB    @LPCI           ;READY IN SET?
4577  021776  100410                          BMI     2$
4578  022000  012737  000200  001124          MOV     #200,$GDDAT             ;/EXPECT 200.
4579  022006  017737  157454  001126          MOV     @LPCI,$BDDAT            ;/SEE WHAT WE GET.
4580  022014  104017                          ERROR   17              ;READY IN NOT SET ON INIT CMND
4581  022016  000456                          BR      TST36           ;;
4582
4583  022020                          2$:                             ;NOTE-WE HAVE ALREADY DETERMINED
4584                                                                  ;DEVICES IN LIST ARE VALID
4585                                                                  ;SO WE SHOULD GET A "000"
4586                                                                  ;STATUS BACK FOR STAT REG.
4587  022020  012737  000000  001124          MOV     #0,$GDDAT               ;/EXPECT 0.
4588  022026  117737  157440  001126          MOVB    @LPCO,$BDDAT            ;/SEE WHAT WE GET.
4589
4590  022034  122737  000000  001126          CMPB    #0,$BDDAT               ;DID AN   ERROR CODE GET RETURNED?
```

```
4591  022042  001405                          BEQ     3$              ;NO-NEXT
4592
4593  022044  117737  157424  001126          MOVB    @LPS0,$BDDAT    ;PUT ERROR CODE IN FOR TYPEOUT.
4594  022052  104017                          ERROR   17              ;KNOWN ERROR CODE RETURN ON INIT COMMAND
4595  022054  000437                          BR      TST36           ;;
4596
4597  022056  012777  054130  157412  3$:     MOV     #DEVLST,@LPADL  ;RESET ADDR.
4598  022064  152777  000001  157374          BISB    #1,@LPCI        ;SET GO-DO ANOTHER INITIALIZE CMND.
4599  022072  005000                          CLR     R0
4600  022074  004737  040542          4$:     JSR PC,SDELAY
4601  022100  105777  157366                  TSTB    @LPCO           ;READY OUT SET
4602  022104  100410                          BMI     5$
4603  022106  012737  000200  001124          MOV     #200,$GDDAT             ;/EXPECT 200.
4604  022114  017737  157352  001126          MOV     @LPCO,$BDDAT            ;/SEE WHAT WE GET.
4605  022122  104017                          ERROR   17                      ;READY OUT NOT SET ON INIT COMMAND
4606  022124  000413                          BR      TST36           ;;
4607
4608  022126                          5$:
4609  022126  012737  000310  001124          MOV     #310,$GDDAT             ;/EXPECT 310.
4610  022134  117737  157334  001126          MOVB    @LPS0,$BDDAT            ;/SEE WHAT WE GET.
4611  022142  122737  000310  001126          CMPB    #310,$BDDAT     ;NOW WE EXPECT ERROR CODE 310
4612  022150  001401                          BEQ     TST36           ;;
4613
4614  022152  104017                          ERROR   17              ;ERROR CODE 310 NOT RETURN ON DOUBLE INIT.
4615
4616                          ;;*****************************************************************
4617                          ;*TEST 36        *TEST INTERRUPTS
4618
4619                                  ;*
4620                                  ;*IN THIS TEST WE'LL MAKE SURE READY IN AND READY OUT
4621                                  ;*CAN GENERATE INTERRUPTS
4622                                  ;*
4623
4624                          ;;*****************************************************************
4625  022154  000004          TST36:  SCOPE
4626
4627  022156  004737  040626          JSR     PC,SRESET
4628  022162  013700  001556          MOV     VECT1,R0        ;GET VECTOR ADDR.
4629  022166  012710  022350          MOV     #4$,(0)         ;SET ADDR FOR READY OUT INTERRUPT.
4630  022172  012760  022270  000004  MOV     #1$,4(0)                ;SET ADDR FOR READY IN INTERRUPT
4631  022200  012777  054130  157270  MOV     #DEVLST,@LPADL  ;SET ADDR. OF DEVICE LIST IN RDA WORD
4632  022206  005037  054130          CLR     DEVLST          ;INDICATE SIZING
4633  022212  113737  001512  054131  MOVB    VERSN,DEVLST+1  ;SET VERSION NUMBER
4634
4635  022220  152777  000001  157240  BISB    #1,@LPCI        ;SET GO.
4636  022226  004737  040542          JSR PC, SDELAY
4637  022232  052777  000100  157226  BIS     #100,@LPCI
                                                              ;INTERRUPT RIGHT AWAY.
4638
4639  022240  004737  040542          JSR     PC,SDELAY
```

```
                                                MOV     #300,$GDDAT             ;/EXPECT 300.
                                                MOV     @LPCI,$BDDAT            ;/SEE WHAT WE GET.
                                                CLR     @LPCI           ;CLEAR INTERRUPT ENABLE.
                                                ERROR   17              ;READY IN FAILED TO CAUSE AN INTERRUPT
                                                BR      TST37           ;;
                                        1$:                             ;DELAY SERVICING

                                        2$:     MOV     #DEVLST,@LPADL  ;SET FOR INIT.
                                                BISB    #1,@LPCI
                                                BIS     #BIT6,@LPCO     ;SET READY OUT INTERRUPT ENABLE
                                                CLR     PS
                                        3$:
                                                JSR     PC,SDELAY
                                                BPL     3$
                                                MOV     #300,$GDDAT             ;/EXPECT 300.
                                                MOV     @LPCO,$BDDAT            ;/SEE WHAT WE GET.
                                                CLRB    @LPCO           ;DISABLE ON ERROR.
                                                ERROR   17              ;READY OUT FAILED TO GENERATE AN ERROR
                                                BR      TST37           ;;

                                        4$:

                                        5$:
                                                CLRB    @LPCI
                                                CLRB    @LPCO
                                                MOV     #STACK,SP

                                                                        ;STATE LOGIC TESTS DONE.


                                        ;;**********************************************************************
                                        ;*TEST 37       *TEST THAT THE LPA-SYSTEM CLOCK CAN BE STARTED
                                        ;*IN THIS TEST WE WILL MAKE SURE WE CAN ORDERLY START
                                        ;*THE SYSTEM CLOCK IF ONE EXISTS. IF NO CLOCK SPECIFIED BY
                                        ;*SR1, WE'LL SEE IF AN ANOLOG DEVICE IS SPECIFIED, IF SO AN ERROR WILL
                                        ;*BE REPORTED SINCE WE DO NEED A CLOCK.
                                        ;;**********************************************************************
                                        TST37:  SCOPE
                                                MOV     #1,$TIMES       ;;DO 1 ITERATION

                                                JSR     PC,SRESET
                                                MOV     #DEVLST,@LPADL  ;SET ADDR. OF DEVICE LIST IN RDA WORD
                                                CLR     DEVLST          ;INDICATE SIZING
                                                MOVB    VERSN,DEVLST+1  ;SET VERSION NUMBER

                                                BISB    #1,@LPCI        ;SET GO.
                                                JSR PC, SDELAY
                                        1$:     BIT     #BIT1!BIT10!BIT13,SR1   ;IS A CLOCK SELECTED?
                                                BNE     2$                      ;YES-CONTINUE.
                                                BIT     #BIT0!BIT12,SR1         ;IS A CLOCK NEEDED?
                                                BEQ     TST40           ;;
                                                ERROR   17                      ;ERROR-NO CLOCK SPECIFIED FOR ANOLOG
                                                HALT                            ;USER MAY NOT CONTINUE-FATAL ERROR
```

```
4694  022456  000765                          BR      1$                      ;IF A CLOCK IS IN SYSTEM, THEN SELEC IT.
4695                                                                          ;IF YOU DESiRE TO RUN TIS DIAG. WITH NO CLOCK,
4696                                                                          ;DESELECT ANOLOG OPTIONS.
4697
4698  022460  012737  000001  044112  2$:     MOV     #BIT0,KWT               ;SELECT MODE:START CLOCK
4699  022466  123727  001772  000115          CMPB    $VERSN,#'M              ;MULTI-REQUEST MODE?
4700  022474  001003                          BNE     4$
4701  022476  052737  000010  044112          BIS     #BIT3,KWT
4702  022504                          4$:
4703  022504  012737  000503  044114          MOV     #BIT1!BIT8!BIT0!BIT6,KWT+2      ;RATE:1MHZ,GO BIT, REPEATED INTERRUPT,IN
4704  022512  012737  177766  044116          MOV     #-10.,KWT+4     ;CLOCK INTERRUPT EVERY 10 USEC.
4705  022520  012777  044112  156750          MOV     #KWT,@LPADL     ;PUT ADDR. IN LPA ADDR. REG
4706  022526  052777  000001  156732          BIS     #BIT0,@LPCI     ;SET GO.
4707
4708  022534  004737  040542                  JSR     PC,SDELAY
4709  022540  032777  000200  156720          BIT     #BIT7,@LPCI     ;READY IN SET
4710  022546  001010                          BNE     3$
4711  022550  012737  000200  001124          MOV     #200,$GDDAT             ;/EXPECT 200.
4712  022556  017737  156704  001126          MOV     @LPCI,$BDDAT            ;/SEE WHAT WE GET.
4713  022564  104017                          ERROR   17                      ;READY IN FAILED TO SET WHEN
4714                                                                          ;CLOCK STARTED.
4715  022566  000412                          BR      TST40                   ;;
4716
4717  022570  105777  156676          3$:     TSTB    @LPCO                   ;ERROR CODE RETURNED?
4718  022574  100007                          BPL     TST40                   ;;
4719  022576  012737  000000  001124          MOV     #0,$GDDAT               ;/EXPECT 0.
4720  022604  117737  156664  001126          MOVB    @LPSO,$BDDAT            ;/SEE WHAT WE GET.
4721
4722  022612  104017                          ERROR   17                      ;ERROR CODE RETURNED ON
4723                                                                          ;CLOCK START.
4724
4725
4726                                  ;;**********************************************************************
4727                                  ;*TEST 40       *TEST THAT THE MULTI-USER MICRO-CODE WILL EXERCISE DR11K #1
4728                                  ;*
4729                                  ;*IN THIS TEST WE'LL MAKE SURE THAT DR11K #1 WILL PASS DATA.
4730                                  ;*SR1 BIT2 SELECTS THIS DR11K FOR TEST. SR2 BIT2 INDICTES
4731                                  ;*THAT THE INPUT IS CALLED BACK TO THE OUTPUT.
4732                                  ;*
4733                                  ;*      NOTE: THIS TESt IS A MULTI-USER MICRO-CODE TEST ONLY!
4734                                  ;*            IF DEDICATE USER MICRO-CODE, THIS TEST WILL BE BY-PASSED.
4735                                  ;*
4736                                  ;*THIS TEST WAS DESiGNED IN ORDER TO FIND A WILD LPA-11 SYSTEM FAULT.
4737                                  ;*I DON'T REALLY EXPECT AN ERROR HERE UNLESS YOU'VE GOT A HEAVY FAULT.
4738                                  ;*ARBITRATION LOGIC IS ONE BIG REASON WHY YOU COULD FAIL. ANOTHER
4739                                  ;*IS THAT INTERRRUPT VECTOR COULD BE WRONG. ALSO THE DMC MICRO-CODE MIGHT
4740                                  ;*NOT BE EXECUTING PROPERLY.
4741                                  ;*
4742                                  ;;**********************************************************************
4743  022614  000004                  TST40:  SCOPE
4744  022616  012737  000001  001160          MOV     #1,$TIMES               ;;DO 1 ITERATION
4745
4746                                                                          ;/-DRLT-
4747  022624  122737  000115  001772          CMPB    #'M,$VERSN              ;RUNNING MULTI-USER MICRO-CODE?
```

```
                                                BEQ     1$              ;YES-CONTINUE.
                                                JMP     20$             ;NO-EXIT THIS TEST.

                                        1$:     MOV     #JOB0,R0        ;CLEAR OUT THE RDA TABLES
                                        25$:    CLR     (R0)+
                                                CMP     R0,#JOB4
                                                BNE     25$

                                                BIT     #BIT2,SR1       ;IS THIS DR11K SELECTED FOR TEST?
                                                BNE     2$              ;YES-TEST IT.
                                                JMP     20$             ;NO-EXIT THIS TEST.

                                        2$:     MOV     #JOB2,R0        ;PICK UP THIS JOB ADDRESS.
                                                MOV     #BIT1!BIT3,(0)  ;SET START AND MULTI-USER.
                                                BIS     #BIT4!BIT8,(0) ;SELECT DIGITAL INPUT SINGLE CHAN.
                                        3$:     MOV     #257.,2(0)      ;SET WORD COUNT
                                                MOV     #JOB2U,4(0)     ;USW ADDR.
                                                CLRB    6(0)            ;NO EXT.
                                                MOVB    #201,7(0)       ;MAKE TWO BUFFERS AVAIL.
                                                MOV     #BUFF0,10(0)    ;1ST BUFFER ADDR.
                                                MOV     #BUFF1,14(0)    ;2ND BUFFER ADDR.
                                                CLR     12(0)           ;NO EXT.
                                                MOV     #200,54(0)      ;DELAY =200 TICKS.
                                                MOV     #1-1,56(0)      ;SET CHAN #=DR11K#-1.
                                                MOV     #1,60(0)        ;SAMPLING ONLY ONE CHANNEL.
                                                MOV     #1,62(0)        ;ON PULSE BETWEEN SAMPLE IS NO EXTERNAL TRIG.
                                                CLR     64(0)           ;NO START/EVENT MAKR WORD
                                                CLR     66(0)           ;NO START MASK
                                                CLR     70(0)           ;NO EVENT MASK.

                                                MOV     #JOB1,R1        ;OK-NOW LETS SET UP THE OUTPUT.
                                                MOV     #BIT1!BIT3,(1)  ;SET START AND MULTI USER.
                                                BIS     #BIT4!BIT7!BIT8,(1) ;SELECT DIGITAL OUT.
                                                MOV     #257.,2(1)      ;WORD COUNT
                                                MOV     #JOB1U,4(1)     ;SET USW ADDR.
                                                CLRB    6(1)            ;NO EXT.
                                                MOVB    #202,7(1)
                                                MOV     #BUFF2,10(1)    ;SET BUFFER #1 ADDR.
                                                MOV     #BUFF3,14(1)    ;SET BUFFER #2 ADDR.
                                                MOV     #BUFF2,20(1)    ;THE THIRD OUTPUT IS NECESSARY
                                                                        ;TO COMPENSATE FOR SAMPLES LOST
                                                CLRB    12(1)           ;NO EXT .
                                                MOV     #200,54(1)      ;DELAY 1 TICK.
                                                MOV     #1-1,56(1)      ;SET CHAN #= DR#-1
                                                MOV     #1,60(1)        ;SAMPLE ONLY ONE CHAN.
                                                MOV     #1,62(1)        ;ONE TICK BETWEEN SAMPLE
                                                CLR     64(1)           ;NO START/EVENT
                                                CLR     66(1)           ;NO START MASK.
                                                CLR     70(1)           ;NO EVENT MASK.

                                                MOV     #125252,R0      ;FILL OUTPUT BUFFER W/PATTERN
                                                MOV     #514.,R1
                                                MOV     #BUFF2,R2
```

```
4802  023166  010022                    4$:     MOV     R0,(2)+
4803  023170  005100                            COM     R0
4804  023172  005301                            DEC     R1
4805  023174  001374                            BNE     4$
4806
4807  023176  012700  056246                    MOV     #BUFF0,R0       ;CLEAR INPUT BUFFER
4808  023202  012701  001002                    MOV     #514.,R1
4809  023206  005020                    5$:     CLR     (0)+
4810  023210  005301                            DEC     R1
4811  023212  001375                            BNE     5$
4812
4813  023214  012737  000011  055066            MOV     #11,JOB0        ;OK,NOW WE GOTTO STOP THE CLOCK
4814  023222  012737  000000  055070            MOV     #0,JOB0+2       ;SO THAT IT DOEsN'T START UNTIL
4815  023230  012737  000011  055440            MOV     #11,JOB3        ;DRIIK PARAMETER SET UP,THEN AS
4816  023236  012737  000503  055442            MOV     #503,JOB3+2     ;JOB #4,WE START CLOCK.
4817  023244  012737  177000  055444            MOV     #177000,JOB3+4
4818  023252  012737  000004  001124            MOV     #4,$GDDAT       ;FOUR JOBS.
4819  023260  012737  000002  002014            MOV     #2,FUDGE        ;/SET UP A FUDGE FACTOR TO
4820                                                                    ;/COMPENSATE FOR THE TWO KW11K JOBS
4821  023266  004737  036026                    JSR PC, FLASH           ;/GO DO THEM!
4822
4823  023272  032737  000004  001564            BIT     #BIT2,SR2       ;/CABLED TOGeTHER?
4824  023300  001417                            BEQ     TST41           ;;
4825  023302  012700  001003                    MOV     #515.,R0        ;/NOW  CHECK DATA IF CABLED.
4826  023306  012701  060252                    MOV     #BUFF2,R1
4827  023312  012702  056246                    MOV     #BUFF0,R2
4828  023316  005300                    6$:     DEC     R0
4829  023320  001407                            BEQ     TST41           ;;
4830  023322  022221                            CMP     (R2)+,(R1)+
4831  023324  001774                            BEQ     6$
4832  023326  014137  001124                    MOV     -(R1),$GDDAT
4833  023332  014237  001126                    MOV     -(R2),$BDDAT
4834
4835  023336  104012                            ERROR   12
4836  023340                            20$:
4837
4838
4839                    ;*****************************************************************
4840                    ;*TEST 41       *TEST THAT THE MULTI-USER MICRO-CODE WILL EXERCISE DR11K #2
4841                    ;*
4842                    ;*IN THIS TEST WE'LL MAKE SURE THAT DR11K #2 WILL PASS DATA.
4843                    ;*SR1 BITS SELECTS THIS DR11K FOR TESt. SR2 BITS INDICTES
4844                    ;*THAT THE INPUT IS CALLED BACK TO THE OUTPUT.
4845                    ;*
4846                    ;*      NOTE: THIS TEST IS A MULTI-USER MICRO-CODE TEST ONLY!
4847                    ;*            IF DEDICATE USER MICRO-CODE, THIS TEST WILL BE BY-PASSED.
4848                    ;*
4849                    ;*THIS TESt WAS DESIGNED IN ORDER TO FIND A WILD LPA-11 SYSTEM FAULT.
4850                    ;*I DON'T REALLY EXPECT AN ERROR HERE UNLESS YOU'VE GOT A HEAVY FAULT.
4851                    ;*ARBITRATION LOGIC IS ONE BIG REASON WHY YOU COULD FAIL. ANOTHER
4852                    ;*IS THAT INTERRRUPT VECTOR COULD BE WRONG. ALSO THE DMC MICRO-CODE MIGHT
4853                    ;*NOT BE EXECUTING PROPERLY.
4854                    ;*
4855                    ;*****************************************************************
```

```
4856  023340  000004                    TST41:  SCOPE
4857  023342  012737  000001  001160            MOV     #1,$TIMES       ;;DO 1 ITERATION
4858
4859                                            ;/-DRLT-
4860  023350  122737  000115  001772            CMPB    #'M,$VERSN      ;RUNNING MULTI-USER MICRO-CODE?
4861  023356  001402                            BEQ     1$              ;YES-CONTINUE.
4862  023360  000137  024064                    JMP     20$             ;NO-EXIT THIS TEST.
4863
4864  023364  012700  055066            1$:     MOV     #JOB0,R0        ;CLEAR OUT THE RDA TABLES
4865  023370  005020                    25$:    CLR     (R0)+
4866  023372  020027  055556                    CMP     R0,#JOB4
4867  023376  001374                            BNE     25$
4868
4869
4870  023400  032737  000040  001562            BIT     #BIT5,SR1       ;IS THIS DR11K SELECTED FOR TEST?
4871  023406  001002                            BNE     2$              ;YES-TEST IT.
4872  023410  000137  024064                    JMP     20$             ;NO-EXIT THIS TEST.
4873
4874  023414  012700  055322            2$:     MOV     #JOB2,R0        ;PICK UP THIS JOB ADDRESS.
4875  023420  012710  000012                    MOV     #BIT1!BIT3,(0)  ;SET START AND MULTI-USER.
4876  023424  052710  000420                    BIS     #BIT4!BIT8,(0) ;SELECT DIGITAL INPUT SINGLE CHAN.
4877  023430  012760  000401  000002    3$:     MOV     #257.,2(0)      ;SET WORD COUNT
4878  023436  012760  055420  000004            MOV     #JOB2U,4(0)     ;USW ADDR.
4879  023444  105060  000006                    CLRB    6(0)            ;NO EXT.
4880  023450  112760  000201  000007            MOVB    #201,7(0)       ;MAKE TWO BUFFERS AVAIL.
4881  023456  012760  056246  000010            MOV     #BUFF0,10(0)    ;1ST BUFFER ADDR.
4882  023464  012760  057250  000014            MOV     #BUFF1,14(0)    ;2ND BUFFER ADDR.
4883  023472  005060  000012                    CLR     12(0)           ;NO EXT.
4884  023476  012760  000200  000054            MOV     #200,54(0)      ;DELAY =200 TICKS.
4885  023504  012760  000001  000056            MOV     #2-1,56(0)      ;SET CHAN #=DR11K#-1.
4886  023512  012760  000001  000060            MOV     #1,60(0)        ;SAMPLING ONLY ONE CHANNEL.
4887  023520  012760  000001  000062            MOV     #1,62(0)        ;ON PULSE BETWEEN SAMPLE IS NO EXTERNAL TRIG.
4888  023526  005060  000064                    CLR     64(0)           ;NO START/EVENT MAKR WORD
4889  023532  005060  000066                    CLR     66(0)           ;NO START MASK
4890  023536  005060  000070                    CLR     70(0)           ;NO EVENT MASK.
4891
4892  023542  012701  055204                    MOV     #JOB1,R1        ;OK-NOW LETS SET UP THE OUTPUT.
4893  023546  012711  000012                    MOV     #BIT1!BIT3,(1)  ;SET START AND MULTI USER.
4894  023552  052711  000620                    BIS     #BIT4!BIT7!BIT8,(1) ;SELECT DIGITAL OUT.
4895  023556  012761  000401  000002            MOV     #257.,2(1)      ;WORD COUNT
4896  023564  012761  055302  000004            MOV     #JOB1U,4(1)     ;SET USW ADDR.
4897  023572  105061  000006                    CLRB    6(1)            ;NO EXT.
4898  023576  112761  000202  000007            MOVB    #202,7(1)
4899  023604  012761  060252  000010            MOV     #BUFF2,10(1)    ;SET BUFFER #1 ADDR.
4900  023612  012761  061254  000014            MOV     #BUFF3,14(1)    ;SET BUFFER #2 ADDR.
4901  023620  012761  060252  000020            MOV     #BUFF2,20(1)    ;THE THIRD OUTPUT IS NECESSARY
4902                                            ;TO COMPENSATE FOR SAMPLES LOST
4903  023626  105061  000012                    CLRB    12(1)           ;NO EXT .
4904  023632  012761  000200  000054            MOV     #200,54(1)      ;DELAY 1 TICK.
4905  023640  012761  000001  000056            MOV     #2-1,56(1)      ;SET CHAN #= DR#-1
4906  023646  012761  000001  000060            MOV     #1,60(1)        ;SAMPLE ONLY ONE CHAN.
4907  023654  012761  000001  000062            MOV     #1,62(1)        ;ONE TICK BETWEEN SAMPLE
4908  023662  005061  000064                    CLR     64(1)           ;NO START/EVENT
4909  023666  005061  000066                    CLR     66(1)           ;NO START MASK.
```

```
4910  023672  005061  000070                CLR     70(1)           ;NO EVENT MASK.
4911
4912  023676  012700  125252                MOV     #125252,R0      ;FILL OUTPUT BUFFER W/PATTERN
4913  023702  012701  001002                MOV     #514.,R1
4914  023706  012702  060252                MOV     #BUFF2,R2
4915  023712  010022              4$:       MOV     R0,(2)+
4916  023714  005100                        COM     R0
4917  023716  005301                        DEC     R1
4918  023720  001374                        BNE     4$
4919
4920  023722  012700  056246                MOV     #BUFF0,R0       ;CLEAR INPUT BUFFER
4921  023726  012701  001002                MOV     #514.,R1
4922  023732  005020              5$:       CLR     (0)+
4923  023734  005301                        DEC     R1
4924  023736  001375                        BNE     5$
4925
4926  023740  012737  000011  055066        MOV     #11,JOB0        ;OK,NOW WE GOTTO STOP THE CLOCK
4927  023746  012737  000000  055070        MOV     #0,JOB0+2       ;SO THAT IT DOESN'T START UNTIL
4928  023754  012737  000011  055440        MOV     #11,JOB3        ;DRIIK PARAMETER SET UP,THEN AS
4929  023762  012737  000503  055442        MOV     #503,JOB3+2     ;JOB #4,WE START CLOCK.
4930  023770  012737  177000  055444        MOV     #177000,JOB3+4
4931  023776  012737  000004  001124        MOV     #4,$GDDAT       ;FOUR JOBS.
4932  024004  012737  000002  002014        MOV     #2,FUDGE        ;/SET UP A FUDGE FACTOR TO
4933                                                                ;/COMPENSATE FOR THE TWO KW11K JOBS
4934  024012  004737  036026                JSR PC, FLASH           ;/GO DO THEM!
4935
4936  024016  032737  000040  001564        BIT     #BIT5,SR2       ;/CABLED TOGETHER?
4937  024024  001417                        BEQ     TST42           ;;
4938  024026  012700  001003                MOV     #515.,R0        ;/NOW  CHECK DATA IF CABLED.
4939  024032  012701  060252                MOV     #BUFF2,R1
4940  024036  012702  056246                MOV     #BUFF0,R2
4941  024042  005300              6$:       DEC     R0
4942  024044  001407                        BEQ     TST42           ;;
4943  024046  022221                        CMP     (R2)+,(R1)+
4944  024050  001774                        BEQ     6$
4945  024052  014137  001124                MOV     -(R1),$GDDAT
4946  024056  014237  001126                MOV     -(R2),$BDDAT
4947
4948  024062  104012                        ERROR   12
4949  024064              20$:
4950
4951
4952                                ;*****************************************************************
4953                                ;*TEST 42       *TEST THAT THE MULTI-USER MICRO-CODE WILL EXERCISE DR11K #3
4954                                ;*
4955                                ;*IN THIS TEST WE'LL MAKE SURE THAT DR11K #3 WILL PASS DATA.
4956                                ;*SR1 BIT7 SELECTS THIS DR11K FOR TEST. SR2 BIT7 INDICTES
4957                                ;*THAT THE INPUT IS CALLED BACK TO THE OUTPUT.
4958                                ;*
4959                                ;*      NOTE: THIS TEST IS A MULTI-USER MICRO-CODE TEST ONLY!
4960                                ;*            IF DEDICATE USER MICRO-CODE, THIS TEST WILL BE BY-PASSED.
4961                                ;*
4962                                ;*THIS TEST WAS DESIGNED IN ORDER TO FIND A WILD LPA-11 SYSTEM FAULT.
4963                                ;*I DON'T REALLY EXPECT AN ERROR HERE UNLESS YOU'VE GOT A HEAVY FAULT.
```

```
                                        ;*ARBITRATION LOGIC IS ONE BIG REASON WHY YOU COULD FAIL. ANOTHER
                                        ;*IS THAT INTERRRUPT VECTOR COULD BE WRONG. ALSO THE DMC MICRO-CODE MIGHT
                                        ;*NOT BE EXECUTING PROPERLY.
                                        ;*
                                        ;;*****************************************************************
024064  000004                  TST42:  SCOPE
024066  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION

                                                                ;/-DRLT-
024074  122737  000115  001772          CMPB    #'M,$VERSN      ;RUNNING MULTI-USER MICRO-CODE?
024102  001402                          BEQ     1$              ;YES-CONTINUE.
024104  000137  024610                  JMP     20$             ;NO-EXIT THIS TEST.

024110  012700  055066          1$:     MOV     #JOB0,R0        ;CLEAR OUT THE RDA TABLES
024114  005020                  25$:    CLR     (R0)+
024116  020027  055556                  CMP     R0,#JOB4
024122  001374                          BNE     25$

024124  032737  000200  001562          BIT     #BIT7,SR1       ;IS THIS DR11K SELECTED FOR TEST?
024132  001002                          BNE     2$              ;YES-TEST IT.
024134  000137  024610                  JMP     20$             ;NO-EXIT THIS TEST.

024140  012700  055322          2$:     MOV     #JOB2,R0        ;PICK UP THIS JOB ADDRESS.
024144  012710  000012                  MOV     #BIT1!BIT3,(0)  ;SET START AND MULTI-USER.
024150  052710  000420                  BIS     #BIT4!BIT8,(0) ;SELECT DIGITAL INPUT SINGLE CHAN.
024154  012760  000401  000002  3$:     MOV     #257.,2(0)      ;SET WORD COUNT
024162  012760  055420  000004          MOV     #JOB2U,4(0)     ;USW ADDR.
024170  105060  000006                  CLRB    6(0)            ;NO EXT.
024174  112760  000201  000007          MOVB    #201,7(0)       ;MAKE TWO BUFFERS AVAIL.
024202  012760  056246  000010          MOV     #BUFF0,10(0)    ;1ST BUFFER ADDR.
024210  012760  057250  000014          MOV     #BUFF1,14(0)    ;2ND BUFFER ADDR.
024216  005060  000012                  CLR     12(0)           ;NO EXT.
024222  012760  000200  000054          MOV     #200,54(0)      ;DELAY =200 TICKS.
024230  012760  000002  000056          MOV     #3-1,56(0)      ;SET CHAN #=DR11K#-1.
024236  012760  000001  000060          MOV     #1,60(0)        ;SAMPLING ONLY ONE CHANNEL.
024244  012760  000001  000062          MOV     #1,62(0)        ;ON PULSE BETWEEN SAMPLE IS NO EXTERNAL TRIG.
024252  005060  000064                  CLR     64(0)           ;NO START/EVENT MAKR WORD
024256  005060  000066                  CLR     66(0)           ;NO START MASK
024262  005060  000070                  CLR     70(0)           ;NO EVENT MASK.

024266  012701  055204                  MOV     #JOB1,R1        ;OK-NOW LETS SET UP THE OUTPUT.
024272  012711  000012                  MOV     #BIT1!BIT3,(1)  ;SET START AND MULTI USER.
024276  052711  000620                  BIS     #BIT4!BIT7!BIT8,(1) ;SELECT DIGITAL OUT.
024302  012761  000401  000002          MOV     #257.,2(1)      ;WORD COUNT
024310  012761  055302  000004          MOV     #JOB1U,4(1)     ;SET USW ADDR.
024316  105061  000006                  CLRB    6(1)            ;NO EXT.
024322  112761  000202  000007          MOVB    #202,7(1)
024330  012761  060252  000010          MOV     #BUFF2,10(1)    ;SET BUFFER #1 ADDR.
024336  012761  061254  000014          MOV     #BUFF3,14(1)    ;SET BUFFER #2 ADDR.
024344  012761  060252  000020          MOV     #BUFF2,20(1)    ;THE THIRD OUTPUT IS NECESSARY
                                                                ;TO COMPENSATE FOR SAMPLES LOST
024352  105061  000012                  CLRB    12(1)           ;NO EXT.
024356  012761  000200  000054          MOV     #200,54(1)      ;DELAY 1 TICK.
```

```
5018  024364  012761  000002  000056          MOV     #3-1,56(1)      ;SET CHAN #= DR#-1
5019  024372  012761  000001  000060          MOV     #1,60(1)        ;SAMPLE ONLY ONE CHAN.
5020  024400  012761  000001  000062          MOV     #1,62(1)        ;ONE TICK BETWEEN SAMPLE
5021  024406  005061  000064                  CLR     64(1)           ;NO START/EVENT
5022  024412  005061  000066                  CLR     66(1)           ;NO START MASK.
5023  024416  005061  000070                  CLR     70(1)           ;NO EVENT MASK.
5024
5025  024422  012700  125252                  MOV     #125252,R0      ;FILL OUTPUT BUFFER W/PATTERN
5026  024426  012701  001002                  MOV     #514.,R1
5027  024432  012702  060252                  MOV     #BUFF2,R2
5028  024436  010022                  4$:     MOV     R0,(2)+
5029  024440  005100                          COM     R0
5030  024442  005301                          DEC     R1
5031  024444  001374                          BNE     4$
5032
5033  024446  012700  056246                  MOV     #BUFF0,R0       ;CLEAR INPUT BUFFER
5034  024452  012701  001002                  MOV     #514.,R1
5035  024456  005020                  5$:     CLR     (0)+
5036  024460  005301                          DEC     R1
5037  024462  001375                          BNE     5$
5038
5039  024464  012737  000011  055066          MOV     #11,JOB0        ;OK,NOW WE GOTTO STOP THE CLOCK
5040  024472  012737  000000  055070          MOV     #0,JOB0+2       ;SO THAT IT DOESN'T START UNTIL
5041  024500  012737  000011  055440          MOV     #11,JOB3        ;DR11K PARAMETER SET UP,THEN AS
5042  024506  012737  000503  055442          MOV     #503,JOB3+2     ;JOB #4,WE START CLOCK.
5043  024514  012737  177000  055444          MOV     #177000,JOB3+4
5044  024522  012737  000004  001124          MOV     #4,$GDDAT       ;FOUR JOBS.
5045  024530  012737  000002  002014          MOV     #2,FUDGE        ;/SET UP A FUDGE FACTOR TO
5046                                                                  ;/COMPENSATE FOR THE TWO KW11K JOBS
5047  024536  004737  036026                  JSR PC, FLASH           ;/GO DO THEM!
5048
5049  024542  032737  000200  001564          BIT     #BIT7,SR2       ;/CABLED TOGETHER?
5050  024550  001417                          BEQ     TST43           ;;
5051  024552  012700  001003                  MOV     #515.,R0        ;/NOW  CHECK DATA IF CABLED.
5052  024556  012701  060252                  MOV     #BUFF2,R1
5053  024562  012702  056246                  MOV     #BUFF0,R2
5054  024566  005300                  6$:     DEC     R0
5055  024570  001407                          BEQ     TST43           ;;
5056  024572  022221                          CMP     (R2)+,(R1)+
5057  024574  001774                          BEQ     6$
5058  024576  014137  001124                  MOV     -(R1),$GDDAT
5059  024602  014237  001126                  MOV     -(R2),$BDDAT
5060
5061  024606  104012                          ERROR   12
5062  024610                          20$:
5063
5064
5065                                    ;;*****************************************************************
5066                                    ;*TEST 43       *TEST THAT THE MULTI-USER MICRO-CODE WILL EXERCISE DR11K #4
5067                                    ;*
5068                                    ;*IN THIS TEST WE'LL MAKE SURE THAT DR11K #4 WILL PASS DATA.
5069                                    ;*SR1 BIT8 SELECTS THIS DR11K FOR TEST. SR2 BIT8 INDICTES
5070                                    ;*THAT THE INPUT IS CALLED BACK TO THE OUTPUT.
5071                                    ;*
```

```
5072                                    ;*      NOTE: THIS TEST IS A MULTI-USER MICRO-CODE TEST ONLY!
5073                                    ;*            IF DEDICATE USER MICRO-CODE, THIS TE.  WILL BE BY-PASSED.
5074                                    ;*
5075                                    ;*THIS TEST WAS DESIGNED IN ORDER TO FIND A WILD LPA-11 SYSTEM FAULT.
5076                                    ;*I DON'T REALLY EXPECT AN ERROR HERE UNLESS YOU'VE GOT A HEAVY FAULT.
5077                                    ;*ARBITRATION LOGIC IS ONE BIG REASON WHY YOU COULD FAIL. ANOTHER
5078                                    ;*IS THAT INTERRRUPT VECTOR COULD BE WRONG. ALSO THE DMC MICRO-CODE MIGHT
5079                                    ;*NOT BE EXECUTING PROPERLY.
5080                                    ;*
5081                                    ;**************************************************************************
5082  024610  000004                    TST43:  SCOPE
5083  024612  012737  000001  001160            MOV     #1,$TIMES       ;;DO 1 ITERATION
5084
5085                                            ;/-DRLT-
5086  024620  122737  000115  001772            CMPB    #'M,$VERSN      ;RUNNING MULTI-USER MICRO-CODE?
5087  024626  001402                            BEQ     1$              ;YES-CONTINUE.
5088  024630  000137  025334                    JMP     20$             ;NO-EXIT THIS TEST.
5089
5090  024634  012700  055066            1$:     MOV     #JOB0,R0        ;CLEAR OUT THE RDA TABLES
5091  024640  005020                    25$:    CLR     (R0)+
5092  024642  020027  055556                    CMP     R0,#JOB4
5093  024646  001374                            BNE     25$
5094
5095
5096  024650  032737  000400  001562            BIT     #BIT8,SR1       ;IS THIS DR11K SELECTED FOR TEST?
5097  024656  001002                            BNE     2$              ;YES-TEST IT.
5098  024660  000137  025334                    JMP     20$             ;NO-EXIT THIS TEST.
5099
5100  024664  012700  055322            2$:     MOV     #JOB2,R0        ;PICK UP THIS JOB ADDRESS.
5101  024670  012710  000012                    MOV     #BIT1!BIT3,(0)  ;SET START AND MULTI-USER.
5102  024674  052710  000420                    BIS     #BIT4!BIT8,(0) ;SELECT DIGITAL INPUT SINGLE CHAN.
5103  024700  012760  000401  000002    3$:     MOV     #257.,2(0)      ;SET WORD COUNT
5104  024706  012760  055420  000004            MOV     #JOB2U,4(0)     ;USW ADDR.
5105  024714  105060  000006                    CLRB    6(0)            ;NO EXT.
5106  024720  112760  000201  000007            MOVB    #201,7(0)       ;MAKE TWO BUFFERS AVAIL.
5107  024726  012760  056246  000010            MOV     #BUFF0,10(0)    ;1ST BUFFER ADDR.
5108  024734  012760  057250  000014            MOV     #BUFF1,14(0)    ;2ND BUFFER ADDR.
5109  024742  005060  000012                    CLR     12(0)           ;NO EXT.
5110  024746  012760  000200  000054            MOV     #200,54(0)      ;DELAY =200 TICKS.
5111  024754  012760  000003  000056            MOV     #4-1,56(0)      ;SET CHAN #=DR11K#-1.
5112  024762  012760  000001  000060            MOV     #1,60(0)        ;SAMPLING ONLY ONE CHANNEL.
5113  024770  012760  000001  000062            MOV     #1,62(0)        ;ON PULSE BETWEEN SAMPLE IS NO EXTERNAL TRIG.
5114  024776  005060  000064                    CLR     64(0)           ;NO START/EVENT MAKR WORD
5115  025002  005060  000066                    CLR     66(0)           ;NO START MASK
5116  025006  005060  000070                    CLR     70(0)           ;NO EVENT MASK.
5117
5118  025012  012701  055204                    MOV     #JOB1,R1        ;OK-NOW LETS SET UP THE OUTPUT.
5119  025016  012711  000012                    MOV     #BIT1!BIT3,(1)  ;SET START AND MULTI USER.
5120  025022  052711  000620                    BIS     #BIT4!BIT7!BIT8,(1) ;SELECT DIGITAL OUT.
5121  025026  012761  000401  000002            MOV     #257.,2(1)      ;WORD COUNT
5122  025034  012761  055302  000004            MOV     #JOB1U,4(1)     ;SET USW ADDR.
5123  025042  105061  000006                    CLRB    6(1)            ;NO EXT.
5124  025046  112761  000202  000007            MOVB    #202,7(1)
5125  025054  012761  060252  000010            MOV     #BUFF2,10(1)    ;SET BUFFER #1 ADDR.
```

```
5126  025062  012761  061254  000014          MOV     #BUFF3,14(1)    ;SET BUFFER #2 ADDR.
5127  025070  012761  060252  000020          MOV     #BUFF2,20(1)    ;THE THIRD OUTPUT IS NECESSARY
5128                                                                  ;TO COMPENSATE FOR SAMPLES LOST
5129  025076  105061  000012                  CLRB    12(1)           ;NO EXT .
5130  025102  012761  000200  000054          MOV     #200,54(1)      ;DELAY 1 TICK.
5131  025110  012761  000003  000056          MOV     #4-1,56(1)      ;SET CHAN #= DR#-1
5132  025116  012761  000001  000060          MOV     #1,60(1)        ;SAMPLE ONLY ONE CHAN.
5133  025124  012761  000001  000062          MOV     #1,62(1)        ;ONE TICK BETWEEN SAMPLE
5134  025132  005061  000064                  CLR     64(1)           ;NO START/EVENT
5135  025136  005061  000066                  CLR     66(1)           ;NO START MASK.
5136  025142  005061  000070                  CLR     70(1)           ;NO EVENT MASK.
5137
5138  025146  012700  125252                  MOV     #125252,R0      ;FILL OUTPUT BUFFER W/PATTERN
5139  025152  012701  001002                  MOV     #514.,R1
5140  025156  012702  060252                  MOV     #BUFF2,R2
5141  025162  010022                  4$:     MOV     R0,(2)+
5142  025164  005100                          COM     R0
5143  025166  005301                          DEC     R1
5144  025170  001374                          BNE     4$
5145
5146  025172  012700  056246                  MOV     #BUFF0,R0       ;CLEAR INPUT BUFFER
5147  025176  012701  001002                  MOV     #514.,R1
5148  025202  005020                  5$:     CLR     (0)+
5149  025204  005301                          DEC     R1
5150  025206  001375                          BNE     5$
5151
5152  025210  012737  000011  055066          MOV     #11,JOB0        ;OK,NOW WE GOTTO STOP THE CLOCK
5153  025216  012737  000000  055070          MOV     #0,JOB0+2       ;SO THAT IT DOESN'T START UNTIL
5154  025224  012737  000011  055440          MOV     #11,JOB3        ;DRIIK PARAMETER SET UP,THEN AS
5155  025232  012737  000503  055442          MOV     #503,JOB3+2     ;JOB #4,WE START CLOCK.
5156  025240  012737  177000  055444          MOV     #177000,JOB3+4
5157  025246  012737  000004  001124          MOV     #4,$GDDAT       ;FOUR JOBS.
5158  025254  012737  000002  002014          MOV     #2,FUDGE        ;/SET UP A FUDGE FACTOR TO
5159                                                                  ;/COMPENSATE FOR THE TWO KW11K JOBS
5160  025262  004737  036026                  JSR PC, FLASH           ;/GO DO THEM!
5161
5162  025266  032737  000400  001564          BIT     #BIT8,SR2       ;/CABLED TOGETHER?
5163  025274  001417                          BEQ     TST44           ;;
5164  025276  012700  001003                  MOV     #515.,R0        ;/NOW  CHECK DATA IF CABLED.
5165  025302  012701  060252                  MOV     #BUFF2,R1
5166  025306  012702  056246                  MOV     #BUFF0,R2
5167  025312  005300                  6$:     DEC     R0
5168  025314  001407                          BEQ     TST44           ;;
5169  025316  022221                          CMP     (R2)+,(R1)+
5170  025320  001774                          BEQ     6$
5171  025322  014137  001124                  MOV     -(R1),$GDDAT
5172  025326  014237  001126                  MOV     -(R2),$BDDAT
5173
5174  025332  104012                          ERROR   12
5175  025334                          20$:
5176
5177
5178                                  ;;**********************************************************************
5179                                  ;*TEST 44        *TEST THAT THE MULTI-USER MICRO-CODE WILL EXERCISE DR11K #5
```

```
                                ;*
                                ;*IN THIS TEST WE'LL MAKE SURE THAT DR11K #5 WILL PASS DATA.
                                ;*SR1 BIT9 SELECTS THIS DR11K FOR TEST. SR2 BIT9 INDICTES
                                ;*THAT THE INPUT IS CALLED BACK TO THE OUTPUT.
                                ;*
                                ;*      NOTE: THIS TEST IS A MULTI-USER MICRO-CODE TEST ONLY!
                                ;*            IF DEDICATE USER MICRO-CODE, THIS TEST WILL BE BY-PASSED.
                                ;*
                                ;*THIS TEST WAS DESIGNED IN ORDER TO FIND A WILD LPA-11 SYSTEM FAULT.
                                ;*I DON'T REALLY EXPECT AN ERROR HERE UNLESS YOU'VE GOT A HEAVY FAULT.
                                ;*ARBITRATION LOGIC IS ONE BIG REASON WHY YOU COULD FAIL. ANOTHER
                                ;*IS THAT INTERRRUPT VECTOR COULD BE WRONG. ALSO THE DMC MICRO-CODE MIGHT
                                ;*NOT BE EXECUTING PROPERLY.
                                ;*
                                ;.*****************************************************************
025334  000004          TST44:  SCOPE
025336  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
                                                                ;/-DRLT-
025344  122737  000115  001772          CMPB    #'M,$VERSN      ;RUNNING MULTI-USER MICRO-CODE?
025352  001402                          BEQ     1$              ;YES-CONTINUE.
025354  000137  026060                  JMP     20$             ;NO-EXIT THIS TEST.

025360  012700  055066          1$:     MOV     #JOB0,R0        ;CLEAR OUT THE RDA TABLES
025364  005020          25$:    CLR     (R0)+
025366  020027  055556                  CMP     R0,#JOB4
025372  001374                          BNE     25$

025374  032737  001000  001562          BIT     #BIT9,SR1       ;IS THIS DR11K SELECTED FOR TEST?
025402  001002                          BNE     2$              ;YES-TEST IT.
025404  000137  026060                  JMP     20$             ;NO-EXIT THIS TEST.

025410  012700  055322          2$:     MOV     #JOB2,R0        ;PICK UP THIS JOB ADDRESS.
025414  012710  000012                  MOV     #BIT1!BIT3,(0)  ;SET START AND MULTI-USER.
025420  052710  000420                  BIS     #BIT4!BIT8,(0)  ;SELECT DIGITAL INPUT SINGLE CHAN.
025424  012760  000401  000002  3$:     MOV     #257.,2(0)      ;SET WORD COUNT
025432  012760  055420  000004          MOV     #JOB20,4(0)     ;USW ADDR.
025440  105060  000006                  CLRB    6(0)            ;NO EXT.
025444  112760  000201  000007          MOVB    #201,7(0)       ;MAKE TWO BUFFERS AVAIL.
025452  012760  056246  000010          MOV     #BUFF0,10(0)    ;1ST BUFFER ADDR.
025460  012760  057250  000014          MOV     #BUFF1,14(0)    ;2ND BUFFER ADDR.
025466  005060  000012                  CLR     12(0)           ;NO EXT.
025472  012760  000200  000054          MOV     #200,54(0)      ;DELAY =200 TICKS.
025500  012760  000004  000056          MOV     #5-1,56(0)      ;SET CHAN #=DR11K#-1.
025506  012760  000001  000060          MOV     #1,60(0)        ;SAMPLING ONLY ONE CHANNEL.
025514  012760  000001  000062          MOV     #1,62(0)        ;ON PULSE BETWEEN SAMPLE IS NO EXTERNAL TRIG.
025522  005060  000064                  CLR     64(0)           ;NO START/EVENT MAKR WORD
025526  005060  000066                  CLR     66(0)           ;NO START MASK
025532  005060  000070                  CLR     70(0)           ;NO EVENT MASK.

