183
H-1H-81
(81)

LA-8731-M

Manual

(1)

01.2520

R-3517

Small Sample Assay Station Users Guide

MASTER

University of California

LOS ALAMOS SCIENTIFIC LABORATORY

Post Office Box 1663 Los Alamos, New Mexico 87545

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

SMALL SAMPLE ASSAY STATION USERS GUIDE

by

E. L. Adams, S. Bourret, and M. M. Meier

ABSTRACT

A system for acquisition of delayed neutron data, based on an LSI-11 with 28 K words of memory, is described. Hardware features are a six-channel scaler and level sensor to determine the state of the experiment; and normal peripherals include dual floppy-disk drive, line printer, and CRT terminal. The software for experiment control and for the analysis of data is presented. The protocol for assays that optimally utilize the system is suggested.

I. INTRODUCTION

This guide is to acquaint potential users of the Van de Graaff-based Small Sample Assay Station (SSAS) with the associated support hardware and software. The general operation of the SSAS is described in Ref. 1. In brief, the proton beam is pulsed with a repetition rate of 10 s^{-1} . The 100-ms period is divided into four intervals:

1. 0 to 35 ms beam on target for neutron production, rate 10^{10} - 10^{11} n/s and SSAS preamp biased off;
2. 35 to 50 ms beam off, SSAS preamp biased on and gain stabilizing;
3. 50 to 90 ms beam off, SSAS preamp gain stabilized and scaler electronics counting delayed neutrons; and

4. 90 to 100 ms beam off, SSAS preamp biased off and scaler electronics gated off.

The interrogating neutron flux is monitored by a cadmium-covered ^3He proportional counter mounted on top of the SSAS, a cylindrical fission chamber (FC) that surrounds the sample position, a modified long counter (MLC), and the proton current incident on the neutron producing target. The first three of these signals are supplied to integral discriminators with biases set between the low pulse-height tail and the thermal peak (^3He counters) or fission fragment peak (fission chamber). The target current is supplied to a commercial current integrator and the output pulses, proportional to total charge, are scaled in a scaler board² along with the discriminated monitors' pulse trains.

The SSAS is a very efficient detector based on 39 moderated ^3He proportional counters. During the time that its preamp is biased on and stable, logic pulses corresponding to $^3\text{He}(n,p)\text{T}$ events are supplied to another channel of the scaler. Samples are assayed by measuring the ratio of SSAS events in the delayed neutron counting period to the interrogating neutron flux. A rabbit system is used to transport the samples from the Van de Graaff console to the irradiation position in the SSAS. An associated clock is started when the sample is transmitted and an "enable acquisition" dc level (EAL) shifts from 0 to 5 V after a preset interval in which the irradiated sample comes to equilibrium.

II. HARDWARE

The Van de Graaff LSI-11-related hardware is comprised of the LSI-11 chassis, the interfaces and scaler board, a CRT terminal, and a LA180 printer.

The LSI-11 chassis houses the computer CPU, memory, scaler board, and interfaces. The chassis can hold a total of 16 dual Q bus cards, has a tilt-up backplane, a power-start power-fail and line clock board, and two power supplies, a 5 V, 23 A and a 12 V, 7 A supply. Figure 1 is a block diagram of the chassis and bus interfaces to the peripherals. In slots 1 and 2 of the Q bus is the LSI-11 CPU with 4 K of memory. In slot 3 is a 24 K memory board for a total memory of 28 K words. In slots 4, 5, 6, 7, 8, and 9 are, respectively, the DSD dual-drive floppy-disk interface, the DLV11 serial line to the console terminal, the DLV11 serial line to the LA180 line printer, the DRV11 parallel

Fig. 1.
Van de Graaff LSI-11 block diagram.

interface to the Van de Graaff "enable acquisition" level, the 6-channel scaler to the discriminator outputs, and the REV11 bootstrap-memory refresh board.

III. SOFTWARE CONTROL

There are two types of software control for this system: automatic program control and the semi-manual control by the console keyboard.

The automatic program is comprised of the following sequence: initialize, enter ID, wait for the start signal, start the scalers and timer, timeout, store the data on the disk, do the calculations, and print the results. Appendix A is the control program listing. The executive flowchart is shown in Fig. 2. The first event after start is initialization of the program, disk, scalers, and clocks. The computer then prompts for the sample ID. After the ID has been entered the program goes into a checking routine to see if the Van de Graaff EAL has shifted or if there are any manual commands from the console keyboard. (See Fig. 3.) If a keyboard character is present, it is decoded into one of the following commands and program control is transferred to that subroutine. These commands are:

ID	enter identification data, overriding old ID
HELP	print the console keyboard command menu,

Fig. 2.
Van de Graaff LSI-11 executive program
flowchart.

Fig. 3.
The executive program check routine
flowchart.

START	override the start signal from the Van de Graaff and start (perhaps again),
MONITOR	return to the RT11 operating system monitor,
PAUSE	stop the experiment where it is and/or continue,
READ	read the current value of the scalers and output to display,
INIT	restart the program from the beginning, and
TIME	enter the counting time for the clock.

New commands are easily added to the program.

The interrupt status word is next checked to see if an interrupt has occurred from the foreground program. A 60-Hz clock is the only generator of these interrupts, and there are three possible courses of action when one is triggered. First, a word initially equal to 60 is decremented. If the result is zero, a word initially set to the preset time in seconds is decremented and the original word is reset to 60. This corresponds to the elapsing of one second and is reflected in a change in the time display. If the preset time result is zero, the clock and the scalers are disabled and then a report is made to the background check routine through a bit in the interrupt status word. This corresponds to the end of a run.

After the keyboard and status word are checked and dealt with, return is made to the executive program. The executive program then checks the Van de Graaff EAL and if it is not set, it loops to the top of the executive to call check and try again. If the start signal is there, the scalers clock and interrupts are enabled to count. The check routine is repeatedly called until either a restart or start is given through the keyboard commands or the clock timeout occurs, whereby the scalers are stopped and the data are printed on the printer and stored on the disk. Control is then returned to the beginning of the executive.

The main building blocks of the control portion of the Van de Graaff program are the executive routine, the interrupt handler, and 13 major sub-routines.

