A LABORATORY CONTROL SYSTEM

USING THE PDP 11/03 MICROPROCESSOR

by

David A. Lockner

John D. Weeks

and

James D. Byerlee

Open File Report #78-125

U.S. Geological Survey
345 Middlefield Road
Menlo Park, CA 94025

LABORATORY CONTROL SYSTEM

Contents

I. INTRODUCTION

The ongoing trend in the miniturization of integrated circuits has brought us to a new era in laboratory experimentation. The microprocessor will soon play a major role in expanding the capabilities of a sophisticated laboratory. The lowered cost, increased speed and smaller size of microprocessors now make them available to the experimenter as a valuable analytic tool.

This report describes what the authors consider to be a state-of-the-art controller system based on the Digital Equipment Corp. PDP11/03 microprocessor. The system is designed to monitor and control in real time four independent high pressure rock-testing machines. In addition, the processor is capable of analyzing non-time-critical data at the same time that it is controlling experiments.

A brief description of the decisions we moved through in choosing the PDP 11/03 may prove of value to those currently searching the microprocessor market for the first time. When it became apparent that the analog controllers we had been using to operate our high-pressure machines were no longer adaquate to meet the increasing demands of the experiments being performed, we were faced with the decision of making the new system analog or digital. A digital system seemed more suited to our particular applications for a number of reasons. First, over the long time periods that experiments would run (days to weeks) a digital system should be less susceptible to

drift. Also, with a competent programer, a digital system should be far more versatile and easier to upgrade than a dedicated analog system.

After surveying the digital market, the DEC PDP11/03 seemed the best choice. DEC offered, at a reasonable price, a complete system called DECLAB, specifically designed for laboratory operation. This included the processor, terminal, dual floppy disk drives, 16K-words memory, 12 bit, 16 channel A-D converter, programmable clock and supporting software (including FORTRAN IV and BASIC compilers).

In the remainder of this report we will describe the various components of the controller system and how they interface to produce a powerful and versatile controller for laboratory experiments.

2. OVERVIEW

The primary task of the controller is to monitor and adjust pressures on four separate triaxial rock presses. This is done by turning on and off hydraulic pumps and drains on each of the machines. The processor receives information about the various pressures by digitizing the output of simple strain gauge bridges. While this primary function is being performed, the processor is also able to execute a second, independent program that analyzes data in a batch-like mode.

The current system configuration is shown in Figure 1. Strain gauge bridges are used to measure pressures on the four rock testing machines. The bridge outputs are amplified up to 66 db by high-precision, high-stability differential amplifiers designed by one of the authors (JDW). These high-level signals ($\pm$ 10V) are sent through low-pass filters with 20 Hz cutoff frequencies to a 32 channel, 12 bit, differential A-D converter. The digitized inputs are sampled by the computer once every 2 seconds.

The computer program compares the AD inputs to setpoints entered by the operator. If the setpoint deviation on any channel exceeds the dead band (determined in the program) the appropriate pump or drain is turned on to adjust pressure. TTL level outputs from the computer are sent to a buffered relay-driver unit also designed by one author (JDW). This turns on or off the appropriate relays on each machine. These in turn activate the pumps and drains.

Laboratory Controller Routine (LCR) is a FORTRAN program written by another of the authors (DAL) to control the rock-testing machines. LCR is an interactive program designed for ease of use and flexability. It continually

updates parameters associated with each channel to compensate for changes in response characteristics of each rock press.

LCR currently uses about 30 percent of the available CPU time. The rest can be utilized by a batch-type job which has access to a number of peripheral devices including two 9-track tape drives (a third one can be used by LCR to output AD readings), two floppy disk drives, the console terminal and a line printer/plotter. This gives the processor tremendous potential for acquiring and processing laboratory data.

3. ANALOG INPUTS

3.1 SOURCE TRANSDUCERS

Source transducers are of two general types: pressure transducers and displacement transducers. Pressure transducers are all strain gauge bridges giving balanced impedance outputs isolated from the system ground. Full scale range of the bridges is about 10 mV. Depending on the setup used, a full scale reading can represent a pressure change between 500 bars and 20,000 bars (1 bar = 14.5 psi). To attain the desired 0.05 percent accuracy, voltage changes of $\pm$ 5 μV must be detected. Since the AD converter will only detect changes of a few millivolts, we designed a high gain, low drift DC amplifier to boost the strain gauge output. This is described in section 3.2.

The displacement transducers give full scale balanced output of 5 VDC corresponding to a displacement of approximately 5 cm. Thus these signals may be fed directly into the AD converter without amplification.

3.2 PREAMPLIFIERS

In order to meet the design specifications of 0.05 per cent accuracy, we were required to design a high gain, low drift DC amplifier. This was accomplished using a two-stage design based on the Precision Monolithics OP-07 integrated circuit operational amplifier (Fig. 2). Because the OP-07 op-amps feature extremely low input offset voltage and temperature drift, fundamental circuits using premium-grade components were adequate.

The first stage is a differential amplifier having a fixed gain of 25 and input impedance of 20 kΩ. The gain setting resistors are .1% metal-film resistors chosen in pairs matched to within .01% in order to maintain good common-mode rejection. The second stage consists of a non-inverting amplifier in parallel with an inverting amplifier providing differential outputs with switch-selectable gain between 40 and 66 dB. A .0047 μf capacitor on the first stage acts as a 70 Hz low-pass filter. The low drift characteristics of the op-amps plus resistors having temperature coefficients of ± 25 ppm per degree centigrade give the desired long term .05 percent accuracy. Six of these amplifiers are built on a single PC board and provide the microprocessor with six high-level analog signals from each machine.

3.3 LOWPASS FILTERS

A total of 24 differential analog inputs (six from each rock testing machine) are supplied to the computer. To reduce AC noice picked up between the preamplifiers and the computer (the longest run is 20 meters), input signals are sent on doubly shielded, twisted pair wires. In addition, each input is run through an RC type low pass filter with 20 Hz cutoff frequency (Fig 3).

3.4 AD CONVERTER

The analog-to-digital convertor was designed by ADAC Corporation to be compatible with the LSI-11 bus. Input signals in the range of -10V to + 10V are converted to a 12-bit unsigned integer, giving a resolution of 1 part in

4096 or about .025%. There are 64 multiplexed inputs which can be configured in either 64 channels reference to local ground or 32 channels with differential inputs. Conversions can be performed under program control, or on command by an external clock.

4 PDP 11/03 MICROPROCESSOR

The LSI-11 microprocessor is the smallest of the PDP11 family. It is a 16 bit machine using the same instruction set as the larger PDP-11's. A large number of peripheral devices have been designed to be compatible with it's bus structure in addition to a massive amount of software developed by Digital Equipment Corp. (DEC) as well as other groups, to support it.

Hardware currently on our system includes (Fig 1):

1) program controlled analog to digital converter
2) programable clock
3) VT55 video terminal w/hard copy
4) video display screen
5) dual floppy disk drives
6) 3 9-trk tape drives
7) 2 16-bit DRV11 parallel interfaces (for driving servo-control relays)
8) 28K words MOS memory
9) Line printer/plotter

The available software includes:

1) RT-11 operating system
2) PIP- a mass storage maintenance program
3) a librarian
4) MACRO assembler
5) Basic
6) FORTRAN
7) linkage editor (including an option to create overlay structures).
8) scientific subroutine package.
9) numerous other programs including dump and debugging routines.

The low cost, small size and high speed of the LSI-11 make it well suited for laboratory applications. Another interesting feature that contributes to the power of the LSI-11 is that all periferals and even the CPU registers are address selectable so that all devices look like memory addresses. Peripherals are placed along the bus in a daisy-chained grant structure. Thus the bus could be thought of as the heart of the system, with the CPU essentially being another peripheral. This bus structure along with vectored interrupts allows for fast response without polling of peripherals. Because peripherals can run asynchronously, each component is allowed to run at its fastest speed, thereby increasing the overall system speed.

The foreground/background monitor provided as part of the RT-11 operating system is particularly useful for our application. This feature of RT-11 allows two jobs to run simultaneously, one being the foregound, real-time, job and the other the background job. In this configuration, the foreground job always has priority and has control of the CPU whenever it needs to do calculations. As soon as the foreground job becomes idle, it relinquishes control to the background job but can interrupt the background job at any time. In our application, Laboratory Controller Routine (LCR) is run as a foreground job. This program (described in section 6) samples the strain gauge inputs from the rock testing machines in the lab and controls the pressures during experiments. It must run in real time, sampling the inputs every two seconds. LCR currently uses between 20 and 30 percent of the CPU time, depending on how many experiments are running. This leaves at least 70 percent of the CPU time available for running background jobs that are not time-critical. Thus, while experiments are running, either batch-type jobs can be run that analyze large amounts of data or the operator can do program development. This allows for efficient use of the computer's resources.