025536  012701  055204                  MOV     #JOB1,R1        ;OK-NOW LETS SET UP THE OUTPUT.
025542  012711  000012                  MOV     #BIT1!BIT3,(1)  ;SET START AND MULTI USER.
025546  052711  000620                  BIS     #BIT4!BIT7!BIT8,(1) ;SELECT DIGITAL OUT.
```

```
5234  025552  012761  000401  000002          MOV     #257.,2(1)      ;WORD COUNT
5235  025560  012761  055302  000004          MOV     #JOB1U,4(1)     ;SET USW ADDR.
5236  025566  105061  000006                  CLRB    6(1)            ;NO EXT.
5237  025572  112761  000202  000007          MOVB    #202,7(1)
5238  025600  012761  060252  000010          MOV     #BUFF2,10(1)    ;SET BUFFER #1 ADDR.
5239  025606  012761  061254  000014          MOV     #BUFF3,14(1)    ;SET BUFFER #2 ADDR.
5240  025614  012761  060252  000020          MOV     #BUFF2,20.(1)   ;THE THIRD OUTPUT IS NECESSARY
5241                                                                  ;TO COMPENSATE FOR SAMPLES LOST
5242  025622  105061  000012                  CLRB    12(1)           ;NO EXT .
5243  025626  012761  000200  000054          MOV     #200,54(1)      ;DELAY 1 TICK.
5244  025634  012761  000004  000056          MOV     #5-1,56(1)      ;SET CHAN #= DR#-1
5245  025642  012761  000001  000060          MOV     #1,60(1)        ;SAMPLE ONLY ONE CHAN.
5246  025650  012761  000001  000062          MOV     #1,62(1)        ;ONE TICK BETWEEN SAMPLE
5247  025656  005061  000064                  CLR     64(1)           ;NO START/EVENT
5248  025662  005061  000066                  CLR     66(1)           ;NO START MASK.
5249  025666  005061  000070                  CLR     70(1)           ;NO EVENT MASK.
5250
5251  025672  012700  125252                  MOV     #125252,R0      ;FILL OUTPUT BUFFER W/PATTERN
5252  025676  012701  001002                  MOV     #514.,R1
5253  025702  012702  060252                  MOV     #BUFF2,R2
5254  025706  010022                  4$:     MOV     R0,(2)+
5255  025710  005100                          COM     R0
5256  025712  005301                          DEC     R1
5257  025714  001374                          BNE     4$
5258
5259  025716  012700  056246                  MOV     #BUFF0,R0       ;CLEAR INPUT BUFFER
5260  025722  012701  001002                  MOV     #514.,R1
5261  025726  005020                  5$:     CLR     (0)+
5262  025730  005301                          DEC     R1
5263  025732  001375                          BNE     5$
5264
5265  025734  012737  000011  055066          MOV     #11,JOB0        ;OK,NOW WE GOTTO STOP THE CLOCK
5266  025742  012737  000000  055070          MOV     #0,JOB0+2       ;SO THAT IT DOESN'T START UNTIL
5267  025750  012737  000011  055440          MOV     #11,JOB3        ;DRIIK PARAMETER SET UP,THEN AS
5268  025756  012737  000503  055442          MOV     #503,JOB3+2     ;JOB #4,WE START CLOCK.
5269  025764  012737  177000  055444          MOV     #177000,JOB3+4
5270  025772  012737  000004  001124          MOV     #4,$GDDAT       ;FOUR JOBS.
5271  026000  012737  000002  002014          MOV     #2,FUDGE        ;/SET UP A FUDGE FACTOR TO
5272                                                                  ;/COMPENSATE FOR THE TWO KW11K JOBS
5273  026006  004737  036026                  JSR PC, FLASH           ;/GO DO THEM!
5274
5275  026012  032737  001000  001564          BIT     #BIT9,SR2       ;/CABLED TOGETHER?
5276  026020  001417                          BEQ     TST45           ;;
5277  026022  012700  001003                  MOV     #515.,R0        ;/NOW  CHECK DATA IF CABLED.
5278  026026  012701  060252                  MOV     #BUFF2,R1
5279  026032  012702  056246                  MOV     #BUFF0,R2
5280  026036  005300                  6$:     DEC     R0
5281  026040  001407                          BEQ     TST45           ;;
5282  026042  022221                          CMP     (R2)+,(R1)+
5283  026044  001774                          BEQ     6$
5284  026046  014137  001124                  MOV     -(R1),$GDDAT
5285  026052  014237  001126                  MOV     -(R2),$BDDAT
5286
5287  026056  104012                          ERROR   12
```

```
                                20$:
                                ;*****************************************************************
                                ;*TEST 45        *TEST THE LPA SYSTEM USER MICRO-CODE'S ABILITY TO TAKE SAMPLE SINGLE JOB
                                ;*
                                ;*IN THIS TEST WE'LL VERIFY ANOLOG SAMPLING ABILITIES.
                                ;*DATA WILL BE TAKEN ON ALL CHANNELLS THROUGH ONE JOB.
                                ;*AT THE CONCLUSION, IF A WRAP AROUND MODULE WAS ON THE ANOLOG
                                ;*MODULE, A CHECK WILL BE MADE ON SPECIFIC CHANNELLS
                                ;*FOR SPECIFIC VALUES.
                                ;*
                                ;*
                                ;*THIS TEST WAS DESIGNED IN ORDER TO FIND A WILD LPA-11 SYSTEM FAULT.
                                ;*I DON'T REALLY EXPECT AN ERROR HERE UNLESS YOU'VE GOT A HEAVY FAULT.
                                ;*ARBITRATION LOGIC IS ONE BIG REASON WHY YOU COULD FAIL. ANOTHER
                                ;*IS THAT INTERRRUPT VECTOR COULD BE WRONG. ALSO THE DMC MICRO-CODE MIGHT
                                ;*NOT BE EXECUTING PROPERLY.
                                ;*
                                ;*****************************************************************
026060  000004               TST45:  SCOPE

026062  032737  012001  001562        BIT     #BIT12!BIT10!BIT0,SR1 ;IS THERE AN ANOLOG DEVICE?
026070  001002                       BNE     1$
026072  000137  026416               JMP     20$
026076  012737  170000  044116  1$:  MOV     #170000,KWT+4   ;RESTART CLOCK AT 1 MEGHZ RATE
026104  012777  044112  153364        MOV     #KWT,@LPADL
026112  052777  000001  153346        BIS     #BIT0,@LPCI
026120  012700  055066               MOV     #JOB0,R0        ;CLEAR OUT JOB AREA.
026124  005020                 10$:  CLR     (0)+
026126  022700  056230               CMP     #JOB7R,R0
026132  001374                       BNE     10$

026134  012700  055066               MOV     #JOB0,R0        ;GET SET TO DO SINGLE JOB RDA
026140  012710  000002               MOV     #2,(0)
026144  052710  001000         2$:   BIS     #BIT9,(0)       ;SET SEQUENTIAL CHAN SAMPLE
026150  122737  000104  001772        CMPB    #'D,$VERSN      ;MULTI USER MICRO-CODE?
026156  001402                       BEQ     3$
026160  052710  000010               BIS     #BIT3,(0)       ;YES-TELL RDA MODE WORD.
026164  012760  000401  000002  3$:  MOV     #257.,2(0)      ;SET BUFFER SIZE.
026172  012760  055164  000004        MOV     #JOB0U,4(0)     ;SET USW WORD.
026200  005070  000004               CLR     @4(0)           ;CLEAR USW WORD.
026204  012760  056246  000010        MOV     #BUFF0,10(0)    ;NOW SET BUFFER ADDRESS
026212  012760  057250  000014        MOV     #BUFF1,14(0)
026220  012760  060252  000020        MOV     #BUFF2,20(0)
026226  112760  000002  000007        MOVB    #2,7(0)         ;INDICATE HOW MANY BUFFERS USED.
026234  122737  000104  001772        CMPB    #'D,$VERSN      ;IF THIS IS DEDICATED MODE,
026242  001003                       BNE     31$             ;WE MUST LET BUFFER OVERRUN BE NON-FATAL.
026244  152760  000200  000007        BISB    #BIT7,7(0)      ;DEDICATED MODE IS JUST TO FAST FOR
                                                             ;THE SIZE BUFFER AND STYLE WE USE HERE.
026252                         31$:
026252  012760  000144  000054        MOV     #100.,54(0)     ;SET DELAY BEFORE START
026260  012760  000020  000060        MOV     #16.,60(0)      ;SET # OF CHANNELS
```

```
026266  012760  000001  000062  4$:     MOV     #+1,62(0)               ;# OF TICKS BETWEEN SAMPLES
026274  012760  000400  000056          MOV     #400,56(0)              ;START CH 0, INC=1
026302  016037  000060  034454          MOV     60(0),CHANU
026310  012737  000001  001124          MOV     #1,$GODAT               ;ONLY ONE JOB.
026316  004737  036026                  JSR     PC,FLASH                ;GO-DO IT.
026322  032737  002021  001564          BIT     #BIT0!BIT4!BIT10,SR2 ;G5034 WRAPAROUND MODULE?
026330  001432                          BEQ     TST46                   ;
                                                                        ;YES-LETS CHECK OUT THE RESULTS!
026332  005237  034540                  INC     REPOR
026336  012700  055066                  MOV     #JOB0,R0
026342  012703  055006                  MOV     #LIST10,R3              ;GET LIST OF S/B RESULTS FOR 10 BITS.
026346  032737  002000  001562          BIT     #BIT10,SR1                      ;SEC IF 12 BIT A/D
026354  001002                          BNE     5$
026356  012703  055036                  MOV     #LIST12,R3              ;GET LIST OF 12 BIT A/D RESULTS.
026362  012702  000004          5$:     MOV     #4.,R2                  ;CHECK 4 CHANS
026366  011337  034474          6$:     MOV     (3),CHANS               ;PICK UP CHAN #
026372  012337  034476                  MOV     (3)+,CHANF              ;
026376  012337  034514                  MOV     (3)+,AVEXP
026402  012337  034512                  MOV     (3)+,TOLER
026406  004737  034146                  JSR     PC,AVERR
026412  005302                          DEC     R2
026414  001364                          BNE     6$

026416                          20$:

                                ;;**********************************************************
                                ;*TEST 46       *TEST THE LPA SYSTEM USER MICRO-CODE'S ABILITY TO TAKE RANDOM SAMPLES SI
                                ;*
                                ;*IN THIS TEST WE'LL VERIFY ANOLOG SAMPLING ABILITIES.
                                ;*DATA WILL BE TAKEN AN ALL CHANNELLS THROUGH ONE JOB.
                                ;*
                                ;*THE DIFFERENCE BETWEEN THIS TEST AND THE ONE BEFORE IF IS THAT
                                ;*DATA IS TAKEN IN RANDOM CHANNELL MODE INSTEAD OF SEQUENCAL.
                                ;*
                                ;*AT THE CONCLUSION, IF A WRAP AROUND MODULE WAS ON THE ANOLOG
                                ;*MODULE, A CHECK WILL BE MADE ON SPECIFIC CHANNELLS
                                ;*FOR SPECIFIC VALUES.
                                ;*
                                ;*
                                ;*THIS TEST WAS DESIGNED IN ORDER TO FIND A WILD LPA-11 SYSTEM FAULT.
                                ;*I DON'T REALLY EXPECT AN ERROR HERE UNLESS YOU'VE GOT A HEAVY FAULT.
                                ;*ARBITRATION LOGIC IS ONE BIG REASON WHY YOU COULD FAIL. ANOTHER
                                ;*IS THAT INTERRRUPT VECTOR COULD BE WRONG. ALSO THE DMC MICRO-CODE MIGHT
                                ;*NOT BE EXECUTING PROPERLY.
                                ;*
                                ;;**********************************************************
026416  000004                  TST46:  SCOPE

026420  032737  012001  001562          BIT     #BIT12!BIT10!BIT0,SR1 ;IS THERE AN ANOLOG DEVICE?
026426  001002                          BNE     1$
026430  000137  027014                  JMP     20$
026434  012737  170000  044116  1$:     MOV     #170000,KWT+4   ;RESTART CLOCK AT 1 MEGHZ RATE
026442  012777  044112  153026          MOV     #KWT,@LPADL
026450  052777  000001  153010          BIS     #BIT0,@LPCI
```

```
5396  026456  012700  055066            MOV   #JOBO,RO        ;CLEAR OUT JOB AREA.
5397  026462  005020            10$:    CLR   (O)+
5398  026464  022700  056230            CMP   #JOB7R,RO
5399  026470  001374                    BNE   10$
5400
5401  026472  012700  055066            MOV   #JOBO,RO        ;GET SET TO dO SINGLE JOB RDA
5402  026476  012710  000002            MOV   #2,(O)
5403  026502                    2$:
5404  026502  122737  000104  001772    CMPB  #'D,$VERSN      ;MULTI USER MICRO-CODE?
5405  026510  001402                    BEQ   3$
5406  026512  052710  000010            BIS   #BIT3,(O)       ;YES-TELL RDA MODE WORD.
5407  026516  012760  000401  000002 3$: MOV   #257.,2(O)     ;SET BUFFER SIZE.
5408  026524  012760  055164  000004    MOV   #JOBOU,4(O)     ;SET USW WORD.
5409  026532  005070  000004            CLR   @4(O)           ;CLEAR USW WORD.
5410  026536  012760  056246  000010    MOV   #BUFFO,10(O)    ;NOW SET BUFFER ADDRESS
5411  026544  012760  057250  000014    MOV   #BUFF1,14(O)
5412  026552  012760  060252  000020    MOV   #BUFF2,20(O)
5413  026560  112760  000002  000007    MOVB  #2,7(O)         ;INDICATE HOW MANY BUFFERS USED.
5414  026566  122737  000104  001772    CMPB  #'D,$VErSN      ;IF THIS IS DEDICATED MODE,
5415  026574  001003                    BNE   31$             ;WE MUST LET BUFFER OVERRUN BE NON-FATEL.
5416  026576  152760  000200  000007    BISB  #BIT7,7(O)      ;DEDICATED MODE IS JUST TO FAST FOR
5417                                                          ;THE SIZE BUFFER AND STYLE WE USE HERE.
5418  026604                    31$:
5419  026604  012760  000144  000054    MOV   #100.,54(O)     ;SET DELAY BEFORE START
5420  026612  012760  000007  000060    MOV   #7.,60(O)       ;SET # OF CHANNELS
5421  026620  012760  000010  000062 4$: MOV   #+10,62(O)      ;# OF TICKS BETWEEN SAMPLES
5422  026626  012760  055166  000050    MOV   #JOBOR,50(O)    ;SET ADDR OF RANDOM CHANNEL LIST
5423  026634  016037  000060  034454    MOV   60(O),CHANU
5424  026642  012700  055166            MOV   #JOBOR,RO       ;PICK UP POINT TO RANDOM LIST.
5425  026646  012720  000000            MOV   #0,(O)+         ;CHAN 0 FIRST
5426  026652  012720  000003            MOV   #3,(O)+         ;NEXT CHAN 3
5427  026656  012720  000002            MOV   #2,(O)+         ;CHAN 2 NEXT
5428  026662  012720  000003            MOV   #3,(O)+         ;CHAN 3
5429  026666  012720  000004            MOV   #4,(O)+         ;NEXT CHAN 4
5430  026672  012720  000015            MOV   #15,(O)+        ;CHAN 15
5431  026676  012710  000003            MOV   #3,(O)          ;CHAN 3 NEXT.
5432  026702  052720  140000            BIS   #BIT15!BIT14,(O)+;NOP AND END OF LIST.
5433  026706  012737  000001  001124    MOV   #1,$GDDAT       ;ONLY ONE JOB.
5434  026714  004737  036026            JSR   PC,FLASH        ;GO-DO IT.
5435  026720  032737  002021  001564    BIT   #BITO!BIT4!BIT10,SR2 ;G5034 WRAPAROUND MODULE?
5436  026726  001432                    BEQ   TST47           ;;
5437                                                          ;YES-LETS CHECK OUT THE RESULTS!
5438  026730  005237  034540            INC   REPOR
5439  026734  012700  055066            MOV   #JORO,RO
5440  026740  012703  055006            MOV   #LIST10,R3      ;GET LIST OF S/B RESULTS FOR 10 BITS.
5441  026744  032737  002000  001562    BIT   #BIT10,SR1              ;SEC IF 12 BIT A/D
5442  026752  001002                    BNE   5$
5443  026754  012703  055036            MOV   #LIST12,R3      ;GET LIST OF 12 BIT A/D RESULTS.
5444  026760  012702  000004      5$:   MOV   #4.,R2          ;CHECK 4 CHANS
5445  026764  011337  034474      6$:   MOV   (3),CHANS       ;PICK UP CHAN #
5446  026770  012337  034476            MOV   (3)+,CHANF      ;
5447  026774  012337  034514            MOV   (3)+,AVEXP
5448  027000  012337  034512            MOV   (3)+,TOLER
5449  027004  004737  034146            JSR   PC,AVERR
```

```
                                        DEC     R2
                                        BNE     6$

                                20$:

                                ;;*****************************************************************
                                ;*TEST 47        *TEST THAT MULTI JOBS CAN BE STARTED AND RUN
                                ;*IN THIS TEST, WE'LL ATTEMPT TO STARTUP UP TO EIGHT JOBS WITH
                                ;*MULTIUSER MICRO CODE. THIS TEST WILL ONLY BE EXECUTED IF
                                ;*MULTIUSER MICRO-CODE HAS BEEN LOADED INTO THE KMC.
                                ;*LESS THAN EIGHT JOBS MAY BE RUN, DEPENDING UPON THE
                                ;*PARTICULAR CONFIGURATION OF YOUR SYSTEM.
                                ;*BELOW IS A LIST OF JOBS AND HOW THEY ARE SET UP.
                                ;*
                                ;*      JOB#0   AD11K#1 (IF CONFIGURED) OR LPS A/D OR AR11
                                ;*      JOB#1   DR11K#1 (IF CONFIGURED) OR LPS I/O OR AR11 I/O
                                ;*      JOB#2   AA11K   (IF CONFIGURED) OR LPS OR AR11
                                ;*      JOB#3   AD11K#2 (IF CONFIGURED)
                                ;*      JOB#4   DR11K#2 (IF CONFIGURED)
                                ;*      JOB#5   DR11K#3 (IF CONFIGURED)
                                ;*      JOB#6   DR11K#4 (IF CONFIGURED)
                                ;*      JOB#7   DR11K#5 (IF CONFIGURED)
                                ;*
                                ;*
                                ;*WHEN WE SET UP THESE JOBS, WE'LL SET THEM UP TO START ROUGHLY
                                ;*AT THE SAME TIME BY VARING THE START UP DELAY.
                                ;*
                                ;*WHAT WE'RE GONNA LOOK FOR AN ERROR INDICATION
                                ;*IS FOR ERRORS ON  JOBS, OR SOME JOB TO FAIL TO FINISH.
                                ;*
                                ;;*****************************************************************
000004                          TST47:  SCOPE
012737  000001  001160                  MOV     #1,$TIMES       ;;DO 1 ITERATION
122737  000115  001772                  CMPB    #'M,$VERSN
001402                                  BEQ     1$
000137  030220                          JMP     20$
012700  055066                  1$:     MOV     #JOB0,R0
012703  055204                          MOV     #JOB1,R3                ;GET DIF IN ADDR. BETWEEN JOBS.
160003                                  SUB     R0,R3
005020                          2$:     CLR     (0)+
020027  056230                          CMP     R0,#JOB7R
001374                                  BNE     2$
005037  001124                          CLR     $GDDAT
032737  012001  001562                  BIT     #BIT0!BIT10!BIT12,SR1 ;ANY A/D'S?
001442                                  BEQ     3$                      ;NO-EXIT A/D'S.
012700  055066                          MOV     #JOB0,R0                ;GET JOB POINTER.
012710  000012                          MOV     #12,(0)                 ;MODE WORD-START A/D.
052710  000400                          BIS     #BIT8,(0)               ;SINGLE CHAN.
012760  000512  000002                  MOV     #512,2(0)               ;TAKE 512 SAMPLES
012760  055164  000004                  MOV     #JOBOU,4(0)
005070  000004                          CLR     @4(0)                   ;CLEAR USW
152760  000200  000007                  BISB    #BIT7,7(0)              ;BUFFER OVERRUN NOT FATAL, ONE BUFFER.
012760  056246  000010                  MOV     #BUFF0,10(0)            ;SET BUFFER ADDR.
```

```
5504  027146  012760  000010  000054          MOV   #10,54(0)          ;SET DELAY BEFORE START.
5505  027154  012760  000016  000060          MOV   #16,60(0)          ;SAMPLE 14 CHANNELS
5506  027162  012760  000010  000062          MOV   #10,62(0)          ;SET SAMPLE RATE.
5507  027170  005237  001124                  INC   $GDDAT
5508  027174  012760  000400  000056          MOV   #400,56(0)         ;START CH0, INC=1
5509  027202                          3$:
5510
5511
5512                                                                   ;/-SDRM-
5513
5514  027202  032737  100004  001562          BIT   #BIT2!BIT15,SR1    ;/THIS DR11K #1 OR LPS I/O SEL?
5515  027210  001442                          BEQ   64$
5516  027212  060300                          ADD   R3,R0              ;/UPDAT JOB STORAGE AREA POINTER.
5517  027214  012710  000432                  MOV   #432,(0)           ;/SET I/O START, MULTI-USER, INPUT SINGLE CHAN.
5518  027220  012760  000256  000002          MOV   #256,2(0)          ;/WORD COUNT
5519  027226  012760  055302  000004          MOV   #JOB1U,4(0)
5520  027234  012770  000000  000004          MOV   #0,@4(0)           ;/CLEAR USW
5521  027242  012760  000000  000006          MOV   #0,6(0)
5522  027250  012760  060252  000010          MOV   #BUFF2,10(0)       ;/SET BUFFER ADDR.
5523  027256  005060  000012                  CLR   12(0)
5524  027262  005237  001124                  INC   $GDDAT             ;RECORD THIS JOB
5525  027266  012760  000100  000054          MOV   #100,54(0)         ;DELAY BEFORE START.
5526  027274  012760  000000  000056          MOV   #1-1,56(0)         ;/SELECT CHAN #.
5527  027302  012760  000001  000060          MOV   #1,60(0)           ;/SAMPLING ONLY ONE CHAN.
5528  027310  012760  000001  000062          MOV   #1,62(0)           ;/SAMPLE RATE
5529
5530  027316                          64$:
5531
5532  027316  032737  042010  001562          BIT   #BIT3!BIT10!BIT14,SR1 ;D/A OUTPUT SELECTED?
5533  027324  001440                          BEQ   5$                 ;NO-NEXT SET-UP
5534  027326  005237  001124                  INC   $GDDAT
5535                                                                   ;ALTHOUGH WE'RE GONNA WORK
5536                                                                   ;WITH D/A, NO OUTPUT
5537                                                                   ;WILL RESULT. D/A THROWN IN
5538                                                                   ;JUST TO TRY AND CONFUSE LPA.
5539  027332  060300                          ADD   R3,R0              ;UPDATE JOB STORAGE AREA POINTER
5540  027334  012710  001212                  MOV   #BIT1!BIT3!BIT7!BIT9,(0) ;START, D/A, SEQ. CHAN.
5541  027340  012760  000256  000002          MOV   #256,2(0)          ;SET XFERR COUNT
5542  027346  012760  055420  000004          MOV   #JOB2U,4(0)
5543  027354  012770  000000  000004          MOV   #0,@4(0)           ;CLR USW.
5544  027362  012760  000000  000006          MOV   #0,6(0)
5545  027370  012760  061254  000010          MOV   #BUFF3,10(0)       ;BUFFER ADDR.
5546  027376  012760  000100  000054          MOV   #100,54(0)         ;DELAY BEFORE START.
5547  027404  012760  000000  000056          MOV   #0,56(0)           ;/SAMPLE CHAN 0
5548  027412  012760  000001  000060          MOV   #1,60(0)           ;/SAMPLE ONLY ONE CHAN
5549  027420  012760  000001  000062          MOV   #1,62(0)           ;/SAMPLE RATE
5550
5551  027426  032737  000020  001562  5$:     BIT   #BIT4,SR1          ;SECOND A/D?
5552  027434  001437                          BEQ   6$
5553  027436  060300                          ADD   R3,R0              ;UPDATE JOB STORAGE ARE POINTER.
5554  027440  012710  000012                  MOV   #12,(0)            ;MODE WORD-START A/D.
5555  027444  012760  000512  000002          MOV   #512,2(0)          ;TAKE 512 SAMPLES
5556  027452  012760  055536  000004          MOV   #JOB3U,4(0)
5557  027460  005070  000004                  CLR   @4(0)              ;CLEAR USW
```

```
5558  027464  152760  000200  000007          BISB    #BIT7,7(0)      ;BUFFER OVERRUN NOT FATAL, ONE BUFFER.
5559  027472  012760  056246  000010          MOV     #BUFF0,10(0)    ;SET BUFFER ADDR.
5560  027500  012760  000010  000054          MOV     #10,54(0)       ;SET DELAY BEFORE START.
5561  027506  012760  000420  000056          MOV     #420,56(0)
5562  027514  012760  000016  000060          MOV     #16,60(0)       ;SAMPLE 14 CHANNELS.
5563  027522  012760  000010  000062          MOV     #10,62(0)       ;SET SAMPLE RATE.
5564  027530  005237  001124                  INC     $GDDAT
5565
5566  027534                          6$:
5567
5568                                                                  ;/-SDRM-
5569
5570  027534  032737  000040  001562          BIT     #BIT5,SR1       ;/THIS DR11K #2 SELECTED?
5571  027542  001442                          BEQ     65$
5572  027544  060300                          ADD     R3,R0           ;/UPDAT JOB STORAGE AREA POINTER.
5573  027546  012710  000432                  MOV     #432,(0)        ;/SET I/O START, MULTI-USER, INPUT SINGLE CHAN.
5574  027552  012760  000256  000002          MOV     #256,2(0)       ;/WORD COUNT
5575  027560  012760  055654  000004          MOV     #JOB4U,4(0)
5576  027566  012770  000000  000004          MOV     #0,@4(0)        ;/CLEAR USW
5577  027574  012760  000000  000006          MOV     #0,6(0)
5578  027602  012760  060252  000010          MOV     #BUFF2,10(0)    ;/SET BUFFER ADDR.
5579  027610  005060  000012                  CLR     12(0)
5580  027614  005237  001124                  INC     $GDDAT          ;RECORD THIS JOB
5581  027620  012760  000100  000054          MOV     #100,54(0)      ;DELAY BEFORE START.
5582  027626  012760  000001  000056          MOV     #2-1,56(0)      ;/SELECT CHAN #.
5583  027634  012760  000001  000060          MOV     #1,60(0)        ;/SAMPLING ONLY ONE CHAN.
5584  027642  012760  000001  000062          MOV     #1,62(0)        ;/SAMPLE RATE
5585
5586  027650                          65$:
5587
5588                                                                  ;/-SDRM-
5589
5590  027650  032737  000200  001562          BIT     #BIT7,SR1       ;/THIS DR11K #3 SELECTED?
5591  027656  001442                          BEQ     66$
5592  027660  060300                          ADD     R3,R0           ;/UPDAT JOB STORAGE AREA POINTER.
5593  027662  012710  000432                  MOV     #432,(0)        ;/SET I/O START, MULTI-USER, INPUT SINGLE CHAN.
5594  027666  012760  000256  000002          MOV     #256,2(0)       ;/WORD COUNT
5595  027674  012760  055772  000004          MOV     #JOB5U,4(0)
5596  027702  012770  000000  000004          MOV     #0,@4(0)        ;/CLEAR USW
5597  027710  012760  000000  000006          MOV     #0,6(0)
5598  027716  012760  060252  000010          MOV     #BUFF2,10(0)    ;/SET BUFFER ADDR.
5599  027724  005060  000012                  CLR     12(0)
5600  027730  005237  001124                  INC     $GDDAT          ;RECORD THIS JOB
5601  027734  012760  000100  000054          MOV     #100,54(0)      ;DELAY BEFORE START.
5602  027742  012760  000002  000056          MOV     #3-1,56(0)      ;/SELECT CHAN #.
5603  027750  012760  000001  000060          MOV     #1,60(0)        ;/SAMPLING ONLY ONE CHAN.
5604  027756  012760  000001  000062          MOV     #1,62(0)        ;/SAMPLE RATE
5605
5606  027764                          66$:
5607
5608                                                                  ;/-SDRM-
5609
5610  027764  032737  000400  001562          BIT     #BIT8,SR1       ;/THIS DR11K #4 SELECTED?
5611  027772  001442                          BEQ     67$
```

```
5612  027774  060300                          ADD     R3,R0           ;/UPDAT JOB STORAGE AREA POINTER.
5613  027776  012710  000432                  MOV     #432,(0)        ;/SET I/O START, MULTI-USER, INPUT SINGLE CHAN.
5614  030002  012760  000256  000002          MOV     #256,2(0)       ;/WORD COUNT
5615  030010  012760  056110  000004          MOV     #JOB6U,4(0)
5616  030016  012770  000000  000004          MOV     #0,@4(0)        ;/CLEAR USW
5617  030024  012760  000000  000006          MOV     #0,6(0)
5618  030032  012760  060252  000010          MOV     #BUFF2,10(0)    ;/SET BUFFER ADDR.
5619  030040  005060  000012                  CLR     12(0)
5620  030044  005237  001124                  INC     $GDDAT          ;RECORD THIS JOB
5621  030050  012760  000100  000054          MOV     #100,54(0)      ;DELAY BEFORE START.
5622  030056  012760  000003  000056          MOV     #4-1,56(0)      ;/SELECT CHAN #.
5623  030064  012760  000001  000060          MOV     #1,60(0)        ;/SAMPLING ONLY ONE CHAN.
5624  030072  012760  000001  000062          MOV     #1,62(0)        ;/SAMPLE RATE
5625
5626  030100                          67$:
5627
5628                                                                  ;/-SDRM-
5629
5630  030100  032737  001000  001562          BIT     #BIT9,SR1       ;/THIS DR11K #5 SELECTED?
5631  030106  001442                          BEQ     68$
5632  030110  060300                          ADD     R3,R0           ;/UPDAT JOB STORAGE AREA POINTER.
5633  030112  012710  000432                  MOV     #432,(0)        ;/SET I/O START, MULTI-USER, INPUT SINGLE CHAN.
5634  030116  012760  000256  000002          MOV     #256,2(0)       ;/WORD COUNT
5635  030124  012760  056226  000004          MOV     #JOB7U,4(0)
5636  030132  012770  000000  000004          MOV     #0,@4(0)        ;/CLEAR USW
5637  030140  012760  000000  000006          MOV     #0,6(0)
5638  030146  012760  060252  000010          MOV     #BUFF2,10(0)    ;/SET BUFFER ADDR.
5639  030154  005060  000012                  CLR     12(0)
5640  030160  005237  001124                  INC     $GDDAT          ;RECORD THIS JOB
5641  030164  012760  000100  000054          MOV     #100,54(0)      ;DELAY BEFORE START.
5642  030172  012760  000004  000056          MOV     #5-1,56(0)      ;/SELECT CHAN #.
5643  030200  012760  000001  000060          MOV     #1,60(0)        ;/SAMPLING ONLY ONE CHAN.
5644  030206  012760  000001  000062          MOV     #1,62(0)        ;/SAMPLE RATE
5645
5646  030214                          68$:
5647
5648  030214  004737  036026                  JSR     PC,FLASH                ;FLASH WILL START JOBS
5649                                                                          ;AND REPORT ERRORS,WAIT TILL DONE.
5650  030220                          20$:
5651
5652                                  ;;*****************************************************************
5653                                  ;*TEST 50       *HIGH SPEED A/D SAMPLE TEST (DEDICATED MODE,SPECIAL TEST
5654                                  ;*
5655                                  ;*      IN THIS TEST WE WILL TAKE A/D SAMPLES AS FAST AS WE CAN.
5656                                  ;*      THIS TEST REQUIRES MORE MEMORY THAN THE REST SO UPON ENTRY
5657                                  ;*      WE WILL CHECK TO MAKE SURE THAT WE HAVE 20K OF MEMORY.
5658                                  ;*      ALSO DEDICATED MODE MICRO-CODE MUST BE IN THE KMC. LAST,
5659                                  ;*      WE MUST HAVE TWO A/D S ON THE I/O BUSS,BOTH HAVEING A G5036
5660                                  ;*      WRAP-ARROUND MODULE INSTALLED AND INFORMED TO PROGRAM VIA SR2.
5661                                  ;*      IF ALL REQIREMENTS ARE MENT,WE WILL DO THIS TEST AND TAKE
5662                                  ;*      SAMPLES AT A RATE OF APP. 150 KILOHERTZ.
5663                                  ;*
5664                                  ;;*****************************************************************
5665  030220  000004                  TST50:  SCOPE
```

```
5666
5667  030222  122737  000104  001772          CMPB    #'D,$VERSN      ;RUNNING DEDICATED MODE U-CODE?
5668  030230  001014                          BNE     ETDF            ;NO-THEN EXIT.
5669  030232  032737  000020  001564          BIT     #BIT4,SR2       ;SECOND A/D WITH WRAPARROUND MODULE?
5670  030240  001410                          BEQ     ETDF            ;NO-EXIT.
5671  030242  012737  030262  000004          MOV     #ETDF,@#4       ;SET FOR TIMEOUT IF NOT ENOUGH CORE.
5672  030250  005037  000006                  CLR     @#6
5673  030254  005737  070000                  TST     @#70000         ;ADDR. MEMORY.
5674  030260  000407                          BR      1$              ;GO TO START OF TEST.IF NOT ENOuGH MEMORY
5675                                                                  ;WE WOULD HAVE TRAPPED.
5676
5677          030262                  ETDF=.
5678  030262  012737  000006  000004          MOV     #6,@#4          ;RESTORE LOC 4
5679  030270  012706  001100                  MOV     #STACK,SP
5680  030274  000137  030576                  JMP     20$
5681
5682  030300  012737  000006  000004  1$:     MOV     #6,@#4
5683  030306  012737  000503  044114          MOV     #503,KWT+2      ;START CLOCK AT RATE 1MHZ.
5684  030314  012737  177763  044116          MOV     #-13.,KWT+4     ;INTR. EVERY 13 USEC.
5685  030322  012700  055066                  MOV     #JOBO,R0
5686  030326  012710  004442                  MOV     #BIT11!BIT8!BIT5!BIT1,(0)       ;START WORD.
5687  030332  012760  001750  000002          MOV     #1000.,2(0)             ;BUFFER SIZE
5688  030340  012760  055164  000004          MOV     #JOBOU,4(0)             ;USW ARRD.
5689  030346  005070  000004                  CLR     @4(0)
5690  030352  012760  070000  000010          MOV     #70000,10(0)            ;SET UP BUFFER ADDRS.
5691  030360  012760  070000  000014          MOV     #70000,14(0)
5692  030366  062760  001750  000014          ADD     #1000.,14(0)
5693  030374  012760  070000  000020          MOV     #70000,20(0)
5694  030402  062760  003720  000020          ADD     #2000.,20(0)
5695  030410  112760  000002  000007          MOVB    #2,7(0)
5696  030416  112760  000144  000054          MOVB    #100.,54(0)             ;DELAY BEFORE START.
5697  030424  112760  000001  000060          MOVB    #1.,60(0)               ;NUMBER OF CHANS.
5698  030432  112737  000002  034454          MOVB    #2.,CHANU
5699  030440  012760  000001  000062          MOV     #1,62(0)                ;TICKS BETWEEN SAMPLES.
5700  030446  012760  000400  000056          MOV     #400,56(0)              ;START CH0,INC=1
5701  030454  012737  000001  001124          MOV     #1,$GDDAT               ;ONE JOB.
5702  030462  004737  036026                  JSR     PC,FLASH                ;DO IT.
5703
5704  030466  005237  034540                  INC     REPOR                   ;LOOK AT AD11K #1
5705  030472  012703  055036                  MOV     #LIST12,R3
5706  030476  011337  034474          2$:     MOV     (3),CHANS
5707  030502  012337  034476                  MOV     (3)+,CHANF
5708  030506  012337  034514                  MOV     (3)+,AVEXP
5709  030512  012337  034512                  MOV     (3)+,TOLER
5710  030516  012701  070000                  MOV     #70000,R1
5711  030522  013737  034474  034466          MOV     CHANS,CHAN
5712  030530  004737  034160                  JSR PC,SPEP
5713
5714  030534  012703  055036                  MOV     #LIST12,R3              ;LOOK AT AD11K #2
5715  030540  011337  034474                  MOV     (3),CHANS
5716  030544  012337  034476                  MOV     (3)+,CHANF
5717  030550  012337  034514                  MOV     (3)+,AVEXP
5718  030554  012337  034512                  MOV     (3)+,TOLER
5719  030560  012701  070002                  MOV     #70002,R1
```

```
030564  013737  034474  034466          MOV     CHANS,CHAN
030572  004737  034160                  JSR PC,SPEP

030576                          20$:

030576                          ETEST:
                                ;;*****************************************************************
                                ;*TEST 51                    END OF TESTS
                                ;;*****************************************************************
030576  000004                  TST51:  SCOPE
                                .SBTTL  END OF PASS ROUTINE

                                ;;*****************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO RSTART

030600                          $EOP:
030600  000240                          NOP
030602  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
030606  005037  001160                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
030612  005237  001202                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
030616  042737  100000  001202          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
030624  005327                          DEC     (PC)+           ;;LOOP?
030626  000001                  $EOpCT: .WORD   1
030630  003064                          BGT     $DOAGN          ;;YES
030632  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
030634  000001                  $ENDCT: .WORD   1
030636  030626                          $EOPCT

                                                                ;/-ENDPAS-
030640  122737  000104  001772          CMPB    #'D,$VERSN
030646  001006                          BNE     1$
030650  104401  030656                  TYPE    ,65$            ;;TYPE ASCIZ STRING
030654  000402                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <200>#D-#
030662                          64$:
030662  000405                          BR      2$
030664                          1$:
030664  104401  030672                  TYPE    ,67$            ;;TYPE ASCIZ STRING
030670  000402                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <200>#M-#
030676                          66$:
030676                          2$:
030676  104401  030704                  TYPE    ,69$            ;;TYPE ASCIZ STRING
030702  000405                          BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  #END PASS #
030716                          68$:
030716  013746  001202                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER.
030722  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
030724  104401  030732                  TYPE    ,71$            ;;TYPE ASCIZ STRING
030730  000411                          BR      70$             ;;GET OVER THE ASCIZ
                                ;;71$:  .ASCIZ  # ; TOTAL ERRORS #
```

```
                                70$:
030754                                  MOV     ERCNT,-(SP)     ;;SAVE ERCNT FOR TYPEOUT
030754  013746  001776                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
030760  104405
030762  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
030766  001405                          BEQ     $DOAgN          ;;BRANCH IF NO MONITOR
030770  000005                          RESET                   ;;CLEAR THE WORLD
030772  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
030774  000240                          NOP                     ;;SAVE ROOM
030776  000240                          NOP                     ;;FOR
031000  000240                          NOP                     ;;ACT11
031002                          $DOAgN:
031002  000137                          JMP     @(PC)+          ;;RETURN
031004  003500                  $RTNAD: .WORD   RSTART
031006     377      377      000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
031012                                  .EVEN


                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;;**************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON---ENTER HERE TO tYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT
                                
031012  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
031016  116637  000001  031235          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
031024  112637  031237                  MOVB    (SP)+,$OMoDE+1  ;;NUMBER OF DIGITS TO TYPE
031030  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
031034  000406                          BR      $TYPON
031036  112737  000001  031235  $TYpOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
031044  112737  000006  031237          MOVB    #6,$OMoDE+1     ;;SET FOR SIX(6) DIGITS
031052  112737  000005  031234  $TYPON: MOVB    #5,$OCnT        ;;SET THE ITERATION COUNT
031060  010346                          MOV     R3,-(SP)        ;;SAVE R3
031062  010446                          MOV     R4,-(SP)        ;;SAVE R4
031064  010546                          MOV     R5,-(SP)        ;;SAVE R5
031066  113704  031237                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
```

```
5828  031072  005404                          NEG     R4
5829  031074  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
5830  031100  110437  031236                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
5831  031104  113704  031235                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
5832  031110  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
5833  031114  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
5834  031116  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
5835  031120  000404                          BR      3$              ;;GO DO MSB
5836  031122  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
5837  031124  006105                          ROL     R5
5838  031126  006105                          ROL     R5
5839  031130  010503                          MOV     R5,R3
5840  031132  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
5841  031134  105337  031236                  DECB    $OMODE          ;;TYPE THIS DIGIT?
5842  031140  100016                          BPL     7$              ;;BR IF NO
5843  031142  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
5844  031146  001002                          BNE     4$              ;;TEST FOR 0
5845  031150  005704                          TST     R4              ;;SUPPRESS THIS 0?
5846  031152  001403                          BEQ     5$              ;;BR IF YES
5847  031154  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5848  031156  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5849  031162  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5850  031166  110337  031232                  MOVB    R3,8$           ;;SAVE FOR TYPING
5851  031172  104401  031232                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
5852  031176  105337  031234          7$:     DECB    $OCNT           ;;COUNT BY 1
5853  031202  003347                          BGT     2$              ;;BR IF MORE TO DO
5854  031204  002402                          BLT     6$              ;;BR IF DONE
5855  031206  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
5856  031210  000744                          BR      2$              ;;GO DO THE LAST DIGIT
5857  031212  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
5858  031214  012604                          MOV     (SP)+,R4        ;;RESTORE R4
5859  031216  012603                          MOV     (SP)+,R3        ;;RESTORE R3
5860  031220  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
5861  031226  012616                          MOV     (SP)+,(SP)
5862  031230  000002                          RTI                     ;;RETURN
5863  031232     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
5864  031233     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
5865  031234     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
5866  031235     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
5867  031236  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
5868                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
5869
5870                                  ;;**************************************************************
5871                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
5872                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
5873                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
5874                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
5875                                  ;*REPLACED WITH SPACES.
5876                                  ;*CALL:
5877                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
5878                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
5879
5880  031240                          $TYPDS:
5881  031240  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
```

```
 5882  031242  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 5883  031244  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 5884  031246  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 5885  031250  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 5886  031252  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 5887  031256  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 5888  031262  100004                        BPL     1$              ;;BR IF INPUT IS POS.
 5889  031264  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 5890  031266  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 5891  031274  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 5892  031276  012703  031454                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 5893  031302  112723  000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 5894  031306  005002                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 5895  031310  016001  031444                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 5896  031314  160105                3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 5897  031316  002402                        BLT     4$              ;;BR IF DONe
 5898  031320  005202                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 5899  031322  000774                        BR      3$
 5900  031324  060105                4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 5901  031326  005702                        TST     R2              ;;CHECK IF BCD DIGIT=0
 5902  031330  001002                        BNE     5$              ;;FALL THROUGH IF 0
 5903  031332  105716                        TSTB    (SP)            ;;STILL DOInG LEADING 0'S?
 5904  031334  100407                        BMI     7$              ;;BR IF YES
 5905  031336  106316                5$:     ASLB    (SP)            ;;MSD?
 5906  031340  103003                        BCC     6$              ;;BR IF NO
 5907  031342  116663  000001  177777        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 5908  031350  052702  000060        6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 5909  031354  052702  000040        7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 5910  031360  110223                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 5911  031362  005720                        TST     (R0)+           ;;JUST INCREMENTING
 5912  031364  020027  000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
 5913  031370  002746                        BLT     2$              ;;GO DO THE NEXT DIGIT
 5914  031372  003002                        BGT     8$              ;;GO TO EXIT
 5915  031374  010502                        MOV     R5,R2           ;;GET THE LSD
 5916  031376  000764                        BR      6$              ;;GO CHANGE TO ASCII
 5917  031400  105726                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 5918  031402  100003                        BPL     9$              ;;BR IF NO
 5919  031404  116663  177777  177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 5920  031410  105013                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 5921  031414  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
 5922  031416  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 5923  031420  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 5924  031422  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 5925  031424  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 5926  031426  104401  031454                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 5927  031432  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 5928  031440  012616                        MOV     (SP)+,(SP)
 5929  031442  000002                        RTI                     ;;RETURN TO USER
 5930  031444  023420                $DTBL:  10000.
 5931  031446  001750                        1000.
 5932  031450  000144                        100.
 5933  031452  000012                        10.
 5934  031454  000004                $DBLK:  .BLKW   4
 5935                                .SBTTL  ERROR HANDLER ROUTINE
```

```
                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                 *AND GO TO $ERRTYP ON ERROR
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW15=1         HALT ON ERROR
                                ;*SW13=1         INHIBIT ERROR TYPEOUT
                                ;*SW10=1         BELL ON ERROR
                                ;*SW09=1         LOOP ON ERROR
                                ;*CALL
                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

031464                          $ERROR:
031464  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
031466  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
031472  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
031474  013777  001102  147440          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
031502  032777  002000  147430          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
031510  001402                          BEQ     1$              ;;NO - SKIP
031512  104401  001164                  TYPE    ,$BELL          ;;RING BELL
031516  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
031522  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
031526  162737  000002  001116          SUB     #2,$ERRPC
031534  117737  147356  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
031542  032777  020000  147370          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
031550  001004                          BNE     20$             ;;SKIP TYPEOUTS
031552  004737  031660                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
031556  104401  001171                  TYPE    ,$CRLF
031562                          20$:
031562  122737  000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
031570  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
031572  113737  001114  031604          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
031600  004737  033454                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
031604     000                  21$:    .BYTE   0
031605     000                          .BYTE   0
031606  000777                  22$:    BR      22$             ;;APT ERROR LOOP
031610  005777  147324          2$:     TST     @SWR            ;;HALT ON ERROR
031614  100002                          BPL     3$              ;;SKIP IF CONTINUE
031616  000000                          HALT                    ;;HALT ON ERROR!
031620  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
031622  032777  001000  147310  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
031630  001402                          BEQ     4$              ;;BR IF NO
031632  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
031636  005737  001162          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
031642  001402                          BEQ     5$              ;;BR IF NONE
031644  013716  001162                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
031650                          5$:
031650  005237  001776          10$:    INC     ERCNT
031654  001775                          BEQ     10$
031656  000002                          RTI
                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

                                ;;**********************************************************************
```

```
                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                                ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

031660                          $ERRTYP:
031660  104401  001171                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
031664  010046                      MOV     R0,-(SP)        ;;SAVE R0
031666  005000                      CLR     R0              ;;PICKUP THE ITEM INDEX
031670  153700  001114              BISB    @#$ITEMB,R0
031674  001004                      BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                            ;;TYPE THE PC OF THE ERROR
031676  013746  001116              MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                            ;;ERROR ADDRESS
031702  104402                      TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
031704  000426                      BR      6$              ;;GET OUT
031706  005300                1$:   DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
031710  006300                      ASL     R0              ;;      WORK FOR THE ERROR TABLE
031712  006300                      ASL     R0
031714  006300                      ASL     R0
031716  062700  001256              ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
031722  012037  031732              MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
031726  001404                      BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
031730  104401                      TYPE                    ;;TYPE THE "ERROR MESSAGE"
031732  000000                2$:   .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
031734  104401  001171              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
031740  012037  031750        3$:   MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
031744  001404                      BEQ     5$              ;;SKIP TYPEOUT IF 0
031746  104401                      TYPE                    ;;TYPE THE "DATA HEADER"
031750  000000                4$:   .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
031752  104401  001171              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
031756  011000                5$:   MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
031760  001004                      BNE     7$              ;;GO TYPE THE DATA
031762  012600                6$:   MOV     (SP)+,R0        ;;RESTORE R0
031764  104401  001171              TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
031770  000207                      RTS     PC              ;;RETURN
031772                          7$:
031772  013046                      MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
031774  104402                      TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
031776  005710                      TST     (R0)            ;;IS THERE ANOTHER NUMBER?
032000  001770                      BEQ     6$              ;;BR IF NO
032002  104401  032010              TYPE    ,8$             ;;TYPE TWO(2) SPACES
032006  000771                      BR      7$              ;;LOOP
032010  020040    000         8$:   .ASCIZ  /  /            ;;TWO(2) SPACES
        032014                      .EVEN
                                .SBTTL  SCOPE HANDLER ROUTINE

                                ;;**************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW14=1         LOOP ON TEST
                                ;*SW11=1         INHIBIT ITERATIONS
                                ;*SW09=1         LOOP ON ERROR
```

```
                                        ;*SW08=1        LOOP ON TEST IN SWR<7:0>
                                        ;*CALL
                                        ;*      SCOPE           ;;SCOPE=IOT