IV. ASSAY PROTOCOL

For normal operation in the fast neutron interrogation mode, two kinds of background must be considered. The first, "machine-off" background is the normal, beam-independent type arising from, for example, cosmic rays. The second, "machine-on" background, taken with beam on target and assay chamber empty is the sum of the first plus the delayed neutron response of extraneous material (e.g. the 5 mg of ^{235}U in the FC) that is present in the SSAS.

The calculation software requires that these two background measurements be sequential; that this background pair be the first two and the last two runs of a sequence of assays; and that backgrounds of each type as well as assays have the same time duration. By convention, an assay must have a run ID between 0 and 9999; "machine-off" background between 10 000 and 19 999; and

"machine-on" background between 20 000 and 29 999. At the end of data taking, the "M" option for exiting to the monitor must be used in order to empty the data and calculations buffers onto disk.

V. CALCULATIONS SOFTWARE

When the program encounters a pair of sequential runs with IDs in the range 10 000 to 19 999 and 20 000 to 29 999 (in either order), which are not the first two runs, a series of calculations follows. The program sorts backwards through memory and finds the last pair of such backgrounds. From these two pairs, average backgrounds are calculated and each response relative to the ^3He detector is calculated for assays done in the sequence of runs between the two background pairs.* Ratios of FC to ^3He and MLC to ^3He are also calculated. Both provide a measure of system stability although the FC response

*The relative response, corrected for background, is given by

$$R(A) = \frac{SS(A) - \overline{SS(BG1)}}{{}^3\text{He}(A)} - \bar{E}$$

where

$SS(A)$ is the slab counts for the assay A,

${}^3\text{He}(A)$ is the ${}^3\text{He}$ counts for the assay A,

$\overline{SS(BG1)}$ is the average of the "machine-off" background counts for the two such background runs that bracket the assay A, and

$\bar{E}$ is the average of the two corrected "machine-on" background responses.

$$E = \frac{SS(BG2) - \overline{SS(BG1)}}{{}^3\text{He}(BG2)}$$

where

$SS(BG2)$ is the slab counts for the "machine-on" background run and
 ${}^3\text{He}(BG2)$ is the ${}^3\text{He}$ counts for the "machine-on" background run.

For a typical sample with 160 mg of ^{235}U , $SS(BG1)/SS(A) = 0.03$ and $E/R(A) \approx 0.04$ for a 40 μA , 2.4 MeV beam on a 3.8 mg/cm² lithium target.

is changed by the sample constitution and in particular is sensitive to hydrogen content. After completing these calculations, the responses for run IDs between 0 to 9999 are compared to other runs with the same ID. Specifically, the average response (for all runs with the same ID), the standard deviation of the mean, the relative standard deviation, and chi-squared per degree of freedom are all calculated. The difference between the average and the most recent run with a given ID in units of the 1σ statistical uncertainty of the latter is also given. The program has a capacity of 560 runs. Typical output for a sequence of runs bracketed by two background pairs is reproduced in Appendix B.

For assays in which the FC is absent, as is the case for moderated mode interrogation, no "machine-on" background is necessary. The program then only dumps the scalers of sequential runs on disk and does not perform the calculations described above. An analysis program, MODANL, can then be used to read the raw data from disk and perform the calculations and dump of reduced responses for sequences between "machine-off" backgrounds.

REFERENCES

1. A. E. Evans and M. S. Krick, "Nuclear Safeguards Research with the Los Alamos 3.75-MV Van de Graaff Accelerator," Proc., Fourth Conf. on Scientific and Industrial Applications of Small Accelerators, Denton, Texas (1976), pp. 273-280, and references therein.
2. S. Bourret, T. W. Crane, F. Cverna, G. W. Eccleston, E. Gallegos, D. Garcia, S. Johnson, R. Slice, and C. Spirio, "Model 516 LSI-11 Six-Channel Scaler," Los Alamos National Laboratory report (to be published).

APPENDIX A

VANDG3 PROGRAM LISTING

Los Alamos Identification No. LP-1275

FORTRAN IV V02.1-4 THU 20-MAR-80 01:16:17

PAGE 001

```

C*****
C   PROGRAM FOR THE VANDEGRAAFF
C   STEVE BOURRET 22-AUG-79
C   MOD 4-SEP-79 SCR
C*****
C   START
0001   INTEGER*2 RFLAG,ITO,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0002   LOGICAL*1 FILNAM,NULL,FILTOD,NTOD
0003   INTEGER*4 DATA(7)
0004   DIMENSION ARRAY(560,10),FILNAM(15),FILTOD(15)
0005   COMMON/DISK/DATA
0006   COMMON/MONIT/I100,I102,I104
0007   COMMON/ARRAY/INDEXA,NEWID,LASTID,ARRAY
0008   COMMON IN,ITO,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0009   DATA FILNAM,NULL/'D','K',:,'A','R','C','D','E','F',:,'
1     'D','A','T',0,0,0/
0010   DATA FILTOD,NTOD/'D','K',:,'A','R','C','D','E','F',:,'
1     'D','A','T',0,0,0/
0011   I100=IPEEK('100')   !SAVE MONIT CLOCK VECTOR
0012   I102=IPEEK('102')   !SAVE PSW
0013   I104=IPEEK('104')   !
0014   ICLOCK=800          !INIT TIME TO 10
0015   LASTID=0            !LAST ID #
0016   INDEXA=1            !ARRAY INDEX
0017   RFLAG='167764       !DRV11 INPUT ADDR
0018   INTT = 1

C
C   ENTER FILESPCS FOR THE TWO DATA FILES
C
0019   CALL ASSIGN(INTT,'TT:')
0020   1000 CONTINUE
0021   PAUSE 'MOUNT INITIALIZED DISC ON DK'
0022   TYPE 2
0023   2   FORMAT(3X,'ENTER DATA FILE NAME: RAW DATA'/)
0024   READ(INTT,1100) (FILNAM(I),I=4,9)
0025   1100 FORMAT(6A1)
0026   OPEN(UNIT=2,NAME=FILNAM,ACCESS='SEQUENTIAL',TYPE='NEW',
1     ERR=2100,FORM='FORMATTED',INITIALSIZE=150)
0027   TYPE 2000
0028   2000 FORMAT(3X,'ENTER DATA FILE NAME: RESPONSE DATA'/)
0029   READ(INTT,1100) (FILTOD(I),I=4,9)
0030   OPEN(UNIT=3,NAME=FILTOD,ACCESS='SEQUENTIAL',TYPE='NEW',
1     ERR=2100,FORM='FORMATTED',INITIALSIZE=150)
0031   WRITE(6,3)
0032   3   FORMAT(7X,'ID',9X,'0',9X,'1',9X,'2',9X,'3',9X,'4',9X,'5')
0033   4   ISTART=0          !USED TO TELL IF TIMER RUNNING
0034   IN=1                  !USED IN PAUSE
0035   ITO=1                 !COMPUTED GOTO
0036   STAT=0                !INTERRUPT STATUS
0037   CALL ESCAL('0')       !STOP ALL SCALERS
0038   RTC=ICLOCK            !SET FOR 800 SEC'S
0039   CALL BELL              !MAKE A JOYFUL NOISE
0040   CALL IDEE              !INPUT ID INFOR
0041   TYPE 5