5 OUTPUTS

Computer generated relay controlling signals are sent to the relay driver unit through two DRV-11 16-bit parallel output modules, operated by FORTRAN callable routines. The relay driver accepts the 32 TTL logic signals from the DRV modules and converts them to 12 volt CMOS logic levels. CMOS logic was chosen because the high voltage levels provide greater noise immunity than TTL logic. Noise is potentially a severe problem due to our long cable runs, the longest of which is 20 meters. The schematic diagram of the relay driver is shown in Fig. 4.

One of the 32 control inputs is pulsed during each 2-second cycle of the controller program, triggering a one-shot multivibrator, which then remains on for a period of three seconds. The output of the one-shot drives one of two inputs on each of 30 NAND gates, the other input of each gate being driven by one of the control inputs. Thus, no outputs can be activated unless the one-shot is retriggered on every cycle of the program. In the event of a computer or software malfunction that could leave some of the control outputs on, this feature shuts off all hydraulic servos within three seconds, preventing dangerous uncontrolled increases in pressure.

The outputs of the NAND gates drive pairs of high current inverters, wired in series, that supply current to either end of a dual-color light-emitting diode. The LED's operate as three-state function indicators: red indicates that the particular relay is off, green indicates that it is on, and an unlit lamp indicates either absence of control current, or a malfunction in the circuit. The inverter that drives the green half of the LED also supplies current to the outputs of the relay driver unit to control the servo relays.

The relays themselves are solid state, optically isolated, zero crossing switches. The optical isolation provides complete separation of the driver from the solenoids and motors on the rock-testing machines, while the zero-crossing feature prevents current surges that could be coupled to the sensitive transducers.

6 CONTROLLER PROGRAM

The Laboratory Controller Routine, LCR, is an interactive program designed to control simultaneously four high pressure rock testing machines in our laboratory. It was designed to provide a user, having little prior knowledge of computers, with an easy and versatile means of running experiments in the lab. The program is written in FORTRAN. A source listing appears in Appendix A.

The first step in LCR is to sample the analog inputs from displacement and pressure transducers on up to 24 channels. These are compared every two seconds to user specified set points. When the set point deviation becomes too large, the program turns on output channels connected to servo-controlled pumps and drains to adjust the appropriate pressure or displacement.

A number of commands, typed on the console terminal, provide user control of the program. The main commands are as follows:

HALT	Turns off all pumps and stops execution of LCR until a RUN command is issued.
RUN	Begins execution of LCR. Any output channels that are enabled may now be turned on by LCR.
LIST ii	Lists all parameters associated with input channel ii.
SHOW mm	Lists all parameters associated with input channels on machine mm.
OUT ii jj	Attaches output channel jj to input channel ii. Each input channel is wired to a particular transducer. Each output channel is wired to a particular pump or drain.

ZERO ii jj Sets the zero offset on channel ii to jj. This adjusts the AD reading so that a zero pressure reading gives a zero reading on the AD converter.

SCALE ii jj Sets the multiplying factor, jj, used to convert AD readings on channel ii into real units (PSI, bars, mm).

LABEL ii Assigns a 6 character label to channel ii.

MODE ii Makes channel ii either active or inactive.

MAX ii jj Sets the maximum value on channels ii to jj. If the input reading on channel ii exceeds this value, the channel is shut off along with all other channels associated with the same machine.

SP ii jj Set point on channel ii is set to jj.

Rate ii jj Rate of change of setpoint on channel ii is set to jj.

SAVE Parameters of all channels are saved on a disk file.

STOP Execution of LCR is terminated and the program is removed from core.

LCR cycles once every two seconds. During each cycle, each pump can be turned on for 0.2 seconds, 0.5 or a full two seconds depending on the magnitude of the setpoint deviation. After a pump has been activated, the program updates a variable which estimates 'change in pressure/second of pumping'. This is used to determine how long the pump should be turned on in subsequent cycles to bring the pressure up to the setpoint. Thus the program automatically compensates for the response characteristics of each pump, and does not require any intervention by the user. The use of this control algorithm has provided excellent results in terms of maximum system performance with minimum user intervention.

Set points and AD readings are continuously displayed on an auxiliary video screen. In this way, the user can easily tell what is happening in his experiments.

Another feature of LCR is that all AD readings can be stored on mag tape. These can then be used for plotting and analysis of experiments.

7 SYSTEM DEVELOPMENT AND POTENTIAL

Although the system described in this report is already operating satisfactorily and is currently being used to run experiments in the lab, a number of improvements are scheduled for the near future.

Specialized subroutines can be added to the LCR program to calculate, in real time, normal stresses on sawcuts as well as other parameters of interest in the experiments. Subroutines can also be written to provide complex stress or stress-strain histories in future experiments. This would make possible experiments that are beyond the scope of most of the control systems currently in use.

ACKNOWLEDGEMENTS

We wish to express our thanks to Robert Summers, Peter Solberg and Ken Harper for their assistance in building this system. Their expertise and craftsmanship have contributed to making this a professional and, above all, a working and usable system.

FIGURE CAPTIONS

Figure 1 System configuration based on the PDP11/03 microprocessor. Four triaxial rock presses are controlled simultaneously by the processor. Control parameters can be adjusted by the user by issuing commands at the video terminal to the foreground job (LCR). Background jobs can also be run to process data.

Figure 2 Schematic of two stage differential amplifier used to boost input signals to the AD converter. Use of precision components insures low drift performance.

Figure 3 Schematic of low pass filters on AD imputs.

Figure 4 Schematic of relay driver. This was designed to accept TTL logic input from the processor and turn on or off the appropriate output relays. Failsafe channel must be pulsed every 2 seconds by the processor or outputs will be turned off.

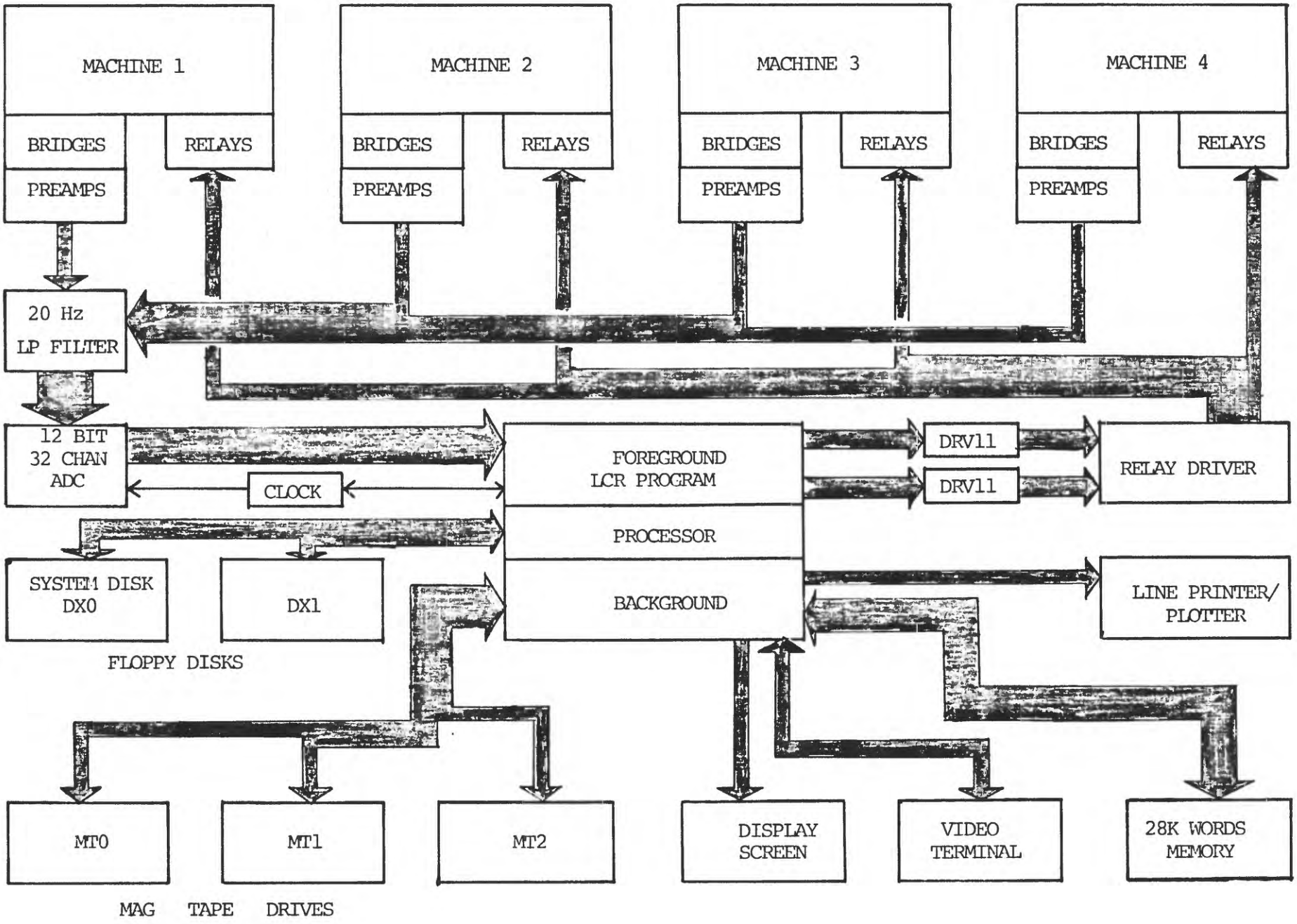
MACHINE 1
MACHINE 2
MACHINE 3
MACHINE 4
BRIDGES
PREAMPS
RELAYS
20 Hz
LP FILTER
12 BIT
32 CHAN
ADC
CLOCK
FOREGROUND
LCR PROGRAM
PROCESSOR
BACKGROUND
DRV11
DRV11
RELAY DRIVER
SYSTEM DISK
DX0
DX1
FLOPPY DISKS
LINE PRINTER/
PLOTTER
MT0
MT1
MT2
MAG TAPE DRIVES
DISPLAY
SCREEN
VIDEO
TERMINAL
28K WORDS
MEMORY