032014                                  $SCOPE:
032014  104407                                  CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
032016  104407                                  CKSWR
032020  032777  040000  147112          1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
032026  001114                                  BNE     $OVER           ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
032030  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
032032  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
032036  012737  032056  000004                  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
032044  005737  177060                          TST     @#177060        ;;TIME OUT ON XOR?
032050  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
032054  000463                                  BR      $SVLAD          ;;GO TO THE NEXT TEST
032056  022626                          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
032060  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
032064  000423                                  BR      7$              ;;LOOP ON THE PRESENT TEST
032066                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
032066  032777  000400  147044                  BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
032074  001404                                  BEQ     2$              ;;BR IF NO
032076  127737  147036  001102                  CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
032104  001465                                  BEQ     $OVER           ;;BR IF YES
032106  105737  001103                  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
032112  001421                                  BEQ     3$              ;;BR IF NO
032114  123737  001115  001103                  CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
032122  101015                                  BHI     3$              ;;BR IF NO
032124  032777  001000  147006                  BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
032132  001404                                  BEQ     4$              ;;BR IF NO
032134  013737  001110  001106          7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
032142  000446                                  BR      $OVER
032144  105037  001103                  4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
032150  005037  001160                          CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
032154  000415                                  BR      1$              ;;ESCAPE TO THE NEXT TEST
032156  032777  004000  146754          3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
032164  001011                                  BNE     1$              ;;BR IF YES
032166  005737  001202                          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
032172  001406                                  BEQ     1$              ;;      INHIBIT ITERATIONS
032174  005237  001104                          INC     $ICNT           ;;INCREMENT ITERATION COUNT
032200  023737  001160  001104                  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
032206  002024                                  BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
032210  012737  000001  001104          1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
032216  013737  032274  001160                  MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
032224  105237  001102                  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
032230  113737  001102  001200                  MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
032236  011637  001106                          MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
032242  011637  001110                          MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
032246  005037  001162                          CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
032252  112737  000001  001115                  MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
032260  013777  001102  146654          $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
032266  013716  001106                          MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
032272  000002                                  RTI                     ;;FIXES PS
```

```
                                $MXCNT: 10                      ;;MAX. NUMBER OF ITERATIONS
                                 $BTTL  TTY INPUT ROUTINE

                                ;;**********************************************************************
                                .ENABL  LSB

                                ;;**********************************************************************
                                ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                ;*WHEN OPERATING IN TTY FLAG MODE.
032276  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
032304  001074                        BNE     15$             ;;BRANCH IF NO
032306  105777  146632                TSTB    @$TKS           ;;CHAR THERE?
032312  100071                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
032314  117746  146626                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
032320  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
032324  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
032330  001062                        BNE     15$             ;;NO, RETURN TO USER
032332  123727  001134  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
032340  001456                        BEQ     15$             ;;BRANCH IF YES

032342  104401  033023                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
032346  104401  033030        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
032352  013746  000176                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
032356  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
032360  104401  033041                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
032364  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
032366  005046                        CLR     -(SP)           ;;THE NEW SWR
032370  105777  146550        7$:     TSTB    @$TKS           ;;CHAR THERE?
032374  100375                        BPL     7$              ;;IF NOT TRY AGAIN

032376  117746  146544                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
032402  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII


032406  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
032412  001005                        BNE     10$             ;;BRANCH IF NOT
032414  104401  033016                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
032420  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
032424  000757                        BR      19$             ;;LET'S TRY IT AGAIN

032426  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
032432  001022                        BNE     16$             ;;BRANCH IF NO
032434  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
032440  001403                        BEQ     11$             ;;BRANCH IF YES
032442  016677  000002  146470        MOV     2(SP),@SWR      ;;SAVE NEW SWR
032450  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
032454  104401  001171        14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
032460  123727  001135  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
032466  001003                        BNE     15$             ;;BRANCH IF NOT
032470  012777  000100  146446        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
```

```
6152  032476  000002                  15$:    RTI                     ;;RETURN
6153  032500  004737  033264          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
6154  032504  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
6155  032510  002420                          BLT     18$             ;;BRANCH IF YES
6156  032512  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
6157  032516  003015                          BGT     18$             ;;BRANCH IF YES
6158  032520  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
6159  032524  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
6160  032530  001403                          BEQ     17$             ;;BRANCH IF YES
6161  032532  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
6162  032534  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
6163  032536  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
6164  032540  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
6165  032544  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
6166  032550  000707                          BR      7$              ;;GET THE NEXT ONE
6167  032552  104401  001170          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
6168  032556  000720                          BR      20$             ;;SIMULATE CONTROL-U
6169                                  .DSABL  LSB
6170
6171
6172                                  ;;*****************************************************************
6173                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
6174                                  ;*CALL:
6175                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
6176                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
6177                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
6178                                  ;
6179
6180  032560  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
6181  032562  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
6182  032570  105777  146350          1$:     TSTB    @$TKS           ;;WAIT FOR
6183  032574  100375                          BPL     1$              ;;A CHARACTER
6184  032576  117766  146344  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
6185  032604  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
6186  032612  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
6187  032620  001013                          BNE     3$              ;;BRANCH IF NO
6188  032622  105777  146316          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
6189  032626  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
6190  032630  117746  146312                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
6191  032634  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
6192  032640  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
6193  032644  001366                          BNE     2$              ;;IF NOT DISCARD IT
6194  032646  000750                          BR      1$              ;;YES, RESUME
6195  032650  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
6196  032656  002407                          BLT     4$              ;;BRANCH IF YES
6197  032660  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
6198  032666  003003                          BGT     4$              ;;BRANCH IF YES
6199  032670  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
6200  032676  000002                  4$:     RTI                     ;;GO BACK TO USER
6201                                  ;;*****************************************************************
6202                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
6203                                  ;*CALL:
6204                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
6205                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
```

```
                                        ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
                                        ;*
                                $RDLIN: MOV     R3,-(SP)                ;;SAVE R3
                                1$:     MOV     #$TTYIN,R3              ;;GET ADDRESS
                                2$:     CMP     #$TTYIN+8.,R3           ;;BUFFER FULL?
                                        BLOS    4$                      ;;BR IF YES
                                        RDCHR                           ;;GO READ ONE CHARACTER FROM THE TTY
                                        MOVB    (SP)+,(R3)              ;;GET CHARACTER
                                10$:    CMPB    #177,(R3)               ;;IS IT A RUBOUT
                                        BNE     3$                      ;;SKIP IF NOT
                                4$:     TYPE    ,$QUES                  ;;TYPE A '?'
                                        BR      1$                      ;;CLEAR THE BUFFER AND LOOP
                                3$:     MOVB    (R3),9$                 ;;ECHO THE CHARACTER
                                        TYPE    ,9$
                                        CMPB    #15,(R3)+               ;;CHECK FOR RETURN
                                        BNE     2$                      ;;LOOP IF NOT RETURN
                                        CLRB    -1(R3)                  ;;CLEAR RETURN (THE 15)
                                        TYPE    ,$LF                    ;;TYPE A LINE FEED
                                        MOV     (SP)+,R3                ;;RESTORE R3
                                        MOV     (SP),-(SP)              ;;ADJUST THE STACK AND PUT ADDRESS OF THE
                                        MOV     4(SP),2(SP)             ;;      FIRST ASCII CHARACTER ON IT
                                        MOV     #$TTYIN,4(SP)
                                        RTI                             ;;RETURN
                                9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
                                        .BYTE   0                       ;;TERMINATOR
                                $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
                                $CNTLU: .ASCIZ  /^U/<15><12>            ;;CONTROL "U"
                                $CNTLG: .ASCIZ  /^G/<15><12>            ;;CONTROL "G"
                                $MSWR:  .ASCIZ  <15><12>/SWR = /
                                $MNEW:  .ASCIZ  /  NEW = /

                                ;       .$SB2D
                                ;       .$DB2D

                                .SBTTL  TYPE ROUTINE

                                ;;**************:*****************************************
                                ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                ;*NOTE1:         $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                ;*NOTE2:         $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                ;*NOTE3:         $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                ;*
                                ;*CALL:
                                ;*1) USING A TRAP INSTRUCTION
                                ;*      TYPE    ,MESADR                 ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                ;*OR
                                ;*      TYPE
                                ;*      MESADR
                                ;*
                                $TYPE:  TSTB    $TPFLG                  ;;IS THERE A TERMINAL?
                                        BPL     1$                      ;;BR IF YES
```

```
6260  033060  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
6261  033062  000430                        BR      3$              ;;LEAVE
6262  033064  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
6263  033066  017600  000002              MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
6264  033072  122737  000001  001214        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
6265  033100  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
6266  033102  132737  000100  001215        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
6267  033110  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
6268  033112  010037  033122              MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
6269  033116  004737  033444              JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
6270  033122  000000                61$:    .WORD   0               ;;MESSAGE ADDRESS
6271  033124  132737  000040  001215  62$:   BITB    #APTCSUP,$ENVm  ;;APT CONSOLE SUPPRESSED
6272  033132  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
6273  033134  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
6274  033136  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
6275  033140  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
6276  033142  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
6277  033144  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
6278  033150  000002                        RTI                     ;;RETURN
6279  033152  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
6280  033156  001430                        BEQ     8$
6281  033160  122716  000200              CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
6282  033164  001006                        BNE     5$
6283  033166  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
6284  033170  104401                        TYPE                    ;;TYPE A CR AND LF
6285  033172  001171                        $CRLF
6286  033174  105037  033330              CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
6287  033200  000755                        BR      2$              ;;GET NEXT CHARACTER
6288  033202  004737  033264          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
6289  033206  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
6290  033212  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
6291  033214  013746  001154              MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
6292                                                                ;;AND THE NULL CHAR.
6293  033220  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
6294  033224  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
6295  033226  004737  033264              JSR     PC,$TYPEC       ;;GO TYPE A NULL
6296  033232  105337  033330              DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
6297  033236  000770                        BR      7$              ;;LOOP
6298
6299                                  ;HORIZONTAL TAB PROCESSOR
6300
6301  033240  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
6302  033244  004737  033264          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
6303  033250  132737  000007  033330        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
6304  033256  001372                        BNE     9$              ;;TAB STOP
6305  033260  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
6306  033262  000724                        BR      2$              ;;GET NEXT CHARACTER
6307  033264  105777  145660        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
6308  033270  100375                        BPL     $TYPEC
6309  033272  116677  000002  145652        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
6310  033300  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
6311  033306  001003                        BNE     1$              ;;BRANCH IF NO
6312  033310  105037  033330              CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
6313  033314  000406                        BR      $TYPEX          ;;EXIT
```

```
 6314  033316  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 6315  033324  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
 6316  033326  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
 6317  033330  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 6318  033332  000207                  $TYPEX: RTS     PC
 6319
 6320                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
 6321
 6322                                  ;;**************************************************************
 6323                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
 6324                                  ;*CHANGE IT TO BINARY.
 6325                                  ;*CALL:
 6326                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
 6327                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
 6328                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
 6329
 6330  033334  011646                  $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
 6331  033336  016666  000004  000002          MOV     4(SP),2(SP)     ;;INPUT NUMBER
 6332  033344  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 6333  033346  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 6334  033350  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 6335  033352  104411                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
 6336  033354  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
 6337  033356  005001                          CLR     R1              ;;CLEAR DATA WORD
 6338  033360  005002                          CLR     R2
 6339  033362  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
 6340  033364  001412                          BEQ     3$              ;;IF ZERO GET OUT
 6341  033366  006301                          ASL     R1              ;;*2
 6342  033370  006102                          ROL     R2
 6343  033372  006301                          ASL     R1              ;;*4
 6344  033374  006102                          ROL     R2
 6345  033376  006301                          ASL     R1              ;;*8
 6346  033400  006102                          ROL     R2
 6347  033402  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
 6348  033406  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
 6349  033410  000764                          BR      2$              ;;LOOP
 6350  033412  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
 6351  033414  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
 6352  033420  010237  033434                  MOV     R2,$HIOCT
 6353  033424  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 6354  033426  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 6355  033430  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 6356  033432  000002                          RTI                     ;;RETURN
 6357  033434  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
 6358                                  .SBTTL  APT COMMUNICATIONS ROUTINE
 6359
 6360                                  ;;**************************************************************
 6361  033436  112737  000001  033702  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 6362  033444  112737  000001  033700  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 6363  033452  000403                          BR      $ATYC
 6364  033454  112737  000001  033702  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 6365  033462                          $ATYC:
 6366  033462  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 6367  033464  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
6368  033466  105737  033700            TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
6369  033472  001450                    BEQ     5$              ;;IF NOT:  BR
6370  033474  122737  000001  001214    CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
6371  033502  001031                    BNE     3$              ;;IF NOT:  BR
6372  033504  132737  000100  001215    BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
6373  033512  001425                    BEQ     3$              ;;IF NOT:  BR
6374  033514  017600  000004            MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
6375  033520  062766  000002  000004    ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
6376  033526  005737  001174      1$:   TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
6377  033532  001375                    BNE     1$              ;;IF NOT:  WAIT
6378  033534  010037  001210            MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
6379  033540  105720            2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
6380  033542  001376                    BNE     2$
6381  033544  163700  001210            SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
6382  033550  006200                    ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
6383  033552  010037  001212            MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
6384  033556  012737  000004  001174    MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
6385  033564  000413                    BR      5$
6386  033566  017637  000004  033612  3$:   MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
6387  033574  062766  000002  000004    ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
6388  033602  013746  177776            MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
6389  033606  004737  033052            JSR     PC,$TYPE        ;;CALL TYPE MACRO
6390  033612  000000            4$:     .WORD   0
6391  033614                    5$:
6392  033614  105737  033702    10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
6393  033620  001416                    BEQ     12$             ;;IF NOT:  BR
6394  033622  005737  001214            TST     $ENV            ;;RUNNING UNDER APT?
6395  033626  001413                    BEQ     12$             ;;IF NOT:  BR
6396  033630  005737  001174      11$:  TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
6397  033634  001375                    BNE     11$             ;;IF NOT:  WAIT
6398  033636  017637  000004  001176    MOV     @4(SP),$FATAL   ;;GET ERROR #
6399  033644  062766  000002  000004    ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
6400  033652  005237  001174            INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
6401  033656  105037  033702    12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
6402  033662  105037  033701            CLRB    $LFLG           ;;CLEAR LOG FLAG
6403  033666  105037  033700            CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
6404  033672  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
6405  033674  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
6406  033676  000207                    RTS     PC              ;;RETURN
6407  033700     000    $MFLG:  .BYTE   0               ;;MESSG. FLAG
6408  033701     000    $LFLG:  .BYTE   0               ;;LOG FLAG
6409  033702     000    $FFLG:  .BYTE   0               ;;FATAL FLAG
6410          033704            .EVEN
6411          000200    APTSIZE=200
6412          000001    APTENV=001
6413          000100    APTSPOOL=100
6414          000040    APTCSUP=040
6415                    .SBTTL  POWER DOWN AND UP ROUTINES
6416
6417                    ;;*****************************************************************
6418                    ;POWER DOWN ROUTINE
6419  033704  012737  034044  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
6420  033712  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
6421  033720  010046                    MOV     R0,-(SP)        ;;PUSH R0 ON STACK
```

```
6422  033722  010146                          MOV     R1,-(SP)          ;;PUSH R1 ON STACK
6423  033724  010246                          MOV     R2,-(SP)          ;;PUSH R2 ON STACK
6424  033726  010346                          MOV     R3,-(SP)          ;;PUSH R3 ON STACK
6425  033730  010446                          MOV     R4,-(SP)          ;;PUSH R4 ON STACK
6426  033732  010546                          MOV     R5,-(SP)          ;;PUSH R5 ON STACK
6427  033734  017746  145200                  MOV     @SWR,-(SP)        ;;PUSH @SWR ON STACK
6428  033740  010637  034050                  MOV     SP,$SAVR6         ;;SAVE SP
6429  033744  012737  033756  000024          MOV     #$PWRUP,@#PWRVEC  ;;SET UP VECTOR
6430  033752  000000                          HALT
6431  033754  000776                          BR      .-2               ;;HANG UP
6432
6433                                  ;;*****************************************************************
6434                                  ;POWER UP ROUTINE
6435  033756  012737  034044  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC  ;;SET FOR FAST DOWn
6436  033764  013706  034050                  MOV     $SAVR6,SP         ;;GET SP
6437  033770  005037  034050                  CLR     $SAVR6            ;;WAIT LOOP FOR THE TTY
6438  033774  005237  034050          1$:     INC     $SAVR6            ;;WAIT FOR THE INC
6439  034000  001375                          BNE     1$                ;;OF  WORD
6440  034002  012677  145132                  MOV     (SP)+,@SWR        ;;POP STACK INTO @SWR
6441  034006  012605                          MOV     (SP)+,R5          ;;POP STACK INTO R5
6442  034010  012604                          MOV     (SP)+,R4          ;;POP STACK INTO R4
6443  034012  012603                          MOV     (SP)+,R3          ;;POP STACK INTO R3
6444  034014  012602                          MOV     (SP)+,R2          ;;POP STACK INTO R2
6445  034016  012601                          MOV     (SP)+,R1          ;;POP STACK INTO R1
6446  034020  012600                          MOV     (SP)+,R0          ;;POP STACK INTO R0
6447  034022  012737  033704  000024          MOV     #$PWRDN,@#PWRVEC  ;;SET UP THE POWER DOWN VECTOR
6448  034030  012737  000340  000026          MOV     #340,@#PWRVEC+2   ;;PRIO:7
6449  034036  104401                          TYPE                      ;REPORT THE POWER FAILURE
6450  034040  034052                  $PWRMG: .WORD   $POWER            ;;POWER FAIL MESSAGE POINTER
6451  034042  000002                          RTI
6452  034044  000000                  $ILLUP: HALT                      ;;THE POWER UP SEQUENCE WAS STARTED
6453  034046  000776                          BR      .-2               ;;  BEFORE THE POWER DOWN WAS COMPLETE
6454  034050  000000                  $SAVR6: 0                         ;;PUT THE SP HERE
6455  034052  005015  047520  042527  $POWER: .ASCIZ   <15><12>"POWER"
6456  034060  000122
6457                                          .EVEN
6458
6459                                          .SBTTL  ;  WIR   ROUTINE TO WAIT FOR "DRLPX0" MICROCODE TO BECOME READY.
6460
6461  034062  005037  034110          WIR:    CLR     WIRT
6462  034066  122777  000377  145376  1$:     CMPB    #377,@KMAD2
6463  034074  001404                          BEQ     2$
6464  034076  005237  034110                  INC     WIRT              ;TIME OUT?
```

```
 6465  034102  001371                          BNE     1$
 6466
 6467  034104  104003                          ERROR   3               ;TIME OUT ERROR,KMC-11 ERROR.
 6468
 6469  034106                          2$:
```

```
6470  034106  000207                          RTS     PC
6471  034110  000000                  WIRT:   .WORD   0
6472
6473                                          ;*
6474                                          ;*SYNC  IN THIS ROUTINE, WE'LL TRY AND INIT
6475                                          ;*      AND RE-SYN LPA-11 SYSTEM.
6476                                          ;*      1ST WE DO A RESET.
6477                                          ;*      2ND WE RE-START CLOCK
6478                                          ;*      NO ERROR CHECKING
6479                                          ;*      CALL=   JSR PC,SYNC
6480
6481  034112  004737  040626          SYNC:   JSR     PC,SRESET       ;RESET SYSTEM
6482
6483  034116  012777  044112  145352          MOV     #KWT,@LPADL     ;CLOCK INFO
6484  034124  052777  000001  145334          BIS     #BIT0,@LPCI     ;START
6485  034132  004737  040542                  JSR     PC,SDELAY       ;DELAY
6486  034136  117737  145332  001126          MOVB    @LPSO,$BDDAT    ;DUMB READ OF STAT REG.
6487  034144  000207                          RTS     PC
6488                                          ;*
6489                                          ;*AVERAGING ROUTINE
6490                                          ;*AT THIS POINT IN TIME ALL SAMPLES FOR ALL CHANNELS HAVE
6491                                          ;*BEEN TAKEN AND STORED IN "BUFFER" IN A
6492                                          ;*SEQUENTIAL INTERLEAVED BUFFER FORM" GIVEN BY THE FOLLOWING FORMULA:
6493                                          ;*"L=2*N+R1" WHERE L EQUALS THE LOCATION OF A SAMPLE,
6494                                          ;*N EQUALS THE NUMBER OF CHANELS SAMPLES WERE TAKEEN ON, AND R1
6495                                          ;*IS THE BUFFER POINTER SET TO "BUFFER" INITIALLY,
6496                                          ;*CAL=JSR PC,AVERR
6497                                          ;*NOTE: CHANU=# OF CHAN.
6498                                          ;*      CHANS=CHAN #
6499                                          ;*      REPOR=0,NO;=1,YES
6500                                          ;*      TOLER=TOLERENCE
6501                                          ;*
6502
6503  034146  013737  034474  034466  AVERR:  MOV     CHANS,CHAN      ;SET TO FIRST CHAN.
6504  034154  012701  056246                  MOV     #BUFFER,R1
6505  034160  013737  034454  034510  SPEP:   MOV     CHANU, SAMOFF
6506  034166  006337  034510                  ASL     SAMOFF
6507  034172  013737  034466  034464          MOV     CHAN,CHAN1
6508  034200  006337  034464                  ASL     CHAN1
6509  034204  012737  000020  034506  AVERRL: MOV     #16., SAMCNT
6510  034212  063701  034464                  ADD     CHAN1,R1        ;PUT CHAN OFFSET IN R1.
6511  034216  005037  034516                  CLR     AVTKN           ;SET INITIAL CONDITIONS FOR THIS CHAN.
6512  034222  005037  034524                  CLR     RTEMP
6513  034226  011137  034520                  MOV     (1),RLOW
6514  034232  011137  034522                  MOV     (1),RHIGH
6515  034236  023711  034522          2$:     CMP     RHIGH,(1)       ;FIND REAL HIGH VALUE.
6516  034242  003002                          BGT     3$
6517  034244  011137  034522                  MOV     (1),RHIGH
6518  034250  021137  034520          3$:     CMP     (1),RLOW        ;FIND REAL LOW VALUE.
6519  034254  003002                          BGT     4$
6520  034256  011137  034520                  MOV     (1),RLOW
6521  034262  011137  034526          4$:     MOV     (1),RTEMP1      ;GET CURRENT SAMPLE.
6522  034266  063737  034526  034516          ADD     RTEMP1,AVTKN    ;"BOOT" ADD ALL SAMPLES.
6523  034274  005537  034524                  ADC     RTEMP
```

```
6524  034300  063701  034510          ADD     SAMOFF,R1
6525  034304  005337  034506          DEC     SAMCNT
6526  034310  001352                  BNE     2$
6527  034312  013737  034514  034526  MOV     AVEXP,RTEMP1
6528  034320  063737  034512  034526  ADD     TOLER,RTEMP1
6529  034326  023737  034526  034522  CMP     RTEMP1,RHIGH
6530  034334  002427                  BLT     ERAV1           ;NO-THEN REPORT ERROR.
6531  034336  163737  034512  034526  SUB     TOLER,RTEMP1    ;YES-OK-CHECK LOWEST READING.
6532  034344  163737  034512  034526  SUB     TOLER,RTEMP1
6533  034352  023737  034520  034526  CMP     RLOW,RTEMP1
6534  034360  002415                  BLT     ERAV1           ;NO-REPORT ERROR.
6535  034362  005737  034512          TST     TOLER           ;DOES OPERATOR WISH "FORCED" TYPEOUT?
6536  034366  001412                  BEQ     ERAV2           ;IF SO-DO IT
6537  034370  062737  000002  034464  AVERRN: ADD     #2,CHAN1        ;SET TO DO NEXT CHAN-BUT
6538  034376  005237  034466          INC     CHAN            ;IF DOEn ALL CHANS-EXIT.
6539  034402  023737  034466  034476  CMP     CHAN,CHANF      ;OTHERWISE LOOP.
6540  034410  003675                  BLE     AVERRL
6541  034412  000207                  RTS     PC
6542
6543                                  ;*ERROR REPORTER
6544
6545  034414                  ERAV1:
6546  034414                  ERAV2:
6547  034414  005737  034540          TST     REPOR           ;REPORT ERROR?
6548  034420  001002                  BNE     1$
6549  034422  000137  034370          JMP     AVERRN
6550  034426                  1$:
6551                                                          ;GO TYPE--OCTAL ASCII(AL)
6552  034426  042737  170000  034520  BIC     #170000,RLOW
6553                                                          ;GO TYPE--OCTAL ASCII(AL)
6554  034434  042737  170000  034516  BIC     #170000,AVTKN
6555                                                          ;GO TYPE--OCTAL ASCII(AL)
6556  034442  042737  170000  034522  BIC     #170000,RHIGH
6557  034450  104020                  ERROR   20
6558  034452  000746                  BR      AVERRN
6559
6560                                  ;*
6561                                  ;*POINTERS USED BY REPEATIBILITY TEST
6562                                  ;*
6563
6564  034454  000000          CHANU:  0
6565  034456  000000          CHAN7:0
6566  034460  000000          NA07:0
6567  034462  000000          NA17:0
6568  034464  000000          CHAN1:  0               ;LEFT JUSTIFIED CURRENT CHANNELL.
6569  034466  000000          CHAN:   0               ;CURRENT CHANNELL
6570  034470  000000          GAIN:   0               ;CURRENT GAIN
6571  034472  000000          ADWD:   0               ;WORD SENT TO A/D
6572  034474  000000          CHANS:  0               ;STARTING CHANNELL
6573  034476  000000          CHANF:  0               ;LAST CHANNELL
6574  034500  000000          CHANSR: 0
6575  034502  000000          CHANFR: 0
6576  034504  000000          CHANNO: 0
6577  034506  000000          SAMCNT: 0               ;SAMPLE COUNT
```

```
 6578  034510  000000                SAMOFF: 0                       ;SAMPLE OFFSET
 6579  034512  000000                TOLER:  0                       ;TOLERANCE BEFORE ERROR IS REPORT
 6580  034514  000000                AVEXP:  0                       ;EXPECTED AVERAGE
 6581  034516  000000                AVTKN:  0                       ;AVERAGE OF SAMPLES TAKEN
 6582  034520  000000                RLOW:   0                       ;LOWEST SAMPLE TAKEN
 6583  034522  000000                RHIGH:  0                       ;HIGHEST SAMPLE TAKEN
 6584  034524  000000                RTEMP:  0
 6585  034526  000000                RTEMP1: 0
 6586  034530  000000                REPMAN: 0                       ;SECTION RESERVED FOR
 6587  034532  000000                        0                       ;MANUAL REPEATIBILITY
 6588  034534  000000                        0                       ;WHERE OPERETOR ENTERS
 6589  034536  000000                        0                       ;IN INFORMATION
 6590  034540  000000                REPOR:  0
 6591  034542     000      000               .BYTE   0,0
 6592  034544  000000                        0
 6593
 6594  034546  104401  043702        MAKEI:  TYPE,M0                 ;WELCOME MESSAGE
 6595  034552  104411                        RDLIN
 6596  034554  005037  001200                CLR     $TESTN
 6597  034560  012600                        MOV     (SP)+,R0
 6598  034562  121027  000117                CMPB    (R0),#'O
 6599  034566  001002                        BNE     1$
 6600  034570  000137  035516                JMP     LOOPJ
 6601  034574  104401  042615        1$:     TYPE,M1
 6602  034600  104411                        RDLIN
 6603  034602  012600                        MOV     (SP)+,R0
 6604  034604  111037  001772                MOVB    (0),$VERSN
 6605  034610  121027  000115                CMPB    (0),#'M
 6606  034614  001403                        BEQ     2$
 6607  034616  121027  000104                CMPB    (0),#'D
 6608  034622  001364                        BNE     1$
 6609  034624  104401  043030        2$:     TYPE,M2
 6610  034630  104412                        RDOCT
 6611  034632  012637  044114                MOV     (SP)+,KWT+2
 6612  034636  104401  043051                TYPE,M3
 6613  034642  104412                        RDOCT
 6614  034644  012637  044116                MOV     (SP)+,KWT+4
 6615  034650  012737  000001  001124  3$:   MOV     #1,$GDDAT
 6616  034656  122737  000104  001772        CMPB    #'D,$VErSN
 6617  034664  001403                        BEQ     4$
 6618  034666  104401  043114                TYPE,M4
 6619  034672  104412                        RDOCT
 6620  034674  012637  001124        4$:     MOV     (SP)+,$GDDAT
 6621  034700  123727  001124  000011        CMPB    $GDDAT,#11
 6622  034706  103360                        BHIS    3$
 6623  034710  013737  001124  001754        MOV     $GDDAT,MYTEMP
 6624  034716  012737  000001  001126        MOV     #1,$BDDAT
 6625  034724  012701  055066                MOV     #JOBO,R1
 6626  034730  012702  066000                MOV     #66000,R2       ;FREE CORE .-
 6627  034734  010237  056246                MOV     R2,FRECOR
 6628  034740  013702  056246                MOV     FRECOR,R2
 6629  034744  104401  043134        L1PED:  TYPE,   M5
 6630  034750  013746  001126                MOV     $BDDAT,-(SP)    ;;SAVE $BDDAT FOR TYPEOUT
 6631  034754  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
6632  034756  104401  043144                    TYPE,M6
6633  034762  104401  043154                    TYPE,M7
6634  034766  104412                            RDOC†
6635  034770  012611                            MOV     (SP)+,(1)          ;STORE MODE WORD
6636  034772  104401  043212                    TYPE,   M8
6637  034776  104412                            RDOC†
6638  035000  012661  000002                    MOV     (SP)+,2(1)
6639  035004  104401  043347          1$:       TYPE,M10                   ;# OF BUFFERS
6640  035010  104412                            RDOC†
6641  035012  112661  000007                    MOVB    (SP)+,7(1)
6642  035016  001772                            BEQ     1$
6643  035020  012737  035664  000004            MOV     #ERTOUT,@#4        ;IN CASE OF BAD BUFFER AREA
6644  035026  116137  000007  001202            MOVB    7(1),$PASS
6645  035034  105361  000007                    DECB    7(1)
6646  035040  010103                            MOV     R1,R3
6647  035042  016104  000002                    MOV     2(1),R4
6648  035046  006304                            ASL     R4
6649  035050  010261  000010          2$:       MOV     R2,10(1)
6650  035054  104401  035062                    TYPE    ,65$               ;;TYPE ASCIZ STRING
6651  035060  000410                            BR      64$                ;;GET OVER THE ASCIZ
6652                                  ;;65$:    .ASCIZ  <200>#BUFFER ADDR. #
6653  035102                        64$:
6654  035102  010246                            MOV     R2,-(SP)           ;;SAVE R2 FOR TYPEOUT
6655  035104  104402                            TYPOC                      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
6656  035106  060402                            ADD     R4,R2
6657  035110  005712                            TST     (2)
6658  035112  005337  001202                    DEC     $PASS
6659  035116  001403                            BEQ     3$
6660  035120  062701  000004                    ADD     #4,R1
6661  035124  000751                            BR      2$
6662  035126  010301                3$:         MOV     R3,R1
6663  035130  104401  043244                    TYPE,   M9                 ;USER SW=
6664  035134  016346  000004                    MOV     4(R3),-(SP)        ;;SAVE 4(R3) FOR TYPEOUT
6665  035140  104402                            TYPOC                      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
6666  035142  104401  043426          5$:       TYPE,M12                   ;BUFFER OVERRUN FATAL?
6667  035146  104411                            RDLIN
6668  035150  012600                            MOV     (SP)+,R0
6669  035152  142710  000240                    BICB    #240,(0)
6670  035156  122710  000131                    CMPB    #'Y,(0)
6671  035162  001004                            BNE     6$
6672  035164  042761  100000  000006            BIC     #BIT15,6(1)
6673  035172  000406                            BR      7$
6674  035174  121027  000116          6$:       CMPB    (0),#'N
6675  035200  001360                            BNE     5$
6676  035202  052761  100000  000006            BIS     #BIT15,6(1)
6677  035210  005071  000004          7$:       CLR     @4(1)              ;CLEAR USER WORD.
6678  035214  104401  043551                    TYPE,M14                   ;DELAY BEFORE START
6679  035220  104412                            RDOC†
6680  035222  012661  000054                    MOV     (SP)+,54(1)
6681  035226  104401  043757                    TYPE,M20                   ;CHAN ADDR WORD
6682  035232  104412                            RDOC†
6683  035234  012661  000056                    MOV     (SP)+,56(1)
6684  035240  104401  043575                    TYPE,M15                   ;# OF CHAN
6685  035244  104412                            RDOC†
```

```
6686  035246  012661  000060                  MOV     (SP)+,60(1)
6687  035252  104401  044003                  TYPE,M21                        ;CLOCK RATE
6688  035256  104412                          RDOCT
6689  035260  012661  000062                  MOV     (SP)+,62(1)
6690  035264  032711  140000                  BIT     #BIT14!BIT15,(1) ;EVENT OR START MASK
6691  035270  001453                          BEQ     9$
6692  035272  104401  044020                  TYPE,M22
6693  035276  104412                          RDOCT
6694  035300  012661  000064                  MOV     (SP)+,64(1)
6695  035304  104401  044051                  TYPE,M23
6696  035310  104412                          RDOCT
6697  035312  012661  000066                  MOV     (SP)+,66(1)
6698  035316  104401  044067                  TYPE,M24
6699  035322  104412                          RDOCT
6700  035324  012661  000070                  MOV     (SP)+,70(1)
6701  035330  032711  001400                  BIT     #BIT8!BIT9,(1)  ;RANDOM CHANNEL?
6702  035334  001031                          BNE     9$              ;NO EXIT THIS LOOP
6703  035336  016037  000060  001202          MOV     60(0),$PASS     ;YES GET NUMBER OF CHANS.
6704  035344  012703  000062                  MOV     #<JOB1-JOB0-34>,R3
6705  035350  060103                          ADD     R1,R3
6706  035352  010361  000050                  MOV     R3,50(1)
6707
6708  035356                          13$:
6709  035356  104401  035364                  TYPE    ,67$            ;;TYPE ASCIZ STRING
6710  035362  000407                          BR      66$             ;;GET OVER THE ASCIZ
6711                                  ;;67$:  .ASCIZ  <200>#RANDOM CH. =#
6712  035402                          66$:
6713  035402  104412                          RDOCT
6714  035404  012623                          MOV     (SP)+,(3)+
6715  035406  005337  001202                  DEC     $PASS
6716  035412  001361                          BNE     13$
6717  035414  012713  140000                  MOV     #BIT15!BIT14,(3)        ;ADD EOC MARKER.
6718
6719  035420                          9$:
6720
6721  035420  104401  043275                  TYPE,M97
6722  035424  104412                          RDOCT
6723  035426  012661  000074                  MOV     (SP)+,74(1)
6724  035432  032711  000200                  BIT     #BIT7,(1)       ;OUTPUT DEVICE?
6725  035436  001404                          BEQ     10$             ;NO-CONTINUE
6726  035440  104401  043660                  TYPE,   M18             ;MAKE BUFFER
6727  035444  104401  043724                  TYPE,   M19             ;PRESS CONTINUE
6728  035450  010237  056246          10$:    MOV     R2,FRECOR
6729  035454  023727  056246  000000G         CMP     FRECOR,#DRLPXO  ;CORE EXCEEDED?
6730  035462  103100                          BHIS    ERTOUT          ;YES-ERROR MESSAGE!
6731  035464  123737  001126  001124          CMPB    $BDDAT,$GDDAT   ;DONE ALL?
6732  035472  001411                          BEQ     11$
6733  035474  005237  001126                  INC     $BDDAT
6734  035500  012703  055204                  MOV     #JOB1,R3        ;GET JOB SIZ
6735  035504  162703  055066                  SUB     #JOB0,R3
6736  035510  060301                          ADD     R3,R1           ;UPDATE JOB POINTER
6737  035512  000137  034744                  JMP     L1PED           ;LOOP FOR NEXT JOBS DATA
6738
6739  035516                          11$:
```

```
6740  035516  004737  036026          LOOPJ:  JSR     PC,FLASH
6741                                          ;;REPORT DATA
6742
6743  035522  104401  035530                  TYPE    ,65$                ;;TYPE ASCIZ STRING
6744  035526  000414                          BR      64$                 ;;GET OVER THE ASCIZ
6745                                  ;;65$:  .ASCIZ  <200>#REPORT DATA FOR JOB 0?#
6746  035560                          64$:
6747  035560  104411                          RDLIN
6748  035562  012600                          MOV     (SP)+,R0
6749  035564  121027  000131                  CMPB    (0),#'Y
6750  035570  001402                          BEQ     2$
6751  035572  000137  034546          1$:     JMP     MAKEI
6752
6753  035576  012701  055066          2$:     MOV     #JOB0,R1
6754  035602  013700  001754                  MOV     MYTEMP,R0
6755  035606  016137  000060  034454  3$:     MOV     60(1),CHANU         ;GET NU. OF CHANS.
6756  035614  005003                          CLR     R3
6757
6758  035616  013702  034454                  MOV     CHANU,R2            ;NU OF CHANS
6759  035622  010337  034474          4$:     MOV     R3,CHANS
6760  035626  010337  034476                  MOV     R3,CHANF
6761
6762  035632  005237  034540                  INC     REPOR
6763  035636  004737  034146                  JSR     PC,AVERR
6764  035642  005203                          INC     R3
6765  035644  005302                          DEC     R2
6766  035646  001365                          BNE     4$
6767  035650  062701  000116                  ADD     #<JOB1-JOB0>,R1
6768  035654  005300                          DEC     R0
6769  035656  001353                          BNE     3$
6770  035660  000744                          BR      1$
6771  035662  000000                          HALT
6772
6773
6774  035664                  ERTOUT:
6775  035664  104401  035672                  TYPE    ,65$                ;;TYPE ASCIZ STRING
6776  035670  000434                          BR      64$                 ;;GET OVER THE ASCIZ
6777                                  ;;65$:  .ASCIZ  <200>#INSUFFICIENT CORE TO DO REQUEST,PLEASE SPECIFY SMALLER#
6778  035762                          64$:
6779  035762  104401  035770                  TYPE    ,67$                ;;TYPE ASCIZ STRING
6780  035766  000415                          BR      66$                 ;;GET OVER THE ASCIZ
6781                                  ;;67$:  .ASCIZ  <200>#BUFFER  OR LESS BUFFERS#
6782  036022                          66$:
6783  036022  000137  034546                  JMP     MAKEI
6784
6785                                          ;*
6786                                          ;*FLASH!  ROUTINE TO EXERCISE 1-8 JOBS UNTIL DONe.
6787                                          ;*      REQUIRED:
6788                                          ;*        $BERSN MUST CONTAIN ASCII DOR M FOR DEDICATED OR MULTIUSER.
6789                                          ;*        KWT PRE SET UP TO CLOCK STATS
6790                                          ;*        JOB0-JOB7 PRE SET UP
6791                                          ;*        $GDDAT=  NO. OF JOBS TO EXERCISE
6792                                          ;*
6793                                          ;*        CALL=  JSR PC, FLASH
```

```
                                        ;*      RETURNS WHEN ALL DONE.
                                        ;
036026                          FLASH:
                                        ;
036026  004737  040626          2$:     JSR     PC,$RESET
036032  012777  054130  143436          MOV     #DEVLST,@LPADL  ;SET ADDR. OF DEVICE LIST IN RDA WORD
036040  005037  054130                  CLR     DEVLST          ;INDICATE SIZING
036044  113737  001512  054131          MOVB    VERSN,DEVIST+1  ;SET VERSION NUMBER
                                        ;
036052  152777  000001  143406          BISB    #1,@LPCI        ;SET GO.
036060  004737  040542                  JSR PC, SDELAY
036064  012777  044112  143404          MOV     #KWT,@LPADL     ;START CLOCK
036072  052777  000001  143366          BIS     #BIT0,@LPCI
036100  012700  054520                  MOV     #TODOQ,R0
                                        ;
036104  005020                  3$:     CLR     (0)+            ;CLEAR QUE AREA.
036106  020027  055004                  CMP     R0,#DONEC
036112  001374                          BNE     3$
036114  013700  001124                  MOV     $GDDAT,R0       ;GET NO. OF JOBS
036120  012701  054520                  MOV     #TODOQ,R1       ;GET QUES AREA
036124  012702  055066                  MOV     #JOB0,R2        ;PICK UP 1ST JOB ADDRESS
036130  012703  055204                  MOV     #JOB1,R3        ;CALCULATE ADDR DIF.
036134  162703  055066                  SUB     #JOB0,R3
036140  010037  054640                  MOV     R0,TODOC        ;REMEMBER HOW MANY JOBS ARE QUEUED
036144  012700  000010                  MOV     #8.,R0
036150  005712                  4$:     TST     (2)             ;ANYTHING IN THIS JOB??
036152  001401                          BEQ     6$
036154  010221                          MOV     R2,(1)+         ;STORE FIRST JOB
036156                          6$:
036156  060302                          ADD     R3,R2           ;POINT TO NEXT JOB
036160  005300                          DEC     R0              ;DONE ALL JOBS?
036162  001372                          BNE     4$              ;NO-LOOP.
036164  005037  001126                  CLR     $BDDAT
036170  063737  002014  001126          ADD     FUDGE,$BDDAT    ;DON'T WAIT FOR CLOCK JOBS(DR11K ONLY)
036176  005037  002014                  CLR     FUDGE
036202  013700  001556                  MOV     VECT1,R0
036206  012720  036410                  MOV     #OUPSRV,(0)+
036212  012720  000340                  MOV     #340,(0)+
036216  012720  036334                  MOV     #INPSRV,(0)+
036222  012720  000340                  MOV     #340,(0)+
036226  052777  000100  143232          BIS     #BIT6,@LPCI
036234  052777  000100  143230          BIS     #BIT6,@LPCO
036242  013737  001124  001774          MOV     $GDDAT,TAXING           ;GET NO OF REQUESTS.
036250  062737  000200  001774          ADD     #200,TAXING
036256  006337  001774                  ASL     TAXING                  ;MUL X2 TO GET TIME.
036262  005037  177776                  CLR     PS
036266  023737  001124  001126  5$:     CMP     $GDDAT,$BDDAT   ;DONE ALL JOBS?
036274  001414                          BEQ     7$
036276  004737  040542                  JSR PC,SDELAY
036302  005337  001774                  DEC     TAXING
036306  001367                          BNE     5$
036310  017737  143152  001124          MOV     @LPCI,$GDDAT
036316  017737  143150  001126          MOV     @LPCO,$BDDAT
036324  104017                          ERROR   17              ;JOB(S) FAILED TO FINISH (LPA FAULT)
```

```
6848  036326                          7$:
6849  036326  004737  040626                JSR     PC,SRESET
6850  036332  000207                        RTS     PC
6851  036334  005737  054640          INPSRV: TST     TODOC           ;ANYTHING TO dO?
6852  036340  001422                        BEQ     IEX
6853  036342  010046                        MOV     R0,-(SP)
6854  036344  012700  054520                MOV     #TODOQ,R0
6855  036350  005720                  1$:   TST     (0)+
6856  036352  001776                        BEQ     1$
6857  036354  014077  143116                MOV     -(0),@LPADL     ;SAVE RDA ADDR.
6858  036360  005010                        CLR     (0)             ;ZERO THIS JOB
6859  036362  005337  054640                DEC     TODOC
6860  036366  001003                        BNE     2$
6861  036370  042777  000100  143070        BIC     #BIT6,@LPCI     ;DON'T ALLOW ANOTHER INTERRUPT.
6862  036376                          2$:
6863  036376  052777  000001  143062        BIS     #BIT0,@LPCI     ;SET GO!
6864  036404  012600                        MOV     (SP)+,R0
6865  036406  000002                  IEX:  RTI
6866
6867  036410  010046                OUPSRV: MOV     R0,-(SP)        ;SAVE REGS.
6868  036412  010146                        MOV     R1,-(SP)
6869  036414  010246                        MOV     R2,-(SP)
6870  036416  010346                        MOV     R3,-(SP)
6871  036420  010446                        MOV     R4,-(SP)
6872  036422  005237  001774                INC     TAXING          ;NO TIME OUT IF ACTIVE.
6873
6874  036426  017737  143044  001770        MOV     @LPADL,ALPADL   ;SAVE RDA ADDR.
6875  036434  017737  143032  001764        MOV     @LPC0,ALPC0     ;GET CONTROL IN REG.
6876  036442  013700  001764                MOV     ALPC0,R0        ;NOW LETS GET USER NUMBER.
6877  036446  042700  177770                BIC     #^C<7>,R0       ;RID OF REST OF JUNK.
6878  036452  010037  001756                MOV     R0,USJNO        ;GET USJNO
6879  036456  010003                        MOV     R0,R3
6880  036460  006300                        ASL     R0
6881  036462  013701  001770                MOV     ALPADL,R1
6882  036466  105737  001765                TSTB    ALPC0+1         ;SEE IF BIT15 SET SHOWING ERROR
6883  036472  100024                        BPL     NER             ;IF NO ERROR,PROCEED.
6884
6885  036474  017737  142766  001762        MOV     @LPCI,ALPCI     ;ELSE REPORT AN ERROR.
6886  036502  113737  001765  001766        MOVB    ALPC0+1,ALPS0   ;FIX STATUS REG SO ITS EASIER TO READ.
6887  036510  105037  001767                CLRB    ALPS0+1         ;ONLY LOW BYTE.
6888  036514  122737  000250  001766        CMPB    #250,ALPS0      ;LEGAL ERROR. BIT 14 SET IN USW
6889  036522  001003                        BNE     40$
6890  036524  005237  001126                INC     $BDDAT
6891  036530  000440                        BR      OUPEX
6892
6893  036532  104021                  40$:  ERROR 21                ;LPA-11 MICRO CODE REPORTED AN ERROR
6894
6895  036534  013737  001124  001126        MOV     $GDDAT,$BDDAT   ;FIX COUNT FOR EXIT TO MAIN LINE FLASH.
6896  036542  000433                        BR OUPEX
6897
6898  036544  001005                  NER:  BNE     10$
6899  036546  005071  000004                CLR     @4(1)           ;CLEAR USW
6900  036552  013760  001770  054642        MOV     ALPADL,TABLE(0) ;SAVE RDA ADDR. FOR FUTURE REF.
6901
```

```
6902   036560                          10$:
6903   036560  016001  054642                  MOV     TABLE(0),R1     ;GET RDA ADDR FROM TABLE
6904   036564  016103  000004                  MOV     4(1),R3         ;GET ADDR OF USW
6905   036570  116302  000001                  MOVB    1(3),R2         ;GET LAST BUFFER USED
6906   036574  005202                          INC     R2              ;UPDATE FOR NEXT BUFFER
6907   036576  042702  177770                  BIC     #177770,R2      ;CLEAN OFF JUNK
6908   036602  116104  000007                  MOVB    VBM(1),R4       ;GET LAST BUFFER MASK.
6909   036606  042704  177770                  BIC     #177770,R4      ;CLEAN OFF JUNK
6910
6911   036612  042702  177770          30$:    BIC     #^C<7>,R2       ;VALID OFFSET.
6912   036616  110263  000001                  MOVB    R2,1(3)
6913   036622  120204                          CMPB    R2,R4           ;LAST BUFFER ?
6914   036624  003402                          BLE     OUPEX           ;NO-BRANCH
6915   036626  052713  040000                  BIS     #40000,(3)      ;SET STOP JOB BIT
6916   036632  042777  000200  142632  OUPEX:  BIC     #BIT7,@LPC0     ;CLEAR READY OUT
6917
6918   036640  012604                          MOV     (SP)+,R4        ;RESTORE REGISTERS
6919   036642  012603                          MOV     (SP)+,R3
6920
6921   036644  012602                          MOV     (SP)+,R2
6922   036646  012601                          MOV     (SP)+,R1
6923   036650  012600                          MOV     (SP)+,R0
6924   036652  000002                          RTI
6925
```

```
                                        ;*
                                        ;*THIS SUB CODE IS USED TO INITIALIZE THE LPA-11
                                        ;*FIRST WE WILL LOAD MICROCODE INTO KMC-11
                                        ;*NEXT WE WILL INIT BOTH UPROCESSORS
                                        ;*THEN WE WILL LOAD DEVICE TABLE IN SLAVE UP.
                                        ;*THE ORDER OF LOAD IS DETERMINED BY THE USER.
                                        ;*
                                        ;*      CALL=   JSR     R5,$LPAI
                                        ;*              .WORD   0               ;ADDR. OF DEVICE ADDRESS.
                                        ;* ROUTINES REQUIRED:   .LOADLP
                                        ;* PROGRAMS REQUIRED:   DRLPX2
                                        ;*
                                        ;*              ;RETURNS WITH $AERR=1 IF SLAVE
                                        ;*              ;MICRO SAYS AN ADDR. DOES NOT EXSIST. IN THE LIST.
                                        ;*
036654                          $LPAI:
036654  013746  000004                  MOV     4,-(SP)
036660  000413                          BR      31$             ;FIELD DOES NOT HAVE A BUS SWITCH TO
                                                                ;WORRY ABOUT,SO WE WILL UNCONDITIONALLY
                                                                ;BRANCH ARROUD THE NEXT CODE THAT
                                                                ;WORKS BASED ON A BUS SWITCH.
                                                                ;CODE LEFT IN HERE FOR IN HOUSE
                                                                ;PERSONAL WHO MAY PATCH THIS BRANCH
                                                                ;INSTRUCTION TO A <NOP> OCTAL <240>
                                                                ;IN ORDER TO rUN PROGRAM WITH A SWITCH.