```

```
0042 5      FORMAT(3X,'TYPE H FOR HELP PACKAGE')
0043 10     CALL CK                      !CHECK ALL I/O
0044 1      GOTO(11,4,17,10)ITO          !A WAY OUT
0045 11     IF((IPEEK(RFLAG).AND.'1')GOTO 10      !START SIGNAL FROM DELAY UNIT
0047      CALL CSCAL                      !CL??S LOL SCALERS
0048 15     CALL CK                      !CHECK ALL I/O
0049      GOTO(16,4,17,15)ITO          !A WAY OUT
0050 16     IF((IPEEK(RFLAG).AND.'1').NE.'1')GOTO 15      !
0052 17     CALL ESCAL('0')              !STOP ALL SCALERS
0053      CALL CSCAL                      !CLEAR ALL SCALERS
0054      RTC=ICLOCK                      !INIT CLOCK
0055      ITO=1                          !INIT ITO
0056      CALL ESCAL('77')                !ENABLE ALL SCALERS
0057      ISTART=1                        !ENABLE SEC TIMER
0058      CALL RTCEN(RTC,STAT,TOD,TOD2,ISTART)          !START TIMER
0059 20     CALL CK                      !CHECK ALL I/O
0060      GOTO(20,4,17,20)ITO !WAIT UNTIL UNTIL SOMETHING
0061 2100    WRITE(INTT,2200)
0062 2200    FORMAT(3X,'EUF ENCOUNTERED IN DISC INITIALIZE')
0063      GOTO1000
0064      END
C*****
```

```

0001      SUBROUTINE CK                      !INTERUPT AND KB CHECK
0002      INTEGER*2 RFLAG,ITO,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0003      INTEGER*4 DATAA(7)
0004      DIMENSION ARRAY(560,10),CNT(7)
0005      COMMON/DISK/DATAA
0006      COMMON/ARRAY/INDEXA,NEWID,LASTID,ARRAY
0007      COMMON IN,ITO,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0008      DATA IH/'110/,IR/'122/,IS/'123/,IT/'124/,IM/'115/
0009      DATA II/'111/,IP/'120/,ID/'104/,IN/'116/,IY/'131/
0010      CALL IPOKE('44','10000.OR.IPEEK('44))      !SET BIT 12 IN JSW
0011      ICHAR=ITINR()      !CHECK FOR INPUT FORM KB
0012      IF(ICHAR.LT.0)GOTO 20      !IF NO CHAR GOTO 20
0014      IF(ICHAR.EQ.ID)CALL IDEE      !INPUT ID I/O
0016      IF(ICHAR.EQ.IH)CALL HELP      !GET HELP PACKAGE
0018      IF(ICHAR.EQ.II)ITO=2      !GOTO START OF PROGRAM
0020      IF(ICHAR.EQ.IM)GO TO 500      !GO TO MONIT
0022      IF(ICHAR.EQ.IP)CALL PASE      !PAUSE/CONTINUE
0024      IF(ICHAR.EQ.IR)CALL SREAD      !READ SCALER
0026      IF(ICHAR.EQ.IS)CALL RSTRT      !RESTART
0028      IF(ICHAR.EQ.IT)CALL TIME      !SET TIME
C
0030  20  CALL IPOKE('44','167777.AND.IPEEK('44))      !CLEAR JSW 12
0031      IF(STAT.LT.0)GOTO 30      !INTERUPT FLAG SET?
0033      RETURN
0034  30  STAT=STAT .AND. '77777      !CLEAR FLAG
0035      IF(STAT.EQ. 0)RETURN      !RE\TQ008 NOTHING TO DO
C
0037      SEC,TIME,SEC
GOTO(400,410,400)STAT
0038  41  TYPE 42,STAT
0039  42  FORMAT(////,3X,'UNDEFINED,STATUS DECODE',07)
0040      RETURN
C*****SEC
0041  400  IF(RTC.LT. 0) RTC = 800      !CHECK FOR NEGATIVE TIME
0043      TYPE 401,RTC
0044  401  FORMAT('+',3X,'TIME='18)
0045      STAT=STAT.AND.'177776.OR.'100000      !CLEAR SEC BIT,SET FLAG
0046      RETURN
C****TIMEOUT
0047  410  CALL ESCAL('0)      !STOP ALL SCALERS
0048      TYPE 411
0049  411  FORMAT(3X,'TIMEOUT')
0050      ITO=2      !RETURN TO START
0051      STAT=0      !CLEAR STAT
C
C      HERE "LASTID" IS SET TO THE PREVIOUSLY OUTPUT RUN ID AND
C      "DATAA(1)" IS SET TO THE CURRENT ID NUMBER.
C
0052      ANEWID = NEWID
0053      NORM = IJCVT(DATAA(1),LASTID)
0054      NORM = JAFIX(ANEWID,DATAA(1))
C
C      SCALERS ARE READ INTO THE REAL ARRAY "ARRAY"
C
0055      DO 420 I=2,7      !READ ALL SCALERS AND PRINT