Figure 1

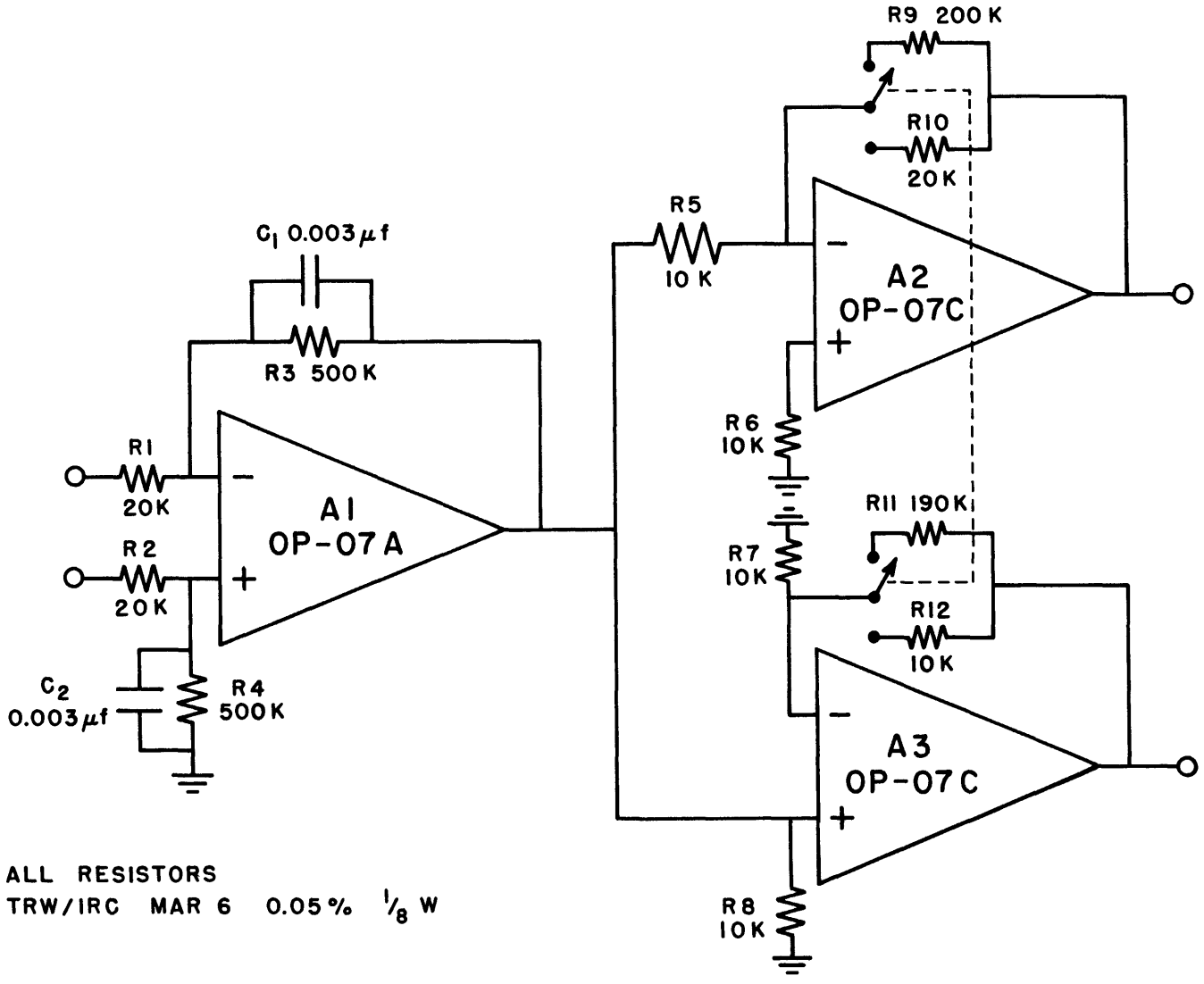
R9 200K
R10
20K
R5
10K
C1 0.003μf
R3 500K
A2
OP-07C
R1
20K
R2
20K
A1
OP-07A
R6
10K
R11 190K
R7
10K
R12
10K
C2
0.003μf
R4
500K
A3
OP-07C
R8
10K
ALL RESISTORS
TRW/IRC MAR 6 0.05% 1/8 W