                                                                ;NOTE THIS "SWITCH" IS A PIECE OF INHOUSE
                                                                ;TEST EQUIPMENT ONLY IT CONNECTS
                                                                ;THE UNIBUS TO THE I/O BUS FOR
                                                                ;CERTAIN TESTING.
036662  012737  036706  000004          MOV     #30$,4
036670  005237  170000                  INC     170000
036674  104401  036702                  TYPE    ,65$            ;;TYPE ASCIZ STRING
036700  000401                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <7>##
036704                          64$:
036704  000401                          BR      31$
036706  022626                  30$:    CMP     (SP)+,(SP)+
036710  012637  000004          31$:    MOV     (SP)+,4         ;ALL THIS JUNK MUST BE REMOVED!!
036714  005037  037532                  CLR     $AERR
036720  004537  037534                  JSR     R5,$LOAD        ;LOAD MICRO-CODE.
036724  000000G                         .WORD   DRLPX2          ;FILE "DRLPX2.OBJ"

036726  052777  040000  142532          BIS     #BIT14,@KMADO   ;ISSUE KMC+DMC INIT.

036734                          1$:
                                                                ;"HANGS" HERE THEN KMC-11 ERROR.
036734  010146                          MOV     R1,-(SP)
036736  005001                          CLR     R1
036740  005201                  2$:     INC     R1              ;STALL FOR DMC-UP
036742  001376                          BNE     2$
```

```
6980  036744  012777  104000  142514          MOV     #BIT15!BIT11,@KMADO     ;SET RUN, AND ENABLE ARBITRATION.
6981  036752  105201                  25$:    INCB    R1
6982  036754  001376                          BNE     25$
6983
6984  036756  032777  000040  142502          BIT     #BIT5,@KMADO    ;SLAVE READY? (READING IPBM SR)
6985  036764  001401                          BEQ     3$
6986                                                                  ;FATAL LPA-11 ERROR SLAVE NOT READY.
6987  036766  104000                          ERROR
6988
6989  036770  012777  000004  142474  3$:     MOV     #4,@KMAD2       ;READ FAST PATH
6990  036776                          4$:
6991  036776  004537  040444                  JSR     R5, $TOUT       ;-TOUT-CHECK FOR TIMEOUT
6992
6993  037002  104000                          ERROR                   ;/TIME-OUT ERROR
6994                                                                  ;/WE FAILED TO COMPLETE
6995                                                                  ;/CURRENT OPERATION.
6996                                                                  ;/CONTINUES IN THIS LOOP
6997                                                                  ;/WOULD MAKE US "HANG" HERE
6998
6999  037004  000774                          BR              4$
7000
7001                                                                  ;/RETURNS HERE-FROM-TIMED OUT.
7002  037006  122777  000377  142456          CMPB    #377,@KMAD2     ;WAIT TILL KMC DONE COMMAND.
7003  037014  001370                          BNE     4$
7004  037016  122777  000377  142452          CMPB    #377,@KMAD4     ;IF FAST PATH=377 THEN ERROR.
7005  037024  001001                          BNE     35$
7006  037026  104000                          ERROR                   ;IPBM ERROR (SLAVE SIDE)
7007                                                                  ;YOU MUST RUN IPBM DIAGNOSTIC.
7008
7009  037030  122777  000004  142440  35$:    CMPB    #4,@KMAD4       ;IS THIS THE CORRECT VERSION OF MICRO-CODE?
7010  037036  001543                          BEQ     5$              ;YES-CONTINUE.
7011  037040  005227  177777                  INC     #-1
7012  037044  001140                          BNE     5$
7013  037046  005227  177777                  INC     #-1
7014  037052  001135                          BNE     5$
7015  037054  104401  037062                  TYPE    ,67$            ;;TYPE ASCIZ STRING
7016  037060  000440                          BR      66$             ;;GET OVER THE ASCIZ
7017                                  ;;67$:  .ASCIZ  <200>"W A R N I N G  THIS PROGRAM WAS DESiGNED TO RUN WITH VERSION 4"
7018  037162                          66$:
7019  037162  104401  037170                  TYPE    ,69$            ;;TYPE ASCIZ STRING
7020  037166  000430                          BR      68$             ;;GET OVER THE ASCIZ
7021                                  ;;69$:  .ASCIZ  <200>"MICRO-CODE. ANOTHER VERSION CODE WAS DETECTED."
7022  037250                          68$:
7023  037250  104401  037256                  TYPE    ,71$            ;;TYPE ASCIZ STRING
7024  037254  000434                          BR      70$             ;;GET OVER THE ASCIZ
7025                                  ;;71$:  .ASCIZ  <200>"THIS MAY OR MAYNOT CAUSE FALSE ERROR TO BE REPORTED."<200><200>
7026  037346                          70$:
7027
7028  037346  112737  177777  037500  5$:     MOVB    #0-1,11$        ;DAC CODE FOR SLAVE.
7029  037354  012501                  6$:     MOV     (5)+,R1         ;GET NEXT DEVICE ADDR.
7030  037356  021127  000000                  CMP     (R1),#0         ;TERM REACHED?
7031  037362  001444                          BEQ     10$
7032  037364  105237  037500                  INCB    11$
7033  037370  113777  037500  142100          MOVB    11$,@KMAD4      ;FIFO DATA
```

```
7034  037376  004737  037502              JSR     PC,20$          ;ISSUE SEND
7035  037402  112177  142070              MOVB    (R1)+,@KMAD4    ;SEND LOW BYTE OF DEVICE ADDR TO SLAVE.
7036  037406  004737  037502              JSR     PC,20$          ;ISSUE SEND
7037  037412  112177  142060              MOVB    (R1)+,@KMAD4    ;SEND HIGH BYTE OF DEVICE ADDR. TO SLAVE.
7038  037416  004737  037502              JSR     PC,20$
7039
7040  037422  032777  000002  142036  7$: BIT     #BIT1,@KMAD0    ;WAIT FOR FIFO DATA
7041  037430  001374                      BNE     7$              ;=1 NO DATA. =0 DATA.
7042  037432  112777  000002  142032      MOVB    #2,@KMAD2       ;READ FIFO.
7043
7044  037440                          8$:
7045  037440  004537  040444              JSR     R5, $TOUt       ;-TOUT-CHECK FOR TIMEOUT
7046
7047  037444  104000                      ERROR                   ;/TIME-OUT ERROR
7048                                                              ;/WE FAILED TO COMPLETE
7049                                                              ;/CURRENT OPERATION.
7050                                                              ;/CONTINUES IN THIS LOOP
7051                                                              ;/WOULD MAKE US "HANG" HERE
7052
7053  037446  000774                      BR              8$
7054
7055                                                              ;/RETURNS HERE-FROM-TIMED OUT.
7056  037450  122777  000377  142014      CMPB    #377,@KMAD2     ;WAIT FOR READ.
7057  037456  001370                      BNE     8$
7058  037460  105777  142012              TSTB    @KMAD4          ;WAS A ZERO RETURNED?
7059  037464  001734                      BEQ     6$              ;YES GET NEXT ADDR.
7060                                                              ;SLAVE WILL RETURN CODE 0 IF
7061  037466  005237  037532              INC     $AERR           ;DEV PRESENT.   ELSE
7062                                                              ;EXIT $AERR=1 IF SLAVE GIVES ERROR.
7063  037472  005041                      CLR     -(1)            ;GET RID OF REFERENCE TO BAD ADDR.
7064  037474  012601                  10$: MOV     (SP)+,R1
7065  037476  000205                      RTS     R5              ;RETURN ALL ADDR. CHECKED.
7066
7067  037500  000000                  11$: .WORD   0               ;HOLDS DAC CODE PLUS OFFSET
7068                                                              ;TO SLAVES ADDR. TABLE.
7069
7070  037502  112777  000003  141762  20$: MOVB    #3,@KMAD2       ;ISSUE FIFO WRITE
7071  037510                          21$:
7072  037510  004537  040444              JSR     R5, $TOUT       ;-TOUT-CHECK FOR TIMEOUT
7073
7074  037514  104000                      ERROR                   ;/TIME-OUT ERROR
7075                                                              ;/WE FAILED TO COMPLETE
7076                                                              ;/CURRENT OPERATION.
7077                                                              ;/CONTINUES IN THIS LOOP
7078                                                              ;/WOULD MAKE US "HANG" HERE
7079
7080  037516  000774                      BR              21$
7081
7082                                                              ;/RETURNS HERE-FROM-TIMED OUT.
7083  037520  122777  000377  141744      CMPB    #377,@KMAD2     ;KMC CODE WILL RETURN A "377"
7084  037526  001370                      BNE     21$             ;WHEN DONE COMMAND.
7085  037530  000207                      RTS     PC
7086
7087  037532  000000              $AERR:  .WORD   0               ;=0 IF ADDR. LIST OK,=1 IF BAD.
```

```
7088
7089                                        ;*
7090                                        ;*THIS SUB CODE USED TO LOAD MICRO-CODE INTO LPA-11.
7091                                        ;*      CALL =  JSR     R5,$LOAD
7092                                        ;*              .WORD   XX              ;ADDR. OF MICRO CODE.
7093                                        ;*              ;RETURNS HERE
7094                                        ;*      NOTE:   MICRO CODE FILE MUST END IN -1 DATA.
7095                                        ;*
7096
7097  037534  010446                $LOAD:  MOV     R4,-(SP)        ;SAVE R4.
7098  037536  010046                        MOV     R0,-(SP)        ;SAVE R0.
7099  037540  012500                1$:     MOV     (5)+,R0         ;GET PROG. ADDR.
7100  037542  005077  141720               CLR     @KMAD0          ;CLEAR CSR
7101  037546  005077  141724               CLR     @KMAD4          ;CLEAR CRAM ADDR.
7102  037552  052777  002000  141706 2$:    BIS     #2000,@KMAD0    ;SELECT CRAM.
7103  037560  012077  141716               MOV     (0)+,@KMAD6     ;WRITE DATA.
7104  037564  052777  020000  141674       BIS     #20000,@KMAD0   ;SET CRAM WRITE
7105  037572  005077  141670               CLR     @KMAD0          ;DISABLE CRAM.
7106  037576  005277  141674               INC     @KMAD4          ;UPDATE CRAM ADDR.
7107  037602  021027  177777               CMP     (0),#-1         ;ALL DONE?
7108  037606  001361                       BNE     2$              ;NO LOOP.
7109  037610  005077  141662               CLR     @KMAD4          ;CLEAR CRAM ADDR.
7110  037614  016500  177776               MOV     -2(5),R0        ;GET MICRO CODE ADDR.
7111
7112  037620  052777  002000  141640 3$:    BIS     #2000,@KMAD0    ;SELECT CRAM
7113  037626  022077  141650               CMP     (R0)+,@KMAD6    ;DATA OK?
7114  037632  001013                       BNE     5$              ;NO - REPORT AN ERROR.
7115  037634  021027  177777               CMP     (0),#-1         ;ALL DONe?
7116  037640  001405                       BEQ     4$              ;YES - EXIT
7117  037642  005077  141620               CLR     @KMAD0          ;NO - DESeLECT CRAM.
7118  037646  005277  141624               INC     @KMAD4          ;UPDATE CRAM ADDR.
7119  037652  000762                       BR      3$
7120
7121  037654  012600                4$:     MOV     (SP)+,R0        ;RESTORE R0
7122  037656  012604                        MOV     (SP)+,R4        ;RESTORE R4
7123  037660  000205                        RTS     R5              ;EXIT
7124
7125  037662                        5$:                             ;COME HERE ON LOAD ERROR
7126  037662  005745                        TST     -(5)
7127  037664  105204                        INCB    R4              ;UPDATE ERROR COUNTER.
7128  037666  100324                        BPL     1$              ;IF NOT TOO MANY, TRY AGAIN.
7129  037670  000000                        HALT                    ;MICRO CODE LOAD ERROR.
7130                                                                ;KMC-11 FAULT. YOU COULD TRY
7131  037672  000722                        BR      1$              ;TO PRESS CONTINUE TO GIVE IT
7132                                                                ;ANOTHER CHANCE, BUT I DOUBT
7133                                                                ;THAT THAT WOULD WORK. SINCE I'VE
7134                                                                ;ALREADY GIVEN IT 177 (OCTAL) CHANCES.
7135                                                                ;TRY RUNNING THE KMC-11 DIAGNOSTIC.
7136
7137
7138                                        ;
7139                                        ;*THIS ROUTINE ISSUES A WRITE COMMAND TO THE LPA-11
7140                                        ;*
7141                                        ;*      CALL =  JSR     R5,$TLKW
```

```
7142                                                ;*              .WORD   0               ;OFFSET OF DEVICE ADDR.
7143                                                ;*              .WORD   0               ;DATA TO BE WRITTEN
7144                                                ;*
7145  037674  010046                        $TLKW:  MOV     R0,-(SP)        ;SAVE R0
7146  037676  012500                                MOV     (5)+,R0         ;GET DEVICE OFFSET
7147  037700  052700  000340                        BIS     #340,R0         ;ADD WRITE CODE.
7148  037704  004737  040156                        JSR     PC,$LPW         ;WAIT FOR FAST PATH READY
7149  037710  010037  040002                        MOV     R0,W1
7150  037714  010077  141556                        MOV     R0,@KMAD4
7151  037720  112777  000005  141544              MOVB    #5,@KMAD2       ;ISSUE FAST PATH WRITE
7152  037726  004737  040156                        JSR     PC,$LPW         ;WAIT FOR RDY
7153  037732  011537  040004                        MOV     (5),W2
7154  037736  112577  141534                        MOVB    (5)+,@KMAD4     ;WRITE LOW BYTE DATA.
7155
7156  037742  112777  000005  141522              MOVB    #5,@KMAD2       ;FP WRITE
7157  037750  004737  040156                        JSR     PC,$LPW
7158  037754  111537  040006                        MOVB    (5),W3
7159  037760  112577  141512                        MOVB    (5)+,@KMAD4     ;WRITE HIGH BYTE
7160  037764  112777  000005  141500              MOVB    #5,@KMAD2
7161  037772  004737  040156                        JSR     PC,$LPW
7162  037776  012600                                MOV     (SP)+,R0
7163  040000  000205                                RTS     R5              ;EXIT DONE.
7164  040002  000000                        W1:     0
7165  040004  000000                        W2:     0
7166  040006  000000                        W3:     0
7167
7168                                                ;*
7169                                                ;*THIS ROUTINE ISSUES A READ COMMAND TO THE LPA-11
7170                                                ;*
7171                                                ;*      CALL =  JSR     R5,$TLKr
7172                                                ;*              .WORD   0               ;OFFSET OF DEVICE
7173                                                ;*              ;RETURNS HERE
7174                                                ;*DATA IN WORD $DATR
7175                                                ;*
7176
7177  040010  010046                        $TLKR:  MOV     R0,-(SP)        ;SAVE R0
7178  040012  012500                                MOV     (5)+,R0         ;GET OFFSET
7179  040014  052700  000300                        BIS     #300,R0         ;ADD READ CODE
7180  040020  004737  040156                        JSR     PC,$LPW         ;WAIT TILL READY
7181  040024  110077  141446                        MOVB    R0,@KMAD4
7182  040030  112777  000005  141434              MOVB    #5,@KMAD2       ;ISSUE WRITE FP
7183  040036  004737  040156                        JSR     PC,$LPW
7184  040042  010037  040152                        MOV     R0,RD1
7185  040046                                1$:
7186  040046  004537  040444                        JSR     R5, $TOUT       ;-TOUT-CHECK FOR TIMEOUT
7187
7188  040052  104000                                ERROR                   ;/TIME-OUT ERROR
7189                                                                        ;/WE FAILED TO COMPLETE
7190                                                                        ;/CURRENT OPERATION.
7191                                                                        ;/CONTINUES IN THIS LOOP
7192                                                                        ;/WOULD MAKE US "HANG" HERE
7193
7194  040054  000774                                BR              1$
7195
```

```
7196                                                    ;/RETURNS HERE-FROM-TIMED OUT.
7197  040056  032777  000040  141402          BIT     #BIT5,@KMADO    ;FAST PATH GOT DATA?
7198  040064  001370                          BNE     1$
7199  040066  112777  000004  141376          MOVB    #4,@KMAD2       ;ISSUE FAST PATH READ
7200  040074  004737  040156                  JSR     PC,$LPW
7201  040100  117737  141372  040154          MOVB    @KMAD4,$DATR    ;GET LOW BYTE
7202  040106                          2$:
7203  040106  004537  040444                  JSR     R5, $TOUT       ;-TOUT-CHECK FOR TIMEOUT
7204
7205  040112  104000                          ERROR                   ;/TIME-OUT ERROR
7206                                                                  ;/WE FAILED TO COMPLETE
7207                                                                  ;/CURRENT OPERATION.
7208                                                                  ;/CONTINUES IN THIS LOOP
7209                                                                  ;/WOULD MAKE US "HANG" HERE
7210
7211  040114  000774                          BR              2$
7212
7213                                                    ;/RETURNS HERE-FROM-TIMED OUT.
7214  040116  032777  000040  141342          BIT     #BIT5,@KMADO    ;FAST PATH READY?
7215  040124  001370                          BNE     2$
7216  040126  112777  000004  141336          MOVB    #4,@KMAD2       ;ISSUE FAST PATH READ
7217  040134  004737  040156                  JSR     PC,$LPW
7218  040140  117737  141332  040155          MOVB    @KMAD4,$DATR+1  ;SAVE HIGH BYTE
7219  040146  012600                          MOV     (SP)+,R0
7220  040150  000205                          RTS     R5
7221  040152  000000                  RD1:    0
7222  040154  000000                  $DATR:  .WORD   0
7223                                          ;
7224                                          ;THIS ROUTINE WAITS FOR KMC-CODE TO BECOME READY AS WELL
7225                                          ;AS FAST PATH TO BE READ.
7226                                          ;
7227                                          ;       CALL =  JSR     PC,$LPW
7228                                          ;
7229                                          ;IT WILL TIME OUT IF TOO MUCH TIME IS TAKEN BY
7230                                          ;THE MICRO-PROCESSORS AND REPORT AN ERROR, THEN HALT.
7231                                          ;
7232
7233  040156  010146                  $LPW:   MOV     R1,-(SP)        ;SAVE R1
7234  040160  005001                          CLR     R1
7235  040162  122777  000377  141302  1$:     CMPB    #377,@KMAD2     ;FINISHED INSTRUCTION?
7236  040170  001403                          BEQ     2$
7237  040172  005201                          INC     R1              ;TIME OUT?
7238  040174  001372                          BNE     1$
7239  040176  000411                          BR      10$
7240
7241  040200  032777  000020  141260  2$:     BIT     #BIT4,@KMADO    ;FAST PATH READ?
7242  040206  001403                          BEQ     3$
7243  040210  005201                          INC     R1              ;NO - TIME OUT?
7244  040212  001372                          BNE     2$
7245  040214  000402                          BR      10$             ;YES - REPORT AN ERROR
7246
7247  040216  012601                  3$:     MOV     (SP)+,R1        ;RESTORE R1
7248  040220  000207                          RTS     PC              ;EXIT
7249
```

```
040222                          10$:
040222  104401  040230                  TYPE    ,65$            ;;TYPE ASCIZ STRING
040226  000407                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <200>#LPA-11 FAULT#
040246                          64$:
                                
040246  000000                  11$:    HALT                    ;LPA-11 FAULT RUN LPA-11
040250  000776                          BR      11$             ;DIAGNOSTICS.
                                
                                
                                ;*
                                ;*THIS ROUTINE PROVIDES THE LINKAGE FROM USER CODE TO
                                ;*A DEVICE ADDRESS ON THE I/O BUSS FOR WRITE ONLY.
                                ;*
                                ;* FIRST WE WILL DETERMINE IF THE ADDRESS HAS BEEN USED
                                ;* BEFORE. IF NOT WE HAVE TO INITIALIZE THE LPA WITH
                                ;* THAT ADDRESS.
                                ;* WHEN THE ADDR. IS KNOWN BY THE LPA, DO THE OUTPUT BY
                                ;* $TLKw
                                ;*
                                
040252  010046          $OUTLP: MOV     R0,-(SP)                ;SAVE R0
040254  010146                  MOV     R1,-(SP)                ;SAVE R1
                                
040256  012700  001514                  MOV     #.DVLS,R0       ;PROGRAM DEFINED LIST.
040262  005001                  CLR     R1
040264  005710          1$:     TST     (0)                     ;TERMINATOR REACHED?
040266  001421                  BEQ     10$                     ;YES NEXT STEP.
040270  027520  000000                  CMP     @(5),(0)+       ;MATCH WITH ADDR IN LIST?
040274  001402                  BEQ     2$
040276  005201                  INC     R1
040300  000771                  BR      1$
                                
040302  010137  040320          2$:     MOV     R1,3$           ;SAVE OFFSET, DEVICE KNOWN.
040306  005725                  TST     (5)+
040310  013537  040322                  MOV     @(5)+,4$        ;GET DATA TO BE WRITTEN
040314  004537  037674                  JSR     R5,$TLKw        ;DO WRITE
040320  000000          3$:     .WORD   0                       ;DEVICE OFFSET
040322  000000          4$:     .WORD   0                       ;DATA TO BE WRITTEN.
040324  012601                  MOV     (SP)+,R1
040326  012600                  MOV     (SP)+,R0
040330  000205                  RTS     R5
040332  017520  000000  10$:     MOV     @(5),(0)+       ;SAVE ADDR.
040336  005010                  CLR     (0)
040340  004537  036654                  JSR     R5,$LPAI
040344  001514                  .WORD   .DVLS
040346  000755                  BR      2$
                                
                                ;*
                                ;*THIS ROUTINE PROVIDES THE LINKAGE FROM USER CODE
                                ;*TO A DEVICE ADDR. ON THE I/O BUSS FOR READ ONLY.
                                ;*
                                ;*FIRST WE WILL DETERMINE IF THE ADDRESS HAS BEEN
```

```
                                        ;*USED BEFORE. IF NOT, WE HAVE TO INITIALIZE THE LPA
                                        ;*WITH THE NEW ADDR.
                                        ;*WHEN THE ADDR IS KNOWN WE CAN DO oUTPUT THROUGH
                                        ;*$TLKR
                                        ;*      CALL THROUGH    MOVEI   DATA,ADDR.
                                        ;*              WHICH EQUALS:
                                        ;*                      JSR     R5,$INLP
                                        ;*                      .WORD   XX      ADDR OF DEVICE
                                        ;*                      .WORD   YY      ADDR TO STORE READ DATA.

040350  010046          $INLP:  MOV     R0,-(SP)        ;SAVE R0
040352  010146                  MOV     R1,-(SP)        ;SAVE R1

040354  012700  001514          MOV     #.DVLS,R0       ;PROG DEFINED ADDR. LIST.
040360  005001                  CLR     R1
040362  005710          1$:     TST     (0)             ;EOL REACHED?
040364  001420                  BEQ     10$             ;YES - DEFINE NEW ADDR.

040366  027520  000000          CMP     @(5),(0)+       ;ADDR. MATCH?
040372  001402                  BEQ     2$
040374  005201                  INC     R1
040376  000771                  BR      1$

040400  010137  040412  2$:     MOV     R1,3$           ;SAVE LIST OFFSET
040404  005725                  TST     (5)+
040406  004537  040010          JSR     R5,$TLKr        ;GO READ DEVICE
        040412          $OFS=.
040412  000000          3$:     .WORD   0               ;OFFSET OF DEVICE

040414  013735  040154          MOV     $DATR,@(5)+     ;STORE DATA.
040420  012601                  MOV     (SP)+,R1        ;RESTORE R1
040422  012600                  MOV     (SP)+,R0        ;RESTORE R2
040424  000205                  RTS     R5              ;EXIT

040426  017520  000000  10$:     MOV     @(5),(0)+
040432  005010                  CLR     (0)
040434  004537  036654          JSR     R5,$LPAI
040440  001514                  .WORD   .DVLS
040442  000756                  BR      2$
                                        ;*
                                        ;*$TOUT ROUTINE USED TO WATCH IF
                                        ;*      WE'RE IN A LOOP TOO-LONG
                                        ;*      CALL=   JSR R5, $TOUT
                                        ;*              ERROR X     ;RETURNS HERE ON TIMEOUT
                                        ;*              BR
                                        ;*              ;RETURNS HERE NO ERROR
                                        ;*

040444  020537  040500  $TOUt:   CMP     R5,$SAD         ;SAME ADDR?
040450  001405                  BEQ     1$
040452  010537  040500          MOV     R5,$SAD         ;NO-SAVE THIS ADDR.
040456  005037  040502          CLR     $CNT            ;CLR CNT AT ADDR.
040462  000403                  BR      2$
040464  005237  040502  1$:     INC     $CNT            ;OVERFLOW?
```

```
7358  040470  100402                    BMI     3$                ;YES-ERROR RETURN
7359  040472  062705  000004  2$:       ADD     #4,R5             ;NO-NON ERROR RETURN
7360  040476  000205            3$:     RTS     R5                ;RETURN.
7361
7362  040500  000000            $SAD:   .WORD   0                 ;CONTAINS LOOP ADDR.
7363  040502  000000            $CNT:   .WORD   0                 ;# OF TIMES AT ADDR.
7364
7365                                    ;*
7366                                    ;* THIS ROUTINE REPLACES WHAT THE USER WOULD ORDINARILY
7367                                    ;*USE FOR A RESET.  FIRST,WE DO A RESET INSTRUCTION.
7368                                    ;*THEN WE CLR ".DVLST" WHICH FORCES US TO RESET BOTH THE
7369                                    ;*KMC AND DMC AS SOON AS A DEVICE IS REFERENCED.
7370                                    ;*
7371                                    ;*      CALL=JSR        PC,$RESET       ;REPLACES "RESET INSTRUCTION
7372                                    ;*              ;RETURNS HERE.
7373                                    ;*
7374  040504  000005            $RESET: RESET                     ;RESET THE WORLD.
7375
7376                            ;*      MOV     @2$,1$  ;/READ DEVICE REG 2$,PUT DATA IN 1$.
7377  040516  005737  037532            TST     $AERR             ;IF NO ERROR,LOOP
7378  040522  001004                    BNE     10$               ;THERE WAS AN ERROR.
7379  040524  062737  000002  040540    ADD     #2,2$             ;UPDATE DEVICE ADDR.
7380                                                              ;YOU SEE ,WE HAVE TO PROTECT OUR SELF!
7381                                                              ;IF 2$ CONTAINED A VALID ADDR,WE
7382                                                              ;MUST KEEP TRYING UNTIL WE GENERATE
7383                                                              ;AN INVALID ADDR.
7384  040532  000764                    BR      $RESET
7385  040534                    10$:
7386  040534  000207                    RTS     PC
7387  040536  000000            1$:     .WORD   0                 ;JUNK LOC.
7388  040540  160000            2$:     .WORD   160000            ;DUMB ADDR. FORCES INIT OF DMC/KMC.
7389
7390
7391                                    ;
7392                                    ;SDELAY- ROUTINE TO GIVE A MINOR DELAY.
7393                                    ;        IS NOT TIME DEPENDENT CODE SENCE
7394                                    ;        NOT USED TO GET SPECIFIC TIME BUT
7395                                    ;        JUST A LITTLE DELAY.
7396                                    ;
7397                                    ;        THAT IS UNLESS A REAL TIME CLOCK IS PRESENT!
7398                                    ;        THEN WE'LL GENERATE A TIME BETWEEN 16MS TO 32 MS
7399                                    ;
7400                                    ;
7401                                    ;        CALL=   JSR PC,  SDELAY
7402                                    ;
7403  040542                    SDELAY:
7404  040542  005737  040624            TST     RTCCSR            ;CLOCK PRESENT?
7405  040546  100016                    BPL     10$
7406  040550  012737  000002  040614    MOV     #2,TIME
7407  040556  052777  000115  000040    BIS     #115,@RTCCSR      ;START CLOCK
7408  040564  005037  177776            CLR     PS
7409  040570  005737  040614    1$:     TST     TIME
7410  040574  001375                    BNE     1$
7411  040576  005077  000022            CLR     @RTCCSR           ;STOP CLOCK
```

```
                                        RTS PC
040602  000207
040604  105237  040614          10$:  INCB    TIME
040610  001375                        BNE     10$
040612  000207                        RTS     PC

040614  000000                  TIME: .WORD   0

040616  005337  040614         CLKINT: DEC    TIME
040622  000002                         RTI
040624  000000                 RTCCSR: .WORD   0                ;CLOCK CSR IF USED.
                                       ;
                                       ;SRESET WILL RESET WHOLE LPA SYSTEM. WITH USER MICRO-CODE.
                                       ;CALL= JSR PC,SRESET
                                       ;
040626  052777  040000  140632 SRESET: BIS     #BIT14,@KMADO           ;SET INIT BIT
040634  012700  000144                 MOV     #100.,R0
040640  004737  040542          1$:    JSR     PC,$DELAY               ;DELAY
040644  005300                         DEC     R0
040646  001374                         BNE     1$
040650  012737  044120  040700         MOV     #MMAST,10$
040656  122737  000115  001772         CMPB    #'M,$VERSN
040664  001403                         BEQ     9$
040666  012737  050124  040700         MOV     #DMAST,10$
040674  004537  037534          9$:    JSR     R5,$LOAD                ;RELOAD MICRO CODE.
040700  044120                  10$:   .WORD   MMAST                   ;CURREN VERSION.

040702  012777  104000  140556         MOV     #BIT15!BIT11,@KMADO     ;START  KMC.
040710  012700  000005                 MOV     #5.,R0
040714  004737  040542          2$:    JSR     PC,$DELAY
040720  005300                         DEC     R0
040722  001374                         BNE     2$
040724  005077  140542                 CLR     @KMAD2
040730  005077  140542                 CLR     @KMAD4
040734  000207                         RTS PC
                                       ;*
                                       ;*THIS MACRO ALLOWS THE OPERATOR TO TALK TO
                                       ;*ANY DEVICE ON THE I/O BUS
                                       ;*USER MUST START AT THIS ADDR.
                                       ;*HE MUST SAY EITHER "E" FOR EXAMINE, OR "D" FOR DEPOSIT.
                                       ;*"E" IS DEFAULT.
                                       ;*NEXT, HE MUST SUPPLY AN ADDR.
                                       ;*NOTE IF ADDR. IS NOT FOUND ON I/O BUS, A HALT
                                       ;*WILL OCCUR.

040736                         $UTK:
040736  005037  001514                 CLR     .DVLS
040742                         21$:
040742  104401  040750                 TYPE    ,65$                    ;;TYPE ASCIZ STRING
040746  000405                         BR      64$                     ;;GET OVER THE ASCIZ
                               ;;65$:  .ASCIZ  <200>#E OR D?#
040762                         64$:
040762  105777  140156          1$:    TSTB    @$TKS
```

```
7466  040766  100375                    BPL     1$
7467  040770  117737  140152  041112    MOVB    @$TKB,20$       ;GET INPUT
7468  040776  104401  041112            TYPE,   20$             ;ECHO, NEXT MESSAGE.
7469  041002  142737  000240  041112    BICB    #240,20$        ;STRIP PARITY, LC
7470  041010  104412                    RDOCT                   ;GET ADDR.
7471  041012  012637  041110            MOV     (SP)+,14$
7472  041016  123727  041112  000104    CMPB    20$,#'D         ;DEPOSIT?
7473  041024  001411                    BEQ     10$
7474
7475  041026  004537  040350            JSR     R5,$INLP        ;GET DATA
7476  041032  041110            2$:     .WORD   14$
7477  041034  041046                    .WORD   5$
7478
7479  041036  013746  041046            MOV     5$,-(SP)        ;;SAVE 5$ FOR TYPEOUT
7480  041042  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
7481  041044  000736                    BR      21$             ;LOOP.
7482  041046  000000            5$:     .WORD   0
7483
7484  041050                    10$:
7485  041050  104401  041056            TYPE    ,67$            ;;TYPE ASCIZ STRING
7486  041054  000404                    BR      66$             ;;GET OVER THE ASCIZ
7487                            ;;67$:  .ASCIZ  <200>#DATA= #
7488  041066                    66$:
7489  041066  104412                    RDOCT
7490  041070  012637  041106            MOV     (SP)+,13$
7491
7492  041074  004537  040252    11$:    JSR     R5,$OUTLP       ;OUTPUT ROUTINE.
7493  041100  041110    12$:            .WORD   14$             ;DEVICE ADDR.
7494  041102  041106                    .WORD   13$             ;DATA
7495  041104  000716                    BR      21$
7496
7497  041106  000000            13$:    .WORD   0
7498  041110  000000            14$:    .WORD   0
7499  041112  100001  042504  044526  20$:    .ASCIZ  <1><200>#DEVICE ADDR= #
7500  041120  042503  040440  042104
7501  041126  036522  000040
7502                                    .EVEN
7503
7504
7505
7506                            ;
7507                            ;THIS ROUTINE LOOKS THROUGH CURENT .DVLS FOR A/D ADDR.
7508                            ;IF UNFOUND,GENERATES IT. THIS ROUTINE'S WHOLE PURPOSE IS
7509                            ;TO SET UP THE USER PROGRAM TO LINK TO FILE "DRLPX2" FOR
7510                            ;SAMPLE TAKEING PURPOSES.
7511                            ;       TO TAKE SAMPLES, THE USER PROGRAM MUST SET UP
7512                            ;       A/D CSR IN BSEL 4,AND 5.
7513                            ;       (2) HE MUST CALL THIS ROUTINE:
7514                            ;               JSR     R5,$PUTS        ;CALL SET UP ROUTINE.
7515                            ;               .WORD   ADCSR           ;ADDR. OF A/D CSR.
7516                            ;               ;RETURNS HERE ;KMC BSEL 3,6,7 PERMINENTLY SET UP
7517                            ;                               ;(UNTILL ONE DOES A RESET)
7518                            ;
7519                            ;       (3)THE USER MUST PUT CODE 006 INTO KMC REG 2 TO
```

```
                                        ;          START CONVERSION  CAUTION*DO WITH MOVB 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.
                                        ;
041132  012537  041142          $PUTS:  MOV     (5)+,1$                 ;GET ADDR OF ADDR. OF A/D
041136  004537  040350                  JSR     R5,$INLP
041142  000000                  1$:     .WORD   0
041144  041240                          .WORD   10$
041146  113777  040412  140326          MOVB    $OFS,@KMAD6
041154  113777  040412  140322          MOVB    $OFS,@KMAD7
041162  013737  041142  041202          MOV     1$,2$
041170  062737  000002  041202          ADD     #2,2$
041176  004537  040350                  JSR     R5,$INLP
041202  000000                  2$:     .WORD   0
041204  041240                          .WORD   10$
041206  113777  040412  140260          MOVB    $OFs,@KMAD3
041214  152777  000340  140260          BISB    #340,@KMAD6
041222  152777  000300  140254          BISB    #300,@KMAD7
041230  152777  000300  140236          BISB    #300,@KMAD3
041236  000205                          RTS     R5
041240  000000                  10$:    .WORD   0

                                .SBTTL  TRAP DECoDER

                                ;;**************************************************************
                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;*GO TO THAT ROUTINE.

041242  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
041244  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
041250  005740                          TST     -(R0)           ;;BACKUP BY 2
041252  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
041254  006300                          ASL     R0              ;;POSITION FOR INDEXING
041256  016000  041276                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
041262  000200                          RTS     R0              ;;GO TO ROUTINE

                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

041264  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
041266  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
041274  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.

                                ;       ROUTINE
                                ;       -------
```

```
 7574  041276  041264                  $TRPAD: .WORD   $TRAP2
 7575  041300  033052                          $TYpE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 7576  041302  031036                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 7577  041304  031012                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 7578  041306  031052                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 7579  041310  031240                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 7580
 7581  041312  032346                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING
 7582
 7583  041314  032276                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
 7584  041316  032560                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
 7585  041320  032700                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
 7586  041322  033334                          $RDOCT  ;;CALL=RDOCT    TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY
 7587
 7588
 7589                                  ;ROUTINE TO READ NUMBERS FROM A BUFFER
 7590  041324                  RBUFR:
 7591  041324  104401  041332                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 7592  041330  000410                          BR      64$             ;;GET OVER THE ASCIZ
 7593                          ;;65$:  .ASCIZ  <200>#BUFFER START? #
 7594  041352                  64$:
 7595  041352  104412                          RDOCT
 7596  041354  012600                          MOV     (SP)+,R0
 7597  041356  001002                          BNE     1$
 7598  041360  012700  056246                  MOV     #BUFFER,R0
 7599  041364                  1$:
 7600  041364  104401  041372                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 7601  041370  000406                          BR      66$             ;;GET OVER THE ASCIZ
 7602                          ;;67$:  .ASCIZ  #CHAN INC? #
 7603  041406                  66$:
 7604  041406  104412                          RDOCT
 7605  041410  012601                          MOV     (SP)+,R1
 7606  041412  001002                          BNE     2$
 7607  041414  012701  000020                  MOV     #16.,R1
 7608  041420  006301                  2$:     ASL R1
 7609  041422  104401  041430                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 7610  041426  000407                          BR      68$             ;;GET OVER THE ASCIZ
 7611                          ;;69$:  .ASCIZ  <200>#ADDR.    DATA#
 7612  041446                  68$:
 7613  041446                  3$:
 7614  041446  104401  041454                  TYPE    ,71$            ;;TYPE ASCIZ STRING
 7615  041452  000401                          BR      70$             ;;GET OVER THE ASCIZ
 7616                          ;;71$:  .ASCIZ  <200>##
 7617  041456                  70$:
 7618  041456  010046                          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
 7619  041460  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 7620  041462  104401  041470                  TYPE    ,73$            ;;TYPE ASCIZ STRING
 7621  041466  000402                          BR      72$             ;;GET OVER THE ASCIZ
 7622                          ;;73$:  .ASCIZ  #  #
 7623  041474                  72$:
 7624  041474  011046                          MOV     (0),-(SP)       ;;SAVE (0) FOR TYPEOUT
 7625  041476  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 7626  041500  060100                          ADD R1,R0
 7627  041502  000761                          BR      3$
```

```
                                                .SBTTL          .ASCIZ  MESSAGES.
041504  046200  040520  040440  EM1:    .ASCIZ  <200>#LPA ADDRESS ERROR#
041512  042104  042522  051523
041520  042440  051122  051117
041526     000
                                        .ASCIZ  <200>#LPA (KMC-11) DATA ERROR#
041527     200  050114  020101  EM2:
041534  045450  041515  030455
041542  024461  042040  052101
041550  020101  051105  047522
041556  000122
                                        .ASCIZ  <200>#LPA (KMC-11) INSTRUCTION ERROR#
041560  046200  040520  024040  EM3:
041566  046513  026503  030461
041574  020051  047111  052123
041602  052522  052103  047511
041610  020116  051105  047522
041616  000122
                                        .ASCIZ  <200>#LPA (M8254) INIT ERROR#
041620  046200  040520  024040  EM4:
041626  034115  032462  024464
041634  044440  044516  020124
041642  051105  047522  000122
                                        .ASCIZ  <200>#LPA (M8254) SILO DATA ERROR#
041650  046200  040520  024040  EM6:
041656  034115  032462  024464
041664  051440  046111  020117
041672  040504  040524  042440
041700  051122  051117     000
                                        .ASCIZ  <200>#LPA (M8254) FAST PATH DATA ERROR#
041705     200  050114  020101  EM7:
041712  046450  031070  032065
041720  020051  040506  052123
041726  050040  052101  020110
041734  040504  040524  042440
041742  051122  051117     000
                                        .ASCIZ  <200>#LPA (AD11K) CSR ERROR#
041747     200  050114  020101  EM10:
041754  040450  030504  045461
041762  020051  051503  020122
041770  051105  047522  000122
                                        .ASCIZ  <200>#LPA (KW11K) CSR ERROR#
041776  046200  040520  024040  EM11:
042004  053513  030461  024513
042012  041440  051123  042440
042020  051122  051117     000
                                        .ASCIZ  <200>#LPA (DR11K) ERROR#
042025     200  050114  020101  EM12:
042032  042050  030522  045461
042040  020051  051105  047522
042046  000122
                                        .ASCIZ  <200>#LPA (AA-11K) ERROR#
042050  046200  040520  024040  EM14:
```

```
 7682  042056  040501  030455  045461
 7683  042064  020051  051105  047522
 7684  042072  000122
 7685
 7686  042074  046200  040520  024040  EM15:   .ASCIZ  <200>#LPA (AR11) ERROR#
 7687  042102  051101  030461  020051
 7688  042110  051105  047522  000122
 7689
 7690  042116  046200  040520  024040  EM16:   .ASCIZ  <200>#LPA (LPS-11) ERROR#
 7691  042124  050114  026523  030461
 7692  042132  020051  051105  047522
 7693  042140  000122
 7694
 7695  042142  046200  040520  030455  EM17:   .ASCIZ  <200>#LPA-11 FUNCTIONAL ERROR#
 7696  042150  020061  052506  041516
 7697  042156  044524  047117  046101
 7698  042164  042440  051122  051117
 7699  042172     000
 7700
 7701  042173     200  042522  042520  EM20:   .ASCIZ  <200>#REPEATIBILITY OR CHAN AVERAGE ERROR#
 7702  042200  052101  041111  046111
 7703  042206  052111  020131  051117
 7704  042214  041440  040510  020116
 7705  042222  053101  051105  043501
 7706  042230  020105  051105  047522
 7707  042236  000122
 7708
 7709  042240  042600  051122  041520  DH1:    .ASCIZ  <200>#ERRPC   ADDRESS#
 7710  042246  020040  040440  042104
 7711  042254  042522  051523     000
 7712
 7713  042261     200  051105  050122  DH2:    .ASCIZ  <200>#ERRPC   EXP'ED   REC'ED#
 7714  042266  020103  020040  054105
 7715  042274  023520  042105  020040
 7716  042302  051040  041505  042447
 7717  042310  000104
 7718
 7719  042312  042600  051122  041520  DH3:    .ASCIZ  <200>#ERRPC#
 7720  042320     000
 7721  042321     200  051105  050122  DH17:   .ASCIZ  <200>#ERRPC   TESTNO  EXP'ED  REC'ED#
 7722  042326  020103  020040  042524
 7723  042334  052123  047516  020040
 7724  042342  054105  023520  042105
 7725  042350  020040  042522  023503
 7726  042356  042105     000
 7727
 7728  042361     200  042524  052123  DH20:   .ASCIZ  <200>#TEST    S/B      CHAN    HIGH    LOW#
 7729  042366  020040  020040  027523
 7730  042374  020102  020040  020040
 7731  042402  041440  040510  020116
 7732  042410  020040  044040  043511
 7733  042416  020110  020040  046040
 7734  042424  053517     000
 7735  042427     200  051105  050122  DH21:   .ASCIZ  <200>#ERRPC   TSTNO   USJNO   USRDA   ALPCO   ALPCI   ALPSO#
```

```
7736  042434  020103  020040  051524
7737  042442  047124  020117  020040
7738  042450  051525  047112  020117
7739  042456  020040  051525  042122
7740  042464  020101  020040  046101
7741  042472  041520  020117  020040
7742  042500  046101  041520  020111
7743  042506  020040  046101  051520
7744  042514  000117
7745
7746                                  .EVEN
7747  042516  001116  001466  000000  DT1:    .WORD   $ERRPC,KMADO,0
7748
7749  042524  001116  001124  001126  DT2:    .WORD   $ERRPC,$GDDAT,$BDDAT,0
7750  042532  000000
7751
7752  042534  001116  000000          DT3:    .WORD   $ERRPC,0
7753  042540  001116  001200  001124  DT17:   .WORD   $ERrPC,$TESTN,$GDDAT,$BDDAT,0
7754  042546  001126  000000
7755
7756  042552  001200  034514  034466  DT20:   $TESiN,AVEXP,CHAN,RHIGH,RLOW,0
7757  042560  034522  034520  000000
7758
7759  042566  001116  001200  001756  DT21:   $ERRPC,$TESTN,USJNO,ALPADL,ALPCO,ALPCI,ALPSO,0
7760  042574  001770  001764  001762
7761  042602  001766  000000
7762  042606  000000  000000          DFO:    .WORD   0,0
7763
7764  042612  020040     000          M2SP:   .ASCIZ  #  #
7765
7766
7767  042615     200  040515  042513  M1:     .ASCII  <200>"MAKE YOUR OWN JOB(S) ROUTINE FOR LPA-11"
7768  042622  054440  052517  020122
7769  042630  053517  020116  047512
7770  042636  024102  024523  051040
7771  042644  052517  044524  042516
7772  042652  043040  051117  046040
7773  042660  040520  030455     061
7774  042665     200  046120  040505  .ASCII  <200>"PLEASE ANSWER ALL QUESTIONS, IF IN DOUbT, REFERR TO DOCUMENTATION"
7775  042672  042523  040440  051516
7776  042700  042527  020122  046101
7777  042706  020114  052521  051505
7778  042714  044524  047117  026123
7779  042722  044440  020106  047111
7780  042730  042040  052517  052102
7781  042736  020054  042522  042506
7782  042744  051122  052040  020117
7783  042752  047504  052503  042515
7784  042760  052116  052101  047511
7785  042766     116
7786  042767     200  042200  042105  .ASCIZ  <200><200>"DEDICTED OR MULTIUSER (D OR M)"
7787  042774  041511  042524  020104
7788  043002  051117  046440  046125
7789  043010  044524  051525  051105
```

```
7790  043016  024040  020104  051117
7791  043024  046440  000051
7792
7793  043030  051600  052105  041440  M2:     .ASCIZ  <200>"SET CLOCK CSR= "
7794  043036  047514  045503  041440
7795  043044  051123  020075     000
7796
7797  043051     200  042523  020124  M3:     .ASCIZ  <200>"SET PRESENT BUFFER (2'S COMP) TO "
7798  043056  051120  051505  047105
7799  043064  020124  052502  043106
7800  043072  051105  024040  023462
7801  043100  020123  047503  050115
7802  043106  020051  047524  000040
7803
7804  043114  044200  053517  046440  M4:     .ASCIZ  <200>"HOW MANY JOBS "
7805  043122  047101  020131  047512
7806  043130  051502  000040
7807
7808  043134  045200  041117  021440  M5:     .ASCIZ  <200>"JOB # "
7809  043142  000040
7810  043144  051440  040524  051524  M6:     .ASCIZ  " STATS:"
7811  043152  000072
7812  043154  044600  047457  042040  M7:     .ASCIZ  <200>"I/O DEVICE START MODE WORD= "
7813  043162  053105  041511  020105
7814  043170  052123  051101  020124
7815  043176  047515  042504  053440
7816  043204  051117  036504  000040
7817  043212  041200  043125  042506  M8:     .ASCIZ  <200>"BUFFER SIZE (2'S COMP)= "
7818  043220  020122  044523  042532
7819  043226  024040  023462  020123
7820  043234  047503  050115  036451
7821  043242  000040
7822  043244  052600  042523  020122  M9:     .ASCIZ  <200>"USER STATUS WORD ADDR= "
7823  043252  052123  052101  051525
7824  043260  053440  051117  020104
7825  043266  042101  051104  020075
7826  043274     000
7827  043275     200  047510  020127  M97:    .ASCII  <200>"HOW MANY TIMES TO FILL BUFFERS (NORM =1) "
7828  043302  040515  054516  052040
7829  043310  046511  051505  052040
7830  043316  020117  044506  046114
7831  043324  041040  043125  042506
7832  043332  051522  024040  047516
7833  043340  046522  036440  024461
7834  043346     040
7835  043347     200  047510  020127  M10:    .ASCIZ  <200>"HOW MANY BUFFERS ? "
7836  043354  040515  054516  041040
7837  043362  043125  042506  051522
7838  043370  037440  000040
7839  043374  044600  020123  042504  M11:    .ASCIZ  <200>"IS DEVICE OVERRUN FATAL?"
7840  043402  044526  042503  047440
7841  043410  042526  051122  047125
7842  043416  043040  052101  046101
7843  043424  000077
```

```
7844  043426  044600  020123  052502  M12:    .ASCIZ  <200>"IS BUFFER OVERRUN FATAL?"
7845  043434  043106  051105  047440
7846  043442  042526  051122  047125
7847  043450  043040  052101  046101
7848  043456  000077
7849  043460  044600  051516  043125  M13:    .ASCIZ  <200>"INSUFFIENT CORE FOR BUFFERS-RETYPE THIS JOBS PRAMETERS?"
7850  043466  044506  047105  020124
7851  043474  047503  042522  043040
7852  043502  051117  041040  043125
7853  043510  042506  051522  051055
7854  043516  052105  050131  020105
7855  043524  044124  051511  045040
7856  043532  041117  020123  051120
7857  043540  046501  052105  051105
7858  043546  037523     000
7859  043551     200  042504  040514  M14:    .ASCIZ  <200>"DELAY BEFORE START"
7860  043556  020131  042502  047506
7861  043564  042522  051440  040524
7862  043572  052122     000
7863  043575     200  052516  041115  M15:    .ASCIZ  <200>"NUMBER OF CHNNELS?"
7864  043602  051105  047440  020106
7865  043610  044103  047116  046105
7866  043616  037523     000
7867  043621     200  044514  052123  M16:    .ASCIZ  <200>"LIST RANDOM CHANNELS"
7868  043626  051040  047101  047504
7869  043634  020115  044103  047101
7870  043642  042516  051514     000
7871  043647     200  040523  050115  M17:    .ASCIZ  <200>"SAMPLE#"
7872  043654  042514  000043
7873  043660  046600  045501  020105  M18:    .ASCIZ  <200>"MAKE BUFFER NOW?"
7874  043666  052502  043106  051105
7875  043674  047040  053517  000077
7876  043702  047200  053505  047440  M0:     .ASCIZ  <200>"NEW OR OLD DATA?"
7877  043710  020122  046117  020104
7878  043716  040504  040524  000077
7879  043724  050200  042522  051523  M19:    .ASCIZ  <200>"PRESS CONTINUE WHEN READY"
7880  043732  041440  047117  044524
7881  043740  052516  020105  044127
7882  043746  047105  051040  040505
7883  043754  054504     000
7884  043757     200  044103  047101  M20:    .ASCIZ  <200>"CHAN ADDRESS WORD?"
7885  043764  040440  042104  042522
7886  043772  051523  053440  051117
7887  044000  037504     000
7888  044003     200  046103  041517  M21:    .ASCIZ  <200>"CLOCK RATE?"
7889  044010  020113  040522  042524
7890  044016  000077
7891  044020  051600  040524  052122  M22:    .ASCIZ  <200>"START/EVENT MARK WORD? "
7892  044026  042457  042526  052116
7893  044034  046440  051101  020113
7894  044042  047527  042122  020077
7895  044050     000
7896  044051     200  052123  051101  M23:    .ASCIZ  <200>"START MASK? "
7897  044056  020124  040515  045523
```

```
 7898  044064  020077     000
 7899  044067     200  053105  047105  M24:    .ASCIZ  <200>"EVENT MARK MASK? "
 7900  044074  020124  040515  045522
 7901  044102  046440  051501  037513
 7902  044110  000040
 7903
 7904                              .EVEN
```

```
                                        ;**
                                        ;** REQUEST DECRIPTOR ARRAY FOR THE KW-11K CLOCK
                                        ;** THIS TABLE WILL BE ALTERED BY THIS PROGRAM
                                        ;** AND BY THE LPA-11 OPTION.
044112  000000                  KWT:    .WORD   0               ;CLOCK START MODE WORD.
044114  000000                          .WORD   0               ;CLOCK PRESET REGISTER.
044116  000000                          .WORD   0               ;CLOCK BUFFER REGISTER.