```

```
0056      J=I-2
0057      CALL RSCAL(J,DATAA(I))
      C
      C      'INDEXA' INDEXES THE POSITION IN 'ARRAY' FOR THE CURRENT RUN
      C
0058      ARRAY(INDEXA,I)=AJFLT(DATAA(I))      !CONVERT 2 WORDS TO FLOAT FT
0059 420    CONTINUE
0060      ARRAY(INDEXA,1) = AJFLT(DATAA(1))
0061      DO 412 I=1,7
0062      CNT(I)=AJFLT(DATAA(I))      !CONVERT TO FP
0063 412    CONTINUE
0064      WRITE(6,415)CNT
0065 415    FORMAT(7F10.0)
0066      DO 417 I=1,4
0067      WRITE(6,416)      !TO CAUSE PRINTER TO PRINT BUFF
0068 416    FORMAT(' ',250X)
0069 417    CONTINUE
0070      CALL DISK      !STORE RECORD
      C
      C      LOOKING FOR SEQUENTIAL BACKGROUND RUNS
      C
0071      IF(DATAA(1).GT.10000.AND.LASTID.GT.10000)CALL CAI C
      C
      C      INDEXA IS INCREMENTED
      C
0073      INDEXA=INDEXA+1
0074      RETURN
0075 430    TYPE 440
0076 440    FORMAT(/,3X,'FILE IS FULL')
0077      CALL ENDA
0078 500    TYPE 502
0079 502    FORMAT(3X,'ARE YOU SURE Y/N?')
0080 503    ICHAR = I'TINK()
0081      IF (ICHR .LT. 0) GO TO 503
0083      IF (ICHR .EQ. IY) CALL ENDA
0085      RETURN
0086      END
      C***GO BACK TO MONIT
```

```

0001      SUBROUTINE SREAD
0002      INTEGER*2 RFLAG, ITO, RTC, STAT, TOD, TOD2, ISTART, ICLOCK
0003      INTEGER*4 DATAA(7)
0004      DIMENSION ARRAY(560,10)
0005      COMMON/DISK/DATAA
0006      COMMON/ARRAY/INDEXA, NEWID, LASTID, ARRAY
0007      COMMON IN, ITO, RTC, STAT, TOD, TOD2, ISTART, ICLOCK
0008      CNT=DATAA(1)
0009      TYPE 10, CNT
0010 10    FORMAT(/,3X,'READ SCAL'////,3X,'ID=',F10.0)
0011      DO 420 J=1,6
0012      K=J-1
0013      CALL RSCAL(K,DAT)
0014      CNT=AJFLT(DAT)
0015      TYPE 415,K,CNT
0016 415   FORMAT(I8,F10.0)
0017 420   CONTINUE
0018      RETURN
0019      END

```

```

0001      SUBROUTINE RSTRT
0002      INTEGER*2 RFLAG, ITO, RTC, STAT, TOD, TOD2, ISTART, ICLOCK
0003      INTEGER*4 DATAA(7)
0004      DIMENSION ARRAY(560,10)
0005      COMMON/DISK/DATAA
0006      COMMON/ARRAY/INDEXA, NEWID, LASTID, ARRAY
0007      COMMON IN, ITO, RTC, STAT, TOD, TOD2, ISTART, ICLOCK
0008      TYPE 10
0009 10    FORMAT(/,3X,'RESTART')
0010      ITO=3
0011      RETURN
0012      END

```

```

0001      SUBROUTINE FASE
0002      INTEGER*2 RFLAG,ITD,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0003      INTEGER*4 DATAA(7)
0004      DIMENSION ARRAY(560,10)
0005      COMMON/DISK/DATAA
0006      COMMON/ARRAY/INDEXA,NEWID,LASTID,ARRAY
0007      COMMON IN,ITD,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0008      TYPE 10
0009  10    FORMAT(/3X,'PAUSE')
0010      IN=1-IN                    !TOGGLE
0011      IF(IN.EQ.1)GOTO 20    !WHICH IS IT?
0013      CALL ESCAL('0')        !DISABLE SCALERS
0014      ISTART=0                !STOP SEC CLOCK
0015      RETURN
0016  20    CALL ESCAL('77')        !ENABLE ALL SCALERS
0017      ISTART=1                !START SEC CLOCK
0018      RETURN
0019      END
      C*****IDEE*****)*+*?*****

```

```

0001      SUBROUTINE IDEE
0002      INTEGER*2 RFLAG,ITD,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0003      INTEGER*4 DATAA(7)
0004      DIMENSION ARRAY(560,10)
0005      COMMON/DISK/DATAA
0006      COMMON/ARRAY/INDEXA,NEWID,LASTID,ARRAY
0007      COMMON IN,ITD,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0008      CALL IPOKE('44','167777.AND.IPEEK('44))    !CLEAR BIT 12 IN JSW
0009      TYPE 10
0010  10    FORMAT(/,3X'INPUT ID INFORMATION')
0011      ACCEPT 20,ANEWID
0012  20    FORMAT(F10.0)
0013      NEWID = ANEWID
0014      RETURN
0015      END
      C*****RING BELL*****

```

```

0001      SUBROUTINE BELL
0002      INTEGER*2 RFLAG,ITO,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0003      INTEGER*4 DATAA(7)
0004      DIMENSION ARRAY(560,10)
0005      COMMON/DISK/DATAA
0006      COMMON/ARRAY/INDEXA,NEWID,LASTID,ARRAY
0007      COMMON IN,ITO,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0008      I2='7'
0009      DO 20 I=1,3          !RING 3 TIMES
0010      K=-3000              !DELAY
0011  12      K=K+1
0012          IF(K.NE.0)GOTO 12
0014          TYPE 10,I2
0015  10      FORMAT('+',A1)
0016  20      CONTINUE
0017          RETURN
0018          END
*****DISK*****