Figure 2

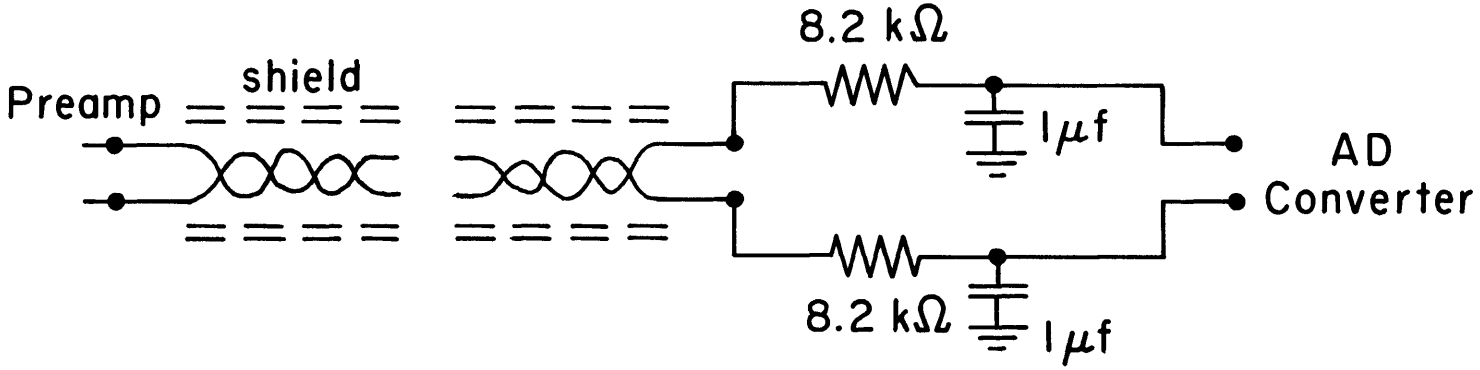

Figure 3

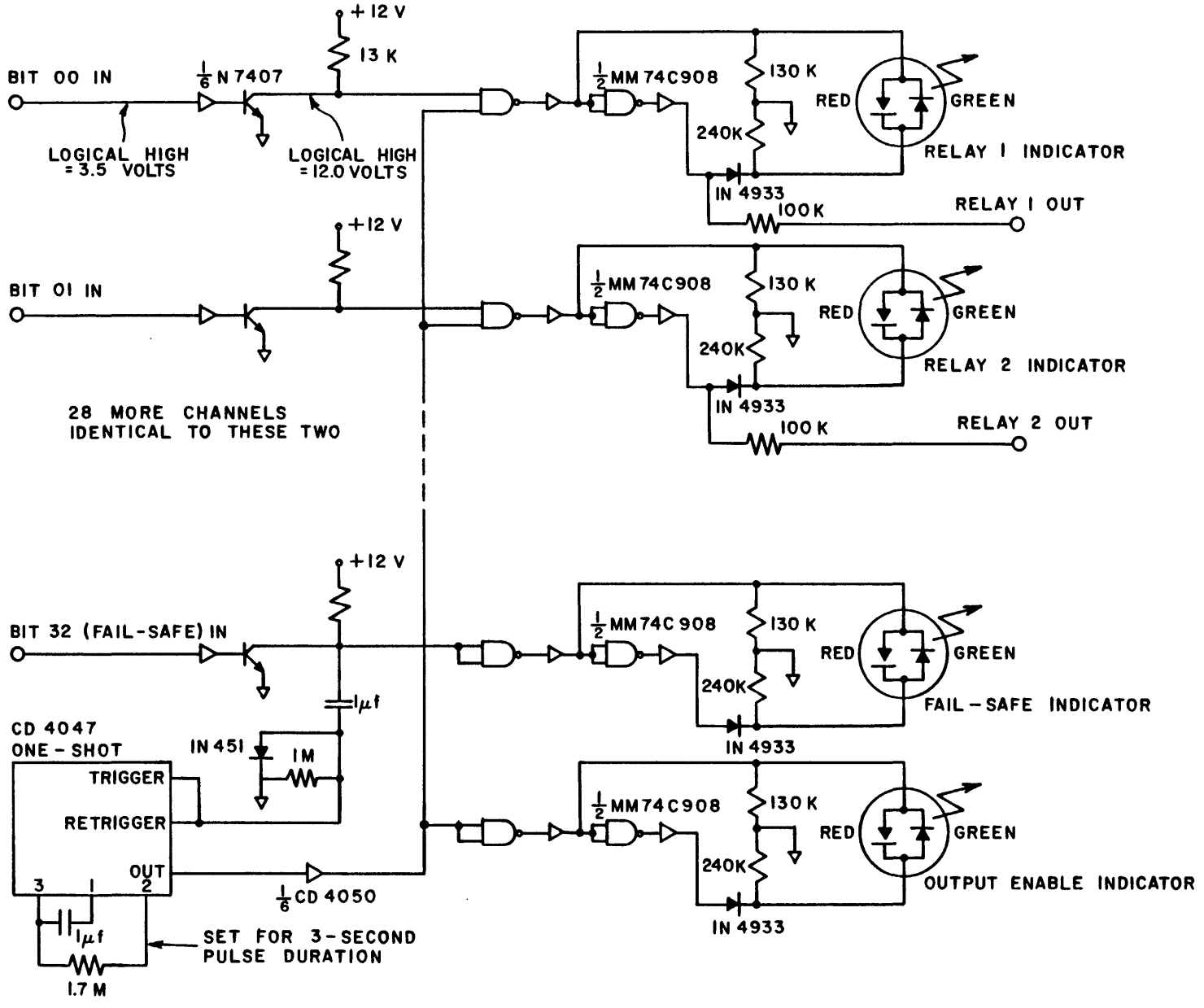
+12 V
13 K
BIT 00 IN
1/6 N 7407
LOGICAL HIGH = 3.5 VOLTS
LOGICAL HIGH = 12.0 VOLTS
1/2 MM 74C908
130 K
240K
RED
GREEN
RELAY 1 INDICATOR
IN 4933
100 K
RELAY 1 OUT
+12 V
BIT 01 IN
1/2 MM 74C908
130 K
240K
RED
GREEN
RELAY 2 INDICATOR
IN 4933
100 K
RELAY 2 OUT
28 MORE CHANNELS IDENTICAL TO THESE TWO
+12 V
BIT 32 (FAIL-SAFE) IN
1μf
CD 4047
ONE - SHOT
IN 451
1M
TRIGGER
RETRIGGER
OUT
3
1
2
1μf
1.7 M
SET FOR 3-SECOND PULSE DURATION
1/6 CD 4050
1/2 MM 74C908
130 K
240K
RED
GREEN
FAIL - SAFE INDICATOR
IN 4933
1/2 MM 74C908
130 K
240K
RED
GREEN
OUTPUT ENABLE INDICATOR
IN 4933

Figure 4

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:49:15          PAGE 001

0001          PROGRAM LCRFG
0002          CALL STRT
0003          CALL LOOP
0004          END



FORTRAN IV      V01C-03A    THU 09-FEB-78 10:49:27          PAGE 001

0001          BLOCK DATA
0002          LOGICAL*1 TIMET(8)
0003          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
             1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
             2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
             3 TIMEA,TIMEO,IDEVA,IDEVB
0004          COMMON /ADCOM/ IADIN(24),ICTABL(24)
0005          COMMON /COMMND/ IHALT,LABFLG,MTFLG,ICOM1,ICHAN,VAL,DATET(3),
             1 TIMET,MTINDX(24),MTNUM
0006          COMMON /CYCLE/ IOSET0,IOSET1
0007          COMMON /VIUBUF/ ISTR(32)
0008          DATA IHALT/0/,LABFLG/0/,DATET/3*'     '/
0009          DATA ADINPT/24*0./,SP/24*0./,DELSP/24*0./,DSPI/24*0./,
             1 DSPIT/24*0./,SCALE/24*1./,SPZERO/24*0./,PRHI/24*0./,
             2 PRLO/24*0./,ITYPE/24*0/,SPMAX/24*1./,SPDEV/24*0./,
             3 IDEVA/5/,IDEVB/7/,IDEVC/2/,IOSET0/0/,IOSET1/0/,MTFLG/0/
0010          END

FORTRAN IV      V01C-03A    THU 09-FEB-78 10:49:43          PAGE 001

0001          SUBROUTINE STRT
0002          LOGICAL*1 TIMET(8)
0003          EXTERNAL TIME.SECNDS
0004          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
             1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
             2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
             3 TIMEA,TIMEO,IDEVA,IDEVB
0005          COMMON /ADCOM/ IADIN(24),ICTABL(24)
0006          COMMON /COMMND/ IHALT,LABFLG,MTFLG,ICOM1,ICHAN,VAL,DATET(3),
             1 TIMET,MTINDX(24),MTNUM
0007          COMMON /CYCLE/ IOSET0,IOSET1
0008          CALL IPOKE("44,"100 .OR. IPEEK("44))
0009          TIMEO=SECNDS(0.)
0010          TIMEA=0.
0011          CALL DATE(DATET)
0012          CALL TIME(TIMET)
      C       CONSTANTS ARE INITALIZED
0013          WRITE (IDEVB,1000)DATET,TIMET
0014    1000  FORMAT (1X,3A4,4X,8A1/
             3 ' LABORATORY CONTROLER ROUTINE (LCR) IS RUNNING.')
0015          CALL ASSIGN(2,'SY:LCRCON.DAT')
0016          READ (2,END=150) LABEL,ITYPE,SP,DELSP,SPZERO,
             1 SCALE,SPMAX,PRHI,PRLO,IOUTCH,JOUTCH
0017          GOTO 160
0018    150   TYPE 1001
0019    1001  FORMAT (' INSUFFICIENT DATA ON INPUT FILE USED TO STORE
             1 PROGRAM CONSTANTS')
0020    160   CALL CLOSE(2)
0021          CALL VDINIT
0022          TYPE 1002
0023    1002  FORMAT (' PLEASE ENTER ANY ADJUSTMENTS. . . .'/
             1 ' THEN ISSUE ''RUN'' COMMAND TO BEGIN EXECUTION.'//)
0024          DO 165 I=1,24
0025          IF (DELSP(I) .EQ. 0.)GOTO 165
0027          DSP(I)=SP(I)
0028          DSPIT(I)=0.
0029    165   IF (ITYPE(I) .GT. 1)ITYPE(I)=1
0031          CALL VDRV1(-1,0)
0032          RETURN
0033          END
```

```
FORTRAN IV      V01C-03A   THU 09-FEB-78 10:50:04                PAGE 001

0001          SUBROUTINE LOOP
0002          LOGICAL*1 TIMET(8)
0003          INTEGER*4 JTIME
0004          EXTERNAL TIME,SECNDS
0005          DIMENSION ITIME(2)
0006          EQUIVALENCE (ITIME,JTIME)
0007          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
             1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
             2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
             3 TIMEA,TIMEO,IDEVA,IDEVB
0008          COMMON /ADCOM/ IADIN(24),ICTABL(24)
0009          COMMON /COMMND/ IHALT,LABFLG,MTFLG,ICOM1,ICHAN,VAL,DATET(3),
             1 TIMET,MTINDX(24),MTNUM
0010          COMMON /CYCLE/ IOSETO,IOSET1
0011          DATA IBLANK/'  '/
0012          CALL IQSET(2)
      C
      C       START LOOP
      C
0013  200     CONTINUE
      C       SET OUTPUT FOR MODE 2
0014          CALL OUTWRT
0015          CALL OUTSET(3)
      C       SLEEP FOR 200 MSEC-TILL END OF MODE 2
0016          CALL ISLEEP(0,0,0,12)
      C       SET OUTPUT FOR MODE 3
0017          CALL OUTWRT
0018          CALL OUTSET(4)
      C       SLEEP FOR 300 MSEC-TILL END OF MODE 3
0019          CALL ISLEEP(0,0,0,18)
      C       SET OUTPUT FOR MODE 4
0020          CALL GTIM(JTIME)
0021          ITIMET=ITIME(2)
0022          CALL OUTWRT
0023  300     DO 320 I=1,24
0024  320     IADIN(I)=0
0025          TIMETM=SECNDS(TIMEO)
0026          IF (TIMETM .LT. TIMEA)TIMEO=-2.-TIMEA
0028          TIMEA=TIMETM
0029          CALL VDRV2
      C       MEANWHILE.....CHECK FOR KEYBOARD COMMAND
0030          ICOM1=ITTINR()
0031          IF (ICOM1 .LT. 0)GOTO 350
0033          CALL TRANS
      C       CHECK FOR HALT COMMAND
0034  350     IF (IHALT .EQ. 1)GOTO 400
0036          CALL OUTSET(1)
0037          CALL OUTWRT
0038  400     CONTINUE
      C       SETUP SLEEP FOR REMAINDER OF 1200 MSEC DELAY
0039          CALL GTIM(JTIME)
0040          ITIMET=ITIMET+71-ITIME(2)
0041          IF (ITIMET .GT. 71 .OR. ITIMET .LE. 0)GOTO 410
0043          CALL ISLEEP(0,0,0,ITIMET)
```