054130  000000                  DEVLST: .WORD   0               ;MODE WORD FOR START.
054132  000012                          .BLKW   10.             ;TEN ADDRESSES ON START.
054156                          DMDT:
                                        .TITLE  DMDT
                                        .SBTTL  DMDT -- DEDICATED MODE DISPATCH TABLE
                                        .IDENT  /LPA.03/

                                ;
                                ; COPYRIGHT    1976, 1977  DIGITAL EQUIPMENT CORP.,   MAYNARD,   MASS.
                                ;
                                ; THIS SOFTWARE IS FURNISHED TO pURCHASER UNDER A LICENSE FOR USE
                                ; ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION
                                ; OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM,
                                ; EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.
                                ;
                                ; THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
                                ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
                                ; EQUIPMENT CORPORATION.
                                ;
                                ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
                                ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                                ;
                                ;
                                ; CHARLES A. SAMUELSON
                                ; FEBRUARY 4, 1977
                                ;
                                ; +
                                ; DMDT -- DEDICATED MODE DISPATCH TABLE
                                ; TABLE FOR LPA11 MICRO PROCESSOR DEDICATED MODE SAMPLING
                                ; -
                                ;
                                ; DEFINED VALUES
                                ;
000022                  DMDSIZ=18.              ;LENGTH OF DMDT BUFFER IN BYTES
000100                  SDT=100                 ;SLAVE DISPATCH TABLE START ADDRESS
000000                  CAINC=0                 ;CHANNEL ADDRESS INCREMENT VALUE
000000                  AD1SRL=0                ;ADC #1 STATUS REGISTER ADDRESS LOW BYTE
000120                  SEX=120                 ;SELECT EXTERNAL CLOCK START AND INTERRUPT ENABLE
000035                  RONPR=35                ;REQUEST OUTPUT NPR IN MICRO-PROCESSOR
000000                  CLR=0                   ;CLEAR AD STATUS REGISTER
000235                  RONPRL=235              ;REQUEST OUTPUT NPR LOW BYTE IN MICRO-PROCESSOR
000002                  AD1DRL=2                ;ADC #1 DATA REGISTER ADDRESS LOW BYTE
000015                  RINPR=15                ;REQUEST INPUT NPR
000040                  SCS=40                  ;SELECT CLOCK OVERFLOW START FOR ADC'S
000040                  AD2SRL=40               ;ADC #2 STATUS REGISTER ADDRESS LOW BYTE
```

```
                    000042                  AD2DRL=42               ;ADC #2 DATA REGISTER ADDRESS LOW BYTE
                    000001                  AD1SRH=1                ;ADC #1 STATUS REGSITER HIGH BYTE ADDRESS
                    000041                  AD2SRH=41               ;ADC #2 STATUS REGISTER HIGH BYTE ADDRESS
                    000020                  SEN=20                  ;SELECT EXTERNAL START, NO INTERRUPT ENABLE
                                            ;
                                            ;
 054156                                     DMDT::
 054156                                     D.OES::                 ;ONE ADC, EXTERNAL TRIGGER, SINGLE CHANNEL
 054156      162     000     120                    .BYTE   DMDSIZ+<3*40>,AD1SRL,SEX,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT
 054161      035     002     015
 054164      000     235     100
                    054201                  .=D.OES+23
 054201                                     D.OEQ::                 ;ONE ADC, EXTERNAL TRIGGER, SEQUENTIAL CHANNEL
 054201      162     000     000                    .BYTE   DMDSIZ+<3*40>,CAINC,AD1SRL,SEX,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT
 054204      120     035     002
 054207      015     000     235
 054212      100
 054213      000                                    .BYTE   CAINC
                    054224                  .=D.OEQ+23
 054224                                     D.OCS::                 ;ONE ADC, CLOCK TRIGGER, SINGLE CHANNEL
 054224      162     000     040                    .BYTE   DMDSIZ+<3*40>,AD1SRL,SCS,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT
 054227      035     002     015
 054232      000     235     100
                    054247                  .=D.OCS+23
 054247                                     D.OCQ::                 ;ONE ADC, CLOCK TRIGGER, SEQUENTIAL CHANNEL
 054247      162     000     000                    .BYTE   DMDSIZ+<3*40>,CAINC,AD1SRL,SCS,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT
 054252      040     035     002
 054255      015     000     235
 054260      100
 054261      000                                    .BYTE   CAINC
                    054272                  .=D.OCQ+23
 054272                                     D.TES::                 ;TWO ADC, EXTERNAL TRIGGER, SINGLE CHANNEL
 054272      162     000     120                    .BYTE   DMDSIZ+<3*40>,AD1SRL,SEX,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT+10
 054275      035     002     015
 054300      000     235     110
 054303      040     120     035                    .BYTE   AD2SRL,SEX,RONPR,AD2DRL,RINPR,CLR,RONPRL,SDT
 054306      042     015     000
 054311      235     100
                    054315                  .=D.TES+23
 054315                                     D.TEQ::                 ;TWO ADC, EXTERNAL TRIGGER, SEQUENTIAL CHANNEL
 054315      162     000     000                    .BYTE   DMDSIZ+<3*40>,CAINC,AD1SRL,SEX,RONPR,AD1DRL,RINPR,CLR,RONPRL
 054320      120     035     002
 054323      015     000     235
 054326      111     000     040                    .BYTE   SDT+11,CAINC,AD2SRL,SEX,RONPR,AD2DRL,RINPR,CLR,RONPRL,SDT
 054331      120     035     042
 054334      015     000     235
 054337      100
                    054340                  .=D.TEQ+23
 054340                                     D.TCS::                 ;TWO ADC, CLOCK TRIGGER, SINGLE CHANNEL
 054340      262     000     040                    .BYTE   DMDSIZ+<5*40>,AD1SRL,SCS,RONPR,AD1DRL,RINPR,CLR,RONPRL
 054343      035     002     015
 054346      000     235
 054350      110     040     040                    .BYTE   SDT+10,AD2SRL,SCS,RONPR,AD2DRL,RINPR,CLR,RONPRL,SDT
 054353      035     042     015
```

```
8013  054356  000  235  100
8014          054363                    .=D.TCS+23
8015  054363                    D.TCQ::                 ;TWO ADC, CLOCK TRIGGER, SEQUENTIAL CHANNEL
8016  054363  262  000  000              .BYTE   DMDSIZ+<5*40>,CAINC,AD1SRL,SCS,RONPR,AD1DRL,RINPR,CLR,RONPRL
8017  054366  040  035  002
8018  054371  015  000  235
8019  054374  111  000  040              .BYTE   SDT+11,CAINC,AD2SRL,SCS,RONPR,AD2DRL,RINPR,CLR,RONPRL,SDT
8020  054377  040  035  042
8021  054402  015  000  235
8022  054405  100
8023          054406                    .=D.TCQ+23
8024                                    ;
8025                                    ; PARALLEL MODE TABLE
8026                                    ;
8027  054406                    D.TESP::                ;TWO ADC, EXTERNAL TRIGGER, SINGLE, PARALLES
8028  054406  022  120  000              .BYTE   DMDSIZ+<0*40>,SEX,AD1SRL,RONPR,AD2SRL,SEN,RONPR,AD1DRL,RINPR
8029  054411  035  040  020
8030  054414  035  002  015
8031  054417  042  015  001              .BYTE   AD2DRL,RINPR,AD1SRH,RONPRL,AD2SRH,RONPRL,SDT+6
8032  054422  235  041  235
8033  054425  106
8034          054431                    .=D.TESP+23
8035  054431                    D.TEQP::                ;TWO ADC, EXTERNAL TRIGGER, SEQUENTIAL, PARALLEL
8036  054431  022  120  000              .BYTE   DMDSIZ+<0*40>,SEX,AD1SRL,RONPR,AD2SRL,SEN,RONPR,CAINC,AD1DRL
8037  054434  035  040  020
8038  054437  035  000  002
8039  054442  015  042  015              .BYTE   RINPR,AD2DRL,RINPR,AD1SRH,RONPRL,AD2SRH,RONPRL,SDT+6
8040  054445  001  235  041
8041  054450  235  106
8042          054454                    .=D.TEQP+23
8043  054454                    D.TCSP::                ;TWO ADC, CLOCK TRIGGER, SINGLE, PARALLEL
8044  054454  222  040  000              .BYTE   DMDSIZ+<4*40>,SCS,AD1SRL,RONPR,AD2SRL,SCS,RONPR,AD1DRL,RINPR
8045  054457  035  040  040
8046  054462  035  002  015
8047  054465  042  015  001              .BYTE   AD2DRL,RINPR,AD1SRH,RONPRL,AD2SRH,RONPRL,SDT+6
8048  054470  235  041  235
8049  054473  106
8050          054477                    .=D.TCSP+23
8051  054477                    D.TCQP::                ;TWO ADC, CLOCK TRIGGER, SEQUENTIAL, PARALLEL
8052  054477  222  040  000              .BYTE   DMDSIZ+<4*40>,SCS,AD1SRL,RONPR,AD2SRL,SCS,RONPR,CAINC,AD1DRL
8053  054502  035  040  040
8054  054505  035  000  002
8055  054510  015  042  015              .BYTE   RINPR,AD2DRL,RINPR,AD1SRH,RONPRL,AD2SRH,RONPRL,SDT+6
8056  054513  001  235  041
8057  054516  235  106
8058  054520  000050            TODOQ:  .BLKW 40.
8059  054640  000000            TODoC:  .WORD 0
8060
8061                                    ;THE FOLLOWING TABLE IS USED BY FLASH TO INDEX THE JOBS BASED
8062                                    ;ON THE INFO THE LPA RETURNS AS ASSIGNED JOB NUMBERS.
8063                                    ;FLASH INDEXES THIS TABLE WITH THE JOB NUMBER TO STORE THE RDA
8064                                    ;ADDR.  WHEN CODE=0,ANY OTHER TIME TABLE IS INDEXED TO GET
8065                                    ;THE RDA ADDR. OF A PARTICULAR JOB.
8066
```

```
8067   054642   000000                    TABLE:   .WORD   0
8068   054644   000000                             .WORD   0
8069   054646   000000                             .WORD   0
8070   054650   000000                             .WORd   0
8071   054652   000000                             .WORD   0
8072   054654   000000                             .WORD   0
8073   054656   000000                             .WORD   0
8074   054660   000000                             .WORD   0
8075   054662   000000                             .WORD   0
8076                                                        ;NOTE THIS AREA FROM    TODOQ: TO DONEC: IS
8077                                                        ;CLEARED EACH TIME FLASH IS ENTERED.
8078   054664   000050                    DONEQ:   .BLKW 40.
8079   055004   000000                    DONeC:   .WORD 0
8080
8081   055006   000000   001000   000004  LIST10: 0,1000,4                    ;AR11 CHO,EXP GR.,TOL,4
8082   055014   000003   001754   000024          3,1754,24          ;       3,EXP 1754,TOL 24
8083   055022   000002   001315   000024          2,1315,24          ;       2,EXP 1315,TOL 24
8084   055030   000003   001754   000024          3,1754,24          ;       3,EXP 1754,TOL 24
8085
8086   055036   000000   004000   000004  LIST12: 0,4000,4                    ;AD11K CHO,EXP GR,TOL 4
8087   055044   000002   004632   000050          2,4632,50          ;        2
8088   055052   000003   006000   000144          3,6000,144
8089   055060   000004   002000   000240          4,2000,240
8090                                               ;**
8091                                               ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #0
8092                                               ;**
8093                                               ;**     THIS TABLE WILL BE ALERED BY THIS PROGRAM
8094                                               ;**     AND BY THE LPA-11 OPTION.
8095                                               ;**
8096
8097   055066   000000                    JOBO: .WORD     0                  ;/MODE INFORMATION.
8098   055070   000000                          .WORd     0                  ;/WORD COUNT
8099   055072   055164                          .WORD     0+JOBOU ;/USW (USER STATUS WORD)
8100   055074      000                          .BYTE     0                  ;/USW (EXTENDED ADDR. BITS)
8101   055075      000                          .BYTE     0                  ;/VBM (VALID BUFFER MASK)
8102   055076   000000                          .WORD     0                  ;/BUFFER ADDRESS #0
8103   055100      000                          .BYTE     0                  ;/EXTENDED ADDRESS BITS
8104   055101      000                          .BYTE     0                  ;/UNUSED
8105   055102   000000                          .WORD     0                  ;/BUFFER ADDRESS #1
8106   055104      000                          .BYTE     0                  ;/EXTENDED ADDRESS BITS
8107   055105      000                          .BYTE     0                  ;/UNUSED
8108   055106   000000                          .WORD     0                  ;/BUFFER ADDRESS #1
8109   055110      000                          .BYTE     0                  ;/EXTENDED ADDRESS BITS
8110   055111      000                          .BYTE     0                  ;/UNUSED
8111   055112   000000                          .WORD     0                  ;/BUFFER ADDRESS #1
8112   055114      000                          .BYTE     0                  ;/EXTENDED ADDRESS BITS
8113   055115      000                          .BYTE     0                  ;/UNUSED
8114   055116   000000                          .WORD     0                  ;/BUFFER ADDRESS #1
8115   055120      000                          .BYTE     0                  ;/EXTENDED ADDRESS BITS
8116   055121      000                          .BYTE     0                  ;/UNUSED
8117   055122   000000                          .WORD     0                  ;/BUFFER ADDRESS #1
8118   055124      000                          .BYTE     0                  ;/EXTENDED ADDRESS BITS
8119   055125      000                          .BYTE     0                  ;/UNUSED
8120   055126   000000                          .WORD     0                  ;/BUFFER ADDRESS #1
```

```
 8121  055130  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8122  055131  000                          .BYTE   0               ;/UNUSED
 8123  055132  000000                       .WORD   0               ;/BUFFER ADDRESS #1
 8124  055134  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8125  055135  000                          .BYTE   0               ;/UNUSED
 8126
 8127  055136  055166                       .WORD   JOBGR   ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
 8128  055140  000                          .BYTE   0               ;/RCL EXTENDED ADR. BITS
 8129  055141  000                          .BYTE   0               ;/UNUSED.
 8130
 8131  055142  000                  JOBOS:  .BYTE   0               ;/CHANNEL START ADDRESS
 8132  055143  000                          .BYTE   0               ;/NUMBER OF CHANNELS
 8133
 8134
 8135  055144  000                          .BYTE   0               ;/CHANNEL INCREMENT
 8136  055145  000                          .BYTE   0               ;/FATAL ERROR MASK
 8137
 8138  055146  000000                       .WORD   0               ;/DELAY
 8139  055150  000000                       .WORD   0               ;/SAMPLE RATE
 8140
 8141  055152  000                          .BYTE   0               ;/STWD
 8142  055153  000                          .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
 8143
 8144  055154  000000                       .WORD   0               ;/ST MSK
 8145  055156  000000                       .WORD   0               ;/EM MSK EVEnT MARK DIGITAL IPUT MASK.
 8146
 8147                                   ;END REG.  TABLE
 8148  055160  000003                       .WORD   03              ;/CONTAINS STOP CODE FOR THIS JOB.
 8149  055162  000000                       .WORD   0               ;/CONTAINS LOOP COUNT FOR JOB.(NO OF TIMES THUR BUFFER.
 8150
 8151  055164  000000                  JOBOU:  .WORD   0               ;/USW USER STATUS WORD
 8152
 8153
 8154  055166  000007                  JOBOR:  .BLKW   7
 8155
 8156                                   ;**
 8157                                   ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #1
 8158                                   ;**
 8159                                   ;**     THIS TABLE WILL BE ALERED BY THIS PROGRAM
 8160                                   ;**     AND BY THE LPA-11 OPTION.
 8161                                   ;**
 8162
 8163  055204  000000                  JOB1:   .WORD   0               ;/MODE INFORMATION.
 8164  055206  000000                          .WORD   0               ;/WORD COUNT
 8165  055210  055302                          .WORD   0+JOB1U ;/USW (USER STATUS WORD)
 8166  055212  000                             .BYTE   0               ;/USW (EXTENDED ADDR. BITS)
 8167  055213  000                             .BYTE   0               ;/VBM (VALID BUFFER MASK)
 8168  055214  000000                          .WORD   0               ;/BUFFER ADDRESS #0
 8169  055216  000                             .BYTE   0               ;/EXTENDED ADDRESS BITS
 8170  055217  000                             .BYTE   0               ;/UNUSED
 8171  055220  000000                          .WORD   0               ;/BUFFER ADDRESS #1
 8172  055222  000                             .BYTE   0               ;/EXTENDED ADDRESS BITS
 8173  055223  000                             .BYTE   0               ;/UNUSED
 8174  055224  000000                          .WORD   0               ;/BUFFER ADDRESS #1
```

```
8175  055226  000                       .BYTE   0               ;/EXTENDED ADDRESS BITS
8176  055227  000                       .BYTE   0               ;/UNUSED
8177  055230  000000                    .WORD   0               ;/BUFFER ADDRESS #1
8178  055232  000                       .BYTE   0               ;/EXTENDED ADDRESS BITS
8179  055233  000                       .BYTE   0               ;/UNUSED
8180  055234  000000                    .WORD   0               ;/BUFFER ADDRESS #1
8181  055236  000                       .BYTE   0               ;/EXTENDED ADDRESS BITS
8182  055237  000                       .BYTE   0               ;/UNUSED
8183  055240  000000                    .WORD   0               ;/BUFFER ADDRESS #1
8184  055242  000                       .BYTE   0               ;/EXTENDED ADDRESS BITS
8185  055243  000                       .BYTE   0               ;/UNUSED
8186  055244  000000                    .WORD   0               ;/BUFFER ADDRESS #1
8187  055246  000                       .BYTE   0               ;/EXTENDED ADDRESS BITS
8188  055247  000                       .BYTE   0               ;/UNUSED
8189  055250  000000                    .WORD   0               ;/BUFFER ADDRESS #1
8190  055252  000                       .BYTE   0               ;/EXTENDED ADDRESS BITS
8191  055253  000                       .BYTE   0               ;/UNUSED
8192
8193  055254  055304                    .WORD   JOB1R   ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
8194  055256  000                       .BYTE   0               ;/RCL EXTENDED ADR. BITS
8195  055257  000                       .BYTE   0               ;/UNUSED.
8196
8197  055260  000             JOB1S:    .BYTE   0               ;/CHANNEL START ADDRESS
8198  055261  000                       .BYTE   0               ;/NUMBER OF CHANNELS
8199
8200
8201  055262  000                       .BYTE   0               ;/CHANNEL INCREMENT
8202  055263  000                       .BYTE   0               ;/FATAL ERROR MASK
8203
8204  055264  000000                    .WORD   0               ;/DELAY
8205  055266  000000                    .WORD   0               ;/SAMPLE RATE
8206
8207  055270  000                       .BYTE   0               ;/STWD
8208  055271  000                       .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
8209
8210  055272  000000                    .WORD   0               ;/ST MSK
8211  055274  000000                    .WORD   0               ;/EM MSK EVENT MARK DIGITAL IPUT MASK.
8212
8213                                    ;END REG. TABLE
8214  055276  000003                    .WORD   03              ;/CONTAINS STOP CODE FOR THIS JOB.
8215  055300  000000                    .WORD   0               ;/CONTAINS LOOP COUNT FOR JOB.(NO OF TIMES THUR BUFFER.
8216
8217  055302  000000          JOB1U:    .WORD   0               ;/USW USER STATUS WORD
8218
8219
8220  055304  000007          JOB1R:    .BLKW   7
8221
8222                                    ;**
8223                                    ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #2
8224                                    ;**
8225                                    ;**     THIS TABLE WILL BE ALERED BY THIS PROGRAM
8226                                    ;**     AND BY THE LPA-11 OPTION.
8227                                    ;**
8228
```

```
8229  055322  000000                JOB2: .WORD   0                ;/MODE INFORMATION.
8230  055324  000000                      .WORD   0                ;/WORD COUNT
8231  055326  055420                      .WORD   0+JOB2U ;/USW (USER STATUS WORD)
8232  055330     000                      .BYTE   0                ;/USW (EXTENDED ADDR. BITS)
8233  055331     000                      .BYTE   0                ;/VBM (VALID BUFFER MASK)
8234  055332  000000                      .WORD   0                ;/BUFFER ADDRESS #0
8235  055334     000                      .BYTE   0                ;/EXTENDED ADDRESS BITS
8236  055335     000                      .BYTE   0                ;/UNUSED
8237  055336  000000                      .WORD   0                ;/BUFFER ADDRESS #1
8238  055340     000                      .BYTE   0                ;/EXTENDED ADDRESS BITS
8239  055341     000                      .BYTE   0                ;/UNUSED
8240  055342  000000                      .WORD   0                ;/BUFFER ADDRESS #1
8241  055344     000                      .BYTE   0                ;/EXTENDED ADDRESS BITS
8242  055345     000                      .BYTE   0                ;/UNUSED
8243  055346  000000                      .WORD   0                ;/BUFFER ADDRESS #1
8244  055350     000                      .BYTE   0                ;/EXTENDED ADDRESS BITS
8245  055351     000                      .BYTE   0                ;/UNUSED
8246  055352  000000                      .WORD   0                ;/BUFFER ADDRESS #1
8247  055354     000                      .BYTE   0                ;/EXTENDED ADDRESS BITS
8248  055355     000                      .BYTE   0                ;/UNUSED
8249  055356  000000                      .WORD   0                ;/BUFFER ADDRESS #1
8250  055360     000                      .BYTE   0                ;/EXTENDED ADDRESS BITS
8251  055361     000                      .BYTE   0                ;/UNUSED
8252  055362  000000                      .WORD   0                ;/BUFFER ADDRESS #1
8253  055364     000                      .BYTE   0                ;/EXTENDED ADDRESS BITS
8254  055365     000                      .BYTE   0                ;/UNUSED
8255  055366  000000                      .WORD   0                ;/BUFFER ADDRESS #1
8256  055370     000                      .BYTE   0                ;/EXTENDED ADDRESS BITS
8257  055371     000                      .BYTE   0                ;/UNUSED
8258
8259  055372  055422                      .WORD   JOB2R   ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
8260  055374     000                      .BYTE   0                ;/RCL EXTENDED ADR. BITS
8261  055375     000                      .BYTE   0                ;/UNUSED.
8262
8263  055376     000                JOB2S: .BYTE   0                ;/CHANNEL START ADDRESS
8264  055377     000                      .BYTE   0                ;/NUMBER OF CHANNELS
8265
8266
8267  055400     000                      .BYTE   0                ;/CHANNEL INCREMENT
8268  055401     000                      .BYTE   0                ;/FATAL ERROR MASK
8269
8270  055402  000000                      .WORD   0                ;/DELAY
8271  055404  000000                      .WORD   0                ;/SAMPLE RATE
8272
8273  055406     000                      .BYTE   0                ;/STWD
8274  055407     000                      .BYTE   0                ;/EMWD EVENT MARK DIGITAL INT WD#
8275
8276  055410  000000                      .WORD   0                ;/ST MSK
8277  055412  000000                      .WORD   0                ;/EM MSK EVENT MARK DIGITAL IPUT MASK.
8278
8279                                ;END REG. TABLE
8280  055414  000003                      .WORD   03               ;/CONTAINS STOP CODE FOR THIS JOB.
8281  055416  000000                      .WORD   0                ;/CONTAINS LOOP COUNT FOR JOB.(NO OF TIMES THUR BUFFER.
8282
```

```
8283  055420  000000                    JOB2U: .WORD    0               ;/USW USER STATUS WORD
8284
8285
8286  055422  000007                    JOB2R:  .BLKW   7
8287
8288                                            ;**
8289                                            ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #3
8290                                            ;**
8291                                            ;**     THIS TABLE WILL BE ALERED BY THIS PROGRAM
8292                                            ;**     AND BY THE LPA-11 OPTION.
8293                                            ;**
8294
8295  055440  000000                    JOB3: .WORD     0               ;/MODE INFORMATION.
8296  055442  000000                          .WORD     0               ;/WORD COUNT
8297  055444  055536                          .WORD     0+JOB3U ;/USW (USER STATUS WORD)
8298  055446     000                          .BYTE     0               ;/USW (EXTENDED ADDR. BITS)
8299  055447     000                          .BYTE     0               ;/VBM (VALID BUFFER MASK)
8300  055450  000000                          .WORD     0               ;/BUFFER ADDRESS #0
8301  055452     000                          .BYTE     0               ;/EXTENDED ADDRESS BITS
8302  055453     000                          .BYTE     0               ;/UNUSED
8303  055454  000000                          .WORD     0               ;/BUFFER ADDRESS #1
8304  055456     000                          .BYTE     0               ;/EXTENDED ADDRESS BITS
8305  055457     000                          .BYTE     0               ;/UNUSED
8306  055460  000000                          .WORD     0               ;/BUFFER ADDRESS #1
8307  055462     000                          .BYTE     0               ;/EXTENDED ADDRESS BITS
8308  055463     000                          .BYTE     0               ;/UNUSED
8309  055464  000000                          .WORD     0               ;/BUFFER ADDRESS #1
8310  055466     000                          .BYTE     0               ;/EXTENDED ADDRESS BITS
8311  055467     000                          .BYTE     0               ;/UNUSED
8312  055470  000000                          .WORD     0               ;/BUFFER ADDRESS #1
8313  055472     000                          .BYTE     0               ;/EXTENDED ADDRESS BITS
8314  055473     000                          .BYTE     0               ;/UNUSED
8315  055474  000000                          .WORD     0               ;/BUFFER ADDRESS #1
8316  055476     000                          .BYTE     0               ;/EXTENDED ADDRESS BITS
8317  055477     000                          .BYTE     0               ;/UNUSED
8318  055500  000000                          .WORD     0               ;/BUFFER ADDRESS #1
8319  055502     000                          .BYTE     0               ;/EXTENDED ADDRESS BITS
8320  055503     000                          .BYTE     0               ;/UNUSED
8321  055504  000000                          .WORD     0               ;/BUFFER ADDRESS #1
8322  055506     000                          .BYTE     0               ;/EXTENDED ADDRESS BITS
8323  055507     000                          .BYTE     0               ;/UNUSED
8324
8325  055510  055540                          .WORD   JOB3R   ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
8326  055512     000                          .BYTE     0               ;/RCL EXTENDED ADR. BITS
8327  055513     000                          .BYTE     0               ;/UNUSED.
8328
8329  055514     000                JOB3S: .BYTE     0               ;/CHANNEL START ADDRESS
8330  055515     000                       .BYTE     0               ;/NUMBER OF CHANNELS
8331
8332
8333  055516     000                          .BYTE     0               ;/CHANNEL INCREMENT
8334  055517     000                          .BYTE     0               ;/FATAL ERROR MASK
8335
8336  055520  000000                          .WORD     0               ;/DELAY
```

```
 8337  055522  000000                          .WORD   0               ;/SAMPLE RATE
 8338
 8339  055524     000                          .BYTE   0               ;/STWD
 8340  055525     000                          .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
 8341
 8342  055526  000000                          .WORD   0               ;/ST MSK
 8343  055530  000000                          .WORD   0               ;/EM MSK EVENT MARK DIGITAL IPUT MASK.
 8344
 8345                                          ;END REG. TABLE
 8346  055532  000003                          .WORD   03              ;/CONTAINS STOP CODE FOR THIS JOB.
 8347  055534  000000                          .WORD   0               ;/CONTAINS LOOP COUNT FOR JOB.(NO OF TIMES THUR BUFFER.
 8348
 8349  055536  000000                  JOB3U:  .WORD   0               ;/USW USER STATUS WORD
 8350
 8351
 8352  055540  000007                  JOB3R:  .BLKW   7
 8353
 8354                                          ;**
 8355                                          ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #4
 8356                                          ;**
 8357                                          ;**     THIS TABLE WILL BE ALERED BY THIS PROGRAM
 8358                                          ;**     AND BY THE LPA-11 OPTION.
 8359                                          ;**
 8360
 8361  055556  000000                  JOB4:   .WORD   0               ;/MODE INFORMATION.
 8362  055560  000000                          .WORD   0               ;/WORD COUNT
 8363  055562  055654                          .WORD   0+JOB4U ;/USW (USER STATUS WORD)
 8364  055564     000                          .BYTE   0               ;/USW (EXTENDED ADDR. BITS)
 8365  055565     000                          .BYTE   0               ;/VBM (VALID BUFFER MASK)
 8366  055566  000000                          .WORD   0               ;/BUFFER ADDRESS #0
 8367  055570     000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8368  055571     000                          .BYTE   0               ;/UNUSED
 8369  055572  000000                          .WORD   0               ;/BUFFER ADDRESS #1
 8370  055574     000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8371  055575     000                          .BYTE   0               ;/UNUSED
 8372  055576  000000                          .WORD   0               ;/BUFFER ADDRESS #1
 8373  055600     000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8374  055601     000                          .BYTE   0               ;/UNUSED
 8375  055602  000000                          .WORD   0               ;/BUFFER ADDRESS #1
 8376  055604     000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8377  055605     000                          .BYTE   0               ;/UNUSED
 8378  055606  000000                          .WORD   0               ;/BUFFER ADDRESS #1
 8379  055610     000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8380  055611     000                          .BYTE   0               ;/UNUSED
 8381  055612  000000                          .WORD   0               ;/BUFFER ADDRESS #1
 8382  055614     000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8383  055615     000                          .BYTE   0               ;/UNUSED
 8384  055616  000000                          .WORD   0               ;/BUFFER ADDRESS #1
 8385  055620     000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8386  055621     000                          .BYTE   0               ;/UNUSED
 8387  055622  000000                          .WORD   0               ;/BUFFER ADDRESS #1
 8388  055624     000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
 8389  055625     000                          .BYTE   0               ;/UNUSED
 8390
```

```
8391  055626  055656                          .WORD   JOB4R   ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
8392  055630     000                          .BYTE   0               ;/RCL EXTENDED ADR. BITS
8393  055631     000                          .BYTE   0               ;/UNUSED.
8394
8395  055632     000                JOB4S:    .BYTE   0               ;/CHANNEL START ADDRESS
8396  055633     000                          .BYTE   0               ;/NUMBER OF CHANNELS
8397
8398
8399  055634     000                          .BYTE   0               ;/CHANNEL INCREMENT
8400  055635     000                          .BYTE   0               ;/FATAL ERROR MASK
8401
8402  055636  000000                          .WORD   0               ;/DELAY
8403  055640  000000                          .WORD   0               ;/SAMPLE RATE
8404
8405  055642     000                          .BYTE   0               ;/STWD
8406  055643     000                          .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
8407
8408  055644  000000                          .WORD   0               ;/ST MSK
8409  055646  000000                          .WORD   0               ;/EM MSK EVENT MARK DIGITAL IPUT MASK.
8410
8411                                          ;END REG. TABLE
8412  055650  000003                          .WORD   03              ;/CONTAINS STOP CODE FOR THIS JOB.
8413  055652  000000                          .WORD   0               ;/CONTAINS LOOP COUNT FOR JOB.(NO OF TIMES THUR BUFFER.
8414
8415  055654  000000                JOB4U:    .WORD   0               ;/USW USER STATUS WORD
8416
8417
8418  055656  000007                JOB4R:    .BLKW   7
8419
8420                                          ;**
8421                                          ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #5
8422                                          ;**
8423                                          ;**     THIS TABLE WILL BE ALERED BY THIS PROGRAM
8424                                          ;**     AND BY THE LPA-11 OPTION.
8425                                          ;**
8426
8427  055674  000000              JOB5:     .WORD   0               ;/MODE INFORMATION.
8428  055676  000000                        .WORD   0               ;/WORD COUNT
8429  055700  055772                        .WORD   0+JOB5U ;/USW (USER STATUS WORD)
8430  055702     000                        .BYTE   0               ;/USW (EXTENDED ADDR. BITS)
8431  055703     000                        .BYTE   0               ;/VBM (VALID BUFFER MASK)
8432  055704  000000                        .WORD   0               ;/BUFFER ADDRESS #0
8433  055706     000                        .BYTE   0               ;/EXTENDED ADDRESS BITS
8434  055707     000                        .BYTE   0               ;/UNUSED
8435  055710  000000                        .WORD   0               ;/BUFFER ADDRESS #1
8436  055712     000                        .BYTE   0               ;/EXTENDED ADDRESS BITS
8437  055713     000                        .BYTE   0               ;/UNUSED
8438  055714  000000                        .WORD   0               ;/BUFFER ADDRESS #1
8439  055716     000                        .BYTE   0               ;/EXTENDED ADDRESS BITS
8440  055717     000                        .BYTE   0               ;/UNUSED
8441  055720  000000                        .WORD   0               ;/BUFFER ADDRESS #1
8442  055722     000                        .BYTE   0               ;/EXTENDED ADDRESS BITS
8443  055723     000                        .BYTE   0               ;/UNUSED
8444  055724  000000                        .WORD   0               ;/BUFFER ADDRESS #1
```

```
8445  055726  000                         .BYTE   0               ;/EXTENDED ADDRESS BITS
8446  055727  000                         .BYTE   0               ;/UNUSED
8447  055730  000000                      .WORD   0               ;/BUFFER ADDRESS #1
8448  055732  000                         .BYTE   0               ;/EXTENDED ADDRESS BITS
8449  055733  000                         .BYTE   0               ;/UNUSED
8450  055734  000000                      .WORD   0               ;/BUFFER ADDRESS #1
8451  055736  000                         .BYTE   0               ;/EXTENDED ADDRESS BITS
8452  055737  000                         .BYTE   0               ;/UNUSED
8453  055740  000000                      .WORD   0               ;/BUFFER ADDRESS #1
8454  055742  000                         .BYTE   0               ;/EXTENDED ADDRESS BITS
8455  055743  000                         .BYTE   0               ;/UNUSED
8456
8457  055744  055774                      .WORD   JOB5R   ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
8458  055746  000                         .BYTE   0               ;/RCL EXTENDED ADR. BITS
8459  055747  000                         .BYTE   0               ;/UNUSED.
8460
8461  055750  000                 JOB5S:  .BYTE   0               ;/CHANNEL START ADDRESS
8462  055751  000                         .BYTE   0               ;/NUMBER OF CHANNELS
8463
8464
8465  055752  000                         .BYTE   0               ;/CHANNEL INCREMENT
8466  055753  000                         .BYTE   0               ;/FATAL ERROR MASK
8467
8468  055754  000000                      .WORD   0               ;/DELAY
8469  055756  000000                      .WORD   0               ;/SAMPLE RATE
8470
8471  055760  000                         .BYTE   0               ;/STWD
8472  055761  000                         .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
8473
8474  055762  000000                      .WORD   0               ;/ST MSK
8475  055764  000000                      .WORD   0               ;/EM MSK EVENT MARK DIGITAL IPUT MASK.
8476
8477                                      ;END REG. TABLE
8478  055766  000003                      .WORD   03              ;/CONTAINS STOP CODE FOR THIS JOB.
8479  055770  000000                      .WORD   0               ;/CONTAINS LOOP COUNT FOR JOB.(NO OF TIMES THUR BUFFER.
8480
8481  055772  000000              JOB5U:  .WORD   0               ;/USW USER STATUS WORD
8482
8483
8484  055774  000007              JOB5R:  .BLKW   7
8485
8486                                      ;**
8487                                      ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #6
8488                                      ;**
8489                                      ;**     THIS TABLE WILL BE ALERED BY THIS PROGRAM
8490                                      ;**     AND BY THE LPA-11 OPTION.
8491                                      ;**
8492
8493  056012  000000              JOB6:   .WORD   0               ;/MODE INFORMATION.
8494  056014  000000                      .WORD   0               ;/WORD COUNT
8495  056016  056110                      .WORD   0+JOB6U ;/USW (USER STATUS WORD)
8496  056020  000                         .BYTE   0               ;/USW (EXTENDED ADDR. BITS)
8497  056021  000                         .BYTE   0               ;/VBM (VALID BUFFER MASK)
8498  056022  000000                      .WORD   0               ;/BUFFER ADDRESS #0
```

```
8499  056024  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
8500  056025  000                          .BYTE   0               ;/UNUSED
8501  056026  000000                       .WORD   0               ;/BUFFER ADDRESS #1
8502  056030  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
8503  056031  000                          .BYTE   0               ;/UNUSED
8504  056032  000000                       .WORD   0               ;/BUFFER ADDRESS #1
8505  056034  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
8506  056035  000                          .BYTE   0               ;/UNUSED
8507  056036  000000                       .WORD   0               ;/BUFFER ADDRESS #1
8508  056040  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
8509  056041  000                          .BYTE   0               ;/UNUSED
8510  056042  000000                       .WORD   0               ;/BUFFER ADDRESS #1
8511  056044  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
8512  056045  000                          .BYTE   0               ;/UNUSED
8513  056046  000000                       .WORD   0               ;/BUFFER ADDRESS #1
8514  056050  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
8515  056051  000                          .BYTE   0               ;/UNUSED
8516  056052  000000                       .WORD   0               ;/BUFFER ADDRESS #1
8517  056054  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
8518  056055  000                          .BYTE   0               ;/UNUSED
8519  056056  000000                       .WORD   0               ;/BUFFER ADDRESS #1
8520  056060  000                          .BYTE   0               ;/EXTENDED ADDRESS BITS
8521  056061  000                          .BYTE   0               ;/UNUSED
8522
8523  056062  056112                       .WORD   JOB6R   ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
8524  056064  000                          .BYTE   0               ;/RCL EXTENDED ADR. BITS
8525  056065  000                          .BYTE   0               ;/UNUSED.
8526
8527  056066  000                  JOB6S:  .BYTE   0               ;/CHANNEL START ADDRESS
8528  056067  000                          .BYTE   0               ;/NUMBER OF CHANNELS
8529
8530
8531  056070  000                          .BYTE   0               ;/CHANNEL INCREMENT
8532  056071  000                          .BYTE   0               ;/FATAL ERROR MASK
8533
8534  056072  000000                       .WORD   0               ;/DELAY
8535  056074  000000                       .WORD   0               ;/SAMPLE RATE
8536
8537  056076  000                          .BYTE   0               ;/STWD
8538  056077  000                          .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
8539
8540  056100  000000                       .WORD   0               ;/ST MSK
8541  056102  000000                       .WORD   0               ;/EM MSK EVENT MARK DIGITAL IPUT MASK.
8542
8543                                       ;END REG.  TABLE
8544  056104  000003                       .WORD   03              ;/CONTAINS STOP CODE FOR THIS JOB.
8545  056106  000000                       .WORD   0               ;/CONTAINS LOOP COUNT FOR JOB.(NO OF TIMES THUR BUFFER.
8546
8547  056110  000000              JOB6U:  .WORD   0               ;/USW USER STATUS WORD
8548
8549
8550  056112  000007              JOB6R:  .BLKW   7
8551
8552                                       ;**
```

```
                                        ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #7
                                        ;**
                                        ;**     THIS TABLE WILL BE ALERED BY THIS PROGRAM
                                        ;**     AND BY THE LPA-11 OPTION.
                                        ;**

056130  000000                  JOB7: .WORD   0                 ;/MODE INFORMATION.
056132  000000                        .WORD   0                 ;/WORD COUNT
056134  056226                        .WORD   0+JOB7U ;/USW (USER STATUS WORD)
056136      000                       .BYTE   0                 ;/USW (EXTENDED ADDR. BITS)
056137      000                       .BYTE   0                 ;/VBM (VALID BUFFER MASK)
056140  000000                        .WORD   0                 ;/BUFFER ADDRESS #0
056142      000                       .BYTE   0                 ;/EXTENDED ADDRESS BITS
056143      000                       .BYTE   0                 ;/UNUSED
056144  000000                        .WORD   0                 ;/BUFFER ADDRESS #1
056146      000                       .BYTE   0                 ;/EXTENDED ADDRESS BITS
056147      000                       .BYTE   0                 ;/UNUSED
056150  000000                        .WORD   0                 ;/BUFFER ADDRESS #1
056152      000                       .BYTE   0                 ;/EXTENDED ADDRESS BITS
056153      000                       .BYTE   0                 ;/UNUSED
056154  000000                        .WORD   0                 ;/BUFFER ADDRESS #1
056156      000                       .BYTE   0                 ;/EXTENDED ADDRESS BITS
056157      000                       .BYTE   0                 ;/UNUSED
056160  000000                        .WORD   0                 ;/BUFFER ADDRESS #1
056162      000                       .BYTE   0                 ;/EXTENDED ADDRESS BITS
056163      000                       .BYTE   0                 ;/UNUSED
056164  000000                        .WORD   0                 ;/BUFFER ADDRESS #1
056166      000                       .BYTE   0                 ;/EXTENDED ADDRESS BITS
056167      000                       .BYTE   0                 ;/UNUSED
056170  000000                        .WORD   0                 ;/BUFFER ADDRESS #1
056172      000                       .BYTE   0                 ;/EXTENDED ADDRESS BITS
056173      000                       .BYTE   0                 ;/UNUSED
056174  000000                        .WORD   0                 ;/BUFFER ADDRESS #1
056176      000                       .BYTE   0                 ;/EXTENDED ADDRESS BITS
056177      000                       .BYTE   0                 ;/UNUSED

056200  056230                        .WORD   JOB7R   ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
056202      000                       .BYTE   0                 ;/RCL EXTENDED ADR. BITS
056203      000                       .BYTE   0                 ;/UNUSED.