```

```

0001      SUBROUTINE DISK
0002      DIMENSION RDATA(7)
0003      INTEGER*4 DATAA(7)
0004      COMMON/DISK/DATAA
0005      TYPE 10
0006  10      FORMAT(/,3X,'DISK')
0007          DO 100 J = 1,7
0008      RDATA(J) = AJFLT(DATAA(J))
0009  100      CONTINUE
0010          WRITE(2,30)(RDATA(J),J=1,7)
0011  30      FORMAT(7(2X,F11.0))
0012          I2='7'          !ASC11 BELL
0013          TYPE 20,I2      !RING BELL
0014  20      FORMAT('+',A1)
0015          RETURN
0016          END

```

```

0001      SUBROUTINE TIME
0002      INTEGER*2 RFLAG,IYD,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0003      INTEGER*4 DATAA(7)
0004      DIMENSION ARRAY(560,10)
0005      COMMON/DISK/DATAA
0006      COMMON/ARRAY/INDEXA,NEWID,LASTID,ARRAY
0007      COMMON IN,IYD,RTC,STAT,TOD,TOD2,ISTART,ICLOCK
0008      CALL IPOKE('44','167777.AND.IPEEK('44))      !CLEAR BIT 12 IN JSW
0009      TYPE 10
0010  10  FORMAT('$',3X,'TIME=')
0011      ACCEPT 20,XRTC      !ACCEPT TIME IN FLOATING P1
0012  20  FORMAT(F10.0)
0013      RTC=XRTC      !CONVERT TO INTEGER
0014      ICLOCK=RTC
0015      RETURN
0016      END
*****

```

```

C      MOD 31-AUG-79
*****
0001      SUBROUTINE HELP
0002      TYPE 10
0003  10  FORMAT(///10X,'KEYBOARD COMMANDS'//
1      5X,'D--INPUT ID INFORMATION'//
1      5X,'I--INITIALIZE PROGRAMM'//
1      5X,'H--HELP PACKAGE'//
1      5X,'M--GO TO MONITOR'//
1      5X,'P--PAUSE/CONTINUE'//
1      5X,'R--READ SCALERS'//
1      5X,'S--RESTART'//
1      5X,'T--SET TIME')
0004      RETURN
0005      END
*****

```

```

0001      SUBROUTINE ENDA
0002      COMMON/MONIT/I100,I102,I104
0003      CALL IPOKE('104,I104)
0004      CALL IPOKE('102,I102)
0005      CALL IPOKE('100,I100)
0006  20  CALL IPOKE('44','167777.AND.IPEEK('44))      !CLEAR JSW 12
0007      CALL CLOSE(2)      !CLOSE DISK FILE
0008      STOP
0009      END