```
      C     SET UP AD SAMPLING AND EXECUTE.....
0044  410   CALL GTIM(JTIME)
0045        ITIMET=ITIME(2)
0046        ICMF=0
0047        CALL SETR(1,9,4167.,ICMF)
0048        CALL ADSUM
0049        CALL SETR(-1,,,)
      C     ADJUST SETPOINT CONSTANTS
0050        IF (MTFLG .EQ. 0)GOTO 420
      C     WRITE AD VALUES TO TAPE
0052        CALL MTDRV2
0053  420   CALL ADFLT
0054        IF (IHALT .EQ. 0)GOTO 430
0056        CALL SETPNT
0057        CALL OUTSET(2)
      C     SETUP SLEEP FOR REMAINDER OF 300 MSEC DELAY
0058  430   CALL GTIM(JTIME)
0059        ITIMET=ITIMET+17-ITIME(2)
0060        IF (ITIMET .GT. 17 .OR. ITIMET .LT. 1)GOTO 440
0062        CALL ISLEEP(0,0,0,ITIMET)
0063  440   IF (IHALT .EQ. 1)GOTO 200
      C     HALT CODE IN EFFECT
0065        CALL ISLEEP(0,0,0,30)
0066        CALL GTIM(JTIME)
0067        ITIMET=ITIME(2)
0068        GOTO 300
0069        END
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:50:34          PAGE 001

0001        SUBROUTINE ADSUM
0002        COMMON /ADCOM/ IADIN(24)
0003        DIMENSION IADSAM(96)
0004        DATA IERR/0/
0005        DO 10 I=1,4
0006        NBUF=1
0007        IBEF=1
0008        J=24*(I-1)+1
0009        CALL SAMGO(24,0,IADSAM(J),NBUF,IBEF,IERR)
0010  5     IF (IERR)5,5,10
0011  10    CONTINUE
0012        DO 20 I=1,24
0013        DO 20 J=I,I+72,24
0014  20    IADIN(I)=IADIN(I)+IADSAM(J)
0015        RETURN
0016        END
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:50:47          PAGE 001

0001        SUBROUTINE ADFLT
      C     COMPLETION ROUTINE FOR AD SAMPLING
      C     FLOATS THE DATA AND SCALES IT.
      C     OUTPUT IS THEN IN GAGUE READING UNITS.
0002        COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
           1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
           2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
           3 TIMEA,TIMEO,IDEVA,IDEVB
0003        COMMON /ADCOM/ IADIN(24)
0004        DO 100 I=1,24
0005        ADTEM=FLOAT(IADIN(I))
0006  90    ADINPT(I)=(ADTEM-SPZERO(I))*SCALE(I)
0007  100   CONTINUE
      D     TYPE 2000,IADIN
      D     CALL ISLEEP(0,0,5,0)
      D2000 FORMAT (1X,12I6)
0008        RETURN
0009        END
```

```
FORTRAN IV      V01C-03A   THU 09-FEB-78 10:51:13                PAGE 001

0001          SUBROUTINE SETPNT
      C       THIS ROUTINE CALCULATES SETPOINT ESTIMATORS,
      C       UPDATES THEM AND SETS TYPE OF OUTPUT ON EACH ACTIVE CHANNEL.
0002          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
             1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
             2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
             3 TIMEA,TIMEO,IDEVA,IDEVB
0003          DIMENSION PMPTM(4),PMPTMR(4)
0004          DATA PMPTM/1.,.20,.5,2.0/,PMPTMR/1.,5.,2.,.5/
0005          DO 500 N=1,24
      C       CHECK IF CHANNEL IS ENABLED
0006          IF (ITYPE(N) .EQ. 0)GOTO 500
      C       ADJUST SETPOINT DUE TO RATE
0008          IF (DELSP(N) .EQ. 0.)GOTO 50
0010          SP(N)=DSPI(N)+DELSP(N)*(TIMEA-DSPIT(N))
      C       CHECK IF OVER SPMAX
0011    50    IF (ADINPT(N) .LT. SPMAX(N))GOTO 80
      C       CHANNEL IS OVERRANGE-SHUT DOWN MACHINE...
0013          MACH=JOUTCH(N)/10
0014          DO 60 I=1,24
0015          IF (JOUTCH(I)/10 .EQ. MACH) ITYPE(I)=0
0017    60    CONTINUE
0018          TYPE 1000,N,MACH
0019          NT=-1-MACH
0020          CALL LISTC(NT)
0021          GOTO 500
0022    80    TEMDEV=SPDEV(N)
0023          SPDEV(N)=ADINPT(N)-SP(N)
      C       CHECK IF CHANNEL WAS ACTIVE LAST CYCLE
0024          IF (ITYPE(N) .EQ. 1)GOTO 200
      C       BRANCH IF CHANNEL WAS HI
0026          TEMPR=(SPDEV(N)-TEMDEV)*PMPTMR(ITYPE(N))
0027          IF (TEMDEV .GT. 0.)GOTO 100
      C       CHANNEL WAS LO
0029          DELPR=.5*(TEMPR-PRLO(N))
0030          DELPRS=SIGN(1.,DELPR)
0031          DELPR=DELPRS*AMIN1(ABS(DELPR),0.2)
0032          PRLO(N)=PRLO(N)+DELPR
0033          GOTO 200
      C       CHANNEL WAS HI
0034    100   DELPR=.5*(TEMPR-PRHI(N))
0035          DELPRS=SIGN(1.,DELPR)
0036          DELPR=DELPRS*AMIN1(ABS(DELPR),0.2)
0037          PRHI(N)=PRHI(N)+DELPR
0038    200   NT=1
0039          TEMDEV=ABS(SPDEV(N))
0040          IF (SPDEV(N) .GT. 0.)GOTO 300
0042          TEMPR=PRLO(N)
0043          GO TO 400
0044    300   TEMPR=ABS(PRHI(N))
      C       CHECK FOR OVERSHOOT
0045    400   DEADBD=AMIN1(ABS(PRHI(N)),PRLO(N))*PMPTM(2)
0046          DEADBD=AMAX1(DEADBD,1.)
0047          IF (TEMDEV .LT. DEADBD)GOTO 450
0049          DO 440 I=3,4
0050          NT=NT+1
0051    440   IF (TEMDEV-PMPTM(I)*TEMPR*2. .LT. 0.)GOTO 450
0053          NT=4
0054    450   ITYPE(N)=NT
0055    500   CONTINUE
0056          RETURN
0057    1000  FORMAT (' INPUT VALUE HAS EXCEEDED MAXIMUM ON CHANNEL',I3/
             1 ' MACHINE',I2,' WAS SHUT DOWN.')
0058          END
```

```
FORTRAN IV      V01C-03A   THU 09-FEB-78 10:51:53              PAGE 001

0001          SUBROUTINE OUTSET(MODE)
      C       OUTPUT CHANNEL ODD => ADC READING IS HI
      C                      EVEN => ADC READING IS LO
0002          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
             1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
             2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
             3 TIMEA,TIMEO,IDEVA,IDEVB
0003          COMMON /CYCLE/ IOSETO,IOSET1
0004          DIMENSION IOMSK(16)
0005          DATA IOMSK/"1,"2,"4,"10,"20,"40,"100,"200,"400,"1000,
             1 "2000,"4000,"10000,"20000,"40000,"100000/
0006          IOSETO=0
0007          IOSET1=0
      D       TYPE 2000,IOUTCH
      D2000   FORMAT (24I2)
0008          IF (MODE .EQ. 1)GOTO 300
0010          DO 200 I=1,24
0011          IF (ITYPE(I) .LT. MODE)GOTO 200
0013          IF (IOUTCH(I) .LT. 1)GOTO 200
      C       CHANNEL MUST BE TURNED ON- NOW CHECK HI OR LO
0015          J=2*IOUTCH(I)
      D       TYPE 2000,J
0016          IF (SPDEV(I) .GT. 0.)J=J-1
0018          IF (J-16)150,150,180
0019  150     IOSETO=IOSETO .OR. IOMSK(J)
0020          GOTO 200
0021  180     J=J-16
      D       TYPE 2000,J
0022          IOSET1=IOSET1 .OR. IOMSK(J)
0023  200     CONTINUE
      D       TYPE 2001,IOSETO,IOSET1
      D2001       FORMAT (2010)
0024          IF (MODE .GT. 2)RETURN
0026          IOSET1=IOSET1 .OR. IOMSK(16)
0027  300     RETURN
0028          END
```