056204      000             JOB7S: .BYTE    0                 ;/CHANNEL START ADDRESS
056205      000                       .BYTE   0                 ;/NUMBER OF CHANNELS


056206      000                       .BYTE   0                 ;/CHANNEL INCREMENT
056207      000                       .BYTE   0                 ;/FATAL ERROR MASK

056210  000000                        .WORD   0                 ;/DELAY
056212  000000                        .WORD   0                 ;/SAMPLE RATE

056214      000                       .BYTE   0                 ;/STWD
056215      000                       .BYTE   0                 ;/EMWD EVENT MARK DIGITAL INT WD#

056216  000000                        .WORD   0                 ;/ST MSK
```

```
 8607  056220  000000                          .WORD   0               ;/EM MSK EVENT MARK DIGITAL IPUT MASK.
 8608
 8609                                          ;END REG. TABLE
 8610  056222  000003                          .WORD   03              ;/CONTAINS STOP CODE FOR THIS JOB.
 8611  056224  000000                          .WORD   0               ;/CONTAINS LOOP COUNT FOR JOB.(NO OF TIMES THUR BUFFER.
 8612
 8613  056226  000000                JOB7U:  .WORD   0               ;/USW USER STATUS WORD
 8614
 8615
 8616  056230  000007                JOB7R:  .BLKW   7
 8617
 8618  056246                                FRECOR::
 8619  056246                        BUFFER:
 8620  056246  000401                BUFF0:  .BLKW   257.
 8621  057250  000401                BUFF1:  .BLKW   257.
 8622  060252  000401                BUFF2:  .BLKW   257.
 8623  061254  000401                BUFF3:  .BLKW   257.
 8624  062256  000200                BUFF4:  .BLKW   128.
 8625  062656  000200                BUFF5:  .BLKW   128.
 8626
 8627
 8628
 8629                                          ;       W A R N I N G    !!!!!!!!!!!!
 8630                                          ;
 8631                                          ;       THE FREE CORE AREA BETWEEN HERE AND PROGRAM
 8632                                          ;       "DRLPX0" MAY BE USED BY CERTAIN PARTS OF THIS
 8633                                          ;       PROGRAM.  YOU MAY NOT PLACE ANY CODE HERE.
 8634                                          ;       LAST AVAILIBLE SPACE TO PUT CODE IS BEFORE
 8635                                          ;       LOCATION "FRECOR"   OR AFTER ADDRESS "65500".
 8636                                          ;
 8637                                          ;       IF YOU HAVE TO VIOLATE THIS,DON'T ALLOW ROUTINE
 8638                                          ;       "MAKEI" TO BE RUN!
 8639                                          ;
 8640          000001                          .END
```

```
AACSR   001624            600#    822*   2987   2998   3007   3009   3020   3022   3032   3034   4213   4221   4229
                         4236
AA11K   001574            575#    821    4445
ABASE = 170460            185#    269     310    464    467    469    471    473    475    477    479
ACDW1 = 000000            269     312
ACDW2 = 000000            269
ACPUOP= 000000            269     284
ADAC    001616            594#    768*    769*
ADAC2   001622            597#    834*    835*
ADBUF1  001616            593#   1820    1867
ADBUF2  001622            596#   1982    2029
ADDW0 = 000000            269
ADDW1 = 000000            269
ADDW10= 000000            269
ADDW11= 000000            269
ADDW12= 000000            269
ADDW13= 000000            269
ADDW14= 000000            269
ADDW15= 000000            269
ADDW2 = 000000            269
ADDW3 = 000000            269
ADDW4 = 000000            269
ADDW5 = 000000            269
ADDW6 = 000000            269
ADDW7 = 000000            269
ADDW8 = 000000            269
ADDW9 = 000000            269
ADEVCT= 000000            269     275
ADEVM = 000000            269     311
ADWD    034472           6571#
AD1DRL= 000002           7955#   7967   7972   7979   7984   7991   7999   8008   8016   8028   8036   8044   8052
AD1SRH= 000001           7960#   8031   8039   8047   8055
AD1SRL= 000000           7950#   7967   7972   7979   7984   7991   7999   8008   8016   8028   8036   8044   8052
AD11K   001566            572#    766    4425
AD11K2  001576            576#    832    4439
AD2DRL= 000042           7959#   7994   8002   8011   8019   8031   8039   8047   8055
AD2SRH= 000041           7961#   8031   8039   8047   8055
AD2SRL= 000040           7958#   7994   8002   8011   8019   8028   8036   8044   8052
AENV  = 000000            269     280
AENVM = 000000            269     281
AFATAL= 000000            269     272
ALPADL  001770            682#   6874*   6881   6900   7759
ALPCI   001762            679#   6885*   7759
ALPCO   001764            680#   6875*   6876   6882   6886   7759
ALPSO   001766            681#   6886*   6887*  6888   7759
AMADR1= 000000            269     297
AMADR2= 000000            269     301
AMADR3= 000000            269     304
AMADR4= 000000            269     307
AMAMS1= 000000            269     291
AMAMS2= 000000            269     299
AMAMS3= 000000            269     302
AMAMS4= 000000            269     305
AMSGAD= 000000            269     277
```

```
AMSGLG= 000000            269     278
AMSGTY= 000000            269     271
AMTYP1= 000000            269     292
AMTYP2= 000000            269     300
AMTYP3= 000000            269     303
AMTYP4= 000000            269     306
APASS = 000000            269     274
APRIOR= 000000            269
APTCSU= 000040           6271    6414#
APTENV= 000001           5966    6264    6370    6412#
APTSIZ= 000200            743    6411#
APTSPO= 000100           6266    6372    6413#
ARADB   001712            652#    839*    846*    3253
ARADS   001710            651#    838*   3190    3205    3216    3218    3229    3231    3241    3243    3256    3260
ARCB    001716            654#    841*    848*    3365
ARCC    001726            658#    845*    852*
ARCS    001714            653#    840*    847*    3321    3330    3340    3342    3352    3354    3367    3369    3380    3382
ARDS    001720            655#    842*    849*    3418    3429    3438    3440    3451    3453    3463    3465    3560    3568
                         3576    3583
ARXB    001722            656#    843*    850*    3474    3476    3496    3498    3517    3519    3558    3574
ARYB    001724            657#    844*    851*    3485    3487    3506    3508    3528    3530    3566    3581
AR11K   001610            581#     837    4407    4429    4449
ASWREG= 000000            269     282
ATESTN= 000000            269     273
AUNIT = 000000            269     276
AUSWR = 000000            269     283
AVECT1= 000300            186#    269     308     481     482     491     493
AVECT2= 000000            269     309
AVERR   034146           5361    5449    6503#   6763
AVERRL  034204           6509#   6540
AVERRN  034370           6537#   6549    6558
AVEXP   034514           5359*   5447*   5708*   5717*   6527    6580#   7756
AVTKN   034516           6511*   6522*   6554*   6581#
BITO  = 000001            170#    1749    1820    1835    1982    1997    2168    2195    2207    3253    3361    3665    3680
                         3810    4423    4497    4690    4698    4703    4706    5311    5316    5347    5390    5395    5435
                         5494    6484    6806    6863
BIT00 = 000001            160#     170
BIT01 = 000002            159#     169
BIT02 = 000004            158#     168
BIT03 = 000010            157#     167
BIT04 = 000020            156#     166
BIT05 = 000040            155#     165
BIT06 = 000100            154#     164
BIT07 = 000200            153#     163
BIT08 = 000400            152#     162    6065
BIT09 = 001000            151#     161    5977    6073
BIT1  = 000002            169#    1452    1657    2083    2168    2177    2195    2207    3361    3370    3810    3819    4310
                         4401    4688    4703    4762    4780    4875    4893    4988    5006    5101    5119    5214    5232
                         5540    5686    7040
BIT10 = 002000            150#    1792    1954    3180    3225    3311    3411    3547    3561    3569    3637    4251    4405
                         4427    4447    4688    5311    5347    5353    5390    5435    5441    5494    5532    5954
BIT11 = 004000            149#    1443    1779    1941    2133    3212    3624    4270    5686    6080    6980    7440
BIT12 = 010000            148#    1792    1954    3549    3592    3637    4160    4168    4214    4222    4431    4690    5311
                         5390    5494
```

H01


```
BIT13 = 020000   147#  1779  1941  2110  3212  3624  3760  3787  4410  4688  5961
BIT14 = 040000   146#   972  1437  2089  2281  2427  2573  2719  2865  3317  3766  3889  4003
                4146  4148  4265  4452  5432  5532  6051  6690  6717  6972  7427
BIT15 = 100000   145#  1443  1772  1785  1798  1845  1934  1947  1960  2007  3617  3630  3643
                3690  3858  3911  4270  4463  5432  5514  6672  6676  6690  6717  6980  7440
BIT2  = 000004   168#  2110  2133  2249  2303  3337  3637  3787  4459  4757  4823  5514
BIT3  = 000010   167#   977  2089  2145  2960  3317  3766  4200  4202  4443  4701  4762  4780
                4875  4893  4988  5006  5101  5119  5214  5232  5327  5406  5532  5540
BIT4  = 000020   166#  1792  1911  1954  2133  3225  3317  3337  3354  3370  3382  4310  4437
                4763  4781  4876  4894  4989  5007  5102  5120  5215  5233  5347  5435  5551
                5669  7241
BIT5  = 000040   165#  1448  1580  2145  2177  2395  2449  4470  4870  4936  5570  5686  6984
                7197  7214
BIT6  = 000100   164#  1779  1941  2089  2133  2281  2427  2573  2719  2865  3212  3317  3624
                3766  3889  4310  4650  4703  6834  6835  6861
BIT7  = 000200   163#   977  1772  1785  1798  1810  1823  1845  1859  1934  1947  1960  1972
                1985  2007  2021  2177  2207  2313  2459  2541  2595  2605  2751  2897  2998
                3010  3022  3034  3205  3218  3231  3243  3256  3370  3429  3441  3453  3465
                3617  3630  3643  3655  3668  3690  3703  3819  3921  4021  4033  4045  4057
                4274  4279  4284  4310  4475  4500  4709  4781  4894  4983  5007  5049  5120
                5233  5337  5416  5502  5540  5558  5590  6724  6916
BIT8  = 000400   162#  1792  1954  2110  2687  2741  3225  3337  3637  3787  4480  4703  4763
                4781  4876  4894  4989  5007  5096  5102  5120  5162  5215  5233  5498  5610
                5686  6701
BIT9  = 001000   161#  1631  1779  1941  2089  2833  2887  3212  3624  3766  4261  4485  5209
                5275  5324  5540  5630  6701
BPTVEC= 000014   177#
BUFFER 056246   6504  7598  8619#
BUFF0  056246   4768  4807  4827  4881  4920  4940  4994  5033  5053  5107  5146  5166  5220
                5259  5279  5331  5410  5503  5559  8620#
BUFF1  057250   4769  4882  4995  5108  5221  5332  5411  8621#
BUFF2  060252   4786  4788  4801  4826  4899  4901  4914  4939  5012  5014  5027  5052  5125
                5127  5140  5165  5238  5240  5253  5278  5333  5412  5522  5578  5598  5618
                5638  8622#
BUFF3  061254   4787  4900  5013  5126  5239  5545  8623#
BUFF4  062256   8624#
BUFF5  062656   8625#
CAINC = 000000  7949#  7972  7976  7984  7988  7999  8002  8016  8019  8036  8052
CHAN   034466   5711*  5720*  6503*  6507  6538*  6539  6569#  7756
CHANF  034476   5358*  5446*  5707*  5716*  6539  6573#  6760*
CHANFR 034502   6575#
CHANNO 034504   6576#
CHANS  034474   5357*  5445*  5706*  5711  5715*  5720  6503  6572#  6759*
CHANSR 034500   6574#
CHANU  034454   5344*  5423*  5698*  6505  6564#  6755*  6758
CHAN1  034464   6507*  6508*  6510  6537*  6568#
CHAN7  034456   6565#
CKSWR = 104407  5950  5976  6049  6050  7583#
CLKINT 040616   889  890  7420#
CLKTMP 002006   697  698#
CLR   = 000000  7953#  7967  7972  7979  7984  7991  7994  7999  8002  8008  8011  8016  8019
CR    = 000015   85#  6310  6320
CRLF  = 000200   86#  909  6281  6320
DAC0   001626   601#  823*  827*  3043  3045  3087  3089  3132  3134  4211  4227
```

```
DAC1    001630          602#    824#    828#   3054   3056   3098   3100   3143   3145   4219   4234
DAC2    001632          603#    825#    829#   3065   3067   3109   3111   3154   3156
DAC3    001634          604#    826#    830#   3076   3078   3120   3122   3165   3167
DDISP = 177570           92#    250     731
DEVLST  054130         4354    4390    4400   4493*  4496   4569   4570*  4571*  4597   4631   4632*  4633*  4648
                       4682    4683*   4684*  6799   6800*  6801*  7914#
DFO     042606          337     344     351    358    365    372    379    386    393    400    414    421    428
                        435     442     448   7762#
DH1     042240          335    7709#
DH17    042321          433    7721#
DH2     042261          342     363     370    377    384    391    398    412    419    426   7713#
DH20    042361          440    7728#
DH21    042427          446    7735#
DH3     042312          349     356    7719#
DISPLA  001142          250#    731*    739*  5953*  6095*
DISPRE  000174           62#    739
DMAsT   050124         4259    7435    7914#
DMDSIZ= 000022         7947#   7967    7972   7979   7984   7991   7999   8008   8016   8028   8036   8044   8052
DMDT    054156         7916#   7965#
DONEC   055004         6810    8079#
DONeQ   054664         8078#
DRCR1   001636          608#    787*   2257   2270   2284   2286   2295   2297   2313   2334   2371
DRCR2   001644          613#    794*   2403   2416   2430   2432   2441   2443   2459   2480   2517
DRCR3   001652          618#    801*   2549   2562   2576   2578   2587   2589   2605   2626   2663
DRCR4   001660          623#    808*   2695   2708   2722   2724   2733   2735   2751   2772   2809
DRCR5   001666          628#    815*   2841   2854   2868   2870   2879   2881   2897   2918   2955
DRIA1   001642          610#    789*    791*  2281   2325   2343   2348   2356   2362
DRIA2   001650          615#    796*    798*  2427   2471   2489   2494   2502   2508
DRIA3   001656          620#    803*    805*  2573   2617   2635   2640   2648   2654
DRIA4   001664          625#    810*    812*  2719   2763   2781   2786   2794   2800
DRIA5   001672          630#    817*    819*  2865   2909   2927   2932   2940   2946
DRLPX0= ******  G      1434#   1435    6729
DRLPX1= ******  G       970#    971
DRLPX2= ******  G        14#   6970
DROA1   001640          609#    788*    790*  2279   2311   2346   2360
DROA2   001646          614#    795*    797*  2425   2457   2492   2506
DROA3   001654          619#    802*    804*  2571   2603   2638   2652
DROA4   001662          624#    809*    811*  2717   2749   2784   2798
DROA5   001670          629#    816*    818*  2863   2895   2930   2944
DR11K1  001572          574#    786    4461
DR11K2  001600          577#    793    4472
DR11K3  001602          578#    800    4477
DR11K4  001604          579#    807    4482
DR11K5  001606          580#    814    4487
DSWR  = 177570           91#    249     730
DT1     042516          336    7747#
DT17    042540          434    7753#
DT2     042524          343     364     371    378    385    392    399    413    420    427   7749#
DT20    042552          441    7756#
DT21    042566          447    7759#
DT3     042534          350     357    7752#
D.OCQ   054247  G      7983#   7989
D.OCS   054224  G      7978#   7982
D.OeQ   054201  G      7971#   7977
```

```
D.QES   054156  G       7966#   7970
D.TCQ   054363  G       8015#   8023
D.TCQP  054477  G       8051#
D.TCS   054340  G       8007#   8014
D.TCSP  054454  G       8043#   8050
D.TEQ   054315  G       7998#   8006
D.TEQP  054431  G       8035#   8042
D.TES   054272  G       7990#   7997
D.TESP  054406  G       8027#   8034
EMTVEC= 000030           180#    715*    716*
EM1     041504           334    7630#
EM10    041747           383    7666#
EM11    041776           390    7671#
EM12    042025           397    7676#
EM14    042050           411    7681#
EM15    042074           418    7686#
EM16    042116           425    7690#
EM17    042142           432     445    7695#
EM2     041527           341    7635#
EM20    042173           439    7701#
EM3     041560           348    7641#
EM4     041620           355     362    7648#
EM6     041650           369    7653#
EM7     041705           376    7659#
ERAV1   034414          6530    6534    6545#
ERAV2   034414          6536    6546#
ERCNT   001776           685#    911*    5775    5984*
ERRVEC= 000004           173#    728     729*    740*    877*    881*    887*    927     928*    929     930*    950*    951*    -
                        6056    6057*   6059*   6062*
ERTOUT  035664          6643    6730    6774#
ETDF  = 030262          5668    5670    5671    5677#
ETEST   030576          4420    5725#
FLASH   036026          4821    4934    5047    5160    5273    5346    5434    5648    5702    6740    6796#
FRECOR  056246  G       6627*   6628    6728*   6729    8618#
FUDGE   002014           701#    874*   4819*   4932*   5045*   5158*   5271*   6827    6828*
GAIN    034470          6570#
GNS   = ****** U          61     908     5755    5761    5766    5773    6652    6711    6745    6777    6781    6963    7017
                        7021    7025    7253    7463    7487    7575    7576    7577    7578    7579    7581    7583    7584
                        7585    7586    7593    7602    7611    7616    7622
GTSWR = 104406           903    7581#
HT    = 000011            83#    6279    6320
IEX     036406          6852    6865#
INPSRV  036334          6832    6851#
IOTVEC= 000020           178#    713*    714*
JOBO    055066          4533*   4535*   4536    4751    4813*   4814*   4864    4926*   4927*   4977    5039*   5040*   5090
                        5152*   5153*   5203    5265*   5266*   5317    5322    5351    5396    5401    5439    5487    5496
                        5685    6625    6704    6735    6753    6767    6814    6816    8097#
JOBOR   055166          5422    5424    8127    8154#
JOBOS   055142          8131#
JOBOU   055164          4535    5329    5408    5500    5688    8099    8151#
JOB1    055204          4779    4892    5005    5118    5231    5488    6704    6734    6767    6815    8163#
JOB1R   055304          8193    8220#
JOB1S   055260          8197#
JOB1U   055302          4783    4896    5009    5122    5235    5519    8165    8217#
```

```
JOB2    055322  4761   4874   4987   5100   5213   8229#
JOB2R   055422  8259   8286#
JOB2S   055376  8263#
JOB2U   055420  4765   4878   4991   5104   5217   5542   8231   8283#
JOB3    055440  4815*  4816*  4817*  4928*  4929*  4930*  5041*  5042*  5043*  5154*  5155*  5156*  5267*
                5268*  5269*  8295#
JOB3R   055540  8325   8352#
JOB3S   055514  8329#
JOB3U   055536  5556   8297   8349#
JOB4    055556  4753   4366   4979   5092   5205   8361#
JOB4R   055656  8391   8418#
JOB4S   055632  8395#
JOB4U   055654  5575   8363   8415#
JOB5    055674  8427#
JOB5R   055774  8457   8484#
JOB5S   055750  8461#
JOB5U   055772  5595   8429   8481#
JOB6    056012  8493#
JOB6R   056112  8523   8550#
JOB6S   056066  8527#
JOB6U   056110  5615   8495   8547#
JOB7    056130  8559#
JOB7R   056230  5319   5398   5491   8589   8616#
JOB7S   056204  8593#
JOB7U   056226  5635   8561   8613#
KMAD0   001466   464#   749    932*   934*   972*   1001   1026*  1056   1077*  1131   1177*  1179   1248*
                1251   1318*  1322   1386*  1399*  1437*  1443*  1448   1452   1565   1580   1637   1657
                4265*  4270*  6972*  6980*  6984   7040   7100*  7102*  7104*  7105*  7112*  7117*  7197
                7214   7241   7427*  7440*  7747
KMAD1   001470   467#   977*
KMAD2   001472   469#   935*   1008   1062   1139   1187   1468*  1470   1552*  1574*  1590*  1649*  1666*
                6462   6989*  7002   7042*  7056   7070*  7083   7151*  7156*  7160*  7182*  7199*  7216*
                7235   7445*
KMAD3   001474   471#   7537*  7540*
KMAD4   001476   473#   936*   1014   1067   1473   1551*  1572*  1594   1647*  1670   7004   7009   7033*
                7035*  7037*  7058   7101*  7106*  7109*  7118*  7150*  7154*  7159*  7181*  7201   7218
                7446*
KMAD5   001500   475#
KMAD6   001502   477#   937*   1020   1072   7103*  7113   7530*  7538*
KMAD7   001504   479#   752    7531*  7539*
KMCSR   001466   454#
KWADR0  001674   636#   772*   2093   2103   2113   2115   2125   2127   2174   2176   2187   2189
KWADR1  001676   638#   773*   778*   2172
KWADR2  001700   640#   774*   779*
KWADR4  001702   642#   775*   780*   2136   2138   2149   2151   2160   2162   2201   2207   2217   2219
KWADR5  001704   644#   776*   781*   2199
KWADR6  001706   646#   777*   782*
KWT     044112  4310*  4313   4698*  4701*  4703*  4704*  4705   5314*  5315   5393*  5394   5683*  5684*
                6483   6611*  6614*  6805   7909#
KWVECL  002010   699#   889*
KWVECP  002012   700#   890*
KW11K   001570   573#   771    4403
KW11L   002000   694#   878    879
KW11P   002002   695#   882    883
```

```
LF     = 000012            84#   6314   6320
LIST10   055006          5352    5440   8081#
LIST12   055036          5355    5443   5705   5714   8086#
LOOPJ    035516          6600    6740#
LOPC   = 000074           691#
LPADBR   001732           664#    856*   865*   3665   3711
LPADH    001500           474#
LPADL    001476           472#   4316*  4356*  4496*  4536*  4569*  4597*  4631*  4648*  4682*  4705*  5315*  5394*
                         6483*  6799*  6805*  6857*  6874
LPADSR   001730           663#    855*  3601   3617   3628   3630   3641   3643   3653   3655   3668   3672   3687
                         3690   3701   3703
LPCI     001466           463#   4274   4280   4317*  4358*  4497*  4500   4537*  4573*  4576   4579   4598*  4635*
                         4637*  4642   4643*  4649*  4664*  4686*  4706*  4709   4712   5316*  5395*  6484*  6803*
                         6806*  6834*  6845   6861*  6863*  6885
LPCKBR   001736           666#    858*   867*  3814
LPCKSR   001734           665#    857*   866*  3770   3780   3790   3792   3802   3804   3816   3818   3829   3831
LPCO     001472           468#   4284   4286   4319   4321   4336*  4361   4366   4509   4540   4546   4588   4601
                         4604   4650*  4656   4657*  4665*  4717   6835*  6846   6875   6916*
LPDASR   001746           670#    862*   871*  4010   4021   4030   4032   4043   4045   4055   4057   4159   4167
                         4175   4182
LPDAXR   001750           671#    863*   872*  4066   4068   4089   4091   4113   4115   4157   4173
LPDAYR   001752           672#    864*   873*  4077   4079   4100   4102   4124   4126   4165   4180
LPIOIR   001742           668#    860*   869*  3888   3933   3956   3965   3970
LPIOOR   001744           669#    861*   870*  3886   3919   3951   3954   3968
LPIOSR   001740           667#    859*   868*  3865   3878   3892   3894   3903   3905   3921   3942   3979
LPMR     001470           466#
LPMS1    001502           476#   4519
LPMS2    001504           478#
LPSO     001474           470#   4291   4294   4328   4338   4342   4372   4512   4552   4593   4610   4720   6486
LPS11    001612           582#    854   4412   4433   4454   4465
L1PED    034744          6629#   6737
MAKEI    034546            70    6594#  6751   6783
MMAST    044120          4255    7432   7437   7914#
MYTEMP   001754           676#   1820   1867   1982   2029   3253   3554*  3560   3568   3576   3583   3665   3711
                         4153*  4159   4167   4175   4182   4207*  4213   4221   4229   4236   6623*  6754
M0       043702          6594   7876#
M1       042615          6601   7767#
M10      043347          6639   7835#
M11      043374          7839#
M12      043426          6666   7844#
M13      043460          7849#
M14      043551          6678   7859#
M15      043575          6684   7863#
M16      043621          7867#
M17      043647          7871#
M18      043660          6726   7873#
M19      043724          6727   7879#
M2       043030          6609   7793#
M2SP     042612          7764#
M20      043757          6681   7884#
M21      044003          6687   7888#
M22      044020          6692   7891#
M23      044051          6695   7896#
M24      044067          6698   7899#
```

```
M3        043051            6612    7797#
M4        043114            6618    7804#
M5        043134            6629    7808#
M6        043144            6632    7810#
M7        043154            6633    7812#
M8        043212            6636    7817#
M9        043244            6663    7822#
M97       043275            6721    7827#
NA07      034460            6566#
NA17      034462            6567#
NER       036544            6883    6898#
NOCLK     002004             697#    885
OUPEX     036632            6891    6896    6914    6916#
OUPSRV    036410            6830    6867#
PC      =%000007             104#   1560*   1575*   1592*   1633*   1650*   1668*   4248*   4267*   4276*   4318*   4495*   4499*
                            4567*   4574*   4600*   4627*   4636*   4639*   4653*   4681*   4687*   4708*   4821*   4934*   5047*
                            5160*   5273*   5346*   5361*   5434*   5449*   5648*   5702*   5712*   5721*   5743*   5746*   5780*
                            5785    5963*   5969*   6024*   6153*   6269*   6288*   6295*   6302*   6316*   6318*   6389*   6406*
                            6470*   6481*   6485*   6487*   6541*   6740*   6763*   6798*   6804*   6842*   6849*   6850*   7034*
                            7036*   7038*   7085*   7148*   7152*   7157*   7161*   7180*   7183*   7200*   7217*   7248*   7386*
                            7413*   7416*   7429*   7442*   7447*
PIRQ    = 177772             90#
PIRQVE= 000240              184#
PR0     = 000000              107#
PR1     = 000040              108#
PR2     = 000100              109#
PR3     = 000140              110#
PR4     = 000200              111#
PR5     = 000240              112#
PR6     = 000300              113#
PR7     = 000340              114#
PS      = 177776               87#     88    1385*   1398*   4651*   6839*   7408*
PSW     = 177776               88#
PWRVEC= 000024              179#    719*    720*   6419*   6420*   6429*   6435*   6447*   6448*
RBUFR     041324               73    7590#
RDCHR   = 104410            6212    7584#
RDLIN   = 104411            6335    6595    6602    6667    6747    7585#
RDOCT   = 104412            6610    6613    6619    6634    6637    6640    6679    6682    6685    6688    6693    6696    6699
                            6713    6722    7470    7489    7586#   7595    7604
RD1       040152            7184*   7221#
REPMAN    034530            6586#
REPOR     034540            5350*   5438*   5704*   6547    6590#   6762*
RESVEC= 000010              174#
RHIGH     034522            6514*   6515    6517*   6529    6556*   6583#   7756
RINPR   = 000015            7956#   7967    7972    7979    7984    7991    7994    7999    8002    8008    8011    8016    8019
                            8028    8031    8039    8044    8047    8055
RLOW      034520            6513*   6518    6520*   6533    6552*   6582#   7756
RONPR   = 000035            7952#   7967    7972    7979    7984    7991    7994    7999    8002    8008    8011    8016    8019
                            8028    8036    8044    8052
RONPRL= 000235              7954#   7967    7972    7979    7984    7991    7994    7999    8002    8008    8011    8016    8019
                            8031    8039    8047    8055
RSTART    003500             912#   5786
RTCCSR    040624             879*    883*    885*   7404    7407*   7411*   7422#
RTEMP     034524            6512*   6523*   6584#
```

```
                    2798*  2800*  2809*  2841*  2854*  2863*  2865*  2868*  2870*  2879*  2881*  2895*  2897*
                    2909*  2918*  2927*  2930*  2932*  2940*  2944*  2946*  2955*  2987*  2998*  3007*  3009*
                    3020*  3022*  3032*  3034*  3043*  3045*  3054*  3056*  3065*  3067*  3076*  3078*  3087*
                    3089*  3098*  3100*  3109*  3111*  3120*  3122*  3132*  3134*  3143*  3145*  3154*  3156*
                    3165*  3167*  3190*  3205*  3216*  3218*  3229*  3231*  3241*  3243*  3253*  3256*  3260*
                    3321*  3330*  3340*  3342*  3352*  3354*  3365*  3367*  3369*  3380*  3382*  3418*  3429*
                    3438*  3440*  3451*  3453*  3463*  3465*  3474*  3476*  3485*  3487*  3496*  3498*  3506*
                    3508*  3517*  3519*  3528*  3530*  3558*  3560*  3566*  3568*  3574*  3576*  3581*  3583*
                    3601*  3617*  3628*  3630*  3641*  3643*  3653*  3655*  3665*  3668*  3672*  3687*  3690*
                    3701*  3703*  3711*  3770*  3780*  3790*  3792*  3802*  3804*  3814*  3816*  3818*  3829*
                    3831*  3865*  3878*  3886*  3888*  3892*  3894*  3903*  3905*  3919*  3921*  3933*  3942*
                    3951*  3954*  3956*  3965*  3968*  3970*  3979*  4010*  4021*  4030*  4032*  4043*  4045*
                    4055*  4057*  4066*  4068*  4077*  4079*  4089*  4091*  4100*  4102*  4113*  4115*  4124*
                    4126*  4157*  4159*  4165*  4167*  4173*  4175*  4180*  4182*  4211*  4213*  4219*  4221*
                    4227*  4229*  4234*  4236*  4254*  4258*  5826   5832*  5834*  5836*  5837*  5838*  5839
                    5857*  5885   5887*  5889*  5896*  5900*  5915   5921*  6426   6441*  6969*  6991*  7045*
                    7065*  7072*  7123*  7163*  7186*  7203*  7220*  7287*  7292*  7295*  7329*  7336*  7340*
                    7352   7354   7359*  7360*  7377*  7436*  7475*  7492*  7527*  7534*  7541*
R6      =%000006     101#   707*   708*   709
R7      =%000007     102#
SAMCNT  034506      6509*  6525*  6577#
SAMOFF  034510      6505*  6506*  6524   6578#
SCS     = 000040    7957#  7979   7984   8008   8011   8016   8019   8044   8052
SDELAY  040542      4267   4276   4318   4499   4574   4600   4636   4639   4653   4687   4708   6485   6804
                    6842   7403#  7429   7442
SDT     = 000100    7948#  7967   7972   7979   7984   7991   7994   8002   8011   8019   8031   8039   8047
                    8055
SEN     = 000020    7962#  8028   8036
SEX     = 000120    7951#  7967   7972   7991   7994   7999   8002   8028   8036
SP      =%000006     103#   711*   728*   736*   740    886*   927*   929*   948*   950    951   1394*  1408*
                    4666*  5679*  5768*  5775*  5816*  5817   5818   5819*  5824*  5825*  5826*  5832   5857
                    5858   5859   5860*  5861*  5881*  5882*  5883*  5884*  5885*  5886*  5887   5890*  5903
                    5905*  5907   5917   5919   5921   5922   5923   5924   5925   5927*  5928*  5958   5979*
                    5982*  5996*  6001*  6022   6026*  6056*  6059   6061   6062   6091   6092   6096*  6113*
                    6114*  6115   6122*  6125*  6126*  6130*  6131*  6135   6138*  6142   6144   6146   6147*
                    6154   6156   6158*  6159   6161*  6162*  6163*  6164*  6165*  6180*  6181*  6184*  6185*
                    6186   6190*  6191*  6192   6195   6197   6199*  6208*  6213   6224   6225*  6226*  6227*
                    6262*  6263   6273*  6275   6276   6277*  6279   6281   6283   6289   6291*  6293*  6301*
                    6305   6309   6310   6314   6330*  6331*  6332*  6333*  6334*  6336   6339*  6347*  6348
                    6350   6351*  6353   6354   6355   6366*  6367*  6374   6375*  6386   6387*  6388*  6398
                    6399*  6404   6405   6421*  6422*  6423*  6424*  6425*  6426*  6427*  6428   6436*  6440
                    6441   6442   6443   6444   6445   6446   6597   6603   6611   6614   6620   6630*  6635
                    6638   6641   6654*  6664*  6668   6680   6683   6686   6689   6694   6697   6700   6714
                    6723   6748   6853*  6864   6867*  6868*  6869*  6870*  6871*  6918   6919   6921   6922
                    6923   6944*  6966   6967   6976*  7064   7097*  7098*  7121   7122   7145*  7162   7177*
                    7219   7233*  7247   7272*  7273*  7290   7291   7314*  7315*  7334   7335   7471   7479*
                    7490   7552*  7553   7563*  7564*  7596   7605   7618*  7624*
SPEP    034160      5712   5721   6505#
SRESET  040626      4495   4567   4627   4681   6481   6798   6849   7427#
SR1     001562       529#  1749   1911   2083   2249   2395   2541   2687   2833   2980   3180   3311   3411
                    3547   3592   3760   3858   4003   4146   4200   4401   4405   4410   4423   4427   4431
                    4437   4443   4447   4452   4459   4463   4470   4475   4480   4485   4688   4690   4757
                    4870   4983   5096   5209   5311   5353   5390   5441   5494   5514   5532   5551   5570
                    5590   5610   5630
```

```
SR2       001564          560*   2303   2449   2595   2741   2887   3549   3911   4148   4202   4823   4936   5049
                         5162   5275   5347   5435   5669
STACK  =  001100            78*    711    886   4666   5679
START     002016            68    704*
STKLMT=   177774            89*
STPCOD=   000073           690*
STREG1    001614           591*    767*   1757   1772   1783   1785   1796   1798   1808   1810   1823   1827   1842
                         1845   1857   1859
STREG2    001620           595*    833*   1919   1934   1945   1947   1958   1960   1970   1972   1985   1989   2004
                         2007   2019   2021
SWR       001140           249*    709    730*    732    738*    745*    901   4251   4261   5954   5961   5973   5977
                         6051   6065   6067   6073   6080   6109   6146*  6427   6440*
SWREG     000176            63*    738    901   6109   6122
SW0    =  000001           142*
SW00   =  000001           132*    142
SW01   =  000002           131*    141
SW02   =  000004           130*    140
SW03   =  000010           129*    139
SW04   =  000020           128*    138
SW05   =  000040           127*    137
SW06   =  000100           126*    136
SW07   =  000200           125*    135
SW08   =  000400           124*    134
SW09   =  001000           123*    133
SW1    =  000002           141*
SW10   =  002000           122*
SW11   =  004000           121*
SW12   =  010000           120*
SW13   =  020000           119*
SW14   =  040000           118*
SW15   =  100000           117*
SW2    =  000004           140*
SW3    =  000010           139*
SW4    =  000020           138*
SW5    =  000040           137*
SW6    =  000100           136*
SW7    =  000200           135*
SW8    =  000400           134*
SW9    =  001000           133*
SYNC      034112          6481*
TABLE     054642          6900*  6903   8067*
TAXING    001774           684*  6836*  6837*  6838*  6843*  6872*
TBITVE=   000014           175*
TIME      040614          7406*  7409   7414*  7418*  7420*
TKVEC  =  000060           182*
TODOC     054640          6817*  6851   6859*  8059*
TODOQ     054520          6807   6813   6854   8058*
TOLER     034512          5360*  5448*  5709*  5718*  6528   6531   6532   6535   6579*
TPVEC  =  000064           183*
TRAPVE=   000034           181*   717*   718*
TRTVEC=   000014           176*
TST1      003504           918*
TST10     007254          2143   2155   2166   2193   2220   2248*
TST11     007762          2305   2320   2329   2338   2352   2366   2394*
```

```
TST12   010470          2451    2466    2475    2484    2498    2512    2540#
TST13   011176          2597    2612    2621    2630    2644    2658    2686#
TST14   011704          2743    2758    2767    2776    2790    2804    2832#
TST15   012412          2889    2904    2913    2922    2936    2950    2978#
TST16   013442          3082    3093    3104    3115    3127    3138    3149    3160    3169    3178#
TST17   014012          3211    3223    3236    3248    3262    3309#
TST2    003644           966#
TST20   014354          3384    3409#
TST21   015132          3502    3512    3523    3545#
TST22   015336          3548    3550    3590#
TST23   016024          3623    3635    3648    3660    3678    3758#
TST24   016346          3832    3857#
TST25   017044          3898    3909    3913    3928    3937    3946    3960    3974    4001#
TST26   017622          4095    4106    4119    4127    4143#
TST27   020026          4147    4149    4196#
TST3    004464          1137    1143    1185    1328    1392    1406    1430#
TST30   020240          4201    4203    4246#
TST31   020502          4282    4289    4293    4307#
TST32   020666          4324    4334    4339    4353#
TST33   020774          4368    4374    4378    4387#
TST34   021616          4508    4510    4518    4531#
TST35   021732          4548    4554    4565#
TST36   022154          4581    4595    4606    4612    4625#
TST37   022364          4645    4659    4678#
TST4    004634          1460    1466    1479    1484    1540#
TST40   022614          4691    4715    4718    4743#
TST41   023340          4824    4829    4856#
TST42   024064          4937    4942    4969#
TST43   024610          5050    5055    5082#
TST44   025334          5163    5168    5195#
TST45   026060          5276    5281    5309#
TST46   026416          5348    5388#
TST47   027014          5436    5482#
TST5    005272          1698    1747#
TST50   030220          5665#
TST51   030576          5729#
TST6    005760          1778    1790    1803    1815    1833    1909#
TST7    006446          1940    1952    1965    1977    1995    2081#
TYPDS = 104405          5770    5776    7579#
TYPE  = 104401           895    5753    5759    5764    5771    5851    5926    5956    5964    5995    6012    6014    6017
                        6019    6023    6030    6120    6121    6124    6137    6148    6167    6216    6219    6223    6284
                        6449    6594    6601    6609    6612    6618    6629    6632    6633    6636    6639    6650    6663
                        6666    6678    6681    6684    6687    6692    6695    6698    6709    6721    6726    6727    6743
                        6775    6779    6961    7015    7019    7023    7251    7461    7468    7485    7575#   7591    7600
                        7609    7614    7620
TYPOC = 104402          6003    6027    6123    6631    6655    6665    7480    7576#   7619    7625
TYPON = 104404          7578#
TYPOS = 104403          7577#
USJNO   001756           677#   6878*   7759
USRDA   001760           678#
USWH  = 000007           693#
VBM   = 000007           692#   6908
VECTOR  001506           481#
VECTPS  001510           482#
```

```
VECT1   001556          491#    757*    758*    759    1384*   1396*   1397*   4628    6829
VECT2   001560          493#    759*    760*   1395*   1409*   1410*
VERSN   001512          484#   1472    4264    4493    4571    4633    4684    6801
WIR     034062         1560    1575    1592    1633    1650    1668    6461#
WIRT    034110         6461*   6464*   6471#
W1      040002         7149*   7164#
W2      040004         7153*   7165#
W3      040006         7158*   7166#
$AERR   037532         1765    1927    2094    2257    2403    2549    2695    2841    2987    3198    3322    3418    3609
                       3771    3865    4010    6968*   7061*   7087#   7377
$APTHD  001000          209     215#
$ASTAT= ****** U       6392    6407
$ATYC   033462         6363    6365#
$ATY1   033436         6361#
$ATY3   033444         6269    6362#
$ATY4   033454         5969    6364#
$AUTOB  001134          246#    905*    6117    6238
$BASE   001250          310#    748
$BDADR  001122          241#
$BDDAT  001126          243#    999*   1001*   1002    1008*   1009    1014*   1015    1020*   1021    1056*   1058*   1062*
                       1067*   1072*   1247*   1258    1315*   1330    1473*   1481    1564*   1565*   1567    1594*   1597
                       1636*   1637*   1640    1670*   1673    1754*   1757    1772*   1774    1785*   1786    1798*   1799
                       1810*   1811    1827    1828    1845    1847    1859*   1860    1916*   1919    1934*   1936    1947*
                       1948    1960*   1961    1972*   1973    1989    1990    2007    2009    2021*   2022    2103    2104
                       2115    2116    2127    2138    2139    2151    2162    2169*   2172    2176    2179    2189    2196*
                       2199    2207    2208    2219    2270    2286    2297    2313    2314    2325    2334    2348    2362
                       2416    2432    2443    2459    2460    2471    2480    2494    2508    2562    2578    2589    2605
                       2606    2617    2626    2640    2654    2708    2724    2735    2751    2752    2763    2772    2786
                       2800    2854    2870    2881    2897    2898    2909    2918    2932    2946    2998*   3009    3010*
                       3011    3022*   3023    3034*   3045    3056    3067    3078    3089    3100    3111    3122    3134
                       3145    3156    3167    3186*   3190    3205*   3207    3218*   3219    3231*   3232    3243*   3244
                       3260    3261    3330    3331    3342    3343    3354    3355    3362*   3365    3369    3372    3382
                       3383    3429*   3440    3441*   3442    3453*   3454    3465*   3476    3487    3498    3508    3519
                       3530    3553*   3566    3568*   3569    3581    3583*   3598*   3601    3617*   3619    3630*   3631
                       3643*   3644    3655*   3656    3672    3673    3690    3692    3703*   3704    3780    3781    3792
                       3793    3804    3811*   3814    3818    3821    3831    3878    3894    3905    3921    3922    3933
                       3942    3956    3970    4021*   4032    4033*   4034    4045*   4046    4057*   4068    4079    4091
                       4102    4115    4126    4152*   4165    4167*   4168    4180    4182*   4206*   4219    4221*   4222
                       4234    4236*   4280*   4286*   4294*   4321*   4326*   4328*   4329    4342*   4366*   4372*   4373
                       4505*   4512*   4514    4546*   4552*   4553    4579*   4588*   4590    4593*   4604*   4610*   4611
                       4642*   4656*   4712*   4720*   4833*   4946*   5059*   5172*   5285*   6486*   6624*   6630    6731
                       6733*   6826*   6827*   6840    6846*   6890*   6895*   7749    7753
$BELL   001164          261#    5956    5987
$CDWI   001254          312#
$CHARC  033330         6286*   6296*   6303    6312*   6317#
$CKSWR  032276         6109#   7583
$CMTAG  001100          229#    706     707     715     721     722
$CM3    =  000000       259#
$CNT    040502         7355*   7357*   7363#
$CNTLG  033023         6120    6233#
$CNTLU  033016         6137    6232#
$CPUOP  001222          284#
$CRLF   001171          263#    5964    5987    5995    6014    6019    6023    6148    6232    6285    6320
$DATR   040154         7201*   7218*   7222#   7333
```

```
$DBLK   031454          5892    5926    5934#
$DEVCT  001204           275#
$DEVm   001252           311#
$DOAGN  031002          5745    5778    5784#
$DTBL   031444          5895    5930#
$ENDAD  030772           194    5780#
$ENDCT  030634          5747#
$ENULL  031006          5787#
$ENv    001214           280#    899    5966    6264    6370    6394
$ENVM   001215           281#    743    6266    6271    6372
$EOP    030600          5737#
$EOpCT  030626          5744#   5748
$ERFLG  001103           232#   5951*   5987    6039    6069    6071    6077*   6099
$ERMAX  001115           238#    723*   6071    6094*   6099
$ERROR  031464           715    5949#
$ERrPC  001116           239#   5958*   5959*   5960    5987    6001    7747    7749    7752    7753    7759
$ERRTB  001256           329#   6009
$ERRTY  031660          5963    5994#
$ERtTL  001112           236#   5957*   5987
$EScAP  001162           260#    722*   5980    5982    5987    6093*
$ETABL  001214           279#
$ETEND  001256           221     313#
$FATAL  001176           272#   6398#
$FFLG   033702          6361*   6364*   6392    6401*   6409#
$FILLC  001156           257#   6289    6320
$FILLS  001155           256#   6320
$GDADR  001120           240#
$GDDAT  001124           242#   1000*   1002    1007*   1009    1015    1021    1055*   1246*   1258    1316*   1330    1472*
                        1481    1563*   1567    1595*   1597    1635*   1640    1671*   1673    1773*   1779*   1783    1786
                        1792*   1796    1799    1805*   1808    1820*   1823*   1828    1835*   1842    1845*   1847    1854*
                        1857    1935*   1941*   1945    1948    1954*   1958    1961    1967*   1970    1982*   1985*   1990
                        1997*   2004    2007*   2009    2016*   2019    2089*   2093    2104    2110*   2113    2116    2122*
                        2125    2133*   2136    2139    2145*   2149    2151    2157*   2160    2168*   2174    2177*   2179
                        2184*   2187    2195*   2201    2207*   2208    2214*   2217    2254*   2257    2279    2281*   2284
                        2286    2292*   2295    2308*   2311    2313*   2314    2322*   2325    2343*   2346    2348    2356
                        2357*   2360    2371    2400*   2403    2425    2427*   2430    2432    2438*   2441    2454*   2457
                        2459*   2460    2468*   2471    2489*   2492    2494    2502    2503*   2506    2517    2546*   2549
                        2571    2573*   2576    2578    2584*   2587    2600*   2603    2605*   2606    2614*   2617    2635*
                        2638    2640    2648    2649*   2652    2663    2692*   2695    2717    2719*   2722    2724    2730*
                        2733    2746*   2749    2751*   2752    2760*   2763    2781*   2784    2786    2794    2795*   2798
                        2809    2838*   2841    2863    2865*   2868    2870    2876*   2879    2892*   2895    2897*   2898
                        2906*   2909    2927*   2930    2932    2940    2941*   2944    2955    2984*   2987    3004*   3007
                        3011    3017*   3020    3023    3029*   3032    3040*   3043    3045    3054    3056    3065    3067
                        3076    3078    3084*   3087    3089    3098    3100    3109    3111    3120    3122    3129*   3132
                        3134    3143    3145    3154    3156    3165    3167    3187*   3206*   3212*   3216    3219    3225*
                        3229    3232    3238*   3241    3253*   3256*   3261    3317*   3321    3331    3337*   3340    3343
                        3349*   3352    3354*   3355    3361*   3367    3370*   3372    3377*   3380    3382*   3383    3415*
                        3418    3435*   3438    3442    3448*   3451    3454    3460*   3463    3471*   3474    3476    3485
                        3487    3493*   3496    3498    3506    3508    3514*   3517    3519    3528    3530    3552*   3558
                        3560*   3561    3574    3576*   3618*   3624*   3628    3631    3637*   3641    3644    3650*   3653
                        3665*   3668*   3673    3680*   3687    3690*   3692    3698*   3701    3766*   3770    3781    3787*
                        3790    3793    3799*   3802    3810*   3816    3819*   3821    3826*   3829    3862*   3865    3886
                        3888    3889*   3892    3894    3900*   3903    3916*   3919    3921*   3922    3930*   3933    3951*
                        3954    3956    3965*   3968    3979    4007*   4010    4027*   4030    4034    4040*   4043    4046
```

```
$MTYP4  001241          306#
$MXCNT  032274         6088    6098#
$NULL   001154          255#    6291    6320
$NWtST= 000001          915#     954#     956    1413#    1415    1495#    1497    1707#    1709    1869#    1871    2031#    2033
                       2228#    2230    2374#    2376    2520#    2522    2666#    2668    2812#    2814    2959#    2961    3175#
                       3269#    3271    3390#    3392    3537#    3539    3587#    3713#    3715    3837#    3839    3982#    3984
                       4135#    4137    4188#    4190    4240#    4242    4298#    4300    4345#    4347    4380#    4382    4524#
                       4526     4558#    4560    4616#    4618    4671#    4673    4726#    4728    4839#    4841    4952#    4954
                       5065#    5067    5178#    5180    5292#    5294    5367#    5369    5455#    5457    5652#    5654    5726#
$OCNT   031234         5823#    5852#    5865#
$OFS   = 040412         7330#    7530    7531    7537
$OMoDE  031236         5818#    5822#    5827    5830#    5841#    5867#
$OUTLP  040252          1757     1783     1796    1808     1823     1842    1857    1919    1945    1958    1970    1985    2004
                        2019     2093     2113    2125     2136     2149    2160    2172    2174    2187    2199    2201    2217
                        2257     2284     2295    2311     2343     2346    2356    2360    2371    2403    2430    2441    2457
                        2489     2492     2502    2506     2517     2549    2576    2587    2603    2635    2638    2648    2652
                        2663     2695     2722    2733     2749     2781    2784    2794    2798    2809    2841    2868    2879
                        2895     2927     2930    2940     2944     2955    2987    3007    3020    3032    3043    3054    3065
                        3076     3087     3098    3109     3120     3132    3143    3154    3165    3190    3216    3229    3241
                        3256     3321     3340    3352     3365     3367    3380    3418    3438    3451    3463    3474    3485
                        3496     3506     3517    3528     3558     3560    3566    3568    3574    3576    3581    3583    3601
                        3628     3641     3653    3668     3687     3701    3770    3790    3802    3814    3816    3829    3865
                        3892     3903     3919    3951     3954     3965    3968    3979    4010    4030    4043    4055    4066
                        4077     4089     4100    4113     4124     4157    4159    4165    4167    4173    4175    4180    4182
                        4211     4213     4219    4221     4227     4229    4234    4236    7272#   7492
$OVER   032260         6052     6068     6076    6086    6095#
$PASS   001202          274#     742*    4249    5741*    5742*    5768    5787    6082    6099    6644*    6658*    6703*    6715*
$PASTM  001006          219#
$POWER  034052         6450    6455#
$PUTS   041132         7526#
$PWRDN  033704          719     6419#    6447
$PWRMG  034040         6450#
$PWRUP  033756         6429    6435#
$QUES   001170          262#    5987    6167    6216    6232    6320
$RDCHR  032560         6180#    7584
$RDDEC= ****** U       7587
$RDLIN  032700         6208#    7585
$RDOCT  033334         6330#    7586
$RDSZ  = 000010         6201#
$RESET  040504          4248    7374#    7384
$RTNAD  031004         5786#
$R2A   = ****** U       7587
$SAD    040500         7352    7354*    7362#
$SAVRE= ****** U        7587
$SAVR6  034050         6428*    6436    6437*    6438*    6454#
$SCOPE  032014          713     6048#
$SETUP= 000117          703#     712     713     715     717     719     721     722     724     895     896    5739    5950
                       5976    5984    6049    6104    6238
$STUP  = 177777         703#
$SVLAD  032224         6060    6089#
$SVPC  = 000244         192#     197
$SWR   = 167400          31#      41      47      48      49      50      51      52      53     259     260     261     721
                        722     724     725     919     967    1431    1541    1748    1910    2082    2249    2395    2541
                       2687    2833    2979    3179    3310    3410    3546    3591    3759    3858    4002    4144    4197
```