```



```
0001      SUBROUTINE CALC
      C
      C      IS ENTERED WHEN, IN SUBROUTINE CK, A TIMEOUT OCCURS FOR A RUN
      C      WHICH IS THE 2ND OF TWO CONSECUTIVE BACKGROUND RUNS.
      C      THIS RUN IS REFERRED TO BELOW AS THE "CURRENT" RUN. THIS PAIR
      C      OF RUNS IS THE 2ND OR MOST RECENT PAIR. THE PAIR OF BACKGROUND
      C      RUNS WHICH PROCEEDS THE MOST RECENT IS THE 1ST PAIR, AND IS
      C      CHARACTERIZED BY THE "ARRAY" INDICES "JBK1" AND "JMB1", AND
      C      THE RUN ID'S "IRBK1" AND "IKMB1".
0002      COMMON/ARRAY/ INDEXA,NEWID,LASTID,ARRAY
0003      DIMENSION ARRAY(560,10)
0004      REAL MBBAR
0005      INTEGER*2 TEMPID
      C
      C      ASSIGNMENT FOR LINEPRINTER
      C
0006      LP = 6
      C
      C      BEAM CURRENT INTEGRATOR IS 2ND SCALER-AUXILIARY INTERROGATING
      C      FLUX MONITOR
      C
0007      IQMON = 2
      C
      C      HELIUM-3 MONITOR IS 3RD SCALER-PRIMARY INTERROGATING FLUX
      C      MONITOR
      C
0008      IHE3 = 3
      C
      C      FISSION CHAMBER IS 4TH SCALER-AUXILIARY INTERROGATING FLUX
      C      MONITOR
      C
0009      IFC = 4
      C
      C      BREITMAN LONG COUNTER IS 5TH SCALER-AUXILIARY INTERROGATING
      C      FLUX MONITOR
      C
0010      IBM = 5
      C
      C      SLAB IS 6TH SCALER-PRIMARY DELAYED NEUTRON COUNTER
      C
0011      ISLAB = 6
      C
      C      THE FOLLOWING FOUR STEPS INITIALIZE 2ND (MOST RECENT)
      C      BACKGROUND PAIR "ARRAY" INDICES CLEAR FLAGS
      C
0012      JBK2 = 0
0013      JMB2 = 0
0014      JBFLAG = 0
0015      JMFLAG = 0
      C
      C      LABEL "ARRAY" POSITION FOR CURRENT RUN
      C      CURRENT INDEX OF "ARRAY" =1-560; KINDEX, THE CONSIDERED RUN
      C      IS INITIALLY SET TO THIS CURRENT RUN
      C
```

```
0016      KINDEX = INDEXA
      C
      C      RETURN IF THESE ARE THE FIRST TWO BACKGROUNDS
      C
0017      IF(INDEXA .LT. 3) RETURN
0019 1000  CONTINUE
      C
      C      RUN ID UNDER CONSIDERATION
      C
0020      TEMPID = ARRAY(KINDEX,1) + 0.1
      C
      C      THE TWO BACKGROUND RUNS HAVE BEEN IDENTIFIED, PROCEED
      C      TO 1600
      C
0021      IF(JMB2 .NE. 0 .AND. JBK2 .NE. 0) GO TO 1600
      C
      C      FALSE ENTRY: THE CURRENT RUN IS NOT A BACKGROUND
      C
0023      IF(TEMPID .LT. 10000) GO TO 1314
      C
      C      THIS IS A MACHINE OFF BK
      C
0025      IF(TEMPID .LT. 20000) GO TO 1200
      C
      C      THIS IS A MACHINE ON BK
      C
0027      IF(TEMPID .LT. 30000) GO TO 1400
0029      IF(TEMPID .GT. 30000) GO TO 1313
0031 1200  CONTINUE
      C
      C      FOLLOWING CONDITION IS TRUE FOR 2 CONSECUTIVE MACHINE OFF BK
      C      YOU HAVE BEEN THROUGH THIS SECTION BEFORE
      C
0032      IF(JBFLAG .EQ. 1) GO TO 1315
      C
      C      THIS IS THE MOST RECENT MACHINE OFF BKGD RUN
      C      JBK2 LABELS THE POSITION IN "ARRAY" OF THE SECOND
      C      MACHINE OFF BKGD
      C
0034      JBK2 = KINDEX
      C
      C      IRBK2 LABELS THE IS OF THE SECOND MACHINE OFF BKGD
      C
0035      IRBK2 = TEMPID
      C
      C      CONSIDER THE PRECEDING RUN
      C
0036      KINDEX = KINDEX - 1
      C
      C      SET THE MACHINE OFF BACKGROUND FLAG
      C
0037      JBFLAG = 1
0038      GO TO 1000
0039 1400  CONTINUE
```

```
      C
      C      THIS CONDITION TRUE FOR 2 CONSECUTIVE MACHINE ON BKGS
      C      YOU HAVE BEEN THROUGH THIS SECTION BEFORE
      C
0040      IF(JMFLAG .EQ. 1) GO TO 1316
      C
      C      THIS IS THE MOST RECENT MACHINE ON BACKGROUND
      C      JMB2 LABELS THE POSITION IN 'ARRAY' OF THE SECOND MACHINE
      C      ON BACKGROUND
      C
0042      JMB2 = KINDEX
      C
      C      IRMB2 LABELS THE ID OF SECOND MACHINE BACKGROUND
      C
0043      IRMB2 = TEMPID
      C
      C      CONSIDER THE PRECEDING RUN
      C
0044      KINDEX = KINDEX - 1
      C
      C      SET MACHINE ON BACKGROUND FLAG
      C
0045      JMFLAG = 1
0046      GO TO 1000
      C
      C      ONWARD TO A BACKGROUND PAIR TAKEN IN THE PAST (FIRST PAIR)
      C
0047 1600 CONTINUE
      C
      C      INITIALIZE 1ST (PRECEDING MOST RECENT) BACKGROUND
      C      PAIR 'ARRAY' INDICE
      C
0048      JMB1 = 0
0049      JKB1 = 0
      C
      C      ITEMP INITIALLY SET TO THE RUN PRECEDING THE MOST RECENT
      C      BACKGROUND PAIR
      C
0050      JTEMP = INDEXA - 2
0051 195 CONTINUE
      C
      C      RUN ID UNDER CONSIDERATION
      C
0052      TEMPID = ARRAY(JTEMP,1)
      C
      C      ASSAY RUN; UNINTERESTING
      C
0053      IF(TEMPID .LT. 10000) GO TO 800
      C
      C      MACHINE OFF BACKGROUND
      C
0055      IF(TEMPID .LT. 20000) GO TO 200
      C
      C      MACHINE ON BACKGROUND
```

```
      C
0057      IF(TEMPID .LT. 30000) GO TO 400
      C
      C      MISLABELED RUN, UNINTERESTING
      C
0059      IF(TEMPID .GT. 30000) GO TO 800
0061      200      CONTINUE
      C
      C      THIS IS THE PROCEEDING MACHINE OFF BACKGROUND; RUN
      C      IRBK1 IDENTIFIES IS OF FIRST MACHINE OFF BACKGROUND; RUN
      C
0062      IRBK1 = TEMPID
      C
      C      JBK1 IS THE 'ARRAY' INDEX MACHINE OFF BACKGROUND; RUN
      C
0063      JBK1 = JTEMP
      C
      C      SEQUENTIAL MACHINE ON AND MACHINE OFF BACKGROUNDS; UNWARD
      C      TO CALCULATIONS
      C
0064      IF(JMB1 .EQ. JTEMP+1) GO TO 600
      C
      C      OTHERWISE ANOTHER CYCLE
      C
0066      GO TO 800
0067      400      CONTINUE
      C
      C      THIS IS THE PROCEEDING MACHINE ON BACKGROUND
      C      IRMB1 IDENTIFIES ID OF FIRST MACHINE ON BKGD RUN
      C?
0068      IRMB1 = TEMPID
      C
      C      JMB1 IS THE 'ARRAY' INDEX MACHINE OFF BKGD RUN
      C
0069      JMB1 = JTEMP
      C
      C      SEQUENTIAL MACHINE ON AND MACHINE OFF BACKGROUNDS;
      C      ONWARD TO CALCULATIONS
      C
0070      IF(JBK1 .EQ. JTEMP+1) GO TO 600
      C
      C      DECREMENT THE RUN NUMBER ('ARRAY' INDEX) AND TRY AGAIN
      C
0072      800      JTEMP = JTEMP-1
0073      GO TO 195
      C
      C      LOCATION 600 IS REACHED WHEN TWO SETS OF SEQUENTIAL BK AND
      C      MACHINE ON BK ARE FOUND. THE FIRST SET HAS ARRAY INDICES
      C      'JBK1' AND 'JMB1', AND ID'S 'IRBK1' AND 'IRMB1' AND THE
      C      SECOND MOST RECENT, HAS 'ARRAY' INDICES 'JBK2' AND 'JMB1'
      C      AND ID'S 'IRBK2' AND 'IRMB2'
      C
0074      600      WRITE(LP,131)
0075      WRITE(LP,610) IRBK1,IRMB1,IRBK2,IRMB2
```

```

0076 610  FORMAT(5X,'CALCULATIONS FOR RUNS BETWEEN',I6,'(BK1)',I6,'(MR1)',
1      ,I6,'(BK2)',I6,'(MR2).')
      C
      C      CALCULATION OF BACKGROUND AVERAGES
      C
0077      BKBAR = (ARRAY(JBK1,ISLAB) + ARRAY(JBK2,ISLAB))/2.
0078      MBBAR = (ARRAY(JMR1,ISLAB) + ARRAY(JBK1,ISLAB))/ARRAY(JMB1,IHE3)
0079      MBBAR = MBBAR + (ARRAY(JMR2,ISLAB) -
1      ARRAY(JBK2,ISLAB))/ARRAY(JMB2,IHE3)
0080      MBBAR = MBBAR/2.
      C
      C      CALCULATION OF STATISTICAL UNCERTAINTIES
      C
0081      DELBK = (BKBAR/2.)*0.5
0082      A = ARRAY(JMB1,ISLAB) + ARRAY(JBK1,ISLAB)
0083      A = A + (ARRAY(JMR1,ISLAB) -
1      ARRAY(JBK1,ISLAB))**2/ARRAY(JMB1,IHE3)
0084      A = A/(4.0*ARRAY(JMR1,IHE3)**2)
0085      B = ARRAY(JMR2,ISLAB) + ARRAY(JBK2,ISLAB)
0086      B = B + (ARRAY(JMR2,ISLAB) -
1      ARRAY(JBK2,ISLAB))**2/ARRAY(JMB2,IHE3)
0087      B = B/(4.0*ARRAY(JMR2,IHE3)**2)
0088      DELMB = (A + B)*0.5
      C
      C      WRITE AVERAGE BACKGROUNDS AND UNCERTAINTIES
      C?
0089      WRITE(LP,120) BKBAR,DELBK,MBBAR,DELMB
0090 120  FORMAT(5X,'AVG. BK = ',F10.2,'+',F10.2,'',AVG. MR = ',E13.6,
1      ',E13.6)
0091      WRITE(LP,125)
0092 125  FORMAT(20X,'3HE RESPONSE',12X,'FC RESPONSE',13X,'HM RESPONSE')
0093      ICALC = JBK1 + 1
      C
      C      "ARRAY" INDEX OF 1ST ASSAY FOLLOWING 1ST BKGD PAIR
      C
0094      IF(JMB1 .GT. JBK1) ICALC = ICALC + 1
      C
      C      DO FOR ALL ASSAYS BETWEEN THE TWO PAIRS
      C
0096      DO 150 I = ICALC,JBK2
      C
      C      CALCULATE SLAB RESPONSE REL TO HE-3 MONITOR STATISTICAL ERROR
      C
0097      RESP = (ARRAY(I,ISLAB) - BKBAR)/ARRAY(I,IHE3) - MBBAR
0098      A = (ARRAY(I,ISLAB)/ARRAY(I,IHE3))**2
0099      A = A*(1.0/ARRAY(I,ISLAB) + 1./ARRAY(I,IHE3))
0100      B = (BKBAR/ARRAY(I,IHE3))**2
0101      B = B*((DELBK/BKBAR)**2 + 1.0/ARRAY(I,IHE3))
0102      DELRES = (A + B + DELMB**2)*0.5
      C
      C      CALCULATE FISSION CHAMBER/HE-3 MONITOR RATIO AND
      C      STATISTICAL ERROR
      C
0103      FCRES = ARRAY(I,IFC)/ARRAY(I,IHE3)