```
FORTRAN IV      V01C-03A   THU 09-FEB-78 10:51:42              PAGE 001

0001          SUBROUTINE OUTWRT
      C       WRITES CONTROL BIT PATTERN TO DRV11'S
0002          COMMON /CYCLE/ IOSETO,IOSET1
0003          CALL IDOR (0,0,"177777,IOSETO)
0004          CALL IDOR (0,1,"177777,IOSET1)
      D       TYPE 1000,IOSETO,IOSET1
0005          RETURN
      D1000   FORMAT (2010)
0006          END
```

```
FORTRAN IV      V01C-03A   THU 09-FEB-78 10:53:40                PAGE 001

0001          SUBROUTINE TRANS
      C       IHALT=0 => HALT
      C            =1 => RUN
      C
0002          LOGICAL*1 TIMET(8),LLDUM(2),LLDUM1(2)
0003          EXTERNAL TIME,DATE,SECNDS
0004          EQUIVALENCE (LLDUM(1),ILDUM),(LLDUM1(1),ILDUM1)
0005          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
             1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
             2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
             3 TIMEA,TIMEO,IDEVA,IDEVB
0006          COMMON /COMMND/ IHALT,LABFLG,MTFLG,ICOM1,ICHAN,VAL,DATET(3),
             1 TIMET,MTINDX(24),MTNUM
0007          DIMENSION KLABEL(3),KCOM(20,2)
0008          DATA KCOM/"122,"110,"123,"123,"122,"123,"132,"115,"110,
             1 "114,"115,"115,"114,"114,"117,"123,"123,"40,"40,"40,
             2 "125,"101,"124,"120,"101,"103,"105,"101,"111,"117,
             3 "117,"124,"111,"101,"125,"101,"110,"40,"40,"40/,
             4 IBLANK/'  '/
      D       IHALT=0
0009          IF (LABFLG .EQ. 1) GOTO 285
0011          VAL=0.
0012          VALS=1.
0013          ICHAN=0
0014          J=0
      C       READ COMMAND
      D2000   FORMAT (1X,508)
0015  100     ICOM2=ITTINR()
0016          IF (ICOM2 .EQ. "12)GOTO 800
      D       TYPE 2000,ICOM1,ICOM2
      C       SKIP TO CHANNEL
0018  102     I=ITTINR()
      D       TYPE 2000,ICOM1,ICOM2,I
0019          IF (I .EQ. "12)GOTO 108
0021          IF (I .EQ. "40)GOTO 103
0023          GOTO 102
      C       READ CHANNEL
0024  103     I=ITTINR()
0025          IF (I .EQ. "40)GOTO 104
0027          IF (I .EQ. "15)GOTO 108
0029          ICHAN=10*ICHAN+I-"60
0030          IF (ICHAN .GT. 1000)GOTO 108
0032          GOTO 103
      C       READ VALUE
0033  104     I=ITTINR()
0034          IF (I .EQ. "15 .OR. I .EQ. "40)GOTO 107
0036          IF (I .NE. "55)GOTO 105
0038          VALS=-1.
0039          GOTO 104
0040  105     IF (I .NE. "56)GOTO 106
0042          J=1
0043          GOTO 104
0044  106     VAL=10.*VAL+FLOAT(I-"60)
0045          IF (J .GT. 0)J=J+1
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:53:40                PAGE 002

0047            GOTO 104
0048  107       VAL=VALS*VAL*10.**(1-J)
0049  108       IF (ITTINR() .GE. 0)GOTO 108
      D         TYPE 2001, ICHAN, VAL
      D2001     FORMAT (I5, F10.2)
0051            IF (ICHAN .GE. 0 .AND. ICHAN .LE. 24)GOTO 110
0053            IF (ICOM1 .EQ. "114 .AND. ICOM2 .EQ. "111 .AND. ICHAN .EQ. 99)
               1 GOTO 310
0055  109       WRITE (IDEVB,1003)
0056            GO TO 600
0057  110       J=0
0058            DO 120 I=1,20
0059            J=J+1
0060            IF (ICOM1 .EQ. KCOM(I,1) .AND. ICOM2 .EQ. KCOM(I,2))GOTO 140
0062  120       CONTINUE
0063  130       WRITE (IDEVB,1005)
0064            GOTO 600
0065  140       IF (ICHAN .GT. 0)GOTO 145
0067            IF (J .LE. 3 .OR. J .EQ. 13 .OR. J .EQ. 16)GOTO 145
0069            GOTO 109
0070  145       GOTO (160,150,900,210,220,230,250,240,260,270,190,180,
               1 310,280,320,340,360,600,600,600) J
      C HALT
0071  150       IHALT=0
0072            WRITE (IDEVB,1001) TIMEA
0073            GOTO 600
      C RUN
0074  160       IHALT=1
0075            GOTO 600
      C MT
0076  180       CALL MTDRV1(ICHAN)
0077            GOTO 600
      C MODE
0078  190       I=INT(VAL)
0079            IF (I .LT. 0 .OR. I .GT. 4)GOTO 200
0081            ITYPE(ICHAN)=I
0082            GOTO 500
0083  200       WRITE (IDEVB,1004)
0084            GOTO 600
      C SETPOINT
0085  210       SP(ICHAN)=VAL
0086            ITYPE(ICHAN)=1
0087            DSPI(ICHAN)=SP(ICHAN)
0088            DSPIT(ICHAN)=TIMEA
0089            GOTO 500
      C RATE
0090  220       DELSP(ICHAN)=VAL
0091            DSPI(ICHAN)=SP(ICHAN)
0092            DSPIT(ICHAN)=TIMEA
0093            GOTO 510
      C SCALE
0094  230       SCALE(ICHAN)=VAL
0095            ITYPE(ICHAN)=1
0096            GOTO 510
```

```
FORTRAN IV      V01C-03A   THU 09-FEB-78 10:53:40              PAGE 003

      C MAX
0097  240   SPMAX(ICHAN)=VAL
0098        GOTO 510
      C ZERO
0099  250   SPZERO(ICHAN)=VAL
0100        GOTO 510
      C HI RATE
0101  260   PRHI(ICHAN)=VAL
0102        GOTO 510
      C LO RATE
0103  270   PRLO(ICHAN)=VAL
0104        GOTO 510
      C LABEL
0105  280   TYPE 1008
0106        LABFLG=1
0107        GOTO 800
0108  285   DO 290 I=1,3
0109  290   KLABEL(I)=IBLANK
0110  291   DO 292 I=1,3
0111        ILDUM1=ITTINR()
0112        IF (ILDUM1 .LT. 0)GOTO 800
0114        IF (ILDUM1 .EQ. "12 .OR. ILDUM1 .EQ. "15)GOTO 293
0116        LLDUM(1)=LLDUM1(1)
0117        ILDUM1=ITTINR()
0118        IF (ILDUM1 .EQ. "15)ILDUM1="40
0120        LLDUM(2)=LLDUM1(1)
0121  292   KLABEL(I)=ILDUM
0122  293   DO 300 I=1,3
0123  300   LABEL(ICHAN,I)=KLABEL(I)
0124        LABFLG=0
0125  301   IF (ITTINR() .GE. 0)GOTO 301
0127        CALL VDRV1(0,ICHAN)
0128        GOTO 500
      C LIST
0129  310   IF (ICHAN .EQ. 99)ICHAN=-1
0131        GOTO 510
      C OUTPUT CHANNEL
0132  320   I=INT(10.*VAL)
0133        J=0
0134        IF (I .LT. 1 .OR. I .GT. 53)GOTO 109
0136        IF (MOD(I,10) .EQ. 0)GOTO 325
0138        J=MOD(I,10)+3*((I-10)/10)
0139  325   IOUTCH(ICHAN)=J
0140        JOUTCH(ICHAN)=I
0141        CALL VDRV1(-1,0)
0142        GOTO 500
      C     SAVE CONSTANTS
0143  340   CALL SAVEC
0144        GOTO 600
      C     SHOW  (BY MACHINE)
0145  360   I=-1-ICHAN
0146        CALL LISTC (I)
0147        GOTO 600
0148  500   CALL LISTC(ICHAN+100)
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:53:40                PAGE 004

0149            GOTO 600
0150    510     CALL LISTC(ICHAN)
0151            GOTO 600
0152    600     WRITE (IDEVB,1000)
0153    800     RETURN
0154    900     CALL DATE(DATET)
0155            CALL TIME(TIMET)
0156            WRITE (IDEVB,1002)DATET,TIMET
0157            CALL LISTC (-1)
0158            CALL ENDRTN
0159    1000    FORMAT (' ?',$)
0160    1001    FORMAT (' LCR EXECUTION HALTED AT',F9.0)
0161    1002    FORMAT (' LCR ROUTINE STOPPED ',3A4,'     ',8A1)
0162    1003    FORMAT ('+ERROR - ILLEGAL CHANNEL')
0163    1004    FORMAT ('+ERROR - ILLEGAL CHANNEL MODE')
0164    1005    FORMAT ('+UNRECOGNIZABLE COMMAND')
0165    1007    FORMAT (3A2)
0166    1008    FORMAT ('+LABEL? ',$)
0167            END