```
                    4247    4308    4354    4388    4532    4566    4626    4679    4744    4857    4970    5083    5196
                    5310    5389    5483    5666    5730    5734    5740    5779    5785    5787    5941    5942    5943
                    5944    5945    5954    5961    5973    5977    5987    6040    6041    6042    6043    6044    6051
                    6063    6065    6066    6069    6070    6071    6078    6079    6080    6092    6095    6098    6451
$SWREG   001216       282#     745
$SWRMK=  000000        53      54    6044    6045    6067
$TESTN   001200       273#    923#   6090#   6596#   7753    7756    7759
$TIMES   001160       259#    721#    919#    967#   1431#   1541#   4197#   4308#   4679#   4744#   4857#   4970#   5083#
                    5196#   5483#   5740#   6078#   6085    6088#   6098
$TKb     001146       252#   6102    6113    6130    6184    6190    7467
$TKS     001144       251#   6102    6111    6127   6151#    6182    6188    7465
$TLKR    040010      7177#   7329
$TLKW    037674      7145#   7287
$TN    = 000052        41#    187#    915     919#    954     967#   1137    1143    1185    1328    1392    1406    1413
                    1431#   1460    1466    1479    1484    1495   1541#   1698    1707    1748#   1778    1790    1803
                    1815    1833    1869    1910#   1940    1952    1965    1977    1995    2031    2082#   2143    2155
                    2166    2193    2220    2228    2249#   2305    2320    2329    2338    2352    2366    2374    2395#
                    2451    2466    2475    2484    2498    2512    2520   2541#   2597    2612    2621    2630    2644
                    2658    2666   2687#    2743    2758    2767    2776    2790    2804    2812   2833#    2889    2904
                    2913    2922    2936    2950    2959   2979#    3082    3093    3104    3115    3127    3138    3149
                    3160    3169    3175   3179#    3211    3223    3236    3248    3262    3269   3310#    3384    3390
                   3410#    3502    3512    3523    3537   3546#    3548    3550    3587   3591#    3623    3635    3648
                    3660    3678    3713   3759#    3832    3837   3858#    3898    3909    3913    3928    3937    3946
                    3960    3974    3982   4002#    4095    4106    4119    4127    4135   4144#    4147    4149    4188
                   4197#    4201    4203    4240   4247#    4282    4289    4293    4298   4308#    4324    4334    4339
                    4345   4354#    4368    4374    4378    4380   4388#    4508    4510    4518    4524   4532#    4548
                    4554    4558   4566#    4581    4595    4606    4612    4616   4626#    4645    4659    4671   4679#
                    4691    4715    4718    4726   4744#    4824    4829    4839   4857#    4937    4942    4952   4970#
                    5050    5055    5065   5083#    5163    5168    5178   5196#    5276    5281    5292   5310#    5348
                    5367   5389#    5436    5455   5483#    5652   5666#    5726   5730#
$TOUT    040444      6991    7045    7072    7186    7203   7352#
$TPB     001152       254#   6309*   6320
$TPFLG   001157       258#    6258    6320
$TPS     001150       253#    6307    6320
$TR    = 000001      8102#   8105#   8108#   8111#   8114#   8117#   8120#   8123#   8126#   8168#   8171#   8174#   8177#
                    8180#   8183#   8186#   8189#   8192#   8234#   8237#   8240#   8243#   8246#   8249#   8252#   8255#
                    8258#   8300#   8303#   8306#   8309#   8312#   8315#   8318#   8321#   8324#   8366#   8369#   8372#
                    8375#   8378#   8381#   8384#   8387#   8390#   8432#   8435#   8438#   8441#   8444#   8447#   8450#
                    8453#   8456#   8498#   8501#   8504#   8507#   8510#   8513#   8516#   8519#   8522#   8564#   8567#
                    8570#   8573#   8576#   8579#   8582#   8585#   8588#
$TRAP    041242       717    7552#
$TRAP2   041264      7563#   7574
$TRP   = 000013      7567#   7576#   7577#   7578#   7579#   7580#   7581    7582#   7583    7584#   7585#   7586#   7587#
$TRPAD   041276      7557    7574#
$TSTM    001004       218#
$TSTNM   001102       231#    922*   5739*   5953    5987    6039    6067    6089*   6090    6095    6099
$TTYIN   033006      6209    6210    6227    6231#
$TYPBN=  ****** U    7580
$TYPDS   031240      5880#   7579
$TYPE    033052      6258#   6389    7567    7575
$TYPEC   033264      6153    6288    6295    6302   6307#   6308
$TYPEX   033332      6313    6315    6318#
$TYPOC   031036      5821#   7576
$TYpON   031052      5820    5823#   7578
```

```
$TYPOS  031012          5816#   7577
$UNIT   001206           276#
$UNITM  001010           220#
$USWR   001220           283#
$UTK    040736          7458#
$VECT1  001244           308#    757
$VECT2  001246           309#
$VERSN  001772           683#   4256*  4260*  4264*  4699   4747   4860   4973   5086   5199   5325   5335   5404
                        5414    5484   5667   5751   6604*  6616   7433
$XTSTR  032030          6054#
$$GET4= 000000          5779#
$OFILL  031235          5817*   5821*  5831   5866#
$40CAT= ****** U        5963    6051
.     = 063256            57#     61#    64#    67#    69#    72#   192    193#   195#   197#   199#   205    206#
                         208#    210#   228#   265    487#   710   724    725    909#   5677   5774#  5787   5788#
                        5934#   5987   6033#  6098   6099   6102  6231#  6232   6238   6320   6410#  6431   6453
                        6653#   6782#  7330   7464#  7603#  7623# 7915#  7970#  7977#  7982#  7989#  7997#  8006#
                        8014#   8023#  8034#  8042#  8050#  8058# 8078#  8154#  8220#  8286#  8352#  8418#  8484#
                        8550#   8616#  8620#  8621#  8622#  8623# 8624#  8625#  8628
.DVLS   001514           486#    755*   913*  2253*  2399*  2545* 2691*  2837*  7275   7296   7317   7341   7459*
.$ASTA= ****** U        6362    6365
.$X   = 001000           205#    210
```

```
BAHE    8102#  8105   8108   8111   8114   8117   8120   8123   8168#  8171   8174   8177   8180   8183   8186
        8189   8234#  8237   8240   8243   8246   8249   8252   8255   8300#  8303   8306   8309   8312   8315
        8318   8321   8366#  8369   8372   8375   8378   8381   8384   8387   8432#  8435   8438   8441   8444
        8447   8450   8453   8498#  8501   8504   8507   8510   8513   8516   8519   8564#  8567   8570   8573
        8576   8579   8582   8585
COMMEN   185#
DAAT    4186#  4187
DFC      331#   337    344    351    358    365    372    379    386    393    400    414    421    428    435
         442    448
DRLT    4724#  4725   4838   4951   5064   5177
ENDCOM   185#
ENDPAS  5730#  5749
ERROR     79#   942   1005   1012   1018   1024   1060   1065   1070   1075   1136   1142   1184   1190   1256
        1261   1327   1333   1391   1405   1458   1463   1476   1486   1570   1586   1600   1643   1663   1676
        1767   1777   1789   1802   1814   1831   1851   1862   1929   1939   1951   1964   1976   1993   2013
        2024   2097   2107   2119   2130   2142   2154   2165   2182   2192   2211   2222   2260   2273   2289
        2300   2317   2328   2337   2351   2365   2406   2419   2435   2446   2463   2474   2483   2497   2511
        2552   2565   2581   2592   2609   2620   2629   2643   2657   2698   2711   2727   2738   2755   2766
        2775   2789   2803   2844   2857   2873   2884   2901   2912   2921   2935   2949   2990   3001   3014
        3026   3037   3048   3059   3070   3081   3092   3103   3114   3125   3137   3148   3159   3170   3200
        3210   3222   3235   3247   3264   3325   3334   3346   3358   3375   3386   3421   3432   3445   3457
        3468   3479   3490   3501   3511   3522   3533   3611   3622   3634   3647   3659   3676   3696   3706
        3774   3784   3796   3807   3824   3834   3868   3881   3897   3908   3925   3936   3945   3959   3973
        4013   4024   4037   4049   4060   4071   4082   4094   4105   4118   4129   4281   4288   4296   4323
        4333   4343   4367   4376   4416   4507   4516   4520   4547   4555   4580   4594   4605   4614   4644
        4658   4692   4713   4722   4835   4948   5061   5174   5287   6467   6557   6847   6893   6987   6993
        7006   7047   7074   7188   7205
ESCAPE   185#
GETPRI   185#
GETSWR   185#   896#
INSTR2  5935#  5984
LDGD    4186#  4327   4341   4365   4371   4545   4550   4578   4587   4603   4608   4641   4655   4711   4719
LPWT    4670#
MOVEI     20#  1770   1783   1796   1808   1817   1825   1843   1857   1865   1932   1945   1958   1970   1979
        1987   2005   2019   2027   2101   2113   2125   2136   2149   2160   2174   2187   2205   2217   2267
        2276   2279   2284   2295   2311   2323   2331   2346   2360   2413   2422   2425   2430   2441   2457
        2469   2477   2492   2506   2559   2568   2571   2576   2587   2603   2615   2623   2638   2652   2705
        2714   2717   2722   2733   2749   2761   2769   2784   2798   2851   2860   2863   2868   2879   2895
        2907   2915   2930   2944   2996   3007   3020   3032   3043   3054   3065   3076   3087   3098   3109
        3120   3132   3143   3154   3165   3203   3216   3229   3241   3250   3258   3328   3340   3352   3367
        3380   3427   3438   3451   3463   3474   3485   3496   3506   3517   3528   3615   3628   3641   3653
        3662   3670   3688   3701   3709   3778   3790   3802   3816   3829   3875   3884   3886   3892   3903
        3919   3931   3939   3954   3968   4019   4030   4043   4055   4066   4077   4089   4100   4113   4124
        7375
MOVEM     19#  1755   1781   1794   1806   1821   1840   1855   1917   1943   1956   1968   1983   2002   2017
        2091   2111   2123   2134   2147   2158   2170   2172   2185   2197   2199   2215   2255   2282   2293
        2309   2340   2344   2353   2358   2368   2401   2428   2439   2455   2486   2490   2499   2504   2514
        2547   2574   2585   2601   2632   2636   2645   2650   2660   2693   2720   2731   2747   2778   2782
        2791   2796   2806   2839   2866   2877   2893   2924   2928   2937   2942   2952   2985   3005   3018
        3030   3041   3051   3062   3073   3085   3095   3106   3117   3130   3140   3151   3162   3188   3214
        3227   3239   3254   3319   3338   3350   3363   3365   3378   3416   3436   3449   3461   3472   3482
        3494   3503   3515   3525   3555   3558   3563   3566   3571   3574   3578   3581   3599   3626   3639
        3651   3666   3685   3699   3768   3788   3800   3812   3814   3827   3863   3890   3901   3917   3948
        3952   3962   3966   3976   4008   4028   4041   4053   4064   4074   4087   4097   4111   4121   4154
```

```
        4157    4162    4165    4170    4173    4177    4180    4208    4211    4216    4219    4224    4227    4231    4234
MRDA    7905#   8090    8156    8222    8288    8354    8420    8486    8552
MULT     185#
NEWTST   185#     915     954    1413    1495    1707    1869    2031    2228    2374    2520    2666    2812    2959    3175
        3269    3390    3537    3587    3713    3837    3982    4135    4188    4240    4298    4345    4380    4524    4558
        4616    4671    4726    4839    4952    5065    5178    5292    5367    5455    5652    5726
POP      185#   5921    6353    6404    6405    6440    6441
PUSH     185#   5880    6332    6365    6367    6388    6421    6427
REPORT   185#
RWARN   4670#   4735    4848    4961    5074    5187    5301    5380
R2D2     784#    785     792     799     806     813
SCOPE     80#    966    1430    1540    1747    1909    2081    2248    2394    2540    2686    2832    2978    3178    3309
        3409    3545    3590    3758    3857    4001    4143    4196    4246    4307    4353    4387    4531    4565    4625
        4678    4743    4856    4969    5082    5195    5309    5388    5482    5665    5729
SDRM    5510#   5511    5567    5587    5607    5627
SETPRI   185#
SETTRA  7567#   7576    7577    7578    7579    7581    7583    7584    7585    7586
SETUP    185#    704
SKIP     185#   1137    1143    1185    1328    1392    1406    1460    1466    1479    1484    1698    1778    1790    1803
        1815    1833    1940    1952    1965    1977    1995    2143    2155    2166    2193    2220    2305    2320    2329
        2338    2352    2366    2451    2466    2475    2484    2498    2512    2597    2612    2621    2630    2644    2658
        2743    2758    2767    2776    2790    2804    2889    2904    2913    2922    2936    2950    3082    3093    3104
        3115    3127    3138    3149    3160    3169    3211    3223    3236    3248    3262    3384    3502    3512    3523
        3548    3550    3623    3635    3648    3660    3678    3832    3898    3909    3913    3928    3937    3946    3960
        3974    4095    4106    4119    4127    4147    4149    4201    4203    4282    4289    4293    4324    4334    4339
        4368    4374    4378    4508    4510    4518    4548    4554    4581    4595    4606    4612    4645    4659    4691
        4715    4718    4824    4829    4937    4942    5050    5055    5163    5168    5276    5281    5348    5436
SLASH    185#
SPACE    185#
STANDE  4569#   4631    4682    6799
STARS    185#     190     202     204     211     224     265     268     915     917     954     965    1413    1429    1495
        1539    1707    1746    1869    1908    2031    2080    2228    2247    2374    2393    2520    2539    2666    2685
        2812    2831    2959    2977    3175    3177    3269    3308    3390    3408    3537    3544    3587    3589    3713
        3757    3837    3856    3982    4000    4135    4142    4188    4195    4240    4245    4298    4306    4345    4352
        4380    4386    4524    4530    4558    4564    4616    4624    4671    4677    4726    4742    4839    4855    4952
        4968    5065    5081    5178    5194    5292    5308    5367    5387    5455    5481    5652    5664    5726    5728
        5732    5793    5870    5937    5989    6036    6101    6104    6172    6201    6243    6322    6360    6417    6433
        7546
SWRSU    185#    726#
TDRT    2226#   2227    2373    2519    2665    2811
TOUT     489#   6990    7044    7071    7185    7202
TRMTRP  7567#
TSTE    8627#   8628
TYPBIN   185#
TYPDEC   185#   5768    5775
TYPNAM   185#    891
TYPNUM   185#
TYPOCS    31#    185#
TYPOCT   185#   6001    6025    6122    6630    6654    6664    7479    7618    7624
TYPTXT   185#   5753    5758    5764    5771    6650    6708    6743    6774    6779    6961    7015    7019    7023    7250
        7460    7485    7590    7599    7609    7613    7620
XY       953#    956    1412#    1415    1495#    1497    1706#    1709    1868#    1871    2030#    2033    3268#    3271    3712#
        3715
ZK      1705#   1709    1867#    1871    2034    2230    2376    2522    2668    2814    2961    3272    3392    3716    3839
```

```
            3984
zK1         4240#   4242
ZT          2228#   2230    2374#   2376    2520#   2522    2666#   2668    2812#   2814    2958#   2961    3389#   3392    3536#
            3539    3837#   3839    3981#   3984    4134#   4137    4187#   4190    4298#   4300    4345#   4347    4380#   4382
            4524#   4526    4557#   4560    4616#   4618    4670#   4673    4725#   4728    4838#   4841    4951#   4954    5064#
            5067    5177#   5180    5291#   5294    5366#   5369    5454#   5457
ZZZ         5652#   5654
$ADTS       1705#   1867
$CAL.         30#   6926
$DMAST        30#   7914
$DMDT         30#   7914
$DRAG        606#    607     612     617     622     627
$MMAST        30#   7914
$$CMRE       222#
$$CMTM       222#
$$ESCA       185#
$$NEWT       185#    915     954    1413    1495    1707    1869    2031    2228    2374    2520    2666    2812    2959    3175
            3269    3390    3537    3587    3713    3837    3982    4135    4188    4240    4298    4345    4380    4524    4558
            4616    4671    4726    4839    4952    5065    5178    5292    5367    5455    5652    5726
$$SET       7567#   7576    7577    7578    7579    7581    7583    7584    7585    7586
$$SETM       742#
$$SKIP       185#   1137    1143    1185    1328    1392    1406    1460    1466    1479    1484    1698    1778    1790    1803
            1815    1833    1940    1952    1965    1977    1995    2143    2155    2166    2193    2220    2305    2320    2329
            2338    2352    2366    2451    2466    2475    2484    2498    2512    2597    2612    2621    2630    2644    2658
            2743    2758    2767    2776    2790    2804    2889    2904    2913    2922    2936    2950    3082    3093    3104
            3115    3127    3138    3149    3160    3169    3211    3223    3236    3248    3262    3384    3502    3512    3523
            3548    3550    3623    3635    3648    3660    3678    3832    3898    3909    3913    3928    3937    3946    3960
            3974    4095    4106    4119    4127    4147    4149    4201    4203    4282    4289    4293    4324    4334    4339
            4368    4374    4378    4508    4510    4518    4548    4554    4581    4595    4606    4612    4645    4659    4691
            4715    4718    4824    4829    4937    4942    5050    5055    5163    5168    5276    5281    5348    5436
.EQUAT        31#     75
.HEADE        31#
.KMADR        17#    455
.KSIS         22#
.LOADL        25#   7089
.LPAIN        22#   6927
.PUTCS        24#   7506
.RESET        22#   7364
.SETTR        31#
.SETUP        31#    703
.SWRHI        31#     43
.SWRLO        54#
.TRMTR        31#
.UTK          29#   7448
.$ACT1        31#    188
.$APTB        31#    266#
.$APTH        31#    200
.$APTY        31#   6358
.$CATC        31#     55
.$CMTA        31#    222
.$DB2D        31#
.$EOP         31#   5730
.$ERRO        31#   5935
.$ERRT        31#   5987
```

```
.$INLP      28#   7299
.$MMAC      18#
.$OUTL      27#   7260
.$POWE      31#   6415
.$RDOC      31#   6320
.$READ      31#   6099
.$SB2D      31#
.$SCOP      31#   6034
.$TLKW      26#   7138
.$TOUT     489#   7343
.$TRAP      31#   7544
.$TYPD      31#   5868
.$TYPE      31#   6241
.$TYPO      31#   5791
```

```
        4102  4115  4126  4329  4753  4830  4866  4943  4979  5056  5092  5169  5205  5282  5319
        5398  5491  5912  6061  6085  6109  6115  6135  6142  6154  6156  6186  6192  6195  6197
        6210  6515  6518  6529  6533  6539  6729  6810  6840  6966  7030  7107  7113  7115  7279
        7322  7352
CMPB     899  1187  1470  1481  1567  1597  1640  1673  4373  4514  4553  4590  4611  4699  4747
        4860  4973  5086  5199  5325  5335  5404  5414  5484  5667  5751  5966  6067  6071  6117
        6149  6214  6220  6264  6279  6281  6289  6310  6314  6370  6462  6598  6605  6607  6616
        6621  6670  6674  6731  6749  6888  6913  7002  7004  7009  7056  7083  7235  7433  7472
COM     4803  4916  5029  5142  5255
DEC     3576  3583  4175  4182  4229  4236  4268  4277  4398  4804  4810  4828  4917  4923  4941
        5030  5036  5054  5143  5149  5167  5256  5262  5280  5362  5450  5743  6005  6525  6658
        6715  6765  6768  6824  6843  6859  7420  7430  7443
DECb    5841  5852  6293  6296  6645
EMT       79
HALT      61  4693  5975  6260  6430  6452  6771  7129  7256
INC      751   893   974  1133  1181  1253  1324  1440  1445  1573  1648  3560  3568  4159  4167
        4213  4221  4355  5350  5438  5507  5524  5534  5564  5580  5600  5620  5640  5704  5741
        5847  5855  5898  5957  5984  6084  6164  6400  6438  6464  6538  6733  6762  6764  6872
        6890  6906  6960  6978  7011  7013  7061  7106  7118  7237  7243  7281  7324  7357
INCB    1388  1402  1582  1602  1659  1679  4363  4502  4542  5951  6089  6316  6981  7032  7127
        7414
IOT       80
JMP       68    70    73  1752  1768  1914  1930  2086  2098  2108  2120  2131  2251  2265  2274
        2290  2301  2397  2411  2420  2436  2447  2543  2557  2566  2582  2593  2689  2703  2712
        2728  2739  2835  2849  2858  2874  2885  2982  2993  3002  3015  3027  3038  3049  3060
        3071  3184  3201  3314  3326  3335  3347  3359  3413  3424  3433  3446  3458  3469  3480
        3491  3596  3612  3763  3775  3785  3797  3808  3860  3873  3882  4005  4016  4025  4038
        4050  4061  4072  4083  4420  4749  4759  4862  4872  4975  4985  5088  5098  5201  5211
        5313  5392  5486  5680  5785  6549  6600  6737  6751  6783
JSR      969  1433  1560  1575  1592  1633  1650  1668  1757  1772  1783  1785  1796  1798  1808
        1810  1820  1823  1827  1842  1845  1857  1859  1867  1919  1934  1945  1947  1958  1960
        1970  1972  1982  1985  1989  2004  2007  2019  2021  2029  2093  2103  2113  2115  2125
        2127  2136  2138  2149  2151  2160  2162  2172  2174  2176  2187  2189  2199  2201  2207
        2217  2219  2257  2270  2279  2281  2284  2286  2295  2297  2311  2313  2325  2334  2343
        2346  2348  2356  2360  2362  2371  2403  2416  2425  2427  2430  2432  2441  2443  2457
        2459  2471  2480  2489  2492  2494  2502  2506  2508  2517  2549  2562  2571  2573  2576
        2578  2587  2589  2603  2605  2617  2626  2635  2638  2640  2648  2652  2654  2663  2695
        2708  2717  2719  2722  2724  2733  2735  2749  2751  2763  2772  2781  2784  2786  2794
        2798  2800  2809  2841  2854  2863  2865  2868  2870  2879  2881  2895  2897  2909  2918
        2927  2930  2932  2940  2944  2946  2955  2987  2998  3007  3009  3020  3022  3032  3034
        3043  3045  3054  3056  3065  3067  3076  3078  3087  3089  3098  3100  3109  3111  3120
        3122  3132  3134  3143  3145  3154  3156  3165  3167  3190  3205  3216  3218  3229  3231
        3241  3243  3253  3256  3260  3321  3330  3340  3342  3352  3354  3365  3367  3369  3380
        3382  3418  3429  3438  3440  3451  3453  3463  3465  3474  3476  3485  3487  3496  3498
        3506  3508  3517  3519  3528  3530  3558  3560  3566  3568  3574  3576  3581  3583  3601
        3617  3628  3630  3641  3643  3653  3655  3665  3668  3672  3687  3690  3701  3703  3711
        3770  3780  3790  3792  3802  3804  3814  3816  3818  3829  3831  3865  3878  3886  3888
        3892  3894  3903  3905  3919  3921  3933  3942  3951  3954  3956  3965  3968  3970  3979
        4010  4021  4030  4032  4043  4045  4055  4057  4066  4068  4077  4079  4089  4091  4100
        4102  4113  4115  4124  4126  4157  4159  4165  4167  4173  4175  4180  4182  4211  4213
        4219  4221  4227  4229  4234  4236  4248  4254  4258  4267  4276  4318  4495  4499  4567
        4574  4600  4627  4636  4639  4653  4681  4687  4708  4821  4934  5047  5160  5273  5346
        5361  5434  5449  5648  5702  5712  5721  5780  5963  5969  6153  6269  6288  6295  6302
        6389  6481  6485  6740  6763  6798  6804  6842  6849  6969  6991  7034  7036  7038  7045
```

```
                7072    7148    7152    7157    7161    7180    7183    7186    7200    7203    7217    7287    7295    7329    7340
                7377    7429    7436    7442    7475    7492    7527    7534
MOV              707     711     713     714     715     716     717     718     719     720     724     725     728     729     730
                 731     736     738     739     740     745     748     749     750     757     759     766     767     768     771
                 772     773     774     775     776     777     786     787     788     789     793     794     795     796     800
                 801     802     803     807     808     809     810     814     815     816     817     821     822     823     824
                 825     826     832     833     834     837     838     839     840     841     842     843     844     845     854
                 855     856     857     858     859     860     861     862     863     864     877     879     881     883     885
                 886     887     889     890     919     920     922     923     924     925     927     928     929     930     950
                 951     967     972    1000    1007    1008    1014    1020    1062    1067    1072    1247    1316    1384    1395
                1396    1409    1431    1443    1472    1541    1544    1545    1546    1563    1626    1635    1779    1792    1820
                1823    1835    1845    1941    1954    1982    1985    1997    2007    2089    2110    2133    2145    2168    2169
                2177    2195    2196    2207    2281    2308    2313    2322    2343    2427    2454    2459    2468    2489    2573
                2600    2605    2614    2635    2719    2746    2751    2760    2781    2865    2892    2897    2906    2927    3004
                3017    3040    3084    3129    3212    3225    3253    3256    3317    3337    3354    3361    3362    3370    3382
                3435    3448    3471    3493    3514    3554    3624    3637    3665    3668    3680    3690    3766    3787    3810
                3811    3819    3889    3916    3921    3930    3951    4027    4040    4063    4086    4110    4153    4197    4207
                4256    4260    4266    4270    4272    4279    4280    4286    4308    4310    4313    4314    4316    4321    4322
                4327    4341    4354    4356    4365    4366    4371    4389    4390    4393    4400    4403    4407    4412    4422
                4425    4429    4433    4435    4439    4441    4445    4449    4454    4457    4461    4465    4468    4472    4474
                4477    4479    4482    4484    4487    4489    4496    4513    4519    4533    4535    4536    4545    4551    4569
                4578    4579    4587    4597    4603    4604    4609    4628    4629    4630    4631    4641    4642    4648    4655
                4656    4666    4679    4682    4698    4703    4704    4705    4711    4712    4719    4744    4751    4761    4762
                4764    4765    4768    4769    4771    4772    4773    4774    4779    4780    4782    4783    4786    4787    4788
                4791    4792    4793    4794    4799    4800    4801    4802    4807    4808    4813    4814    4815    4816    4817
                4818    4819    4825    4826    4827    4832    4833    4857    4864    4874    4875    4877    4878    4881    4882
                4884    4885    4886    4887    4892    4893    4895    4896    4899    4900    4901    4904    4905    4906    4907
                4912    4913    4914    4915    4920    4921    4926    4927    4928    4929    4930    4931    4932    4938    4939
                4940    4945    4946    4970    4977    4987    4988    4990    4991    4994    4995    4997    4998    4999    5000
                5005    5006    5008    5009    5012    5013    5014    5017    5018    5019    5020    5025    5026    5027    5028
                5033    5034    5039    5040    5041    5042    5043    5044    5045    5051    5052    5053    5058    5059    5083
                5090    5100    5101    5103    5104    5107    5108    5110    5111    5112    5113    5118    5119    5121    5122
                5125    5126    5127    5130    5131    5132    5133    5138    5139    5140    5141    5146    5147    5152    5153
                5154    5155    5156    5157    5158    5164    5165    5166    5171    5172    5196    5203    5213    5214    5216
                5217    5220    5221    5223    5224    5225    5226    5231    5232    5234    5235    5238    5239    5240    5243
                5244    5245    5246    5251    5252    5253    5254    5259    5260    5265    5266    5267    5268    5269    5270
                5271    5277    5278    5279    5284    5285    5314    5315    5317    5322    5323    5328    5329    5331    5332
                5333    5340    5341    5342    5343    5344    5345    5351    5352    5355    5356    5357    5358    5359    5360
                5393    5394    5396    5401    5402    5407    5408    5410    5411    5412    5419    5420    5421    5422    5423
                5424    5425    5426    5427    5428    5429    5430    5431    5433    5439    5440    5443    5444    5445    5446
                5447    5448    5483    5487    5488    5496    5497    5499    5500    5503    5504    5505    5506    5508    5517
                5518    5519    5520    5521    5522    5525    5526    5527    5528    5540    5541    5542    5543    5544    5545
                5546    5547    5548    5549    5554    5555    5556    5559    5560    5561    5562    5563    5573    5574    5575
                5576    5577    5578    5581    5582    5583    5584    5593    5594    5595    5596    5597    5598    5601    5602
                5603    5604    5613    5614    5615    5616    5617    5618    5621    5622    5623    5624    5633    5634    5635
                5636    5637    5638    5641    5642    5643    5644    5671    5678    5679    5682    5683    5684    5685    5686
                5687    5688    5690    5691    5693    5699    5700    5701    5705    5706    5707    5708    5709    5710    5711
                5714    5715    5716    5717    5718    5719    5720    5746    5768    5775    5777    5816    5824    5825    5826
                5832    5839    5857    5858    5859    5860    5861    5881    5882    5883    5884    5885    5886    5887    5892
                5895    5915    5921    5922    5923    5924    5925    5927    5928    5953    5958    5979    5982    5996    6001
                6010    6015    6020    6022    6026    6056    6057    6059    6062    6075    6087    6088    6091    6092    6095
                6096    6122    6146    6151    6180    6181    6208    6209    6224    6225    6226    6227    6262    6263    6268
                6276    6291    6330    6331    6332    6333    6334    6336    6351    6352    6353    6354    6355    6366    6367
                6374    6378    6383    6384    6386    6388    6398    6404    6405    6419    6420    6421    6422    6423    6424
```

```
        6425  6426  6427  6428  6429  6435  6436  6440  6441  6442  6443  6444  6445  6446  6447
        6448  6483  6503  6504  6505  6507  6509  6513  6514  6517  6520  6521  6527  6597  6603
        6611  6614  6615  6620  6623  6624  6625  6626  6627  6628  6630  6635  6638  6643  6646
        6647  6649  6654  6662  6664  6668  6680  6683  6686  6689  6694  6697  6700  6703  6704
        6706  6714  6717  6723  6728  6734  6748  6753  6754  6755  6758  6759  6760  6799  6805
        6807  6812  6813  6814  6815  6817  6818  6821  6829  6830  6831  6832  6833  6836  6845
        6846  6853  6854  6857  6864  6867  6868  6869  6870  6871  6874  6875  6876  6878  6879
        6881  6885  6895  6900  6903  6904  6918  6919  6921  6922  6923  6944  6959  6967  6976
        6980  6989  7029  7064  7097  7098  7099  7103  7110  7121  7122  7145  7146  7149  7150
        7153  7162  7177  7178  7184  7219  7233  7247  7272  7273  7275  7284  7286  7290  7291
        7293  7314  7315  7317  7327  7333  7334  7335  7338  7354  7406  7428  7432  7435  7440
        7441  7471  7479  7490  7526  7532  7552  7553  7557  7563  7564  7596  7598  7605  7607
        7618  7624
MOVB     723   905   977  1001  1056  1077  1177  1248  1318  1386  1399  1468  1473  1551  1552
        1565  1572  1574  1590  1594  1595  1637  1647  1649  1666  1670  1671  4264  4294  4328
        4342  4372  4493  4512  4546  4552  4571  4588  4593  4610  4633  4684  4720  4767  4785
        4880  4898  4993  5011  5106  5124  5219  5237  5334  5413  5695  5696  5697  5698  5817
        5818  5821  5822  5823  5827  5830  5831  5850  5890  5893  5907  5910  5919  5960  5968
        6090  6094  6113  6130  6184  6190  6213  6218  6273  6301  6309  6339  6361  6362  6364
        6486  6604  6641  6644  6801  6886  6905  6908  6912  7028  7033  7035  7037  7042  7070
        7151  7154  7156  7158  7159  7160  7181  7182  7199  7201  7216  7218  7467  7530  7531
        7537  7555
NEG     5828  5889
NOP      918  1053  1054  5738  5781  5782  5783
RESET   5779  7374
ROL     5834  5836  5837  5838  5840  6342  6344  6346
RTI      737  5862  5929  5986  6097  6152  6200  6228  6278  6356  6451  6865  6924  7421  7565
RTS     6024  6318  6406  6470  6487  6541  6850  7065  7085  7123  7163  7220  7248  7292  7336
        7360  7386  7413  7416  7447  7541  7558
SUB     5489  5896  5959  6381  6531  6532  6735  6816
TRAP    7567  7576  7577  7578  7579  7581  7583  7584  7585  7586
TST      878   882   897  1547  1765  1774  1811  1860  1927  1936  1973  2022  2094  2127  2162
        2189  2219  2257  2270  2297  2334  2362  2403  2416  2443  2480  2508  2549  2562  2589
        2626  2654  2695  2708  2735  2772  2800  2841  2854  2881  2918  2946  2987  3198  3207
        3244  3322  3418  3609  3619  3656  3704  3771  3804  3831  3865  3878  3905  3942  3970
        4010  5673  5845  5901  5911  5973  5980  6028  6058  6082  6144  6159  6275  6283  6305
        6350  6376  6394  6396  6535  6547  6657  6819  6851  6855  7126  7277  7285  7319  7328
        7377  7404  7409  7554
TSTB    1131  1139  1179  1251  1322  1558  4291  4319  4338  4361  4509  4540  4576  4601  4717
        5903  5917  6069  6111  6127  6182  6188  6258  6307  6368  6379  6392  6882  7058  7465
.ASCII   262   263  7767  7774  7827
.ASCIZ   261   264   909  5756  5762  5767  5774  6032  6232  6233  6234  6236  6455  6653  6712
        6746  6778  6782  6964  7018  7022  7026  7254  7464  7488  7499  7594  7603  7612  7617
        7623  7630  7635  7641  7648  7653  7659  7666  7671  7676  7681  7686  7690  7695  7701
        7709  7713  7719  7721  7728  7735  7764  7786  7793  7797  7804  7808  7810  7812  7817
        7822  7835  7839  7844  7849  7859  7863  7867  7871  7873  7876  7879  7884  7888  7891
        7896  7899
.ASECT    14
.BLKB   6231
.BLKW    487  5934  7915  8058  8078  8154  8220  8286  8352  8418  8484  8550  8616  8620  8621
        8622  8623  8624  8625
.BYTE    231   232   237   238   246   247   255   256   257   258   280   281   291   292   299
         300   302   303   305   306   683  5787  5863  5864  5865  5866  5970  5971  6229  6230
        6407  6408  6409  6591  7967  7972  7976  7979  7984  7988  7991  7994  7999  8002  8008
```

```
             8011    8016    8019    8028    8031    8036    8039    8044    8047    8052    8055    8100    8101    8103    8104
             8106    8107    8109    8110    8112    8113    8115    8116    8118    8119    8121    8122    8124    8125    8128
             8129    8131    8132    8135    8136    8141    8142    8166    8167    8169    8170    8172    8173    8175    8176
             8178    8179    8181    8182    8184    8185    8187    8188    8190    8191    8194    8195    8197    8198    8201
             8202    8207    8208    8232    8233    8235    8236    8238    8239    8241    8242    8244    8245    8247    8248
             8250    8251    8253    8254    8256    8257    8260    8261    8263    8264    8267    8268    8273    8274    8298
             8299    8301    8302    8304    8305    8307    8308    8310    8311    8313    8314    8316    8317    8319    8320
             8322    8323    8326    8327    8329    8330    8333    8334    8339    8340    8364    8365    8367    8368    8370
             8371    8373    8374    8376    8377    8379    8380    8382    8383    8385    8386    8388    8389    8392    8393
             8395    8396    8399    8400    8405    8406    8430    8431    8433    8434    8436    8437    8439    8440    8442
             8443    8445    8446    8448    8449    8451    8452    8454    8455    8458    8459    8461    8462    8465    8466
             8471    8472    8496    8497    8499    8500    8502    8503    8505    8506    8508    8509    8511    8512    8514
             8515    8517    8518    8520    8521    8524    8525    8527    8528    8531    8532    8537    8538    8562    8563
             8565    8566    8568    8569    8571    8572    8574    8575    8577    8578    8580    8581    8583    8584    8586
             8587    8590    8591    8593    8594    8597    8598    8603    8604
.DSABL       6169
.ENABL         31      54    6102
.END         8640
.ENDC          36      50      52      53      54      64      79     171     185     191     195     197     203     205     212
             225     229     231     259     260     261     262     266     269     291     299     302     305     308     309
             310     311     312     315     484     703     711     712     715     717     719     721     722     724     726
             747     895     901     907     909     916     917     918     919     920     921     955     956     965     966
             967     968    1138    1144    1186    1329    1393    1407    1414    1415    1429    1430    1431    1432    1461
            1467    1480    1485    1496    1497    1539    1540    1541    1542    1699    1708    1709    1746    1747    1748
            1779    1791    1804    1816    1834    1870    1871    1908    1909    1910    1941    1953    1966    1978    1996
            2032    2033    2080    2081    2082    2144    2156    2167    2194    2221    2229    2230    2247    2248    2249
            2306    2321    2330    2339    2353    2367    2375    2376    2393    2394    2395    2452    2467    2476    2485
            2499    2513    2521    2522    2539    2540    2541    2598    2613    2622    2631    2645    2659    2667    2668
            2685    2686    2687    2744    2759    2768    2777    2791    2805    2813    2814    2831    2832    2833    2890
            2905    2914    2923    2937    2951    2960    2961    2977    2978    2979    3083    3094    3105    3116    3128
            3139    3150    3161    3170    3176    3177    3178    3179    3212    3224    3237    3249    3263    3270    3271
            3308    3309    3310    3385    3391    3392    3408    3409    3410    3503    3513    3524    3538    3539    3544
            3545    3546    3549    3551    3588    3589    3590    3591    3624    3636    3649    3661    3679    3714    3715
            3757    3758    3759    3833    3838    3839    3856    3857    3858    3899    3910    3914    3929    3938    3947
            3961    3975    3983    3984    4000    4001    4002    4096    4107    4120    4128    4136    4137    4142    4143
            4144    4148    4150    4189    4190    4195    4196    4197    4198    4202    4204    4241    4242    4245    4246
            4247    4283    4290    4294    4299    4300    4306    4307    4308    4309    4325    4329    4335    4340    4343
            4346    4347    4352    4353    4354    4367    4369    4373    4375    4379    4381    4382    4386    4387    4388
            4509    4511    4519    4525    4526    4530    4531    4532    4547    4549    4553    4555    4559    4560    4564
            4565    4566    4580    4582    4589    4596    4605    4607    4611    4613    4617    4618    4624    4625    4626
            4643    4646    4657    4660    4672    4673    4677    4678    4679    4680    4692    4713    4716    4719    4721
            4727    4728    4742    4743    4744    4745    4825    4830    4840    4841    4855    4856    4857    4858    4938
            4943    4953    4954    4968    4969    4970    4971    5051    5056    5066    5067    5081    5082    5083    5084
            5164    5169    5179    5180    5194    5195    5196    5197    5277    5282    5293    5294    5308    5309    5310
            5349    5368    5369    5387    5388    5389    5437    5456    5457    5481    5482    5483    5484    5515    5571
            5591    5611    5631    5653    5654    5664    5665    5666    5727    5728    5729    5730    5733    5734    5736
            5739    5745    5748    5749    5756    5762    5767    5774    5777    5779    5785    5787    5788    5794    5871
            5938    5941    5951    5958    5963    5964    5965    5973    5984    5987    5990    6005    6034    6037    6040
            6045    6051    6053    6064    6067    6068    6069    6071    6073    6080    6084    6089    6091    6095    6098
            6099    6102    6103    6105    6133    6169    6173    6201    6202    6209    6211    6214    6216    6232    6238
            6244    6273    6323    6325    6358    6361    6362    6365    6392    6407    6418    6427    6428    6434    6440
            6441    6451    6458    6653    6712    6746    6778    6782    6964    6994    7001    7018    7022    7026    7048
            7055    7075    7082    7189    7196    7206    7213    7254    7448    7464    7488    7547    7553    7556    7575
            7576    7577    7578    7579    7580    7581    7582    7583    7584    7585    7586    7587    7594    7603    7612
```

```
                7617    7623    8628
                  79      80      88
.EQUIV           133     134     135     136     137     138     139     140     141     142     161     162
                 163     164     165     166     167     168     169     170
.EVEN            269     909    5756    5762    5767    5774    5788    6033    6410    6457    6653    6712    6746    6778    6782
                6964    7018    7022    7026    7254    7464    7488    7502    7594    7603    7612    7617    7623    7746    7904
.GLOBL             14     970    1434
.IDENT           7919
.IF                32      50      51      52      53      54      64      77     143     171     190     193     195     202     204
                  211     224     228     230     259     260     261     265     266     268     291     299     302     305     308
                  309     310     311     312     313     315     481     703     706     711     713     715     717     719     721
                  722     724     742     894     895     896     899     908     915     917     919     920     921     954     956
                  965     967     968    1137    1143    1185    1328    1392    1406    1413    1415    1429    1431    1432    1460
                 1466    1479    1484    1495    1497    1539    1541    1542    1698    1707    1709    1746    1748    1778    1790
                 1803    1815    1833    1869    1871    1908    1910    1940    1952    1965    1977    1995    2031    2033    2080
                 2082    2143    2155    2166    2193    2220    2228    2230    2247    2249    2305    2320    2329    2338    2352
                 2366    2374    2376    2393    2395    2451    2466    2475    2484    2498    2512    2520    2522    2539    2541
                 2597    2612    2621    2630    2644    2658    2666    2668    2685    2687    2743    2758    2767    2776    2790
                 2804    2812    2814    2831    2833    2889    2904    2913    2922    2936    2950    2959    2961    2977    2979
                 3082    3093    3104    3115    3127    3138    3149    3160    3169    3175    3177    3179    3211    3223    3236
                 3248    3262    3269    3271    3308    3310    3384    3390    3392    3408    3410    3502    3512    3523    3537
                 3539    3544    3546    3548    3550    3587    3589    3591    3623    3635    3648    3660    3678    3713    3715
                 3757    3759    3832    3837    3839    3856    3858    3898    3909    3913    3928    3937    3946    3960    3974
                 3982    3984    4000    4002    4095    4106    4119    4127    4135    4137    4142    4144    4147    4149    4188
                 4190    4195    4197    4198    4201    4203    4240    4242    4245    4247    4282    4289    4293    4298    4300
                 4306    4308    4309    4324    4328    4334    4339    4342    4345    4347    4352    4354    4366    4368    4372
                 4374    4378    4380    4382    4386    4388    4508    4510    4518    4524    4526    4530    4532    4546    4548
                 4552    4554    4558    4560    4564    4566    4579    4581    4588    4595    4604    4606    4610    4612    4616
                 4618    4624    4626    4642    4645    4656    4659    4671    4673    4677    4679    4680    4691    4712    4715
                 4718    4720    4726    4728    4742    4744    4745    4824    4829    4839    4841    4855    4857    4858    4937
                 4942    4952    4954    4968    4970    4971    5050    5055    5065    5067    5081    5083    5084    5163    5168
                 5178    5180    5194    5196    5197    5276    5281    5292    5294    5308    5310    5348    5367    5369    5387
                 5389    5436    5455    5457    5481    5483    5484    5514    5570    5590    5610    5630    5652    5654    5664
                 5666    5726    5728    5730    5732    5733    5734    5735    5736    5738    5744    5747    5749    5755    5761
                 5766    5773    5777    5779    5785    5787    5788    5793    5870    5937    5940    5951    5954    5961    5963
                 5964    5966    5973    5977    5984    5987    5989    6004    6020    6036    6039    6044    6050    6051    6063
                 6065    6066    6067    6069    6070    6071    6080    6082    6090    6092    6097    6098    6099    6101    6103
                 6104    6105    6133    6172    6173    6201    6209    6210    6214    6215    6231    6232    6238    6243    6264
                 6322    6325    6337    6360    6362    6365    6392    6407    6417    6427    6428    6433    6440    6441    6449
                 6451    6455    6652    6711    6745    6777    6781    6963    6993    6999    7017    7021    7025    7047    7053
                 7074    7080    7188    7194    7205    7211    7253    7424    7463    7487    7546    7552    7556    7567    7576
                 7577    7578    7579    7580    7581    7583    7584    7585    7586    7587    7593    7602    7611    7616    7622
                 8628
.IFF               50      52      53      54      77     191     195     197     203     205     212     225     228     231     259
                  266     269     711     894     895     915     916     917     918     919     955     956     966     967    1138
                 1144    1186    1329    1393    1407    1414    1415    1430    1431    1461    1467    1480    1485    1496    1497
                 1540    1541    1699    1708    1709    1747    1748    1779    1791    1804    1816    1834    1870    1871    1909
                 1910    1941    1953    1966    1978    1996    2032    2033    2081    2082    2144    2156    2167    2194    2221
                 2229    2230    2248    2249    2306    2321    2330    2339    2353    2367    2375    2376    2394    2395    2452
                 2467    2476    2485    2499    2513    2521    2522    2540    2541    2598    2613    2622    2631    2645    2659
                 2667    2668    2686    2687    2744    2759    2768    2777    2791    2805    2813    2814    2832    2833    2890
                 2905    2914    2923    2937    2951    2960    2961    2978    2979    3083    3094    3105    3116    3128    3139
                 3150    3161    3170    3176    3177    3178    3179    3212    3224    3237    3249    3263    3270    3271    3309
                 3310    3385    3391    3392    3409    3410    3503    3513    3524    3538    3539    3545    3546    3549    3551
                 3588    3589    3590    3591    3624    3636    3649    3661    3679    3714    3715    3758    3759    3833    3838
```