```

```
0104      DELFC = FCRES*(1./AKRAY(I,IFC) + 1./AKRAY(I,IHE3))**.5
      C
      C      CALCULATE BREITMAN/HE-3 MONITORS RATIO AND STATISTICAL
      C      ERROR
0105      STAB = ARRAY(I,IBM)/AKRAY(I,IHE3)
0106      DELSTA = STAB*(1./ARRAY(I,IBM) + 1./ARRAY(I,IHE3))**.5
      C
      C      STORE SLAB RELATIVE RESPONSE AND UNCERTAINTY IN 'AKRAY'
      C
0107      ARRAY(I,8) = RESP
0108      ARRAY(I,9) = DELRES
      C
      C      WRITE RESULTS
      C
0109      WRITE(LP,129) AKRAY(I,1)
0110      WRITE(LP,130) RESP,FCRES,STAB
0111      IDUM = ARRAY(I,1)
0112      WRITE(3,132) IDUM,RESP,DELRES
0113 132   FORMAT(15,2E13.6)
0114      DELRES = DELRES/RESP
0115      DELFC = DELFC/FCRES
0116      DELSTA = DELSTA/STAB
0117 129   FORMAT(5X, 'CALCULATIONS FOR RUN 1D' ,F10.0)
0118 130   FORMAT(3X,'R',5X,3(10X,E13.6))
0119 131   FORMAT(/)
0120      WRITE(LP,133) DELRES,DELFC,DELSTA
0121 133   FORMAT(3X,'DELR',2X,3(10X,E13.6))
0122      WRITE(LP,134) DELRES,DELFC,DELSTA
0123 134   FORMAT(3X,'DELR/R',3(10X,E13.6))
      C
      C      EXIT LOOP IF THE NEXT ASSAY IS ONE OF THE BACKGROUND
      C
0124      IF(I+1 .EQ. JBK2 .OR. I+1 .EQ. JMB2) GO TO 160
0126 150   CONTINUE
0127 160   CONTINUE
0128      ILAST = I
      C
      C      DO FOR ALL ASSAYS
      C
0129      DO 190 K = ICALC,ILAST
      C
      C      SETS INITIAL CONDITIONS
      C
0130      AVG = 0.0
0131      ICT = 0.0
0132      SDMEAN = 0.0
0133      RSD = 0.0
0134      CHISQ = 0.0
      C
      C      DO AVERAGE AND STANDARD DEVIATION FOR EACH RUN
      C
0135      DO 170 I = 1,ILAST
```

```
      C      EXIT UPON COMPLETION OF CALCULATION
      C
0136      IF (ARRAY(I,1) .NE. ARRAY(K,1)) GO TO 170
0138      ICT = ICT + 1
      C
      C      CALCULATE AVERAGE
      C
0139      AVG = AVG + ARRAY(I,8)/(ARRAY(I,9)**2)
      C
      C      CALCULATE MEAN STANDARD DEVIATION
      C
0140      SDMEAN = SDMEAN + ARRAY(I,9)**(-2)
0141      170 CONTINUE
      C
      C      CALCULATE AVERAGE FOR RUN
      C
0142      AVG = AVG/SDMEAN
      C
      C      CALCULATE STANDARD DEVIATION FOR RUN
      C
0143      SDMEAN = SDMEAN**(-0.5)
      C
      C      DO REL STANDARD DEVIATION AND CHISQ FOR EACH RUN
      C
0144      DO 175 I = 1, ILAST
      C
      C      EXIT UPON COMPLETION OF CALCULATIONS
      C
0145      IF (ARRAY(I,1) .NE. ARRAY(K,1)) GO TO 175
      C
      C      CALCULATE REL STANDARD DEVIATION FOR EACH RUN
      C
0147      RSD = RSD + (ARRAY(I,8) - AVG)**2
0148      CHISQ = CHISQ + ((ARRAY(I,8) - AVG)/ARRAY(I,9))**2
0149      175 CONTINUE
      C
      C      SAME AS ABOVE 'IF'S'
      C
0150      IF (ICT .LE. 1) GO TO 190
      C
      C      SETS NUMBER OF RUNS
      C
0152      FCT = ICT
      C
      C      FINIAL CALCULATION FO REL STANDARD DEVIATION
      C
0153      RSD = RSD/(FCT - 1.)
0154      RSD = RSD**0.5
      C
      C      CALCULATE FINIAL CHISQ/DEG OF FREEDOM
      C
0155      CHISQ = CHISQ/(FCT - 1.)
      C
      C      CALCULATE HOW FAR FORM SIGMA THIS RUN IS OFF
```

```
      C
0156      PEE = (ARRAY(K,8) - AVG)/ARRAY(K,9)
      C
      C      WRITE SAMPLE INFORMATION
      C
0157      WRITE(LP,131)
0158      WRITE(LP,2001) ARRAY(K,1),ICT,AVG,SDMEAN,RSD,CHISQ,PEE
0159      2001  FORMAT(5X,'RUN ',F6.0,' IS ', I3,'TH RUN
1          WITH THAT ID.',/,5X,'WT. AVG. = ',E13.6,'+',E13.6,' (SD
2          OF MEAN), RSD = 'E13.6,/,5X,'CHISQ/(N-1) = ',
3          E13.6,'CURRENT VALUE IS ',F7.3,' SIGMAS FROM MEAN.')
0160      190  CONTINUE
0161      DO 417 I = 1,4
0162      417  WRITE(LP,416)
0163      416  FORMAT('+',250X)
0164      WRITE(LP,2002)
0165      2002  FORMAT('1')
      C
      C      RETURN TO CONTROLLING PROGRAM UPON COMPLETION OF CALC
      C
0166      RETURN
0167      1314 WRITE(LP,140)
0168      140  FORMAT(5X,'FALSE ENTRY TO CALC; RETURN.')
0169      RETURN
0170      1313 WRITE(LP,141)
0171      141  FORMAT(5X,'PRESUMED BK OR MB HAS .GT. 30000; RETURN.')
0172      RETURN
0173      1315 WRITE(LP,142)
0174      142  FORMAT(5X,'TWO CONSECUTIVE BKS; NEED AN MB. RETURN')
0175      RETURN
0176      1316 WRITE(LP,143)
0177      143  FORMAT(5X,'TWO CONSECUTIVE MBS; NEED A BK. RETURN')
0178      RETURN
0179      END
```