FORTRAN IV      V01C-03A    THU 09-FEB-78 10:56:29                PAGE 001

0001            SUBROUTINE FLTASC(IBUF,V,N)
        C       CONVERTS FLT PNT TO ASCII EVEN IF >30000
0002            DIMENSION IBUF(7)
0003            ABSV=ABS(V)
0004            IF (ABSV .LT. 32000.)GOTO 20
0006            ILO=INT(AMOD(ABSV,1000.))
0007            IHI=INT(.001*V)
0008            IF (ABSV .GT. 3.E7)IHI=30000
0010            CALL INTASC(IBUF,ILO,N)
0011            DO 10 ILO=N-2,N
0012    10      IF(IBUF(ILO) .EQ. "40)IBUF(ILO)="60
0014            CALL INTASC(IBUF,IHI,N-3)
0015            RETURN
0016    20      ILO=INT(V)
0017            CALL INTASC(IBUF,ILO,N)
0018            RETURN
0019            END


FORTRAN IV      V01C-03A    THU 09-FEB-78 10:56:43                PAGE 001

0001            SUBROUTINE LABASC(IBUF,I1,I2,I3)
0002            IMPLICIT LOGICAL*1 (L)
0003            DIMENSION IBUF(7),JBUF(3),LBUF(6),LT(2)
0004            EQUIVALENCE (JBUF(1),LBUF(1)),(JT,LT(1))
0005            JBUF(1)=I1
0006            JBUF(2)=I2
0007            JBUF(3)=I3
0008            DO 20 I=1,6
0009            LT(1)=LBUF(I)
0010    20      IBUF(I)=JT
0011            RETURN
0012            END
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:54:41                PAGE 001

0001            SUBROUTINE LISTC(ICODE)
0002            COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
               1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
               2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
               3 TIMEA,TIMEO,IDEVA,IDEVB
0003            DATA LSTFLG/5/
0004            IENBLE=1
0005            IF (ICODE-99)10,70,70
0006  10        IF (ICODE)30,20,40
0007  20        IENBLE=0
0008  30        IF (ICODE .LT. -1)GOTO 200
0010            IL=1
0011            IH=24
0012            GOTO 50
0013  40        IL=ICODE
0014            IH=IL
0015  50        WRITE (IDEVB,1000) TIMEA
0016            IF (ICODE .LE. 0) LSTFLG=5
0018            LSTFLG=LSTFLG+1
0019            IF (LSTFLG .LT. 5)GOTO 53
0021            LSTFLG=0
0022            WRITE (IDEVB,1004)
0023  53        DO 60 I=IL,IH
0024  55        IF (IENBLE+ITYPE(I) .EQ. 0)GOTO 60
0026            WRITE (IDEVB,1002) I,(LABEL(I,J),J=1,3),ITYPE(I),JOUTCH(I),
               1 ADINPT(I),SP(I),DELSP(I),SPZERO(I),SCALE(I),
               2 SPMAX(I),PRHI(I),PRLO(I)
0027  60        CONTINUE
0028            RETURN
0029  70        IF (ICODE-100)90,80,100
0030  80        IENBLE=0
0031  90        IL=1
0032            IH=24
0033            GOTO 110
0034  100       IL=ICODE-100
0035            IH=IL
0036  110       WRITE (IDEVB,1001) TIMEA
0037            DO 120 I=IL,IH
0038            IF (IENBLE+ITYPE(I) .EQ. 0)GOTO 120
0040            WRITE (IDEVB,1003)I,(LABEL(I,J),J=1,3),ITYPE(I),JOUTCH(I),
               1 ADINPT(I),SP(I)
0041  120       CONTINUE
0042            RETURN
0043  200       IL=-1-ICODE
0044            WRITE (IDEVB,1000)TIMEA
0045            WRITE (IDEVB,1004)
0046            DO 210 I=1,24
0047            IF (JOUTCH(I)/10 .EQ. IL)WRITE (IDEVB,1002)I,
               1 (LABEL(I,J),J=1,3),ITYPE(I),JOUTCH(I),ADINPT(I),SP(I),
               2 DELSP(I),SPZERO(I),SCALE(I),SPMAX(I),PRHI(I),PRLO(I)
0049  210       CONTINUE
0050            RETURN
0051  1000      FORMAT (' TIME=',F9.0,' SEC')
0052  1001      FORMAT (' TIME=',F9.0,' SEC'//' CH LABEL MODE OUT VALUE     SP')
0053  1002      FORMAT (1X,I2,1X,3A2,I3,I5,2F7.0,F8.5,F7.0,F7.4,F7.0,2F6.2)
0054  1003      FORMAT (1X,I2,1X,3A2,I3,I5,2F7.0)
0055  1004      FORMAT (' CH LABEL MODE OUT VALUE     SP     RATE     ZERO
               1 'SCALE    MAX    HIRA   LORA')
0056            END
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:55:08              PAGE 001

0001            SUBROUTINE MTDRV1(ICODE)
      C         OPENS OR CLOSES MAGTAPE FILE ON MT0
0002            LOGICAL*1 TIMET(8),LYES,LT
0003            COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
               1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
               2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
               3 TIMEA,TIMEO,IDEVA,IDEVB
0004            COMMON /COMMND/ IHALT,LABFLG,MTFLG,ICOM1,ICHAN,VAL,DATET(3),
               1 TIMET,MTINDX(24),MTNUM
0005            DATA LYES/'Y'/
0006            GOTO (100,500)ICODE
      C         BAD CHANNEL
0007            TYPE 1
0008            RETURN
      C         OPEN FILE
0009  100       IF (MTFLG .EQ. 0)GOTO 110
0011            TYPE 2
0012            RETURN
0013  110       TYPE 3
0014            ACCEPT 4,LT
0015            IF (LT .EQ. LYES)GOTO 120
0017            TYPE 5
0018            RETURN
0019  120       DO 130 I=1,24
0020  130       MTINDX(I)=0
0021            MTNUM=0
      C         SET UP CHANNEL TABLE
0022            TYPE 7
0023            DO 140 I=1,24
0024            IF (ITYPE(I) .EQ. 0)GOTO 140
0026            TYPE 8,I
0027            ACCEPT 4,LT
0028            IF (LT .NE. LYES)GOTO 140
0030            MTNUM=MTNUM+1
0031            MTINDX(MTNUM)=I
0032  140       CONTINUE
0033            IF (MTNUM .NE. 0)GOTO 150
0035            TYPE 9
0036            RETURN
0037  150       TYPE 6
0038            CALL ASSIGN(3,,-1)
0039            CALL TIME(TIMET)
0040            CALL DATE(DATET)
0041            WRITE (3) DATET,TIMET,TIMEA,LABEL,SCALE,MTINDX
0042            MTFLG=1
0043            RETURN
0044  500       IF (MTFLG-1)520,510,520
0045  510       CALL CLOSE(3)
0046            MTFLG=0
0047  520       RETURN
0048  1         FORMAT ('+BAD ARGUEMENT')
0049  2         FORMAT ('+TAPE FILE ALREADY OPEN')
0050  3         FORMAT ('+HAS TAPE HANDLER ''MT:SYS'' BEEN LOADED?',$)
0051  4         FORMAT (A1)
0052  5         FORMAT ('+ABORT OPENING OF MT FILE')
0053  6         FORMAT (' MOUNT TAPE WITH WRITE RING ON MT0'/
               1 ' ENTER FILE NAME (MT0:LCR___.DAT):'/)
0054  7         FORMAT ('+REPLY ''Y'' TO EACH CHANNEL YOU WANT RECORDED:'/)
0055  8         FORMAT ('+',I2,$)
0056  9         FORMAT ('+NO OUTPUT CHANNELS SPECIFIED -- ABORT'/)
0057            END
```

```
FORTRAN IV      V01C-03A   THU 09-FEB-78 10:51:00              PAGE 001

0001          SUBROUTINE MTDRV2
0002          LOGICAL*1 TIMET(8)
0003          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
             1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
             2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
             3 TIMEA,TIMEO,IDEVA,IDEVB
0004          COMMON /COMMND/IHALT,LABFLG,MTFLG,ICOM1,ICHAN,VAL,DATET(3),
             1 TIMET,MTINDX(24),MTNUM
0005          DIMENSION ADOUT(24)
0006          DO 100 I=1,MTNUM
0007          J=MTINDX(I)
0008  100     ADOUT(I)=ADINPT(J)
0009          WRITE (3) TIMEA,(ADOUT(I),I=1,MTNUM)
0010          RETURN
0011          END