```
        3839  3857  3858  3899  3910  3914  3929  3938  3947  3961  3975  3983  3984  4001  4002
        4096  4107  4120  4128  4136  4137  4143  4144  4148  4150  4189  4190  4196  4197  4198
        4202  4204  4241  4242  4246  4247  4283  4290  4294  4299  4300  4307  4308  4309  4325
        4328  4335  4340  4342  4346  4347  4353  4354  4367  4369  4372  4375  4379  4381  4382
        4387  4388  4509  4511  4519  4525  4526  4531  4532  4546  4549  4552  4555  4559  4560
        4565  4566  4580  4582  4588  4596  4605  4607  4610  4613  4617  4618  4625  4626  4643
        4646  4657  4660  4672  4673  4678  4679  4680  4692  4713  4716  4719  4720  4727  4728
        4743  4744  4745  4825  4830  4840  4841  4856  4857  4858  4938  4943  4953  4954  4969
        4970  4971  5051  5056  5066  5067  5082  5083  5084  5164  5169  5179  5180  5195  5196
        5197  5277  5282  5293  5294  5309  5310  5349  5368  5369  5388  5389  5437  5456  5457
        5482  5483  5484  5515  5570  5590  5610  5630  5653  5654  5665  5666  5727  5728  5729
        5730  5733  5735  5738  5745  5748  5787  5794  5871  5938  5940  5954  5984  5990  6005
        6034  6037  6064  6067  6068  6071  6098  6102  6105  6173  6175  6180  6201  6202  6211
        6215  6232  6244  6323  6361  6418  6434  6451  6993  6999  7047  7053  7074  7080  7188
        7194  7205  7211  7547  7553
.IFT     909  5756  5762  5767  5774  5964  6079  6175  6180  6341  6357  6358  6653  6712  6746
        6778  6782  6964  7018  7022  7026  7254  7464  7488  7594  7603  7612  7617  7623
.IFTF    909  5756  5762  5767  5774  5963  6077  6120  6173  6176  6337  6341  6357  6653  6712
        6746  6778  6782  6964  7018  7022  7026  7254  7464  7488  7594  7603  7612  7617  7623
.IIF      31    36    41    42    47    48    49    50    53    54    61   265   269   712   715
         721   722   724   725   895  5734  5739  5740  5769  5776  5787  5788  5941  5942  5943
        5944  5945  5950  5976  5984  5987  6002  6027  6040  6041  6042  6043  6044  6045  6049
        6078  6079  6095  6098  6099  6102  6123  6224  6232  6238  6320  6631  6655  6665  7480
        7575  7576  7577  7578  7579  7581  7583  7584  7585  7586  7619  7625
.IRP     703   915   954  1413  1495  1707  1869  2031  2228  2374  2520  2666  2812  2959  3175
        3269  3390  3537  3587  3713  3837  3982  4135  4188  4240  4298  4345  4380  4524  4558
        4616  4671  4726  4839  4952  5065  5178  5292  5367  5455  5652  5726  5738  5881  5921
        5984  6050  6332  6353  6366  6367  6388  6404  6405  6421  6427  6440  6441
.LIST     14    31    53    61   185   259   266   269   703   726   895   896   909   915   919
         954   967  1413  1431  1495  1541  1707  1748  1757  1772  1783  1785  1796  1798  1808
        1810  1820  1823  1827  1842  1845  1857  1859  1867  1869  1910  1919  1934  1945  1947
        1958  1960  1970  1972  1982  1985  1989  2004  2007  2019  2021  2029  2031  2082  2093
        2103  2113  2115  2125  2127  2136  2138  2149  2151  2160  2162  2172  2174  2176  2187
        2189  2199  2201  2207  2217  2219  2228  2249  2257  2270  2279  2281  2284  2286  2295
        2297  2311  2313  2325  2334  2343  2346  2348  2356  2360  2362  2371  2374  2395  2403
        2416  2425  2427  2430  2432  2441  2443  2457  2459  2471  2480  2489  2492  2494  2502
        2506  2508  2517  2520  2541  2549  2562  2571  2573  2576  2578  2587  2589  2603  2605
        2617  2626  2635  2638  2640  2648  2652  2654  2663  2666  2687  2695  2708  2717  2719
        2722  2724  2733  2735  2749  2751  2763  2772  2781  2784  2786  2794  2798  2800  2809
        2812  2833  2841  2854  2863  2865  2868  2870  2879  2881  2895  2897  2909  2918  2927
        2930  2932  2940  2944  2946  2955  2959  2979  2987  2998  3007  3009  3020  3022  3032
        3034  3043  3045  3054  3056  3065  3067  3076  3078  3087  3089  3098  3100  3109  3111
        3120  3122  3132  3134  3143  3145  3154  3156  3165  3167  3175  3179  3190  3205  3216
        3218  3229  3231  3241  3243  3253  3256  3260  3269  3310  3321  3330  3340  3342  3352
        3354  3365  3367  3369  3380  3382  3390  3410  3418  3429  3438  3440  3451  3453  3463
        3465  3474  3476  3485  3487  3496  3498  3506  3508  3517  3519  3528  3530  3537  3546
        3558  3560  3566  3568  3574  3576  3581  3583  3587  3591  3601  3617  3628  3630  3641
        3643  3653  3655  3665  3668  3672  3687  3690  3701  3703  3711  3713  3759  3770  3780
        3790  3792  3802  3804  3814  3816  3818  3829  3831  3837  3858  3865  3878  3886  3888
        3892  3894  3903  3905  3919  3921  3933  3942  3951  3954  3956  3965  3968  3970  3979
        3982  4002  4010  4021  4030  4032  4043  4045  4055  4057  4066  4068  4077  4079  4089
        4091  4100  4102  4113  4115  4124  4126  4135  4144  4157  4159  4165  4167  4173  4175
        4180  4182  4188  4197  4211  4213  4219  4221  4227  4229  4234  4236  4240  4247  4298
        4308  4345  4354  4380  4388  4524  4532  4558  4566  4616  4626  4671  4679  4726  4744
```

```
.PAGE     222     315
.PSECT     14
.REM        1      14      31     499     535     566     585     980    1032    1080    1146    1194    1264    1336
.REPT      61    8102    8168    8234    8300    8366    8432    8498    8564
.SBTTL     43      55      75     188     200     222     266     315     495     496     497     531     532     533     562
          563     564     588     598     607     612     617     622     627     634     648     661     705     891     896
          915     954     968     979    1030    1079    1145    1192    1263    1335    1413    1494    1495    1703    1707
         1869    2031    2228    2374    2520    2666    2812    2959    3175    3269    3390    3537    3587    3713    3837
         3982    4135    4188    4240    4298    4345    4380    4524    4558    4616    4671    4726    4839    4952    5065
         5178    5292    5367    5455    5652    5726    5730    5791    5868    5935    5987    6034    6099    6241    6320
         6358    6415    6459    7544    7567    7629    7918
.TITLE     31    7917
.WORD      61      62      63      65     196     216     217     218     219     220     221     230     233     234     235
          236     239     240     241     242     243     244     245     248     249     250     271     272     273     274
          275     276     277     278     282     283     284     297     301     304     307     308     309     310     311
          312     464     467     469     471     473     475     477     479     481     482     484     486     491     493
          529     560     572     573     574     575     576     577     578     579     580     581     582     591     594
          595     597     600     601     602     603     604     608     609     610     613     614     615     618     619
          620     623     624     625     628     629     630     636     638     640     642     644     646     651     652
          653     654     655     656     657     658     663     664     665     666     667     668     669     670     671
          672     676     677     678     679     680     681     682     684     685     694     695     697     698     699
          700     701     971    1435    1700    1701    1757    1772    1783    1785    1796    1798    1808    1810    1820
         1823    1827    1842    1845    1857    1859    1867    1919    1934    1945    1947    1958    1960    1970    1972
         1982    1985    1989    2004    2007    2019    2021    2029    2093    2103    2113    2115    2125    2127    2136
         2138    2149    2151    2160    2162    2172    2174    2176    2187    2189    2199    2201    2207    2217    2219
         2257    2270    2279    2281    2284    2286    2295    2297    2311    2313    2325    2334    2343    2346    2348
         2356    2360    2362    2371    2403    2416    2425    2427    2430    2432    2441    2443    2457    2459    2471
         2480    2489    2492    2494    2502    2506    2508    2517    2549    2562    2571    2573    2576    2578    2587
         2589    2603    2605    2617    2626    2635    2638    2640    2648    2652    2654    2663    2695    2708    2717
         2719    2722    2724    2733    2735    2749    2751    2763    2772    2781    2784    2786    2794    2798    2800
         2809    2841    2854    2863    2865    2868    2870    2879    2881    2895    2897    2909    2918    2927    2930
         2932    2940    2944    2946    2955    2987    2998    3007    3009    3020    3022    3032    3034    3043    3045
         3054    3056    3065    3067    3076    3078    3087    3089    3098    3100    3109    3111    3120    3122    3132
         3134    3143    3145    3154    3156    3165    3167    3190    3205    3216    3218    3229    3231    3241    3243
         3253    3256    3260    3321    3330    3340    3342    3352    3354    3365    3367    3369    3380    3382    3418
         3429    3438    3440    3451    3453    3463    3465    3474    3476    3485    3487    3496    3498    3506    3508
         3517    3519    3528    3530    3558    3560    3566    3568    3574    3576    3581    3583    3601    3617    3628
         3630    3641    3643    3653    3655    3665    3668    3672    3687    3690    3701    3703    3711    3770    3780
         3790    3792    3802    3804    3814    3816    3818    3829    3831    3865    3878    3886    3888    3892    3894
         3903    3905    3919    3921    3933    3942    3951    3954    3956    3965    3968    3970    3979    4010    4021
         4030    4032    4043    4045    4055    4057    4066    4068    4077    4079    4089    4091    4100    4102    4113
         4115    4124    4126    4157    4159    4165    4167    4173    4175    4180    4182    4211    4213    4219    4221
         4227    4229    4234    4236    5744    5747    5786    5867    6013    6018    6270    6317    6357    6390    6450
         6471    6970    7067    7087    7222    7288    7289    7296    7331    7341    7362    7363    7377    7387    7388
         7418    7422    7437    7476    7477    7482    7493    7494    7497    7498    7528    7529    7535    7536    7542
         7574    7747    7749    7752    7753    7762    7909    7910    7911    7914    8059    8067    8068    8069    8070
         8071    8072    8073    8074    8075    8079    8097    8098    8099    8102    8105    8108    8111    8114    8117
         8120    8123    8127    8138    8139    8144    8145    8148    8149    8151    8163    8164    8165    8168    8171
         8174    8177    8180    8183    8186    8189    8193    8204    8205    8210    8211    8214    8215    8217    8229
         8230    8231    8234    8237    8240    8243    8246    8249    8252    8255    8259    8270    8271    8276    8277
         8280    8281    8283    8295    8296    8297    8300    8303    8306    8309    8312    8315    8318    8321    8325
         8336    8337    8342    8343    8346    8347    8349    8361    8362    8363    8366    8369    8372    8375    8378
         8381    8384    8387    8391    8402    8403    8408    8409    8412    8413    8415    8427    8428    8429    8432
         8435    8438    8441    8444    8447    8450    8453    8457    8468    8469    8474    8475    8478    8479    8481
```

000000

ERRORS DETECTED: 0

```
*DRLPA,DRLPA/SOL/CRF=DRLPA.MAC,DRLPA
RUN-TIME:  40 34 5 SECONDS
CORE USED:  46K
```

```
                                        .LIST   MC,BIN,BEX,MEB
                                        .NLIST  MD,CND,ME

            177777                      ADDRESS=-1
                                ;       MACRO DEFFINITIONS FOR M800 AND M8204 MICR-PROCESSOR
                                ;       INSTRUCTION SET.
                                ;       TO BE USED WITH RSX MACRO-11 ASSEMBLER
                                ;
                                ;       26-MAY-1976
000000'                                 $BEGIN
000000                                  $LOC    64000
                                        .GLOBL  MRCODE,UCODEE,IMAGE
                                        .ENABL  GBL

                                        ;*
                                        ;*MICRO CODE FOR KMC-11
```

```
                                        ;*THIS CODE WILL BE DOWN LOADED INTO BOTH
                                        ;*KMC-11'S. THE CODE RUNS ASYNCRONOUS TO THE PDP-11 CODE
                                        ;*WE SYNC THROUGH COMMANDS PASSED VIA THE OUT*/IBUS* REGS.
                                        ;*


064000                          IMAGE:  ;JUMP TABLE USED FOR COMMANDS
064000                                  BR      STARTU          ;GOTO START
064000  100407                          .WORD   .$$$.
064002                                  BR      CMNOP           ;NOP=1
064002  100420                          .WORD   .$$$.
064004                                  BR      RDSILO          ;=2 READ SILO PUT IN BSEL4
064004  100430                          .WORD   .$$$.
064006                                  BR      WRSILO          ;=3 READ BSEL4 PUT IN SILO.
064006  100432                          .WORD   .$$$.
064010                                  BR      RDCMND          ;=4 READ FAST PATH PUT IN BSEL4
064010  100434                          .WORD   .$$$.
064012                                  BR      WRCMND          ;=5 READ BSEL4, PUT IN FAST PATH.
064012  100436                          .WORD   .$$$.
064014                                  BR      SAMP            ;=6 TAKE AN A/D SAMPLE
064014  100440                          .WORD   .$$$.

                                        ;START OF U CODED


064016                          STARTU:
064016                                  MOVE    # 0,BREG
064016  000400                          .WORD   .$$$.
064020                                  MOVE    BREG,OUT1 <0>   ;CLEAR UNIBUS CSRS
064020  061220                          .WORD   .$$$.
064022                                  MOVE    BREG,OUT1 <2>
064022  061222                          .WORD   .$$$.
064024                                  MOVE    BREG,OUT1 <3>
064024  061223                          .WORD   .$$$.
064026                                  MOVE    BREG,OUT1 <4>
064026  061224                          .WORD   .$$$.
064030                                  MOVE    BREG,OUT1 <5>
064030  061225                          .WORD   .$$$.
064032                                  MOVE    BREG,OUT1 <6>
064032  061226                          .WORD   .$$$.
064034                                  MOVE    BREG,OUT1 <7>
064034  061227                          .WORD   .$$$.
064036                                  MOVE    BREG,SPAD <6>
064036  063226                          .WORD   .$$$.

064040                          CMNOP:  MOVE    INP0 <12>,OUT1 <0> ;READ STATUS
064040  021240                          .WORD   .$$$.
064042                                  MOVE    # 377,BREG
064042  000777                          .WORD   .$$$.
064044                                  MOVE    BREG,OUT1 <2>   ;INDICATE READY FOR COMMAND.
064044  061222                          .WORD   .$$$.

064046                          LOOP:   MOVE    INP0 <12>,OUT1 <0> ;READ STATUS
064046  021240                          .WORD   .$$$.
```

```
                                        MOVE    INP1 <2>,SPAD <0> ;READ COMMAND REG.
064050                                  .WORD   .$$$.
064050  123040                         BZ      LOOP              ;NO COMMAND THEN LOOP
064052                                  .WORD   .$$$.
064052  101423
                                        MOVE    INP1 <2>,SPAD <0> ;RE-READ COMMAND.
064054                                  .WORD   .$$$.
064054  123040
                                        BR      SPAD <0>          ;BR BASED ON CMND.
064056                                  .WORD   .$$$.
064056  160600                                                    ;NO-USER PROTECTION OFFERED.
                                                                  ;IF YOU ENTER WRONG CODE -
                                                                  ;YOU LOSE.


                                        ;
                                        ;ROUTINE TO READ THE SILO, PUT IN
                                        ;*BUS REG 4
                                        ;CMD=2
                                        ;
                                RDSILO: MOVE    INP0 <10>,OUT1 <4> ;READ SILO.
064060                                  .WORD   .$$$.
064060  021204                                                    ;WRITE *BUS
                                        BR      CMNOP             ;RETURN.
064062                                  .WORD   .$$$.
064062  100420
                                        ;
                                        ;ROUTINE TO WRITE SILO, READ DATA FROM
                                        ;*BUS REG 4
                                        ;CMD=3
                                        ;
                                WRSILO: MOVE    INP1 <4>,OUT0 <10> ;READ DATA IN *BUS
064064                                  .WORD   .$$$.
064064  122110                                                    ;WRITE SILO.
                                        BR      CMNOP
064066                                  .WORD   .$$$.
064066  100420
                                        ;
                                        ;ROUTINE TO READ FAST PATH (CMND) REG.
                                        ;PUT IN *BUS REG 4
                                        ;CMD=4
                                        ;
                                RDCMND: MOVE    INP0 <11>,OUT1 <4> ;READ FAST PATH
064070                                  .WORD   .$$$.
064070  021224                                                    ;WRITE *BUS.
                                        BR      CMNOP             ;RETURN
064072                                  .WORD   .$$$.
064072  100420
                                        ;
                                        ;ROUTINE TO WRITE FAST PATH (CMND) REG.
                                        ;TAKE DATA FROM *BUS REG 4.
                                        ;CMD=5
```

```
                                        ;
                                WRCMND: MOVE    INP1 <4>,OUT0 <11> ;READ DATA IN *BUS
064074  122111                          .WORD   .$$$.
                                                        ;WRITE INTO FAST PATH.
                                        BR      CMNOP   ;RETURN.
064076  100420                          .WORD   .$$$.

                                        ;
                                        ;THIS ROUTINE TAKES AN A/D SAMPLE.
                                        ;CALL= CMND 6 IN BSEL2
                                        ;THESE REGS. MUST BE SET UP IN ADVANCE.
                                        ;BSEL 3 MUST CONTAIN READ CODE FOR A/D BUFFER.
                                        ;BSEL 4,5 MUST CONTAIN  A/D CSR SETTING.
                                        ;BSEL 6 MUST CONTAIN WRITE CODE FOR A/D CSR
                                        ;BSEL 7 MUST CONTAIN READ CODE FOR A/D CSR
                                        ; BSEL 3,6,7 WILL REMAIN  UNEFFECTED.
                                        ; BSEL 4,5 WILL CONTAIN A/D SAMPLE.
                                        ;BSEL2 WILL CONTAIN CODE 377 WHEN DONE.

                                        WTMC    SAMP
064100  020640                          .WORD   .$$$.
064102  103040                          .WORD   .$$$.
064104                                  MOVE    INP1 <6>,OUT0 <11>      ;SEND A/D WRITE CODE.
064104  122151                          .WORD   .$$$.
064106                                  WTMC    SAMP1
064106  020640                          .WORD   .$$$.
064110  103043                          .WORD   .$$$.
064112                                  MOVE    INP1 <4>,OUT0 <11>      ;SEND LOW BYTE CSR INFO.
064112  122111                          .WORD   .$$$.
064114                                  WTMC    SAMP2
064114  020640                          .WORD   .$$$.
064116  103046                          .WORD   .$$$.
064120                                  MOVE    INP1 <5>,OUT0 <11>      ;SEND HIGH BYTE CSR INFO.
064120  122131                          .WORD   .$$$.
064122                                  WTMC    SLOOP
064122  020640                          .WORD   .$$$.
064124  103051                          .WORD   .$$$.
064126                                  MOVE    INP1 <7>,OUT0 <11>      ;SEND READ CODE TO GET A/D CSR.
064126  122171                          .WORD   .$$$.
064130                                  WTMC    SAMP3
064130  020640                          .WORD   .$$$.
064132  103054                          .WORD   .$$$.
064134                                  WTMM    SLOOP1
064134  020640                          .WORD   .$$$.
064136  061620                          .WORD   .$$$.
064140  103056                          .WORD   .$$$.
064142                                  MOVE    INP0 <11>,BREG
064142  020620                          .WORD   .$$$.
064144                                  MOVE    BREG,SPAD <0>
064144  063220                          .WORD   .$$$.
064146                                  WTMM    SLOOP2
```

```
  181  064146  020640                .WORD   .$$$.
  182  064150  061620                .WORD   .$$$.
  183  064152  103063                .WORD   .$$$.
  184  064154                        MOVE    INP0 <11>,BREG
  185  064154  020620                .WORD   .$$$.
  186  064156                        BB7     CMNOP                   ;ABORT IF A/D BIT 15=1
  187  064156  103420                .WORD   .$$$.
  188  064160                        MOVE    SPAD <0>,BREG
  189  064160  060600                .WORD   .$$$.
  190  064162                        BB7     LOPE
  191  064162  103473                .WORD   .$$$.
  192  064164                        BR      SLOOP                   ;IF A/D NOT DONE,EXIT.
  193  064164  100451                .WORD   .$$$.
  194  064166                LOPE:   MOVE    INP1 <3>,OUT0 <11>      ;ISSUE READ A/B BUFFER.
  195  064166  122071                .WORD   .$$$.
  196  064170                        WTMM    SLOOP3
  197  064170  020640                .WORD   .$$$.
  198  064172  061620                .WORD   .$$$.
  199  064174  103074                .WORD   .$$$.
  200  064176                        MOVE    INP0 <11>,OUT1 <4>
  201  064176  021224                .WORD   .$$$.
  202  064200                        WTMM    SLOOP4
  203  064200  020640                .WORD   .$$$.
  204  064202  061620                .WORD   .$$$.
  205  064204  103100                .WORD   .$$$.
  206  064206                        MOVE    INP0 <11>,OUT1 <5>
  207  064206  021225                .WORD   .$$$.
  208  064210                        BR      CMNOP
  209  064210  100420                .WORD   .$$$.
  210  064212  177777                .WORD   -1
  211          000001                .END
```

```
ADDRES= 177777          CLK    = 000020          CMNOP   064040          IMAGE   064000  G
LOOP    064046          LOPE   064166          MARHLD= 000000          MARINC= 014000
MARLD  = 010000         MARLDX= 004000          MRCODE= ******  G       PAGE0  = 000000
PAGE1  = 001000         PAGE2  = 002000         PAGE3  = 003000         PC     =%000007
RDCMND  064070          RDSILO  064060          R0     =%000000         R1     =%000001
R2     =%000002         R3     =%000003         R4     =%000004         R5     =%000005
SAMP    064100          SAMP1   064106          SAMP2   064114          SAMP3   064130
SLOOP   064122          SLOOP1  064134          SLOOP2  064146          SLOOP3  064170
SLOOP4  064200          SP     =%000006         STARTU  064016          UCODEE= ******  G
WRCMND  064074          WRSILO  064064          $$$SER= 000001          .ADC   = 000100
.ADD   = 000000         .ADDWC= 000020          .AND   = 000260         .BB0   = 002000
.BB1   = 002400         .BB4   = 003000         .BB7   = 003400         .BC    = 001000
.BR    = 000400         .BSBRG= 160000          .BSIMM= 100000          .BSMEM= 140000
.BZ    = 001400         .CO    = 000400         .DBR   = 000400         .DBRSH= 001400
.DEC   = 000160         .DMEM  = 002400         .DNOP  = 000000         .DOUT0= 002000
.DOUT1= 001000          .DSPAD= 003000          .DSPBR= 003400          .DO    = 000400
.F0    = 000020         .INC   = 000060         .LORN  = 000240         .MINUS= 000360
.M0    = 004000         .OR    = 000300         .PLUS  = 000000         .SBREG= 060000
.SELA  = 000200         .SELB  = 000220         .SIMM  = 000000         .SIN0  = 020000
.SIN1  = 120000         .SMEM  = 040000         .SUB   = 000340         .SUBWC= 000040
.SUB2C= 000360          .S0    = 020000         .XOR   = 000320         .$$$.  = 100420
..LOC  = 064040         .2A    = 000120         .2AWC  = 000140         .      = 064214

        000000
ABCODE  004000      002

 ERRORS DETECTED:  0
```

```
*DRLPX0,DRLPX0/SOL=DRLPX0
RUN-TIME:  3 4 0 SECONDS
CORE USED:  6K
```

```
                        .LIST   MC,BIN,BEX,MEB
                        .NLIST  MD,CND,ME

                        .REM    %

                COPYRIGHT (C) 1975, 1976, 1977
                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 ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

                        %

                        .REM    %

                        THIS CODE WAS DEVELOPED FOR USE WITH THE
                        LPA-11 DIAGNOSTIC, BY EDWARD C. BADGER.

                        %

177777                  ADDRESS=-1
                        MACRO DEFFINITIONS FOR M800 AND M8204 MICR-PROCESSOR
                        INSTRUCTION SET.
                        TO BE USED WITH RSX MACRO-11 ASSEMBLER

                        26-MAY-1976
                THIS IS A MODIFIED VERSION OF THE MACROS WRITTEN BY W C BROWN
                MODIFIED FOR USE WITH MACY11 AND LNKX11.
                WARNING: "BAB" BITS TAKEN OUT OF BRANCH INSTRUCTIONS.

000000'                 $BEGIN
000000                  $LOC    65000
                        .GLOBL  MRCODE,UCODEE,KMDIAG
                        .ENABL  GBL

                        ;*
                        ;*MICRO CODE FOR KMC-11
065000          KMDIAG:
```

```
                                        ;
                                        ;THIS MICRO CODE WILL BE LOADED INTO THE
                                        ;KMC-11 VIA THE LPA-11 DIAG.
                                        ;ALL TESTS MUST BE EXECUTED IN ORDER BY THE
                                        ;PDP-11

                                        ;TEST#1
                                        ;       CRAM ONES WRITE TEST
                                        ;  ENTERED ON INIT.

065000                          TST1:   MOVE    # 377,BREG      ;GET ALL ONES
065000  000777                          .WORD   .$$$.
065002                                  MOVE    BREG,OUT1 <0>   ;PUT ALL ONES INTO CRAM
065002  061220                          .WORD   .$$$.
065004                                  MOVE    BREG,OUT1 <2>
065004  061222                          .WORD   .$$$.
065006                                  MOVE    BREG,OUT1 <3>
065006  061223                          .WORD   .$$$.
065010                                  MOVE    BREG,OUT1 <4>
065010  061224                          .WORD   .$$$.
065012                                  MOVE    BREG,OUT1 <5>
065012  061225                          .WORD   .$$$.
065014                                  MOVE    BREG,OUT1 <6>
065014  061226                          .WORD   .$$$.
065016                                  MOVE    BREG,OUT1 <7>
065016  061227                          .WORD   .$$$.

065020                          A1:     MOVE    INP1 <0>,BREG   ;WHEN PDP-11 DONE VERIFICATION
065020  120400                          .WORD   .$$$.
065022                                  BZ      A1              ;IT WILL CLEAR CARM ADDR. #0.
065022  101410                          .WORD   .$$$.

                                        ;TEST #2
                                        ;       CRAM ZEROS WRITE TEST
                                        ;
                                        ;       ENTERED WHEN PDP-11 ZEROS CRAM ADDR. 0
                                        ;

065024                          TST2:   MOVE    # 0,BREG        ;GET ZERO
065024  000400                          .WORD   .$$$.
065026                                  MOVE    BREG,OUT1 <0>   ;PUT INTO ALL CRAM
065026  061220                          .WORD   .$$$.
065030                                  MOVE    BREG,OUT1 <2>
065030  061222                          .WORD   .$$$.
065032                                  MOVE    BREG,OUT1 <3>
065032  061223                          .WORD   .$$$.
065034                                  MOVE    BREG,OUT1 <4>
065034  061224                          .WORD   .$$$.
065036                                  MOVE    BREG,OUT1 <5>
065036  061225                          .WORD   .$$$.
065040                                  MOVE    BREG,OUT1 <6>
065040  061226                          .WORD   .$$$.
065042                                  MOVE    BREG,OUT1 <7>
065042  061227                          .WORD   .$$$.
```

```
                                                ;
                                                ;TEST #3
                                                ;       BRANCH TEST
                                                ;
                                                ;       ENTERED WHEN PDP-11 RETURNS 377 TO ADDR. 0
                                                ;
                                                ;NOTE WE HAVE TO ASSUME BR AND BZ WORK
                                                ;OR WE COULDN'T HAVE GOT THIS FAR.
                                                ;UPON SUCCESSFUL COMPLETION OF THIS TEST WE
                                                ;PUT CODE 377 INTO CRAM ADDR. 2
                                                ;AND 0 INTO CRAM ADDR. 0. ON FAILURE
                                                ;WE ONLY ZERO CRAM ADDR. 0.
                                                ;
065052                          TST3:   MOVE    # 376,BREG      ;GET ZERO BIT 0
065052  000776                          .WORD   .$$$.
065054                                  BBO     T3E             ;ERROR IF BR.
065054  102062                          .WORD   .$$$.
065056                                  MOVE    # 375,BREG      ;BIT 1 = 0
065056  000775                          .WORD   .$$$.
065060                                  BB1     T3E             ;ERROR IF BR
065060  102462                          .WORD   .$$$.
065062                                  MOVE    # 357,BREG      ;BIT 4 = 0
065062  000757                          .WORD   .$$$.
065064                                  BB4     T3E             ;ERROR IF BR
065064  103062                          .WORD   .$$$.
065066                                  MOVE    # 177,BREG      ;BIT 7 = 0
065066  000577                          .WORD   .$$$.
065070                                  BB7     T3E             ;ERROR IF BR
065070  103462                          .WORD   .$$$.
065072                                  MOVE    # 1,BREG        ;POSITIVE BR TEST
065072  000401                          .WORD   .$$$.
065074                                  BBO     A3              ;SHOULD BRANCH
065074  102040                          .WORD   .$$$.
065076                                  BR      T3E             ;IF NOT, ERROR
065076  100462                          .WORD   .$$$.
065100                          A3:     MOVE    # 2,BREG        ;BR ON BIT 1
065100  000402                          .WORD   .$$$.
065102                                  BB1     B3              ;SHOULD BR.
065102  102443                          .WORD   .$$$.
065104                                  BR      T3E             ;IF NOT, ERROR.
065104  100462                          .WORD   .$$$.
065106                          B3:     MOVE    # 20,BREG       ;BR ON BIT 4
065106  000420                          .WORD   .$$$.
065110                                  BB4     C3              ;SHOULD BR.
065110  103046                          .WORD   .$$$.
065112                                  BR      T3E             ;IF NOT, ERROR.
```

```
162  065112  100462          .WORD   .$$$.
163  065114          C3:     MOVE    # 200,BREG      ;BR IF BIT 7
164  065114  000600          .WORD   .$$$.
165  065116                  BB7     D3              ;SHOULD BR.
166  065116  103451          .WORD   .$$$.
167  065120                  BR      T3E             ;IF NOT, ERROR.
168  065120  100462          .WORD   .$$$.
169  065122          D3:     MOVE    # 0,BREG        ;TEST BZ NEGATIVE
170  065122  000400          .WORD   .$$$.
171  065124                  BZ      T3E             ;ERROR IF BR.
172  065124  101462          .WORD   .$$$.
173  065126                  MOVE    # 377,BREG      ;POS BR
174  065126  000777          .WORD   .$$$.
175  065130                  BZ      E3              ;SHOULD BR
176  065130  101456          .WORD   .$$$.
177  065132                  BR      T3E             ;ERROR IF NOT
178  065132  100462          .WORD   .$$$.
179  065134          E3:
180  065134                  MOVE    BREG,OUT1 <2>   ;PUT 377 IN OUTPUT REG#2
181  065134  061222          .WORD   .$$$.
182  065136                  BR      F3              ;IF BR INSTR FAILS TO WORK,
183  065136  100462          .WORD   .$$$.
184  065140                  MOVE    # 0,BREG        ;THEN A ZERO GETS PUT IN #2
185  065140  000400          .WORD   .$$$.
186  065142                  MOVE    BREG,OUT1 <2>   ;A SIGN OF AN ERROR.
187  065142  061222          .WORD   .$$$.
188  065144          F3:
189  065144          T3E:    MOVE    # 0,BREG        ;SIGNAL PDP-11 THAT WE ARE
190  065144  000400          .WORD   .$$$.
191  065146                  MOVE    BREG,OUT1 <0>   ;THROUGH BR-TEST.
192  065146  061220          .WORD   .$$$.
193
194                          ;
195                          ;TEST #4
196                          ;
197                          ;       ALU TEST
198                          ;
199                          ;       ENTERED WHEN PDP-11 PUTS A CODE 377 INTO
200                          ;       CRAM ADDR 0
201                          ;       ADDR 2 = 0 IF BAD
202                          ;
203  065150          TST4:   MOVE    INP1 <0>,BREG   ;GET CRAM ADDR 0
204  065150  120400          .WORD   .$$$.
205  065152                  BZ      A4              ;WHEN = 377 DO TEST.
206  065152  101467          .WORD   .$$$.
207  065154                  BR      TST4
208  065154  100464          .WORD   .$$$.
209
210  065156          A4:     MOVE    # 0,BREG        ;GET = 0
211  065156  000400          .WORD   .$$$.
212  065160                  MOVE    BREG,OUT1 <2>
213  065160  061222          .WORD   .$$$.
214  065162                  MOVE    BREG,SPAD <4>
215  065162  063224          .WORD   .$$$.
```

```
                                        DEC     SPAD <4>        ;MAKE = 377
065164                                  .WORD   .$$$.!.DSPAD
065164  063164
065166                                  BZ      B4              ;BR IF GOOD SEC.
065166  101475                         .WORD   .$$$.
065170                                  BR      T4E
065170  100503                         .WORD   .$$$.
065172                          B4:     DEC     SPAD <4>        ;SEC = 376
065172  063164                         .WORD   .$$$.!.DSPAD
065174                                  INC     SPAD <4>        ;INC = 377
065174  063064                         .WORD   .$$$.!.DSPAD
065176                                  BZ      C4              ;BR IF = 377
065176  101501                         .WORD   .$$$.
065200                                  BR      T4E
065200  100503                         .WORD   .$$$.
065202                          C4:     MOVE    # 377,BREG
065202  000777                         .WORD   .$$$.
065204                                  MOVE    BREG,OUT1 <2>   ;RETURN GOOD CODE
065204  061222                         .WORD   .$$$.

065206                          T4E:    MOVE    # 0,BREG        ;RETURN CODE 0
065206  000400                         .WORD   .$$$.
065210                                  MOVE    BREG,OUT1 <0>
065210  061220                         .WORD   .$$$.

                                        ;
                                        ;TEST 5
                                        ;
                                        ;       NPR TEST
                                        ;
                                        ;       ENTERED IN LINE WHEN PDP-11 PUTS CODE 377 INTO
                                        ;       CRAM ADDR. 0.
                                        ;       DOES AN INPUT DATA XFER FROM LOC 1124
                                        ;       IN PDP-11 MEM TO LOC 1126
                                        ;       DOES NO DATA CHECKING RETURNS
                                        ;       CODE 0 TO CRAM ADDR. 0 WHEN DONE.
                                        ;
065212                          TST5:   MOVE    INP1 <0>,BREG   ;WAIT FOR PDP-11
065212  120400                         .WORD   .$$$.
065214                                  BZ      A5
065214  101510                         .WORD   .$$$.
065216                                  BR      TST5
065216  100505                         .WORD   .$$$.

065220                          A5:     MOVE    # 2,BREG        ;SET TO DO NPR IN
065220  000402                         .WORD   .$$$.
065222                                  MOVE    BREG,OUT0 <5>   ;SET HIGH ADDR.
065222  062225                         .WORD   .$$$.
065224                                  MOVE    # 124,BREG      ;GET LOW ADDR. OF ADDR. 1124
065224  000524                         .WORD   .$$$.
065226                                  MOVE    BREG,OUT0 <4>   ;SET LOW ADDR.
065226  062224                         .WORD   .$$$.
065230                                  MOVE    # 1,BREG        ;SET TO nPR IN
065230  000401                         .WORD   .$$$.
```

```
                                        MOVE    BREG,OUT1 <10>  ;CAUSE NPR IN
065232  061230                          .WORD   .$$$.
065232                          B5:     MOVE    INP1 <10>,BREG  ;WAIT FOR NPR DONE.
065234  120600                          .WORD   .$$$.
065234                                  BBO     B5
065236  102116                          .WORD   .$$$.

065240                                  MOVE    INP0 <0>,BREG   ;GET LOW BYTE DATA.
065240  020400                          .WORD   .$$$.
065242                                  MOVE    BREG,OUT0 <2>   ;OUTPUT NPR DATA REG.
065242  062222                          .WORD   .$$$.
065244                                  MOVE    INP0 <1>,BREG   ;GET HIGH BYTE.
065244  020420                          .WORD   .$$$.
065246                                  MOVE    BREG,OUT0 <3>   ;SAVE FOR OUTPUT.
065246  062223                          .WORD   .$$$.
065250                                  MOVE    # 2,BREG        ;GET OUTPUT ADDR. = 1126
065250  000402                          .WORD   .$$$.
065252                                  MOVE    BREG,OUT0 <7>
065252  062227                          .WORD   .$$$.
065254                                  MOVE    # 126,BREG      ;LOW PART
065254  000526                          .WORD   .$$$.
065256                                  MOVE    BREG,OUT0 <6>
065256  062226                          .WORD   .$$$.
065260                                  MOVE    INP1 <11>,SPAD <4>      ;/-STN-
065260  123224                          .WORD   .$$$.
065262                                  AND     SPAD <4>,BREG,SPAD
065262  063264                          .WORD   .$$$.
065264                                  MOVE    # 0,BREG
065264  000400                          .WORD   .$$$.
065266                                  OR      SPAD <4>,BREG,BREG
065266  060704                          .WORD   .$$$.
065270                                  MOVE    BREG,OUT1 <11>
065270  061231                          .WORD   .$$$.
065272                                  MOVE    # 21,BREG
065272  000421                          .WORD   .$$$.
065274                                  MOVE    BREG,OUT1 <10>  ;SET NPR OUT.
065274  061230                          .WORD   .$$$.

065276                          C5:     MOVE    INP1 <10>,BREG  ;WAIT TILL DONE
065276  120600                          .WORD   .$$$.
065300                                  BBO     C5
065300  102137                          .WORD   .$$$.

065302                                  MOVE    # 0,BREG        ;TELL PDP-11 WE'RE DONE.
065302  000400                          .WORD   .$$$.
065304                                  MOVE    BREG,OUT1 <0>
065304  061220                          .WORD   .$$$.


                                        ;TEST 6
                                        ;
                                        ;       NPR TEST
                                        ;
                                        ;       ENTERED IN LINE WHEN PDP-11 PUTS CODE 377 INTO
```

```
                                        ;       CRAM ADDR. 0.
                                        ;       DOES AN INPUT DATA XFER FROM LOC 1124
                                        ;       IN PDP-11 MEM TO 10C 1126
                                        ;       DOES NO DATA CHECKING RETURNS
                                        ;       CODE 0 TO CRAM ADDR. 0 WHEN DONe.

065306                          TST6:   MOVE    INPi <0>,BREG   ;WAIT FOR PDP-11
065306  120400                          .WORD   .$$$.
065310                                  BZ      A6
065310  101546                          .WORD   .$$$.
065312                                  BR      TST6
065312  100543                          .WORD   .$$$.

065314                          A6:     MOVE    # 2,BREG        ;SET TO DO NPR IN
065314  000402                          .WORD   .$$$.
065316                                  MOVE    BREG,OUT0 <5>   ;SET HIGH ADDR.
065316  062225                          .WORD   .$$$.
065320                                  MOVE    # 124,BrEG      ;GET LOW ADDR. OF ADDR. 1124
065320  000524                          .WORD   .$$$.
065322                                  MOVE    BREG,OUT0 <4>   ;SET LOW ADDR.
065322  062224                          .WORD   .$$$.
065324                                  MOVE    # 1,BREG        ;SET TO NPR IN
065324  000401                          .WORD   .$$$.
065326                                  MOVE    BREG,OUT1 <10>  ;CAUSE NPR IN
065326  061230                          .WORD   .$$$.
065330                          B6:     MOVE    INP1 <10>,BREG  ;WAIT FOR NPR DONE.
065330  120600                          .WORD   .$$$.
065332                                  BB0     B6
065332  102154                          .WORD   .$$$.

065334                                  MOVE    INP0 <0>,BREG   ;GET LOW BYTE DATA.
065334  020400                          .WORD   .$$$.
065336                                  MOVE    BREG,OUT0 <2>   ;OUTPUT NPR DATA REG.
065336  062222                          .WORD   .$$$.
065340                                  MOVE    INP0 <1>,BREG   ;GET HIGH BYTE.
065340  020420                          .WORD   .$$$.
065342                                  MOVE    BREG,OUT0 <3>   ;SAVE FOR OUTPUT.
065342  062223                          .WORD   .$$$.
065344                                  MOVE    # 2,BREG        ;GET OUTPUT ADDR. = 1126
065344  000402                          .WORD   .$$$.
065346                                  MOVE    BREG,OUT0 <7>
065346  062227                          .WORD   .$$$.
065350                                  MOVE    # 126,BREG      ;LOW PART
065350  000526                          .WORd   .$$$.
065352                                  MOVE    BREG,OUT0 <6>
065352  062226                          .WORD   .$$$.
065354                                  MOVE    INP1 <11>,SPAD <4>      ;/-STN-
065354  123224                          .WORD   .$$$.
065356                                  AND     SPAD <4>,BREG,SPAD
065356  063264                          .WORD   .$$$.
065360                                  MOVE    # 0,BREG
065360  000400                          .WORD   .$$$.
065362                                  OR      SPAD <4>,BREG,BREG
065362  060704                          .WORD   .$$$.
```

```
                                        MOVE    BREG,OUT1 <11>  ;SET NPR OUT.
065364 061231                           .WORD   .$$$.
065366                                  MOVE    # 21,BREG
065366 000421                           .WORD   .$$$.
065370                                  MOVE    BREG,OUT1 <10>
065370 061230                           .WORD   .$$$.

065372                          C6:     MOVE    INP1 <10>,BREG  ;WAIT TILL DONE
065372 120600                           .WORD   .$$$.
065374                                  BBO     C6
065374 102175                           .WORD   .$$$.

065376                                  MOVE    # 0,BREG        ;TELL PDP-11 WE'RE DONE.
065376 000400                           .WORD   .$$$.
065400                                  MOVE    BREG,OUT1 <0>
065400 061220                           .WORD   .$$$.

                                        ;TEST 7
                                        ;
                                        ;       INTERRUPT TEST
                                        ;
                                        ;       THIS TEST GENERATES TWO INTERRUPTS
                                        ;
                                        ;       THE 1ST INTERRUPT TO VECTOR XX0 WHEN
                                        ;       CRAM ADDR. 0 = 377 BY PDP-11 CONTROL.
                                        ;       THE 2ND INTERRUPT TO VECTOR XX4 WHEN
                                        ;       CRAM ADDR. 0 = 377 (1ST MODE = 0 AGAIN).
                                        ;
065402                          TST7:   MOVE    INP1 <0>,BREG   ;WAIT TILL PDP-11 READY!
065402 120400                           .WORD   .$$$.
065404                                  BZ      A7
065404 101604                           .WORD   .$$$.
065406                                  BR      TST7
065406 100601                           .WORD   .$$$.

065410                          A7:     MOVE    INP1 <11>,SPAD <4>
065410 123224                           .WORD   .$$$.
065412                                  AND     SPAD <4>,BREG,SPAD
065412 063264                           .WORD   .$$$.
065414                                  MOVE    # 200,BREG
065414 000600                           .WORD   .$$$.
065416                                  OR      SPAD <4>,BREG,BREG
065416 060704                           .WORD   .$$$.
065420                                  MOVE    BREG,OUT1 <11>  ;SET INTR TO ADDR. XX0
065420 061231                           .WORD   .$$$.
065422                          B7:     MOVE    INP1 <11>,BREG  ;WAIT TILL DONE
065422 120620                           .WORD   .$$$.
065424                                  BB7     B7
065424 103611                           .WORD   .$$$.

065426                                  MOVE    # 0,BREG        ;TELL PDP-11 WE INTERRUPTED
065426 000400                           .WORD   .$$$.
065430                                  MOVE    BREG,OUT1 <0>   ;IN CASE WE DIDN'T
065430 061220                           .WORD   .$$$.
```

```
                                C7:     MOVE    INP1 <0>,BREG   ;NOW WAIT FOR PDP-11 TO tELL
065432  120400                          .WORD   .$$$.
065434                                  BZ      D7              ;US TO iNTR. TO VECTOR XX4
065434  101620                          .WORD   .$$$.
065436                                  BR      C7
065436  100615                          .WORD   .$$$.

065440                          D7:     MOVE    INP1 <11>,SPAD <4>
065440  123224                          .WORD   .$$$.
065442                                  AND     SPAD <4>,BREG,SPAD
065442  063264                          .WORD   .$$$.
065444                                  MOVE    # 300,BREG
065444  000700                          .WORD   .$$$.
065446                                  OR      SPAD <4>,BREG,BREG
065446  060704                          .WORD   .$$$.
065450                                  MOVE    BREG,OUT1 <11>  ;SET INTR TO ADDR. XX4
065450  061231                          .WORD   .$$$.

065452                          E7:     MOVE    INP1 <11>,BREG  ;WAIT TILL DONE.
065452  120620                          .WORD   .$$$.
065454                                  BB7     E7
065454  103625                          .WORD   .$$$.

065456                                  MOVE    # 0,BREG        ;TELL PDP-11 WE THOUGHT
065456  000400                          .WORD   .$$$.
065460                                  MOVE    BREG,OUT1 <0>   ;WE HAD INTERRUPTED!
065460  061220                          .WORD   .$$$.

065462                                  BR      .
065462  100631                          .WORD   .$$$.


065464  177777                  .WORD   -1
        000001                  .END
```

```
ADDRES= 177777          A1      065020          A2      065044          A3      065100
A4      065156          A5      065220          A6      065314          A7      065410
B3      065106          B4      065172          B5      065234          B6      065330
B7      065422          CLK   = 000020          C3      065114          C4      065202
C5      065276          C6      065372          C7      065432          D3      065122
D7      065440          E3      065134          E7      065452          F3      065144
KMDIAG  065000  G       MARHLD= 000000          MARINC= 014000          MARLD  = 010000
MARLDX= 004000          MRCODE= ******  G       PAGE0  = 000000          PAGE1  = 001000
PAGE2  = 002000         PAGE3  = 003000         PC     =%000007          R0     =%000000
R1     =%000001         R2     =%000002         R3     =%000003          R4     =%000004
R5     =%000005         SP     =%000006         TST1    065000          TST2    065024
TST3    065052          TST4    065150          TST5    065212          TST6    065306
TST7    065402          T3E     065144          T4E     065206          UCODEE= ******  G
$$$SER= 000001          .ADC   = 000100          .ADD   = 000000          .ADDWC= 000020
.AND   = 000260         .BBO   = 002000          .BB1   = 002400          .BB4   = 003000
.BB7   = 003400         .BC    = 001000          .BR    = 000400          .BSBRG= 160000
.BSIMM= 100000          .BSMEM= 140000          .BZ    = 001400          .CO    = 000400
.DBR   = 000400         .DBRSH= 001400          .DEC   = 000160          .DMEM  = 002400
.DNOP  = 000000         .DOUiO= 002000          .DOUT1= 001000          .DSPAD= 003000
.DSPBR= 003400          .DO    = 000400          .FO    = 000020          .INC   = 000060
.LORN  = 000240         .MINUS= 000360          .MO    = 004000          .OR    = 000300
.PLUS  = 000000         .SBREG= 060000          .SELA  = 000200          .SELB  = 000220
.SIMM  = 000000         .SINO  = 020000         .SIN1  = 120000          .SMEM  = 040000
.SUB   = 000340         .SUBWC= 000040          .SUB2C= 000360          .SO    = 020000
.XOR   = 000320         .$$$.  = 100631         ..LOC  = 065462          .2A    = 000120
.2AWC  = 000140         .      = 065466

       000000
ABCODE 004000      002
```

ERRORS DETECTED: 0

```
 *DRLPX1,DRLPX1/SOL=DRLPX1
 RUN-TIME:  8  8  0 SECONDS
 CORE USED:  6K

W322    000000

LNKX11   V023   13-DEC-77   13:01

*DRLPA.LDA/B:0,DRLPA.MAP=DRLPA,DRLPX0,DRLPX1/E


LOAD MAP

IDENT:  LPA.03

TRANSFER ADDRESS: 000001
LOW LIMIT: 000000
HIGH LIMIT: 004000
**********
MODULE  DMDT
SECTION ENTRY   ADDRESS SIZE
<. ABS.>        000000  000000
        D.OCQ   054247
        D.OCS   054224
        D.OeQ   054201
        D.OES   054156
        D.TCQ   054363
        D.TCQP  054477
        D.TCS   054340
        D.TCSP  054454
        D.TEQ   054315
        D.TEQP  054431
        D.TES   054272
        D.TESP  054406
        FRECOR  056246
        IMAGE   064000
        KMDIAG  065000
<       >       000000  000000
**********
MODULE  DRLPX0
SECTION ENTRY   ADDRESS SIZE
<       >       000000  000000
<ABCODE>        000000  004000
**********
MODULE  DRLPX1
SECTION ENTRY   ADDRESS SIZE
<       >       004000  000000
**********
UNDEFINED REFERENCES

DRLPX0
DRLPX1
DRLPX2
MRCODE
UCODEE
```

E[illegible]SE[illegible] SECONDS 00010000 780223 PDP10 411