APPENDIX B

VANDG3 SAMPLE OUTPUT

142.	3035092.	250906.	141183.	1726363.	15018.	7997543.
194.	3036572.	251154.	142474.	1732109.	15473.	7997267.
193.	3029827.	251692.	142755.	1724230.	15617.	7997881.
123.	3035993.	250197.	140555.	1718254.	9469.	7998396.
20004.	3040712.	248646.	131093.	1710610.	1589.	7999363.
10004.	0.	13.	0.	23.	833.	7997684.

CALCULATIONS FOR RUNS BETWEEN 10003(BK1) 20003(MR1) 10004(BK2) 20004(MR2).

AVG.BK = 853.00+ 20.65, AVG. MR = 0.278002E-02+ 0.137678E-03

3TH RESPONSE

FC RESPONSE

BM RESPONSE

CALCULATIONS FOR RUN ID 142.

R	0.536754E-01	0.562693E+00	0.688052E+01
DEL R	0.527836E-03	0.187205E-02	0.147005E-01
DEL R/R	0.983385E-02	0.332694E-02	0.213654E-02

CALCULATIONS FOR RUN ID 194.

R	0.554313E-01	0.567277E+00	0.689660E+01
DEL R	0.534951E-03	0.188148E-02	0.147254E-01
DEL R/R	0.965071E-02	0.331669E-02	0.213517E-02

CALCULATIONS FOR RUN ID 193.

R	0.558790E-01	0.567181E+00	0.685056E+01
DEL R	0.536240E-03	0.187926E-02	0.146177E-01
DEL R/R	0.959645E-02	0.331332E-02	0.213379E-02

CALCULATIONS FOR RUN ID 123.

R	0.316568E-01	0.561777E+00	0.686760E+01
DEL R	0.427557E-03	0.187262E-02	0.146954E-01
DEL R/R	0.135060E-01	0.333339E-02	0.213982E-02

RUN 142. IS 3TH RUN WITH THAT ID.

WT. AVG. = 0.536693E-01+ 0.303559E-03 (SD OF MEAN), RSD = 0.480203E-03

CHISQ/(N-1) = 0.837401E+00 CURRENT VALUE IS 0.011 SIGMAS FROM MEAN.

RUN 194. IS 5TH RUN WITH THAT ID.

WT. AVG. = 0.559341E-01+ 0.235232E-03 (SD OF MEAN), RSD = 0.548091E-03

CHISQ/(N-1) = 0.106342E+01 CURRENT VALUE IS -0.940 SIGMAS FROM MEAN.

RUN 193. IS 5TH RUN WITH THAT ID.

WT. AVG. = 0.557869E-01+ 0.234490E-03 (SD OF MEAN), RSD = 0.262910E-03

CHISQ/(N-1) = 0.253027E+00 CURRENT VALUE IS 0.172 SIGMAS FROM MEAN.

RUN 123. IS 3TH RUN WITH THAT ID.

WT. AVG. = 0.313761E-01+ 0.242117E-03 (SD OF MEAN), RSD = 0.538833E-03

CHISQ/(N-1) = 0.162981E+01 CURRENT VALUE IS 0.657 SIGMAS FROM MEAN.