FORTRAN IV      V01C-03A   THU 09-FEB-78 10:55:34              PAGE 001

0001          SUBROUTINE VDRV1(ICODE,ICHAN)
      C       FORTRAN DRIVER INTERFACE FOR VIURAM IN LCR FORMAT.
      C       MACRO CALL LOOKS LIKE:
      C       CALL VIUDIS(IADRS,NSKIP,N,MSK)
      C
      C       IADRS         =STARTING ADDRESS ON VIURAM
      C       NSKIP         =# OF CHARS TO SKIP (UPDATING MASK) BEFORE WRITING
      C       N    =# OF CHARS TO WRITE
      C       MSK = MASK FOR CONTROL CHARS.
      C
      C       VIUBUF        COMMON CONTAINING ASCII STRING (32 WORDS)
      C
      C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      C
      C       ICHAN         = LCR CHANNEL
      C       ICODE         = OP CODE FOR DRIVER FUNCTION:
      C     -1 INITIALIZE SCREEN AND TABLE
      C     0 MODIFY LABEL
      C     * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
0002          COMMON /VIUBUF/ ISTR(32),ICTABL(24)
0003          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),
             1 DELSP(24),DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),
             2 SPMAX(24),SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),
             3 JOUTCH(24),TIMEA,TIMEO,IDEVA,IDEVB
0004          DIMENSION MASKS(4),ISTAD(6,4),MTOPCH(4),
             1 ISTEM(7)
0005          DATA ISP,IDOT/"40,"56/,MASKS/"0,"100,"200,"300/,
             2 ISTAD/"160400,"160600,"161000,"161200,"161400,"161600,
             3       "160500,"160700,"161100,"161300,"161500,"161700,
             4       "162400,"162600,"163000,"163200,"163400,"163600,
             5       "162500,"162700,"163100,"163300,"163500,"163700/
0006          IF (ICODE)10,200,200
      C       INITIALIZE TABLE AND SCREEN
0007  10      DO 20 I=1,32
0008  20      ISTR(I)=ISP
0009          DO 30 I=1,4
0010  30      MTOPCH(I)=0
      C       SET UP TABLE POINTERS
0011          DO 60 I=1,24
0012          J=JOUTCH(I)/10
0013          IF (J .GE. 1 .AND. J .LE. 5)GOTO 50
      C       NO DISPLAY
0015  40      ICTABL(I)=0
0016          GOTO 60
0017  50      IF (MTOPCH(J) .GE. 6)GOTO 40
0019          MTOPCH(J)=MTOPCH(J)+1
0020          MT=MTOPCH(J)
0021          ICTABL(I)=ISTAD(MT,J)
0022  60      CONTINUE
      C       ICTABL NOW CONTAINS START ADDRESS OF CHANNELS
      D       TYPE 1000,ICTABL
      D1000 FORMAT (1X1008)
      C       NOW DISPLAY CH,JOUTCH
0023          DO 90 I=1,24
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:55:34                PAGE 002

0024            IF (ICTABL(I) .EQ. 0)GOTO 90
0026  70        ITEM=I
0027            CALL INTASC(ISTEM,ITEM,2)
0028            DO 80 J=2,3
0029  80        ISTR(J)=ISTEM(J-1)
0030            CALL INTASC(ISTEM,JOUTCH(I),2)
0031            ISTR(12)=ISTEM(1)
0032            ISTR(13)=IDOT
0033            ISTR(14)=ISTEM(2)
      D         TYPE 1001,I,ISTR
      D1001     FORMAT (I3,32A2)
0034            CALL VIUDIS(ICTABL(I),0,32,MASKS(4))
0035  90        CONTINUE
      C         CLEAR REMAINDER OF SCREEN
0036            DO 100 I=1,32
0037  100       ISTR(I)=ISP
0038            DO 110 I=1,4
0039            IF (MTOPCH(I) .EQ. 6)GOTO 110
0041            DO 105 J=MTOPCH(I)+1,6
      D         TYPE 1002,I,J
      D1002     FORMAT(I3,I2,$)
0042  105       CALL VIUDIS(ISTAD(J,I),0,32,MASKS(1))
0043  110       CONTINUE
      C         MODIFY LABEL
0044  200       IL=ICHAN
0045            IH=ICHAN
0046            IF (ICHAN)210,210,220
0047  210       IL=1
0048            IH=24
0049  220       DO 250 I=IL,IH
0050            IF (ICTABL(I) .EQ. 0)GOTO 250
0052  230       CALL LABASC(ISTEM,LABEL(I,1),LABEL(I,2),LABEL(I,3))
0053            DO 240 J=1,6
0054  240       ISTR(J)=ISTEM(J)
0055            CALL VIUDIS(ICTABL(I),4,6,MASKS(4))
0056  250       CONTINUE
0057            RETURN
0058            END
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:56:07                PAGE 001

0001            SUBROUTINE VDRV2
      C         FORTRAN DRIVER INTERFACE FOR VIURAM IN LCR FORMAT.
      C         UPDATES MODE, ADINPT, SP
      C
0002            COMMON /VIUBUF/ ISTR(32), ICTABL(24)
0003            COMMON /PRIME/ LABEL(24,3), ITYPE(24), ADINPT(24), SP(24),
               1 DELSP(24), DSPI(24), DSPIT(24), SPZERO(24), SCALE(24),
               2 SPMAX(24), SPDEV(24), PRHI(24), PRLO(24), IOUTCH(24),
               3 JOUTCH(24), TIMEA, TIMEO, IDEVA, IDEVB
0004            DIMENSION MASKS(4), ISTEM(7)
0005            DATA ISP, IDOT/"40, "56/, MASKS/"0, "100, "200, "300/
      C         MODIFY MODE, VALUE, SP, MASK
0006  500       ISTR(3)=ISP
0007            DO 600 I=1,24
0008            IF (ICTABL(I) .EQ. 0)GOTO 600
      C         SET MASK
0010            IF (ADINPT(I) .GE. .8*SPMAX(I))GOTO 530
0012            IF (ITYPE(I) .EQ. 0)GOTO 510
0014            IF (ITYPE(I) .LT. 4)GOTO 520
0016            MSK=MASKS(1)
0017            GOTO 570
0018  510       MSK=MASKS(4)
0019            GOTO 570
0020  520       MSK=MASKS(2)
0021            GOTO 570
      C         NEAR MAX - SET WARNING
0022  530       MSK=MASKS(3)
      C         NOW SET CHARACTERS
0023  570       CALL INTASC(ISTEM, ITYPE(I), 2)
0024            ISTR(1)=ISTEM(1)
0025            ISTR(2)=ISTEM(2)
0026            CALL FLTASC(ISTEM, ADINPT(I), 6)
0027            DO 580 J=1,6
0028  580       ISTR(J+3)=ISTEM(J)
0029            CALL FLTASC(ISTEM, SP(I), 6)
0030            DO 590 J=1,6
0031  590       ISTR(J+10)=ISTEM(J)
0032            ISTR(10)=IDOT
0033            ISTR(17)=IDOT
0034            CALL VIUDIS(ICTABL(I), 15, 17, MSK)
0035  600       CONTINUE
      C         NOW UPDATE TIME
0036            ISTR(8)=IDOT
0037            CALL FLTASC(ISTEM, TIMEA, 7)
0038            DO 610 I=1,7
0039  610       ISTR(I)=ISTEM(I)
0040            CALL VIUDIS("160076, 0, 8, MASKS(1))
0041            RETURN
0042            END
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:56:56                  PAGE 001

0001          SUBROUTINE SAVEC
0002          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
             1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
             2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
             3 TIMEA,TIMEO,IDEVA,IDEVB
0003          CALL ASSIGN (2,'SY:LCRCON.DAT')
0004          WRITE (2) LABEL,ITYPE,SP,DELSP,SPZERO,SCALE,SPMAX,
             1 PRHI,PRLO,IOUTCH,JOUTCH
0005          ENDFILE 2
0006          CALL CLOSE(2)
0007          RETURN
0008          END
```

```
FORTRAN IV      V01C-03A    THU 09-FEB-78 10:57:10                  PAGE 001

0001          SUBROUTINE ENDRTN
0002          COMMON /PRIME/ LABEL(24,3),ITYPE(24),ADINPT(24),SP(24),DELSP(24),
             1 DSPI(24),DSPIT(24),SPZERO(24),SCALE(24),SPMAX(24),
             2 SPDEV(24),PRHI(24),PRLO(24),IOUTCH(24),JOUTCH(24),
             3 TIMEA,TIMEO,IDEVA,IDEVB
0003          CALL ASSIGN (2,'SY:LCRCON.DAT')
0004          WRITE (2) LABEL,ITYPE,SP,DELSP,SPZERO,SCALE,SPMAX,
             1 PRHI,PRLO,IOUTCH,JOUTCH
0005          CALL CLOSE(2)
0006          STOP 'LCR ENDED'
0007          END
```